3

Deploying Diffserv

3.1 Introduction

In Chapter 1, we defined the key service level agreement (SLA) parameters for IP applications and services as being: delay, delay variation or jitter, packet loss rate, throughput, availability, and per flow sequence preservation. In Chapter 2, we described the QOS component mechanisms and architectures that can be used in engineering a network to meet SLA targets. In this chapter, we build on the context and understanding created by Chapters 1 and 2, to show how the Differentiated Services architecture (Diffserv) can be practically deployed at the network edge and in the network core in order to satisfy defined application SLA requirements. Diffserv is by far the most widely deployed IP QOS architecture; it is widely deployed both in private enterprise networks and in service provider (SP) networks providing virtual private network (VPN) services to enterprises.

The edge of the Diffserv domain represents the provider/customer boundary for the Diffserv-enabled services being offered; Diffserv achieves scalability by performing complex per-customer QOS functions and maintaining per-customer state, only at the edges of the network. Hence, the policies applied at the edge of the network are different from, and generally more complex than, those applied within the core. Many networks are built with a hierarchy consisting of core routers (CRs), which provide connectivity between distribution routers (DRs), which in turn aggregate connections to routers at remote sites, each of which have local access routers (ARs). If, for example, we consider a Diffserv deployment in this type of network,

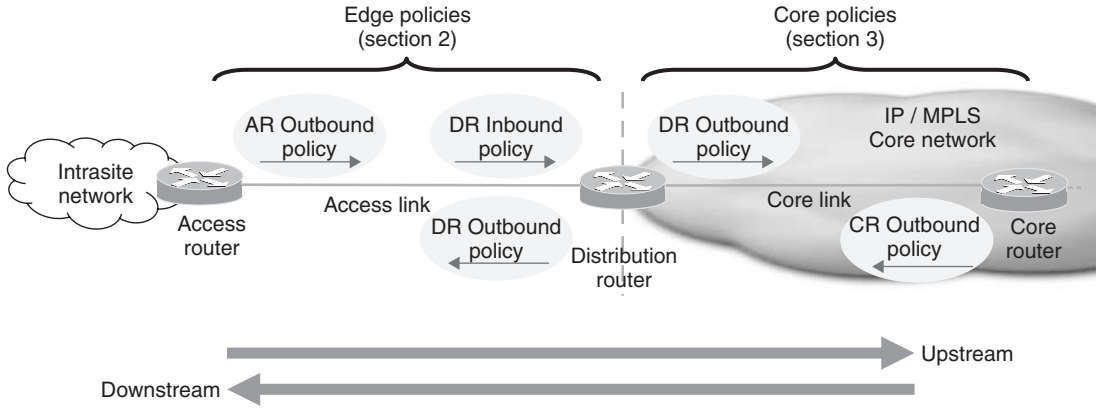

Figure 3.1 Application of Diffserv policies

as shown in Figure 3.1, the access link will normally be the edge of the Diffserv domain, with edge Diffserv policies being applied outbound on the core facing (i.e. upstream) AR interface and outbound on the access facing (i.e. downstream) interface of the DR, which may have inbound policies in some cases also. Core Diffserv policies are then typically applied outbound on the core facing (i.e. upstream) interfaces of the DR and on outbound on all CR interfaces.

In the network core where link bandwidths are high and traffic is highly aggregated, the SLA requirements for a traffic class can be translated into bandwidth requirements, and the problem of SLA assurance can effectively be reduced to that of bandwidth provisioning, which may be on a per-class of service basis. At the network edge, where bandwidth is lower and there is limited aggregation of traffic, different considerations become significant, such as providing isolation between applications. Further, the mechanisms employed on the lower speed access links at the edges of the network to deliver tightly bounded SLA commitments may be different from those used in the core, because factors such as serialization delay become significant at lower speeds.

This chapter focusses on Diffserv designs for the network edge and in the core; in considering Diffserv designs, we apply the following three key design objectives:

- ensuring that the different SLA requirements for each respective class can be met
- optimizing the use of available bandwidth
- keeping the design as simple as possible.

The following sections apply these objectives to Diffserv design case studies supporting applications and services with tightly bounded SLAs for IP service performance. The case studies presented are examples, and as such do not represent the only way of doing things; rather, they aim to describe possible methodologies and the most important aspects to consider with respect to Diffserv designs. In these case studies, where we use the terms service provider and customer, we use them generically. This does not infer, however, that the case studies are only applicable to VPN service providers – the networking department of an enterprise organization is a service provider to their enterprise.

3.2 Deploying Diffserv at the Network Edge

3.2.1 Why is the Edge Key for Tight SLA Services?

In every network, there is a choice about whether or not to deploy QOS mechanisms, such as those defined by Diffserv. Without such QOS mechanisms, however, in order to support services and applications with tightly bound SLA requirements, the available capacity needs to be over-provisioned relative to the peak of the aggregate offered load; this needs to be ensured in the milliseconds timeframe (see Chapter 2, Section 2.1.3). One school of though is that bandwidth will become cheaper and more widely available and hence over-provisioning will be a viable option. At the edge of the network, however, even though access bandwidth speeds have increased and

costs have reduced over time, demands for bandwidth from applications combined with end-systems' capabilities to drive such bandwidth has more than kept pace with the availability of bandwidth, and such over-provisioning has not been viable in practice. Access link costs are a significant component of network costs and to minimize operational expenditure, customers will often delay upgrading these links as long as possible; consequently, access links are often under-dimensioned and prone to congestion.

This may change in the future; however, it is very unlikely that this will change in the foreseeable term. Further, unless we provide for the possibility of dealing with congestion (i.e. assuming that over-provisioning is not always going to be possible) then supporting real-time services with requirements for tight bounds on delay, jitter, and loss, will be precluded in cases where congestion occurs. Hence, in the rest of this section, we consider that peak over-provisioning of access links is not a viable option to support services and applications with stringent SLA requirements, as is the case in practice today, and we consider how Diffserv can be deployed to achieve this end.

3.2.2 Edge Diffserv Case Study

In this section, we work through an example edge Diffserv case study, defining the typical class SLA characteristics and describing how Diffserv can be deployed at the network edge to ensure these SLA characteristics are met. In this study, we refer to the access segment SLA, adopting the segmented SLA approach as described in Chapter 1, Section 1.4.1.

We note that Diffserv achieves scalability by performing complex per-customer QOS functions and maintaining per-customer state, only at the edges of the network; hence, implicitly the Diffserv edge design is more complicated than the core design.

3.2.2.1 SLA Specification

We start by considering a simple case, where a leased line of bandwidth Xkbps provides the connectivity between the AR and the DR, and

where four service classes are supported, each with a different SLA. We define the SLA characteristics for three "customer-facing" classes and one class that is used by the service provider for essential network service control functions, and which is not available to the end customer:

- *VoIP class (VoIP).* This class targets interactive applications with requirements for defined bandwidth, low loss, and tight bounds on delay, such as VoIP bearer traffic.
- *Premium data throughput-optimized class (Prm-th).* This class targets business applications, that should receive priority access to the available bandwidth over standard applications, but which do not have a defined delay requirement; this might include business critical file transfer applications, for example.
- *Standard class (Std).* The *Std* class is used for all other "customer-facing" traffic, which has not already been classified as *VoIP* or *Prm-th*. This class may be used for email and web applications, for example.

  Note that the term "best-effort" has intentionally not been used for this class, as *best-effort* infers no SLA commitments. Whereas in this case, the *Std* class has defined SLA commitments, they just represent the lowest of the SLA commitments for the classes offered.
- *Control class (Ctrl).* The *Ctrl* class is used by the service provider and is dedicated for network control traffic, ensuring that bandwidth on the access link is guaranteed for essential functions such as for routing protocols and for telnet or SNMP access to the AR. This ensures that congestion of the access link caused by the customer cannot impact the SP's ability to manage the delivery of the service. It also ensures that customer traffic is protected from being impacted by management traffic, such as large file transfers due to router software upgrades, for example.

We start by considering these service classes, as illustrated by Figure 3.2, because they represent the baseline of many Diffserv deployments today. In subsequent sections, we consider additional variations to

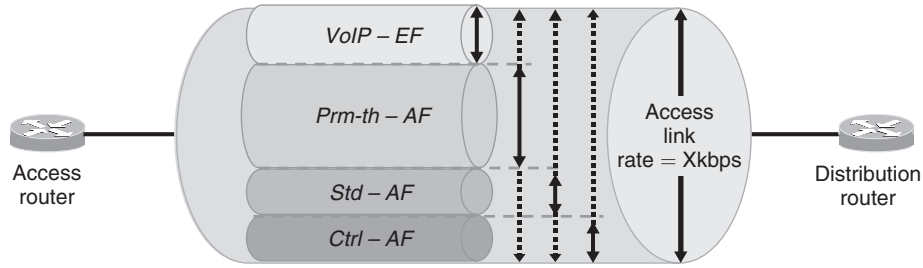

Figure 3.2 Edge case study with four classes

the design, capable of supporting the SLA requirements of the range of applications described in Chapter 1.

The detail of the SLA definitions contracted for each class is defined in the proceeding sections. A summary of the SLA definitions for all of the classes is provided in Section 3.2.2.5.

3.2.2.1.1 VoIP Class (VoIP)

The SLA for the VoIP class is defined in terms of low delay, low jitter, and low loss, and it has a specified bandwidth and availability; attainable throughput is derived from the committed bandwidth and loss rate. The class may support a commitment for per flow sequence preservation.

In agreeing to supply and receive the service, the SP and customer respectively assent to a contract that defines an ingress committed rate (ICR) from the customer site to the SP and an egress committed rate (ECR) from the SP to the customer site (see Chapter 1, Section 1.2.4.2); normally, this relationship is specified symmetrically (i.e. ICR = ECR). The SP enforces the contract by limiting the rate of *VoIP* traffic to/from the customer site using a token bucket policer, with rate R_v and committed burst size B_v; non-conformant traffic will be dropped by the SP. The rate R_v will be selected by the customer and will be offered by the SP up to a defined percentage of the access link speed. The SP will normally limit the maximum percentage of the access link bandwidth that is available for this class in order to ensure that the class latency commitments can be met; for how that maximum

percentage is defined, see Section 3.2.2.8. There will also normally be a defined minimum link speed, below which the SP will not offer this class, for example 256 kbps, because increased serialization delay will mean that the delay target for the class is not achievable at lower rates. B_v will be set based on the offered class delay commitment; the maximum burst value effectively limits the maximum latency of traffic in the class.

For conformant traffic, the SP will commit to a maximum one-way edge segment latency, L_v. The edge segment latency is only one component of the ear-to-mouth delay that impacts a VoIP call. Hence, before defining the edge segment latency, the maximum acceptable ear-to-mouth delay for the particular VoIP service must be defined. A network QOS design should then take this budget and apportion it to the various components of network delay (propagation delay through the backbone, scheduling delay due to congestion, and the access link serialization delay) and end-system delay (due to VoIP codec and de-jitter buffer). Consider a typical example VoIP delay budget allocation as follows.

- The end-to-end (ear-to-mouth) delay target is assumed to be 100 ms, adding a significant 50 ms safety margin to the G.114 target of 150 ms to ensure that most users will be very satisfied (see Chapter 1, Section 1.3.1.1). Note that in many situations there may be additional delays incurred due to tandem encoding (repeated encoding and decoding of the signal), which need to be taken into account in the overall delay budget. There may also be cases where a service may span A multiple networks, and a particular network "owns" only a portion of the overall end-to-end delay budget, hence in practice, it is often important to minimize the delay in all portions of the network when supporting VoIP services.

  Next deduct the significant contributors to the fixed components of delay from the end-to-end delay budget, and deduce the remainder to be apportioned to the variable components of delay:
  - 25 ms is deducted for codec delay, assuming G.711–20 ms (see Chapter 1, Section 1.3.1). Other codecs may incur larger delay

and hence consume more of the end-to-end delay budget; lower bandwidth codecs typically incur larger codec delays.

Note that the packetization interval clearly has an impact on the codec delays, hence larger packetization intervals will implicitly result in larger codec delays.

- Propagation delay is often budgeted for using the widest diameter in the network, which for example, in a national fiber-based network in the US would give a worst-case (coast-to-coast) of approximately 6000 km or ~6000 * 1.25 * 5/1000 = ~40 ms of one-way propagation delay (from Section 1.2.1.1 in Chapter 1).
- In this example we assume that the minimum per-hop delay (switching delay) is relatively negligible, which may not always be the case; if not then it also needs to be included in the above budget.

Hence, the remainder to be apportioned to the variable components of delay is:

Mouth-to-ear budget	100 ms	
Backbone propagation	−40 ms	(assumes ~6000 km)
Example codec delay	−25 ms	(assumes G.711–20 ms)
Variable delay budget	=35 ms	

- We allocate 5 ms of the variable delay budget to the backbone, where links speeds are higher than the access, and hence a smaller portion of the end-to-end delay budget is consumed.
- This leaves 30 ms of the variable delay budget for the access links; 15 ms is allocated to ingress and 15 ms to egress.

L_v is therefore typically in the range 15 ms–50 ms, depending upon the particular ear-to-mouth delay target, and the delay budgeting for the specific network design. L_v is normally specified for a defined packet size.

End-to-end loss rates of typically less than 0.1% (see Chapter 1, Section 1.3.1.3) are offered for the VoIP class.

The contract will stipulate the classification criteria that the SP will use to identify the *VoIP* class at the network edge. Potentially, the classification criteria for any class may use complex classification (which may match on source or destination IP addresses, for example) or simple classification if the end-system or a downstream element is trusted to pre-mark the traffic. Once classified and policed, conformant traffic will be marked with a defined Diffserv codepoint (DSCP) value D_v, if it is not already pre-marked, such that within the network core, traffic classes can be identified by their DSCP markings rather than requiring complex classification.

3.2.2.1.2 Premium Data Throughput-Optimized Class (*Prm-th*)

The SLA for the *Prm-th* class is defined in terms of a specified bandwidth and availability with a commitment for per flow sequence preservation. Jitter is not important for this class and thus it is not defined.

The SP commits to a minimum class bandwidth, R_t, which is typically set to 80–90% of the remaining access link bandwidth after the *VoIP* class has been serviced. As the *Prm-th* class has a higher bandwidth allocation than the *Std* class, if there were the same offered load in both classes, traffic in the *Prm-th* class should receive better service. This gives the end customer the option of allocating some applications to the *Prm-th* class, such that they receive better service than applications in the *Std* class, dependent on managing the relative loads between the classes. The *Prm-th* class is able to re-use unused bandwidth from any other class up to the available link bandwidth and therefore the maximum rate for the class is not enforced with a policer. Consequently, the class delay and loss are dependent upon the customer's actual offered traffic profile for the class, which is outside of the SP's control. Therefore, although the service for the class may have an implied loss and delay commitment, the SP cannot provide explicit commitments for delay and loss for this class at the network edge, although they may provide such commitments across the backbone. Attainable class throughput for TCP-based applications is dependent upon the actual loss rate and RTT experienced by traffic within the class, capped by the access-link bandwidth.

The contract will also define the classification criteria that the SP will use to identify the class and stipulate that conformant traffic will be marked with a defined DSCP value, D_t.

3.2.2.1.3 Standard Data Class (*Std*)

The *Std class* SLA is defined in terms of a specified bandwidth, availability, and commitment for per flow sequence preservation. Jitter is not important for this class and thus it is not defined.

The SP commits to a minimum class bandwidth, R_s, which is typically set to 10–20% of the remaining access link bandwidth once the *VoIP* class has been serviced. This class can re-use any other class's idle bandwidth up to the available link bandwidth. As for the *Prm-th* class, the SP does not provide delay and loss commitments for the *Std* class at the network edge. Attainable class throughput is again dependent upon the actual loss rate and RTT experienced by the class, capped by the access-link bandwidth. The SP may provide a commitment for loss and delay across the core.

The contract will also stipulate that *Std* class traffic will be marked with a defined DSCP value, D_s.

3.2.2.1.4 Control Class (*Ctrl*)

The *Ctrl* class is assured a minimal share of the access-link bandwidth, e.g. ~1%, although normally with a minimum of ~8–16 kbps. The class also has the ability to re-use bandwidth from the other classes that may be idle, up to the available link bandwidth.

3.2.2.2 Diffserv Meta-Language

In the following sections, we describe the detailed Diffserv design required to support these SLA commitments. To ease the description of the Diffserv design we use meta-language defined in Figure 3.3.

3.2.2.3 High-speed Edge Design

We consider "high-speed" access-links as those where the link rate is high enough that link fragmentation and interleaving mechanisms

Diffserv Meta-Language	
Command	Meaning
policy <policy_name>	Defines the start of a Diffserv policy, which may be applied to a particular interface or logical connection.
class <class_name>	Defines the start of the definition of the classification criteria and actions applied to a traffic class within a Diffserv policy.
classify [not] <criteria>	Defines the classification criteria for the particular class; see Chapter 2 Section 2.2.1. Although a number of complex and simple criteria are possible, we define only the simple criteria "DSCP <D>" and "EXP <E>." Where multiple classification criteria are applied, a logical OR operation is assumed between classification criteria.
EF	Indicates that the class will be assigned to a queue serviced with an EF PHB; see Chapter 2 Section 2.3.4.2.1.
AF (<m>)	Indicates that the class will be assigned to a queue serviced with an AF PHB with an assured minimum rate *m*; see Chapter 2 Section 2.3.4.2.2.
mark DSCP (<D>)	Defines the DSCP marking that will be set for packets in the particular class; see Chapter 2 Section 2.2.2.
SR-TCM (<cir>, <cbs>, <ebs>) *green-action <action>* *yellow-action <action>* *red-action <action>*	Applies an RFC2697 single rate three color marker (SR-TCM) to the class with specified committed information rate (CIR), committed burst size (CBS), and excess burst size (EBS); see Chapter 2 Section 2.2.2. Possible resulting actions are 'drop' or 'transmit' or 'transmit-and-mark DSCP (D)'.
TR-TCM (<cir>, <cbs>, <pir>, <pbs>) *green-action <action>* *yellow-action <action>* *red-action <action>*	Applies an RFC2698 two rate three color marker (TR-TCM) to the class with specified committed information rate (CIR), committed burst size (CBS), peak information rate (PIR) and peak burst size (PBS); see Chapter 2 Section 2.2.3.2. Possible resulting actions are 'drop' or 'transmit' or 'transmit-and-mark DSCP (D)'.
drop	This action will drop packets which match the particular condition.
transmit	This action will transmit packets which match the particular condition, without changing the DSCP marking of the packets.
transmit-and-mark DSCP (<D>)	This action will set the DSCP marking of packets which match the particular condition and transmit them.
shape (<r>, <b>)	Applies a token bucket shaper to the class with specified rate *r* and burst *b*; see Chapter 2 Section 2.2.4.3.

Figure 3.3 Diffserv meta-language

Diffserv Meta-Language	
Command	Meaning
tail-drop-limit (<t>)	Applies a tail-drop queue-limit to the class queue, dropping packets received for that particular class when the class queue length exceeds *t* bytes; see Chapter 2 Section 2.2.4.2.1.
RED (DSCP <D>, <minth>, <maxth>, <w>, <pmax>)	Applies a RED profile to traffic with the specified DSCP within the class queue, with defined minimum threshold (*minth*), maximum threshold (*maxth*), exponential weighting constant (w) and probability of packet loss at *maxth* (p_{max}); see Chapter 2 Section 2.2.4.2.3. Multiple RED profiles can be applied to the same queue to effect WRED.
RED (EXP <D>, <minth>, <maxth>, <w>, <pmax>)	Applies a RED profile to traffic with the specified MPLS EXP within the class queue.
<>	Indicates required parameter or parameters.
[]	Indicates optional parameter or parameters.
*	Indicates that the parameter is wild-carded.
{}	Indicates a hierarchy within the Diffserv policy.

Figure 3.3 (*Continued*)

are not required to mitigate the impacts of serialization delay when supporting services with tight SLA bounds for latency; this is typically at link speeds of around 1 Mbps and above; see Section 3.2.2.4.1 for a discussion on low-speed edge designs.

The configuration in Figure 3.4 defines a design to achieve the SLA specification described in Section 3.2.2.1 in terms of the Diffserv meta-language defined in Figure 3.3. This configuration would typically be applied outbound on the core facing (i.e. upstream) AR interface and outbound on the access facing (i.e. downstream) interface of the DR. With this baseline design, there are no inbound policies applied to the AR or DR, although the application of inbound DR policies is required in the design variation discussed in Section 3.2.2.4.3.

Note that within a class the ordering of actions is assumed as described in Chapter 2, Section 2.4.

The following sections describe the design considerations of each class, in order to ensure they support their class SLA commitments.

```
policy outbound-high-speed-edge-policy
 class Voip
   classify DSCP (Dv)
   SR-TCM (Rv, Bv, 0)
     green-action transmit
     red-action drop
   EF
 class Prm-th
   classify <criteria>
   AF (Rt)
   mark DSCP (Dt)
   RED (DSCP(*), <minth>, <maxth>, <w>, <pmax>)
 class Ctrl
   classify DSCP {Dc, 48}
   AF (Rc)
   RED (DSCP(*), <minth>, <maxth>, <w>, <pmax>)
 class Std
   classify *
   AF (Rs)
   mark DSCP (Ds)
   RED (DSCP(*), <minth>, <maxth>, <w>, <pmax>)
```

Figure 3.4 High-speed edge design

3.2.2.3.1 VoIP Class (VoIP)

Considering the *VoIP* class Diffserv configuration in Figure 3.4 it is assumed that VoIP traffic is pre-marked at the source to DSCP D_v, which is used to classify the traffic.

The SLA latency commitment is assured through the following aspects of the design:

- Defining this class as "expedited forwarding" (EF) in order to request the lowest latency service from the scheduler, which would typically be implemented with a strict priority scheduler.

- Ensuring that the arrival rate enforced by the class policer, R_v, is smaller than the servicing rate for the class so that no long-standing buffering can occur. Assuming a strict priority queue EF implementation, the servicing rate for the class would equal the rate of the link where the policy was applied. The class policer, for example, could be the single rate three color marker (SR-TCM) defined in RFC2697, with CIR = R_v, CBS = B_v, with EBS = 0 (i.e. the excess burst is not used in this case) and applying a green action of transmit and a red action of drop. Applied in this way, the SR-TCM would enforce a maximum rate of R_v and a burst of B_v on the traffic stream, and any traffic in excess of this would be dropped.
- Specifying the SLA contract such that the maximum allowed class burst size, B_v, when serviced at the link rate (assuming a strict priority queue EF implementation) ensures that the burst is serviced within the class latency commitment L_v, i.e. $B_v/link\_rate + L_s < L_v$, where L_s represents the worst-case delay impact on EF traffic due to the scheduler and the interface FIFO (see Section 3.2.2.4.1). B_v will in turn determine the maximum percentage of VoIP traffic that can be supported on the access link, as discussed in Section 3.2.2.8.
- With the policer configured, a tail drop queue limit is not required for the *VoIP* class queue. Some vendor implementations, however, may require that one is configured and if so, the queue-limit must be greater than or equal to B_v, to ensure that packets that are within the permitted burst for the class are not dropped. Therefore, the only actual packet loss that conformant traffic within the class should experience is due to layer 1 bit errors or network element failures, which accounts for the committed loss rate for the class offered by the SP.

While this class is targeted at VoIP bearer traffic, there is a design decision to be made as to whether VoIP signaling traffic will share the same class. VoIP signaling traffic generally consists of a small number of small sized packets, hence in some deployments it may be viable to support both VoIP bearer and signaling traffic from the same queue without one having a significant impact on the other.

Some network providers have a policy of ensuring isolation between bearer and signaling, which requires that they are serviced from separate classes, in this case VoIP signaling traffic could be serviced from one of the other AF classes (such as *Prm-th*) or could potentially be assigned its own class.

3.2.2.3.2 Premium Data Throughput-Optimized Class (*Prm-th*)

Considering the *Prm-th* class Diffserv configuration in Figure 3.4, it is assumed that complex classification criteria (which are unspecified in the figure) are used to classify packets into the class. Such criteria could be based upon source or destination addresses, or source or destination UDP/TCP ports, or deep packet inspection/stateful inspection (DPI/SI) for example. The DSCP of all packets classified into the class is set to D_t.

The *Prm-th* class SLA commitment is ensured by treating the class with an AF per-hop behavior (PHB), which provides a minimum bandwidth assurance of R_t. Assuming that a work-conserving scheduler is used, the *Prm-th* class will have access to all unused interface capacity once the *VoIP*, *Std*, and *Ctrl* data classes have been serviced.

The assumption is made that the majority of the *Std* class traffic is TCP/IP [MCCREARY], and hence the random early detection (RED) congestion control mechanism is used within the class queue rather than tail drop to ensure that TCP throughput is maximized when congestion occurs; RED tuning is described in Section 3.4. A tail drop queue limit for the *Prm-th* class queue is not needed when RED is used. Some vendors' implementations, however, require that a queue-limit must also be configured; if one is used, and tail drops were to occur rather than RED drops, then the potential benefit of RED would not be realized. To prevent this, if a queue-limit is used in conjunction to RED, the queue-limit should be set sufficiently greater than the RED maximum threshold so that tail drops do not occur.

3.2.2.3.3 Control Class (*Ctrl*)

Traffic is classified into the *Ctrl* class by matching on the DSCP classification, which is assumed to have been pre-marked at the source: either

to DSCP D_c, by network management end-systems for operation, administration and maintenance (OA&M) functions, or to DSCP 48 by routing protocol end-systems. DSCP 48 is used for routing protocol packets, which is equivalent to an IP precedence marking of 6 and which is the de facto marking used by most routers for routing protocol traffic as a result of being the marking originally specified for Internetwork control traffic in RFC 791.

The *Ctrl* class commitments are assured by treating the class with an AF PHB, with a minimal bandwidth assurance of 1% of the link rate, although normally with a minimum of ~8–16 kbps, to ensure that management access to the AR is available even in the presence of congestion of the access-link caused by the customer. A number of applications used for network control and management use TCP (e.g. BGP, SNMP, Telnet), hence RED is used within the *Ctrl* class queue to maximize TCP throughput; RED tuning is described in detail in Section 3.4.

Some designs may choose to engineer the SLA for the control class more precisely. For example, the class bandwidth may be provisioned to ensure that a defined number of routes can be advertised by a particular routing protocol within a specified time period; this effectively provides an SLA for the propagation of routing updates across the access-link.

Care should be taken to ensure that active SLA-monitoring traffic (see Chapter 5, Section 5.3) is not classified into this class because active monitoring traffic should report on the delay, jitter, and loss characteristics of the actual customer-facing class it is monitoring. Consequently, active SLA-monitoring traffic should be classified based on the packets' DSCP marking, which should be the same as for the class it is monitoring.

3.2.2.3.4 Standard Data Class (*Std*)

The configuration in Figure 3.4 assumes that the ordering of the classes within the Diffserv policy implicitly defines an ordering of the classification criteria, i.e. in the case that a packet matches the classification criteria of multiple classes, it will be classified into the

first matching class listed in the policy. Therefore, the wildcard classification criteria used for the *Std* class ensures that all traffic that is not classified into the *VoIP*, *Prm-th*, or *Ctrl* classes (which come first in the Diffserv policy) is classified into the *Std* class. The DSCP of all packets classified into the class is set to D_s.

The *Std* class SLA commitment is assured by treating the class with an AF PHB with a minimum bandwidth assurance of R_s. Assuming that a work-conserving scheduler is used, the *Std* class will have access to all unused interface capacity once the *VoIP*, *Prm-th*, and *Ctrl* data classes have been serviced. RED is used within the class queue to ensure TCP throughput within the class is maximized when congestion occurs; RED tuning is described in Section 3.4.

3.2.2.4 Design Variations

In this section, we consider design variations to the baseline edge design described in the preceding sections.

3.2.2.4.1 Low-speed Edge Design

The main difference between the Diffserv edge designs for low-speed links compared to high-speed links relates to the additional use of link layer fragmentation and interleaving mechanisms on low-speed links, to ensure that the delay SLA for classes with tightly bounded delay commitments, such as the *VoIP* class, can be met.

An edge SLA service usually targets a maximum one-way edge latency for VoIP of 15 to 50 ms, as discussed in Section 3.2.2.1.1. Even where a strict priority scheduler is used to implement an EF PHB for delay-sensitive traffic such as VoIP, however, a newly arrived priority packet must wait for any non-priority packet currently being serviced, before it can be serviced by the scheduler (see Chapter 2, Section 2.2.4.1.1). In practical implementations, there may be additional delay, with several non-priority packets potentially being queued ahead of a priority packet due to the presence of an interface FIFO (see Chapter 2, Section 2.2.4.1.3). The impact of the scheduler and interface FIFO on the delay of a priority packet is more significant for lower-speed access connections. Therefore, we define "low-speed" links as those

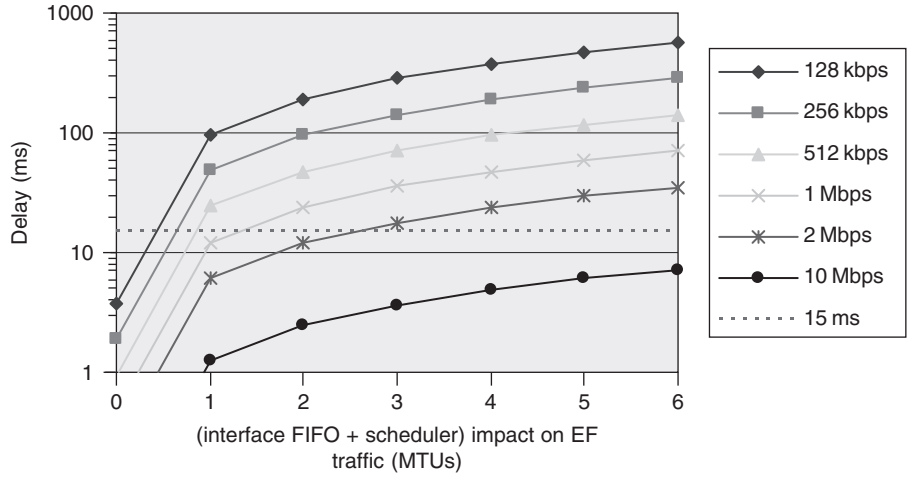

Figure 3.5 Perturbing impact of (Interface FIFO + scheduler) on EF traffic

in which the perturbing impact of non-EF traffic on EF traffic, due to the characteristics of both the scheduler and the interface FIFO, exceeds the latency commitment for the VoIP class, or more generally, the class with the tightest delay commitment. The graph in Figure 3.5 shows the delay of a single 60-byte EF packet (which is the packet size for a G.729A codec with a 20 ms packetization interval) for different rates, with varying impacts due to the interface FIFO and scheduler system, measured in maximum transmission units (MTUs), which are assumed to be 1500 bytes.[1] The actual delay experienced by a VoIP packet may be greater in practice, due to self-induced queuing delay, where there may be multiple VoIP packets in the priority queue, as discussed in Section 3.2.2.8.

As can be seen from Figure 3.5, even where the impact of the interface FIFO and scheduler on EF delay is only 2 * 1500 byte MTUs, which represents a good low-speed implementation in practice, at link speeds of less than 2 Mbps, the perturbing impact on EF delay is sufficient to cause the 15 ms access segment delay budget to be exceeded. In such cases, link layer fragmentation and interleaving mechanisms, such as those provided by Frame-Relay Forum implementation agreement FRF.12 and the multilink point-to-point protocol

(MLPPP) are needed to reduce the impact of non-EF traffic on EF traffic delay. With link-layer fragmentation, large non-EF packets are broken into smaller fragments such that EF packets can be interleaved with the fragments, rather than having to wait for (possibly multiple) whole non-EF packets to be transmitted.

Where link-layer fragment is used, the fragment size F is chosen such that the VoIP latency commitment L_v can be realized. This can be expressed by the following equation:

$$\frac{((T+1) \times F + n \times V)}{link\_rate} < L_v$$

where T is the number of packet buffers in the interface FIFO; $(T + 1)$ accounts for the worst-case scenario of scheduling an AF packet immediately prior to an EF packet; V is the VoIP packet size; n represents the maximum number of concurrent VoIP packets in the VoIP class queue, which accounts for the self-induced queuing delay due to VoIP traffic contention; see Section 3.2.2.8.

For example, with $T = 2$, $n = 1$, and $V = 60$ bytes, with a link rate of 256 kbps and $F = 1500$ bytes the maximum EF latency could be as high as ~143 ms, which consumes most of an 150 ms ear-to-mouth delay budget. Setting the fragmentation size to ~130 bytes decreases the maximum potential EF latency to ~14 ms. Figure 3.6 shows the EF delay versus fragment size for various link speeds where $T = 2$, $n = 1$, and $V = 60$ bytes; as can be seen, even at 128 kbps it may be possible to achieve an EF delay of ~15 ms, albeit with a fragment size of 100 bytes.

Link layer fragmentation and interleaving mechanisms are processor intensive functions that can have an impact on router forwarding performance, hence, in practice for maximum performance the largest possible fragment sizes should be used that can achieve a particular edge segment latency commitment, which is determined by the following formula:

$$F = \frac{(L_v \times link\_rate) - (n \times v)}{(T+1)}$$

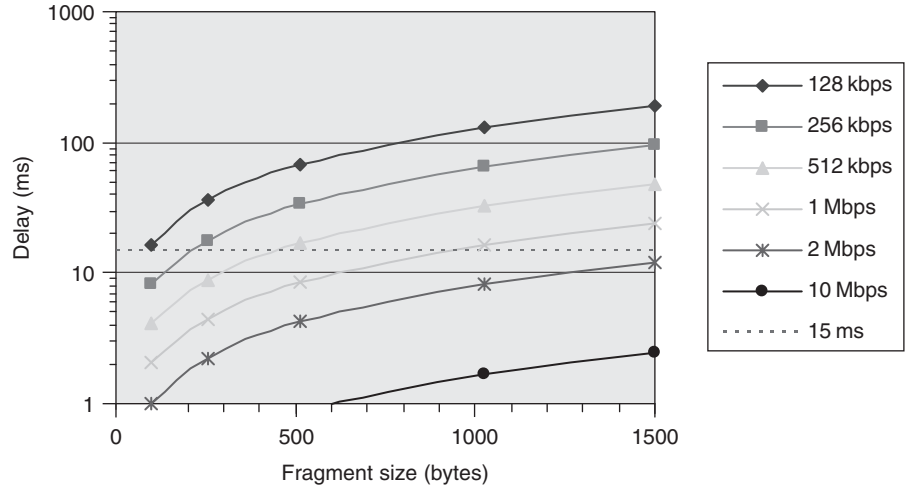

Figure 3.6 EF delay vs fragment size

3.2.2.4.2 Hierarchical Shaping and Scheduling

Distribution routers may potentially aggregate thousands of customer's connections through frame-relay, ATM, TDM (leased line) or metro Ethernet access networks. In the vast majority of deployments, one physical interface will terminate many logical connections using a multiaccess layer 2 technology; each customer site is assigned to one or more virtual links, which would be identified by data link connection identifiers (DLCIs) for frame-relay, virtual circuits (VCs) for ATM, channels for time division multiplexed (TDM) services, or virtual LANs (VLANs) for Ethernet. The distribution and access routers enforce an aggregate rate per-customer site by shaping the virtual link (or links) to a token bucket profile contracted in the SLA to rate R_a and burst B_a. The underlying layer 2 access link virtual connections must guarantee the availability of that traffic profile bidirectionally between the access router and the distribution router with an SLA that is capable of supporting the SLA requirements of the IP services (see Chapter 2, Section 2.5). When the upstream or downstream aggregate offered traffic load is larger than the contracted profile for the access link virtual connections the access router or distribution router shaper delays the packets in the EF/AF scheduler. This creates

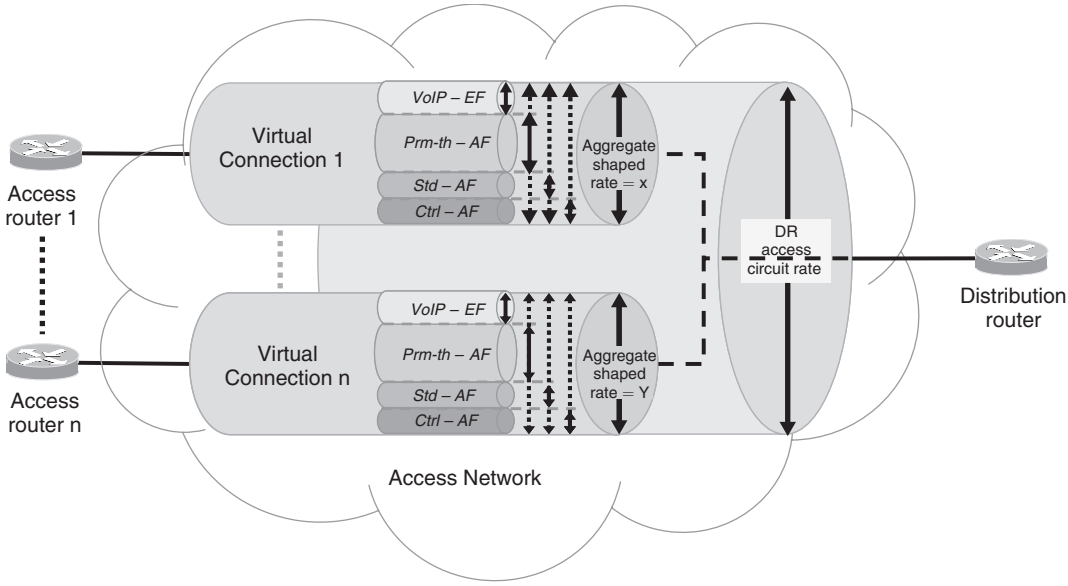

Figure 3.7 Multiaccess DR connectivity: hierarchical shaping and scheduling

a hierarchy where the EF/AF scheduler acts as a child functional block relative to the parent shaper (see Chapter 2, Section 2.4), ensuring that the most important traffic from the customer site gets prioritized access to the committed bandwidth for the logical layer 2 connection. The aggregate rate enforcement serves both to define an aggregate bound on the contract between the SP and the customer, and also to ensure isolation between different customers services which are terminated on the same physical (although different logical) interfaces. This concept is illustrated in the diagram in Figure 3.7.

The configuration in Figure 3.8 augments the design from Section 3.2.2.3 with a parent shaping function, shown in bold text. This configuration would typically be applied outbound on the core facing (i.e. upstream) AR interface and outbound on the access facing (i.e. downstream) interface of the DR. The parent shaping function may need to account for layer 2 overheads, depending upon the layer 2 access technology used, and upon any physical bandwidth constraints (see Chapter 1, Section 1.2.4.1). Where ATM is used as the access

```
policy outbound-shaped-high-speed-edge-policy
  class shaper
    classify *
    shape (Ra, Ba)
      {
       class VoIP
         classify DSCP (Dv)
         SR-TCM (Rv, Bv, 0)
           green-action transmit
           red-action drop
         EF
       class Prm-th
         classify <criteria>
         AF (Rt)
         mark DSCP (Dt)
        RED (DSCP(*), <minth>, <maxth>, <w>, <pmax>)
       class Ctrl
         classify DSCP {Dc, 48}
         AF (Rc)
        RED (DSCP(*), <minth>, <maxth>, <w>, <pmax>)
       class Std
         classify *
         AF (Rs)
         mark DSCP (Ds)
        RED (DSCP(*), <minth>, <maxth>, <w>, <pmax>)
      }
```

Figure 3.8 High-speed edge design with aggregate shaper

network technology, for example, the shaping function will need to take into account the ATM overheads, to ensure that the contracted profile for the ATM VC is not exceeded.

3.2.2.4.3 Unmanaged Access Router Services

The designs discussed so far have been in the context of "managed access router" services – that is, where the network provider (commonly

a VPN provider) owns and manages the access router device, in addition to the core and distribution routers, and consequently commits to the SLA end-to-end from access router-to-access router. With *unmanaged access router* service offerings, the network provider owns and manages the core and distribution routers, but does not own and manage the access router. This option is attractive as a wholesale service offering where a network SP supplies a lowest common denominator service to systems integrators who add their own access router to the service, and offer customized access router configurations. Unmanaged access router services are also attractive for end customers who wish to maintain control of the access router.

There are two major differences with respect to the deployment of unmanaged access router services, when compared to managed service offerings. The first is that as the access router is neither owned nor managed by the network SP, they cannot ensure that the correct configuration and management is applied to the access router to be able to commit to an SLA from the access router to the distribution router. Secondly, in order to protect their network from access routers misconfiguration, the network SP may now need to perform inbound on the access facing (i.e. downstream) interface of the DR, the complex per-customer classification and conditioning functions, in terms of rate enforcement, which were distributed to access routers in the managed service context. Conceptually, the hierarchical shaping with queuing functionality, equivalent to that described in Section 3.2.2.4.2, may be applied on ingress to the DR also; however, this functionality is not widely supported on ingress by network equipment vendor's devices today. Alternatively, an example Diffserv conditioning policy, which would be applied inbound on the access facing (i.e. downstream) interface of the DR to the upstream traffic, is shown in Figure 3.9.

In this case, the classification criteria are the same as that used in the managed AR service; however, if it is assumed that the access router is correctly classifying and marking traffic, simple classification based upon the DSCP marking set by the access router may be used instead.

The policy shown in Figure 3.9 uses two instances of the SR-TCM to condition the traffic received by the distribution router: one

```
policy unmanaged-inbound-dr-policy
  class aggregate
    classify *
      SR-TCM (Ra, Ba, 0)
        green-action transmit
        red-action drop
      {
       class VoIP
         classify DSCP (Dv)
         SR-TCM (Rv, Bv, 0)
           green-action transmit
           red-action drop
       class Prm-th
         classify <criteria>
         mark DSCP (Dt)
       class Ctrl
         classify DSCP {Dc, 48}
       class Std
         classify *
         mark DSCP (Ds)
      }
```

Figure 3.9 Unmanaged AR service – ingress upstream distribution router policy

enforces an aggregate rate R_a for the received traffic, while the other enforces a rate R_v, for the received *VoIP* class traffic. Note that the use of two policing instances in this way has a fundamentally different effect than the shaping/scheduling hierarchy used in Section 3.2.2.4.2. Assuming that the two SR-TCM instances operate independently, if the rate of one of the policers is exceeded, then the resulting behavior is dependent upon the order in which the policers are applied. If the aggregate policer is applied first then *VoIP* traffic may pass the aggregate policer but may be dropped by the *VoIP* class policer, resulting in underutilization of the aggregate rate. If the VoIP class policer is applied first then *VoIP* traffic may pass the class policer but may be

dropped by the *aggregate* policer, resulting in underutilization of the *VoIP* class. Hence, Diffserv policies are generally only applied in this way to protect the SP network from misconfiguration of the AR; if the customer owning the AR has configured the correct policy on the AR (i.e. in this case as described in Section 3.2.2.4.2), this inbound policy on the access facing (i.e. downstream) interface of the DR will have no effect.

3.2.2.4.4 Add Premium Data Delay-optimized Class (*Prm-delay*)

The premium data throughput-optimized class (*Prm-th*) did not support an SLA for delay or loss because the maximum rate for the class was not enforced with a policer and hence the actual class delay and loss are dependent upon the customer's offered traffic profile for that class, which is outside of the SP's control. Therefore, to offer a premium data delay-optimized class (*Prm-delay*) with a defined SLA for loss and latency, the maximum rate and burst for the class must be enforced with a policer. Such a *Prm-delay* class targets business-critical interactive applications with delay requirements such as SNA, SAP R/3, Telnet, and market trading data feed applications.

3.2.2.4.4.1 **Prm-delay *SLA*** The *Prm-delay* SLA is defined in terms of a committed delay and loss rate, with a specified bandwidth and availability. Attainable throughput is derived from the loss rate. Jitter is not important for this service class and is not defined. The class may support a commitment for per flow sequence preservation.

As with the *VoIP* class, the SP and the customer agree to a contract with a defined ICR and ECR, which are specified symmetrically in this case study (that is, ICR = ECR), although that need not necessarily be the case. The SP enforces the contract by limiting the rate of *Prm-delay* traffic to/from the customer site using a token bucket policer of rate R_d and burst B_d; non-conformant traffic in excess of the policer will be dropped by the SP. The rate R_d will be selected by the customer and will be offered by the SP up to a defined percentage of the access link speed. As for the VoIP class, SPs will not offer the *Prm-delay* class below a defined minimum link speed, and the value of B_d is set based on the offered class delay commitment.

For conformant traffic, the SP will commit to a maximum one-way edge segment latency, L_d, typically in the range 30–80 ms (for a defined packet size) and an end-to-end loss rate of typically less than 0.1%.

The contract will also define the classification criteria that the SP will use to identify the class and stipulate that conformant traffic will be marked with a defined DSCP value D_d.

3.2.2.4.4.2 **Prm-delay *Design*** The configuration in Figure 3.10 augments the design from Section 3.2.2.3 with a *Prm-delay* class, which is highlighted in bold text. This configuration would typically be applied outbound on the core facing (i.e. upstream) AR interface and outbound on the access facing (i.e. downstream) interface of the DR.

Considering the *Prm-delay* class Diffserv configuration in Figure 3.10, as per the *Prm-th* class, it is assumed that complex classification criteria (unspecified in the figure) are used to classify packets into the class. The DSCP of all packets classified into the class is set to D_d.

The *Prm-delay* class latency commitment is met by treating the class with an AF per-hop behavior (PHB), which provides a minimum class bandwidth assurance R_d, and by enforcing an average arrival rate of R_d with a policer, such that the arrival rate does not exceed the servicing rate for the class. This could be achieved using the SR-TCM with rate R_d, committed burst size B_d, with EBS = 0 and applying a green action of transmit and mark DSCP and a red action of drop. The SP specifies the SLA contract with the policer's worst-case admitted burst B_d, such that the burst is serviced within the class latency commitment L_d, i.e. $B_d/R_d + L_s < L_d$, where L_s represents the worst-case delay to service the AF traffic (which would be greater than for EF traffic) due to the scheduler and the interface FIFO (see Chapter 2, Section 2.2.4.1.3). A tail drop queue limit is not required for the *Prm-delay* class queue, in addition to the policer.

If a router implementation requires that a queue-limit is configured, the queue-limit must be greater than or equal to B_d, to ensure that packets that are within the permitted burst for the class are not dropped. Therefore, the only actual packet loss that can occur for conformant traffic within the class is due to layer 1 bit errors or network

```
policy high-speed-edge-policy
  class VoIP
    classify DSCP (Dv)
    SR-TCM (Rv, Bv, 0)
      green-action transmit
      red-action drop
    EF
  class Prm-delay
    classify <criteria>
    SR-TCM (Rd, Bd, 0)
      green-action transmit-and-mark DSCP (Dd)
      red-action drop
    AF (Rd)
  class Prm-th
    classify <criteria>
    AF (Rt)
    mark DSCP (Dt)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
  class Ctrl
    classify DSCP {Dc, 48}
    AF (Rc)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
  class Std
    classify *
    AF (Rs)
    mark DSCP (Ds)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
```

Figure 3.10 High-speed edge design with *Prm-delay* class

element failures, which account for the loss rate for the class offered by the SP.

An alternative design is possible using advanced scheduler implementations, which provide support for more than one priority queue as described in Chapter 2, Section 2.2.4.1.4. This would allow the possibility of using the highest priority queue for VoIP class, for

example, and the next priority queue for *Prm-delay* traffic, enabling the VoIP class traffic to receive the lowest delay and jitter, while the *Prm-delay* traffic would have a bounded delay and jitter, independent of the scheduler implementation and the configuration and load of the other AF queues.

3.2.2.4.5 Add Premium Data Throughput-optimized Class with Loss Commitment (*Prm-loss*)

The premium data delay-optimized class (*Prm-delay*) added a policer to the premium Data throughput-optimized class (*Prm-th*) configuration to enforce the maximum rate and burst for the class, dropping excess such that an SLA for loss and delay could be supported. A consequence of the way the policer was used with the *Prm-delay* class, however, is that the class was unable to re-use unused bandwidth from other classes within the same policy. Therefore, to offer a premium data throughput-optimized class which has a loss commitment (*Prm-loss*) and which also has the ability to re-use unused bandwidth from other classes within the same policy, a policer is instead applied to mark a certain amount of a traffic class as in-contract, and everything above that as out-of-contract; the SLA for loss is committed for the in-contract traffic only.

3.2.2.4.5.1 **Prm-loss *SLA*** The *Prm-loss* SLA is defined in terms of a committed loss rate, with a specified bandwidth and availability. Attainable throughput for TCP-based applications is derived from the loss rate and RTT. Jitter is not important for this service class and is not defined. The class may support a commitment for per flow sequence preservation. Such a class targets the same applications as the *Prm-th* class, but where an explicit loss commitment is required. The practical differences between the *Prm-loss* class and the *Prm-th* class are therefore really in terms of the way that the SLA is exposed to the end-customer; this is most commonly provided in the context of services offered by a network service provider, rather than in the case of an intra-enterprise network.

As with the *VoIP* class, the SP and the customer agree to a contract with a defined ICR and ECR; in this case study ICR = ECR. The SP

enforces the contract by limiting the rate of *Prm-loss* traffic to/from the customer site using a token bucket policer of rate R_l and burst B_l; non-conformant traffic will be marked as out-of-contract by the SP. The rate R_l will be selected by the customer and will be offered by the SP possibly up to the access link speed.

For conformant (in-contract) traffic, the SP will commit to an end-to-end loss rate of typically less than 0.1%.

The contract will also define the classification criteria that the SP will use to identify the class and stipulate that conformant traffic within the class be marked with a defined DSCP value D_{lin} and non-conformant traffic will be marked with a defined DSCP value D_{lout}.

In practice, if this class was offered as a service, it would likely be in lieu of the *Prm-th* class, i.e. there is little point in offering both a *Prm-loss* and a *Prm-th* class.

3.2.2.4.5.2 **Prm-loss *Design*** The configuration in Figure 3.10 enhances the design from Section 3.2.2.3 with a *Prm-loss* class, which is shown highlighted in bold, instead of the *Prm-th* class. This configuration would typically be applied outbound on the core facing (i.e. upstream) AR interface. A similar policy may be applied outbound on the access facing (i.e. downstream) interface of the DR, only without the class policer, as the in- and out-of-contract conditioning is performed on ingress to the network.

Considering the *Prm-loss class* Diffserv configuration in Figure 3.11, as per the *Prm-th* class it is assumed that complex classification criteria (unspecified in the figure) are used to classify packets into the class.

The *Prm-loss* class loss commitment is met by treating the class with an AF per-hop behavior (PHB), which provides a minimum class bandwidth assurance R_l, and by enforcing an average arrival rate for in-contract traffic of R_l with a policer, such that the in-contract arrival rate does not exceed the servicing rate for the class. This could be achieved using the SR-TCM with rate R_l, committed burst size B_l, with EBS = 0 and applying a green action of transmit-and-mark DSCP D_{lin} and a red action of transmit-and-mark DSCP D_{lout}. Assuming that most of the traffic within the class is TCP-based, B_l, should be set to

```
policy high-speed-edge-policy
  class VoIP
    classify DSCP (Dv)
    SR-TCM (Rv, Bv, 0)
      green-action transmit
      red-action drop
    EF
  class Prm-loss
    classify <criteria>
    SR-TCM (R1, B1, 0)
      green-action transmit-and-mark DSCP (Dlin)
      red-action transmit-and-mark DSCP (Dlout)
    AF (R1)
    RED (Dlin, <minth>, <maxth>, <w>, <pmax>)
    RED (Dlout, <minth>, <maxth>, <w>, <pmax>)
  class Ctrl
    classify DSCP {Dc, 48}
    AF (Rc)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
  class Std
    classify *
    AF (Rs)
    mark DSCP (Ds)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
```

Figure 3.11 High-speed edge design with *Prm-loss* class

be sympathetic to the behavior of TCP. If B_l is set too small, due to the bursty nature of TCP, most traffic would be marked as out-of-contract, and the rate commitment for the class may not be achieved in practice. If B_l is set too large, most traffic would be marked as in-contract, and the loss commitment for the class may not be achieved in practice. Most TCP tuning focuses on the concept of the TCP "flight size" [RFC2581] or the "pipesize," which is the amount of unacknowledged

data in flight between the TCP sender and the receiver, and is defined by the bottleneck bandwidth * RTT product. For low-speed access with small numbers of TCP flows, setting B_l, equal to R_l * RTT should ensure that the rate and loss commitment for the class can be met for TCP traffic. At higher speeds, as more flows are aggregated, lower values of B_l may be acceptable due to the effect of statistical multiplexing.

Weighted RED (WRED) is then applied to the class queue such that if there is congestion in the queue, then out-of-contract traffic will be preferentially discarded over in-contract traffic. This is achieved by having a more aggressive WRED profile (lower q_{minth} and q_{maxth} settings) for the out-of contract traffic than for the in-contract traffic (as described in Chapter 2, Section 2.2.4.2.4) and by ensuring that sufficient capacity is provisioned to service the in-contract traffic. The WRED profile for in-contract traffic is set such that the in-contract traffic should not be dropped in normal network operation; it may therefore seem to be redundant; however, it is included in order to maximize throughput for in-contract TCP traffic in the presence of unforeseen in-contract load, due to network element failures, for example. Hence, in normal operation, the only actual in-contract packet loss that should occur is due to layer 1 bit errors or network element failures, which account for the loss rate for the class offered by the SP. WRED tuning is described in Section 3.4.

The TR-TCM may be used as an alternative to the SR-TCM, to mark a certain amount of a traffic class as in-contract, and everything above that as out-of-contract, but up to a maximum rate, with a green action of transmit-and-mark DSCP D_{lin} and a yellow action of transmit-and-mark DSCP D_{lout}, and a red action of drop. Applied in this way the TR-TCM would enforce a maximum rate of CIR and a burst of CBS on the traffic stream; any traffic in excess would then be marked out-of-contract up to a maximum rate of PIR and a burst of PBS.

3.2.2.4.6 Add a Video Class (Video)

As discussed in Chapter 1, Section 1.3.2.1.1, worst-case one-way network delays of 100–200 ms are typically targeted for streaming video applications. If there were a requirement to add support for a video

service class there are potentially a number of designs that could be applied to support such a class.

- *Service video from same class as VoIP.* It may be possible to service voice and video from the same class queue, using a design such as that described for the real-time class in Section 3.2.2.3.1. In support of such designs, some vendors have implementations that allow multiple subsets of traffic within a class to be discretely policed to limit the worst-case impact that the subsets of traffic have on each other. Nonetheless, if voice and video are serviced from the same priority queue, large video packets can increase the delay and jitter experienced by the voice traffic; this effect can be significant on low-speed links, but may be acceptable for higher speed access links.
- *Distinct video class using AF queue.* A video service class could be supported using an AF PHB, as per the *Prm–delay* class described in Section 3.2.2.4.4. With this approach, the delay bound that the video traffic can experience may vary depending upon the scheduling algorithm used, may also be dependent upon the number of other AF queues used in the particular implementation and the traffic in those queues, and at low-link speeds it may not be possible to achieve the required delay targets with this approach.
- *Multi-level priority scheduler.* Some advanced scheduler implementations provide support for more than one priority queue as described in Chapter 2, Section 2.2.4.1.4. When supporting voice and video services concurrently, for example, using the highest priority queue for voice traffic, and the next priority queue for video traffic would enable the voice traffic to receive the lowest delay and jitter, while the video traffic would have bounded delay and jitter, independent of the scheduler implementation and the configuration and load of the other AF queues.

Which option is chosen in practice will likely depend upon the nature of the service being offered, and the capabilities of the network equipment used.

Class	Maximum rate (and burst)	Minimum bandwidth assurance	Delay	Loss	DSCP	See Section
VoIP	R_v (B_v)	R_v	L_v	Typically ~0.1%	D_v	3.2.2.3.1
Prm-th	X	R_t	n/a	n/a	D_t	3.2.2.1.2
Std	X	R_s	n/a	n/a	D_s	3.2.2.1.3
Ctrl	X	1%	n/a	n/a	D_c, 48	3.2.2.1.4
Prm-delay	R_d (B_d)	R_d	L_d	Typically ~0.1%	D_d	3.2.2.4.4
Prm-loss: In-contract	R_l (B_l)	R_l	n/a	Typically ~0.1%	D_{lin}	3.2.2.4.5
Out-of-contract	X	n/a	n/a	n/a	D_{lout}	

Figure 3.12 Edge SLA summary

3.2.2.5 Edge SLA Summary

The edge SLA commitments for the different classes described in the preceding sections are summarized in the table in Figure 3.12.

3.2.2.6 How Many Classes are Enough?

At the network's edge, different classes are used for both SLA differentiation and for providing isolation between applications, i.e. ensuring that the behavior of one application cannot impact the SLA committed to another. In the preceding sections, we have described the designs for service classes in support of applications with a number of different SLA requirements. To provide isolation between applications, there may be a requirement to have multiple instances of the same class even though the classes are supporting applications with the same SLA requirements. This leads to the question: how many classes is enough for an edge Diffserv design?

Several efforts in standardization bodies including ITU-T Recommendation G.1010 [G.1010] and [RFC4594] have attempted to categorize applications into QOS services classes based upon their SLA requirements. G.1010 identifies four different categories for user traffic: interactive, responsive, timely, and non-critical; these classes loosely map to the *VoIP, Video, Prm-delay*, and *Std* classes previously defined. [RFC4594] identifies twelve service class categories, together with

recommended traffic forwarding treatments and codepoints for each service class; however, in practice, typical edge Diffserv designs today have less than eight classes [CARTER]. Further, it is not obvious that the requirements for edge classes will naturally increase over time; at some point, adding more classes provides a diminishing return, where the added design and management complexity exceeds the benefit provided. We recommend applying the maxim that the number of classes supported "*should not be multiplied beyond necessity*"[2] [RFC4594] acknowledges: "*[it] is expected that network operators will configure and provide in their networks a subset of the defined service classes*." Hence, the table in Figure 3.13 summarizes the service class categories from [RFC4594] and maps them to the different class designs defined in the previous sections.

There is no general answer to the question of how many classes should be supported in a particular edge Diffserv design; the answer is dependent upon the specific requirements. Some of the main considerations that will drive the number of classes supported are as follows.

- *SLA differentiation*. The need to differentiate SLAs between applications is the primary driver for supporting additional classes. Clearly, there is no need to support a class if there are no applications that will use that class in a particular deployment. Additional classes can be added incrementally as the need arises for them.
- *What level of application isolation is required?* There may be a requirement to deploy different classes for reasons of application isolation rather than SLA differentiation; this may require that multiple classes are deployed supporting the same SLA.
- *Separate classes for different real-time applications?* As discussed in Section 3.2.2.4.6, it may be possible to service different real-time applications, such as voice and video, from the same class, thereby reducing the number of classes needed.
- *Separate classes for routing protocol and management traffic?* Some network designs may choose to use separate classes for routing

Service class name	Example applications	Codepoint	Class design used
Network control	Routing protocols	CS6	*Ctrl* (Section 3.2.2.3.3)
Telephony	IP telephony bearer	EF	*VoIP* (Section 3.2.2.3.1)
Signaling	IP telephony signaling	CS5	*VoIP* (Section 3.2.2.3.1)
Multimedia conferencing	H.323/V2 video conferencing (adaptive)	AF41 AF42 AF43	*Video* (Section 3.2.2.4.6)
Real-time interactive	Video conferencing and Interactive gaming	CS4	*Prm-delay* (Section 3.2.2.4.4)
Multimedia streaming	Streaming video and audio on demand	AF31 AF32 AF33	*Video* (Section 3.2.2.4.6)
Broadcast video	Broadcast TV	CS3	*Video* (Section 3.2.2.4.6)
Low-latency data	Client/server transactions Web-based ordering	AF21 AF22 AF23	*Prm-delay* (Section 3.2.2.4.4)
OAM	Operations, administration & maintenance traffic	CS2	*Ctrl* (Section 3.2.2.3.3)
High throughput data	Store and forward applications	AF11 AF12 AF13	*Prm-th* (Section 3.2.2.1.2) or *Prm-loss* (Section 3.2.2.4.5.1)
Standard	Undifferentiated applications	DF	*Std* (Section 3.2.2.3.4)
Low priority data	Any flow that has no BW assurance	CS1	*Std* (Section 3.2.2.3.4)

Figure 3.13 Traffic service classes summary from [RFC4594]

protocol and management traffic, although in many cases a single edge class is sufficient for this purpose.

- *Separate classes for bearer and signaling?* For applications such as voice and video, there is a choice as to whether application signaling traffic uses the same class as the bearer (media) traffic or is potentially allocated its own class, as discussed in Section 3.2.2.3.1.

3.2.2.7 What Marking Scheme to Use?

Once a decision is made on the number of classes that will be supported, the DSCP marking scheme that will be used to distinctly identify different classes and different drop precedences needs to be decided; a unique codepoint needs to be assigned to each drop precedence within each class.

Although there are recommended codepoints for the EF, AF and default PHBs (see Chapter 2, Section 2.3.4.1), the use of these codepoints is not mandated, which means that while it may make sense to use these values, if there are valid reasons to use DSCP values other than those recommended, then it is up to the network designer's discretion to do so. There is a perception that using recommended marking schemes will facilitate interworking between networks; however, this tends not to be a significant benefit in practice, and the ability to re-mark between different marking schemes at a boundary router is common functionality on routers today. Further, even though two networks may use common DSCP markings, if their SLAs are not aligned for each respective DSCP marking, there may still be a need to re-mark traffic at the network boundary.

There are a number of reasons why markings other than the recommended DSCP markings may be used in practice:

- The use of first three bits of the DS field – which are functionally equivalent to IP precedence bits (see Chapter 2, Section 2.3.2) – can facilitate and simplify mapping of DSCP markings from/to the MPLS Experimental field (see Chapter 2, Section 2.3.6.2), the IEEE 802.1q user_priority field (see Chapter 2, Section 2.5.3), and 802.17 (resilient packet ring), all of which also support 3-bit marking schemes only.
- A particular design may require more than the one EF and four AF classes for which recommended codepoints are specified.
- Some markings, which are not the recommended markings, have become de facto for some applications, and hence using these de facto markings can simplify a design by not requiring that the application's traffic to be re-marked.

When defining a marking scheme for a particular design, there are a few simple rules which can help to avoid pitfalls:

- Only use DSCP 48 for network control traffic such as routing protocols. DSCP 48 is equivalent to an IP precedence value of 6 (i.e. network control), which is the de facto markings used by networking equipment vendors for network control traffic.
- Use DSCP 46 for VoIP traffic, which is the recommended codepoint for EF traffic and is equivalent to an IP precedence value of 5 (i.e. critical), as this is the de facto marking used by VoIP end-system vendors.
- Use DSCP 0 for the "standard class," assuming that this is the majority of traffic, as this avoids unnecessarily re-marking most traffic.

[RFC4594] suggests recommended codepoints for the twelve service classes that it identifies, as shown in Figure 3.13, with the intent of fostering interoperability through consistency between different deployments. It remains to be seen whether the recommendations from [BARBIARZ1] will be widely adopted in networks and by application end-systems; however, if they are, then using this scheme may result in a simplification of designs.

A possible edge-marking scheme, for the different class designs defined in Sections 3.2.2.3 and 3.2.2.4, which aligns with the recommended codepoints in [RFC4594], is given in Figure 3.14.

As the *Prm-loss* and a *Prm-th* classes would not both be supported at the same time, AF11 is used for both classes. Considerations on mapping this edge-marking scheme to a core-marking scheme are provided in Section 3.3.2.5.

3.2.2.8 VoIP – How Much is Enough at the Edge?

In the definition for the *VoIP* class, it was specified that the SP will normally limit the maximum percentage of the access link bandwidth that is available for this class; the reason for applying this limit is to

Edge class		Edge DSCP marking	
VoIP		D_v	= 46 (EF)
Prm-th		D_p	= 10 (AF11)
Std		D_s	= 0
Ctrl:	Routing protocols		= 48 (CS6)
	OA&M	D_c	= 16 (CS2)
Prm-delay		D_d	= 18 (AF21)
Prm-loss:	In-contract	D_{Iout}	= 10 (AF11)
	Out-of-contract	D_{Iout}	= 12 (AF12)

Figure 3.14 Edge-marking scheme

ensure that the delay commitment for the class can be met. Even though the VoIP class may be serviced with an EF PHB implemented using a strict priority queue, there is a possibility that when a packet within that class arrives, there may already be a packet (or packets) from that class in the VoIP class queue waiting to be serviced. As the VoIP class load increases, the probability that there are other packets in the queue increases, and hence the probability of that packet experiencing higher queuing delays, also increases. At some class load, there will be a significant probability that this delay due to VoIP traffic contention, i.e. the self-induced VoIP queuing delay, will be sufficient that the access segment latency commitment is exceeded. Hence, the percentage of the access link bandwidth that is available for this class is limited to ensure that the delay commitment for the class can be met. There are various commonly cited rules of thumb for what this percentage should be, including 30%, 33%, and 50%; however, these are generalizations, which are generally incorrect and in this section we consider the factors that affect the maximum percentage of VoIP traffic supported on an access link in practice.

For traffic which obeys a random packet arrival distribution (i.e. a Poisson distribution), queuing theory can be used to determine the queuing delay due to VoIP traffic contention, i.e. the delay due to the fact that traffic from different calls can arrive at the same time. In this

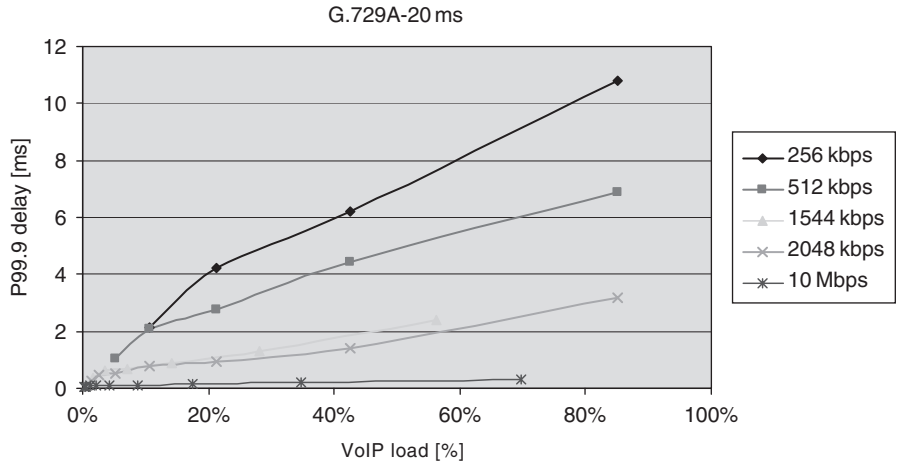

Figure 3.15 Variation of self-induced VoIP queuing delay with rate, assuming a random call arrival distribution

case, the self-induced queuing delay due to VoIP IP traffic contention is a function of two factors:

- the percentage of link utilization due to VoIP traffic
- The VoIP packet size. The serialization delay of a VoIP packet is dependent on the packet size, which in turn is dependent upon the VoIP CODEC used and the packetization interval.

The graph in Figure 3.15 was produced by simulation, where the VoIP self-induced queuing delay was measured when multiple simultaneous calls were established over links of different sizes. The simulation used a random call arrival distribution, with all calls using a G.729A codec (without silence suppression), with a packetization interval of 20 ms (which results in 50 packets per second per call), and a VoIP packet size of 60 bytes. Figure 3.15 shows the variation of the 99.9 percentile VoIP self-induced queuing delay with different levels of VoIP load relative to the access-link rate (i.e. assuming VoIP traffic is serviced from a strict priority queue) for different link rates.

As can be seen from Figure 3.15, for a given codec and packetization interval, the self-induced VoIP queuing delay increases as the

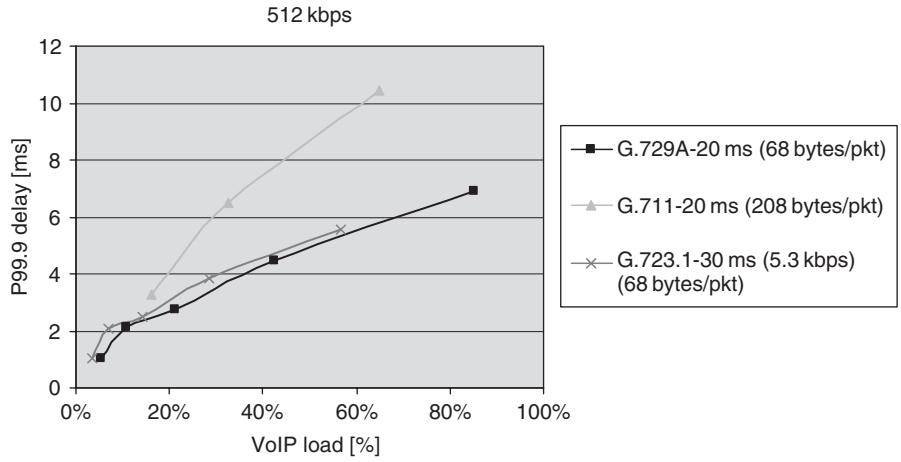

Figure 3.16 Variation of VoIP self-induced queuing delay with codec, assuming a random call arrival distribution

average VoIP traffic increases as a percentage of the link rate. For a given average VoIP traffic load as a percentage of link rate, the self-induced VoIP queuing delay decreases as link rate increases. The graph in Figure 3.16, which is also from simulation, shows the variation of 99.9 percentile VoIP self-induced queuing delay with different codecs (all without silence suppression), for different average VoIP loads, at a link rate of 512 kbps and assuming a random call arrival distribution.

As can be seen from Figure 3.16, for a given percentage of the VoIP traffic average load, the self-induced VoIP queuing delays increase for higher bit rate codecs and for larger packetization intervals.

Therefore, the maximum percentage of VoIP traffic that could be supported on the access link for a particular design is dependent upon the access-link delay budget, from which the worst-case delay through an empty priority queue (i.e. the worst case due to the impact of the interface FIFO and scheduler on EF delay as discussed in Section 3.2.2.4.1, excluding the delay due to VOIP traffic contention itself) should be deducted. The remainder is the delay budget available to VoIP traffic contention, which determines the maximum percentage of VoIP traffic that could be supported on the access link. For example, consider a case where a VoIP class is offered with an access segment

delay commitment of 15 ms, for a 2 Mbps link; if the worst-case delay through an empty priority queue is 2 MTUs due to the impact of the scheduler and the interface FIFO, i.e. 12 ms as per Figure 3.5, then this leaves 3 ms of delay budget available to VoIP traffic contention itself. If the VoIP traffic uses a G.729A codec with a 20 ms packetization interval, then from Figure 3.15 the VoIP class load would need to be less than ~85% of the link rate (i.e. supporting a maximum of ~64 simultaneous calls), to ensure that the access segment delay commitment could still be met.

The previous discussion is valid assuming a random (i.e. Poisson) arrival distribution, which may be valid for a VoIP gateway connected directly to an AR, for example, as telephony call arrivals are well modeled with a Poisson process. If the VoIP traffic has been through a number of levels of aggregation, however, the traffic may no longer follow a random arrival distribution and the previous guidance may not apply. In this case, at the edge of the network, the only options to determine the maximum acceptable VoIP load are either to determine the actual traffic distribution, or to assume worst-case analysis, e.g. that packets from all concurrent calls may arrive at the same time.

The question of how much VoIP traffic load can be supported within the core of the network is discussed in Chapter 6, Section 6.1.3.

3.3 Deploying Diffserv in the Network Backbone

Even where Diffserv is deployed at the edge of the network, there is still a choice as to whether Diffserv is deployed in the network core or not.

3.3.1 Is Diffserv Needed in the Backbone?

Unlike at the network edge where bandwidths are lower, in the backbone where there are high bandwidth links and traffic is highly aggregated, SLA requirements for a traffic class can be translated into

bandwidth requirements, and the problem of SLA assurance can effectively be reduced to that of bandwidth provisioning. With the aid of a suitable capacity planning process, as discussed in Chapter 6, Section 6.1, designing an IP backbone network to ensure that commitments for low delay, low jitter and low loss can be met can be relatively simple; one simply needs to over-provision the bandwidth compared to the measured average load.

For example, from Chapter 6, Figure 6.5, if we assume that Diffserv is not deployed in the core network and want to achieve a target P99.9 queuing delay of 1 ms to support VoIP traffic on a 155 Mbps link, then the provisioned link bandwidth should be at least 2 times the 5-minute average link utilization. This means that the average link utilization should not be higher than approximately 50% of the link capacity or ~77 Mbps, even if the VoIP traffic contributes a relatively small proportion to the aggregate link load. This approach would actually ensure low-delay, low-jitter, and low loss service for all traffic on the link, because with only a single service class, by definition there is no differentiation between different types of traffic. Without core QOS mechanisms, the network may be simpler to design; however, this benefit comes with the cost of aggregate over-provisioning of the core bandwidth. In addition, if the core capacity planning is not accurate or is not reactive enough to new traffic demands, or in the presence of denial of service attacks or network failure conditions, there may be instances when congestion is unavoidable. In cases such as these, without Diffserv, there would be no differentiation between premium and standard services and in unforeseen congestion all traffic will share the same fate and all services will be impacted.

Diffserv provides a solution to this problem, in that it allows per-class virtual backbones to be built on a single physical backbone. Diffserv simply extends the concept of over-provisioning to multiple classes, giving designers the flexibility to have different over-provisioning factors (the ratio of offered load to available capacity) for each service class, thereby providing SLA differentiation and making more efficient use of network capacity. Using the previous example, this could allow the VoIP class capacity to be over-provisioned by a factor of at least 2 relative to the average class load, hence ensuring

that the class receives low-delay, low-jitter and low-loss service, while the aggregate capacity could be over-provisioned by a lower factor, such as 1.2, which is a realistic figure still giving good service. This would result in a bandwidth saving over the non-Diffserv case. In practice as core network links are provisioned in bandwidth increments, this may result in one lower bandwidth increment or one less link being required, which clearly has an associated cost saving. Diffserv also provides isolation between different services classes; in unforeseen congestion; different services no longer need to share the same fate as Diffserv ensures that issues in one service class are isolated from impacting other classes.

Consideration obviously needs to be given to whether the cost of deploying Diffserv outweighs the benefits it provides. There is no generic answer to this question, and the benefits that will be gained will vary deployment by deployment. In the previous example, if the Diffserv deployment cost exceeds the cost of the bandwidth saved (and the router ports, which may be required to terminate that bandwidth) then there may be no sense in deploying Diffserv. For most practical deployments, the maximum potential economic benefit stands to be gained from Diffserv where the traffic requiring the highest SLA targets represents a minor proportion of the overall capacity. As in the previous example, the absence of Diffserv leads the designer to provision capacity equal to the aggregate load across all classes multiplied by the over-provisioning ratio of the class with the tightest SLA; this can be extremely expensive when the tightest-SLA class represents a low proportion of the aggregate traffic. Conversely, when all classes require the same level of service, and hence the same over-provisioning factor, there is no benefit to be gained from deploying Diffserv in the network core. Considering two classes with loads C_1 and C_2, and with over-provisioning factors of OP_1 and OP_2 respectively, the link bandwidth required without Diffserv would be:

$$NoDiffserv\_bandwidth = MAX((OP_1,OP_2) \times (C_1 + C_2))$$

In comparison, with Diffserv, assuming that a work-conserving scheduler is used where class C_1 is serviced with an EF PHB, from a strict

priority queue, and class C_2 is serviced with an AF PHB, from a weighted bandwidth queue, the link bandwidth required with Diffserv would be:

$$Diffserv\_bandwidth = MAX((C_1 \times OP_1), (C_1 + (C_2 \times OP_2)))$$

The first term of this formula is to ensure that the over-provisioning factor for C_1 is met; the class is serviced from a strict priority queue, and therefore is serviced at the interface rate, hence the interface bandwidth needs to be at least C_1 * OP_1. The second term is to ensure that the over-provisioning factor for C_2 is met; as the scheduler is work conserving, this class is able to re-use all unused bandwidth after class C1 has been serviced, therefore the interface bandwidth needs to be at least $C_1 + (C_2 * OP_2)$.

The benefit of Diffserv can be realized either in terms of less bandwidth being required per link to achieve the same SLAs when compared to the non-Diffserv case, or in more aggregate traffic being supported for the same provisioned bandwidth as the non-Diffserv case. Figure 3.17 illustrates the Diffserv bandwidth gain expressed as the bandwidth required without Diffserv divided by the bandwidth required with Diffserv, to achieve the same SLAs, for different relative

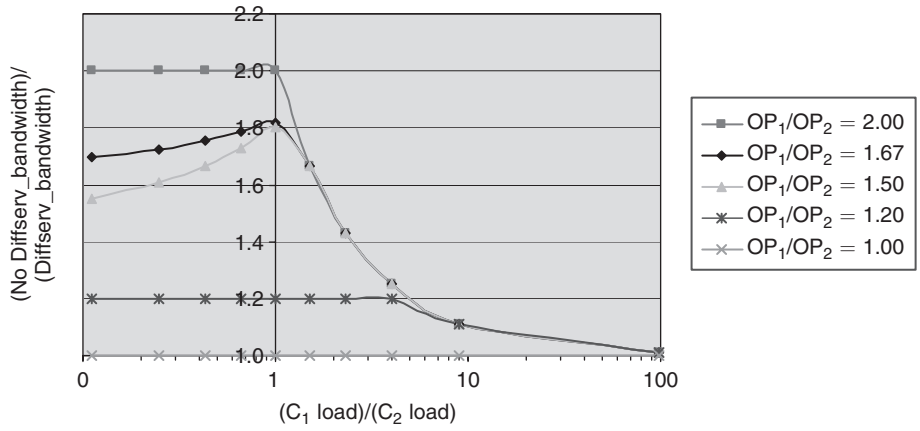

Figure 3.17 Diffserv bandwidth gain

loads of two traffic classes C_1 and C_2, and for different ratios of class over-provisioning factors (i.e. OP_1/OP_2), where $OP_2 = 1$.

As can be seen from Figure 3.17, the most significant relative benefits in terms of bandwidth savings from deploying Diffserv are achieved when the proportion of Class 1 load (the class with the tightest SLA commitment and therefore the highest over-provisioning factor) is low relative to the Class 2 load, and when the ratio of the over-provisioning factor for Class 1 is high relative to the over-provisioning factor for Class 2 (i.e. OP_1/OP_2). Conversely, if all traffic is of the tightest SLA class, intuitively there is no benefit in Diffserv. Chapter 6, Section 6.1.3 discusses the question of what levels of bandwidth over-provisioning are required at different link rates within the core to achieve defined delay and loss commitments.

3.3.2 Core Case Study

We consider a core Diffserv deployment, capable of supporting the SLA requirements of the edge Diffserv classes described in Section 3.2.2.

3.3.2.1 Core Classes of Service and SLA Specification

Clearly, where Diffserv is deployed in the core of the network, the classes supported at the edge need to be able to be mapped to classes in the core, which are capable of meeting the defined classes SLA requirements. At the edge of the network, different classes are used for SLA differentiation and for application isolation, whereas in the core of the network different classes are used for SLA differentiation and for service isolation, where one or more applications may map to a service. Consequently, there may be fewer classes in the backbone than there are at the network edge, with a many-to-one mapping of edge classes to a core class.

Initially we consider support for four core classes, although support for more classes is considered in Section 3.3.2.4, which considers design variations.

- *Real-time (RT)*. This class targets applications such as VoIP and video. The backbone SLA for this class is defined in terms of low

delay, low jitter and low loss. A typical SLA for such a class would commit to a one-way delay across the backbone of less than 5 ms (see VoIP delay budget example in Section 3.2.2.1.1) and an average end-to-end loss rate of typically less than 0.1%.

- *Premium data (Prm).* This class targets business-critical applications, which have requirements for bounded delay (albeit less stringent than that of the real-time class) and low loss. An SLA for such a class may commit to a one-way delay or RTT across the backbone and an average loss rate of typically less than 0.1%.
- *Control class (Ctrl).* The *Ctrl* class is dedicated for network control traffic, ensuring that bandwidth on the core links is guaranteed for essential functions typically including routing protocols and for network management traffic, such as telnet or SNMP.
- *Standard class (Std).* The *Std* class is used for all other data traffic, i.e. traffic other than that which is classified as *RT, Prm,* or *Ctrl*. The SLA commitment for such a class may commit to an average RTT across the core and an average loss rate.

These four classes would allow a one-to-one mapping with the four edge classes described in Section 3.2.2.1, as shown in Figure 3.18.

The detailed core network design supporting classes with these SLA requirements is described in the following sections.

3.3.2.2 "Prioritized" Diffserv Core Model

In practice, most backbone Diffserv deployments today have adopted a "Prioritized Diffserv" deployment model. To illustrate the concepts

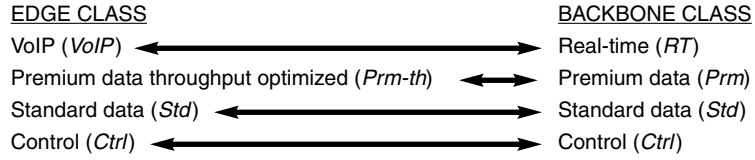

Figure 3.18 Edge to core class mapping

underlying this model, consider a typical design consisting of the four classes defined above, with each class being serviced from its own queue, and serviced with a particular PHB. The *RT* class traffic is serviced with an EF PHB, for example using a strict priority queuing treatment, ensuring that it has the lowest delay and jitter service. The *Ctrl* class is then serviced with an AF PHB, allocated with a minimal bandwidth assurance; let us assume 1% of the interface bandwidth. The *Prm* and *Std* classes are also serviced with AF PHBs, but are configured with a minimum bandwidth allocation such that the remaining bandwidth after the *RT* and *Ctrl* classes have been serviced is allocated with ~90% to the *Prm* class and ~10% to the *Std* class respectively.

Capacity planning is used to ensure that the average measured load for the *RT* and the *Prm* classes is less than their available capacity, i.e. such that their required OP factors are met, hence these classes experience zero packet loss due to congestion, with the only actual packet loss being due to layer 1 bit errors or network element failures. A holistic per-class capacity planning process is essential to ensure this is actually the case; the capacity planning process may take into account network working case conditions (i.e. normal operation) or single or multiple network element failure cases, depending upon the particular goals of the SP; such approaches to capacity planning are discussed in Chapter 6, Section 6.1. Assuming that a work-conserving scheduler is used, even though the standard class is allocated only 10% of the remaining bandwidth, it will have access to all unused interface capacity once the *RT*, *Prm*, and *Ctrl* data classes have been serviced.

A characteristic of this design is that, if interface congestion occurs, assuming the per-class capacity planning process is operating correctly, the loss will be restricted to the *Std* class while the SLAs for the *RT* and *Prm* class will be assured. Further, because the bandwidth available to the *RT* and *Prm* classes is significantly over-provisioned by the scheduler configuration, the scheduler does not need reconfiguring as per-class loads change over time. Rather, when class load thresholds relative to available class bandwidth are such that required over-provisioning factors are not met, the problem is reduced to one of increasing the link bandwidth.

Assuming the per-class capacity planning process is operating correctly, the effect of such a core network Diffserv design is as follows:

- The *RT* class is serviced with priority; capacity planning ensures that the *RT* load is below class thresholds, ensuring that it receives the lowest delay and jitter characteristics and zero packet loss due to congestion.
- The *Prm* class is serviced to its minimum bandwidth assurance; capacity planning ensures that the *Prm* load is below class thresholds, ensuring that it receives zero packet loss due to congestion, and the delay for the class, while not as low as the *RT* class, may be statistically bounded. The class also has the ability to re-use bandwidth from the other classes that may be idle, up to the available link bandwidth, however, if the capacity planning process is operating correctly, it should not need to.
- The *Ctrl* class is serviced to its minimum bandwidth assurance; if the class load is less than the minimum bandwidth assurance for the class it should receive good service with low loss and bounded delay. The class also has the ability to re-use bandwidth from the other classes that may be idle, up to the available link bandwidth, however, it should not need to.
- Due to the work conserving scheduler behavior, the *Std* class is able to re-use unused bandwidth from other classes, up to the available interface rate. The loss and delay characteristics for the class will depend upon the capacity planning load threshold (OP factor) for the class, which need to be managed to achieve the appropriate SLA for the class.

3.3.2.3 Detailed Core Design

The actual core Diffserv policies used to achieve such a design are generally very simple and require relatively minor changes to existing router configurations. Figure 3.19 defines a backbone Diffserv policy designed to achieve the SLA specification described in Section 3.3.2.1 in terms of the Diffserv meta-language defined in Section 3.2.2. This Diffserv policy would be applied outbound to all links in the backbone.

```
policy core-policy
  class RT
    classify DSCP (D_RT)
    classify EXP (E_RT)
    SR-TCM (R_RT, B_RT, 0)
      green-action transmit
      red-action drop
    EF
  class Prm
    classify DSCP (D_p)
    classify EXP (E_p)
    AF (R_p)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
  class Ctrl
    classify DSCP (D_c, 48)
    classify EXP (E_c)
    AF (R_c)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
  class Std
    classify *
    AF (R_s)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
```

Figure 3.19 Backbone Diffserv Policy

As can be seen from the meta-language design described in Figure 3.19, which is representative of real-world router configurations, relatively few lines of configuration are required to implement the Diffserv policy. In backbone Diffserv deployments, these configurations are typically applied once, and then remain static. Furthermore, migrating a backbone to Diffserv can be achieved seamlessly: the backbone configuration can be undertaken independently of the configuration required at the network edge to ensure that traffic is appropriately conditioned and marked on ingress to the network. The benefit of Diffserv for the backbone, however, will not be realized until both edge and backbone components are complete.

The classification criteria used in Figure 3.19 include matching either against DSCP values or against MPLS EXP values. Depending on the particular network deployment and whether MPLS is used in the core network, either or both classification criteria may be required.

3.3.2.3.1 Real-time Class (*RT*)

Considering the *RT* class Diffserv configuration in Figure 3.19, it is assumed that real-time traffic has been marked either at the source or at the edge of the network to one of the DSCP markings within the set D_{RT}, where $D_{RT} = \{D_v, \ldots\}$, or one of the EXP markings within the set E_{RT}.

The per-hop latency commitment is derived from the backbone latency commitment. For example, to achieve an EF backbone delay of less than 5 ms (see VoIP delay budget example in Section 3.2.2.1.1), we assumed 10 router hops through the backbone and apportion a worst-case delay success criterion of 500 μs per hop to ensure that the 5 ms delay target is not exceeded, assuming additive jitter. While jitter might not be additive in practice (see Chapter 6, Figure 6.6, this scenario represents the absolute worst case, and if the deployment can achieve this worst-case per-hop budget, the 5 ms backbone budget will be assured. The per-hop SLA latency commitment is assured by servicing this class with an EF PHB, such that it receives the lowest latency through the scheduler, and by ensuring that the arrival rate enforced by the class policer, R_{RT}, is smaller than the servicing rate for the class.

This could be achieved using the SR-TCM with rate R_{RT}, committed burst size B_{RT}, with EBS = 0 and applying a green action of transmit and a red action of drop. The SP specifies the SLA contract with the policer's worst-case admitted burst B_{RT}, such that the burst is serviced within the class per-hop latency commitment L_{RT}, i.e. $B_{RT}/link\_rate + L_s < L_{RT}$, where L_s represents the worst-case delay impact on EF traffic due to the scheduler and the interface FIFO (see Chapter 2, Section 2.2.4.1.3). Results from [FILSFILS] show that in tests of high-performance routers, L_s was constrained to 125 μs for a 2.5 Gbps interface.

In the backbone, the function of the *RT* class policer is to ensure that the *RT* class cannot starve the other classes of bandwidth,

whereas, at the edge of the network policers are also used to ensure that customers do not send more traffic than their SLA commitment allows. In normal operation, and with correct core network capacity planning, the *RT* traffic demands should be known, capacity should be provisioned, and the policer set such that there is sufficient bandwidth and *RT* traffic should not be dropped by the class policer. Therefore, the *RT* policer is typically set higher than the anticipated class load, at a rate whereby it acts as a safety precaution giving protection for the other classes in the case of unplanned for failures, which lead to adversely high *RT* load. Maximum acceptable *RT* rates are discussed in Chapter 6, Section 6.1.3.

If a router implementation requires that a queue-limit is configured, the queue-limit must be greater than or equal to B_{RT}, to ensure that packets that are within the permitted burst for the class are not dropped. Therefore, the only actual packet loss that can occur for the class is due to layer 1 bit errors or network element failures, which account for the loss rate for the class offered by the SP.

3.3.2.3.2 Premium Data Class (*Prm*)

Considering the *Prm* class Diffserv configuration in Figure 3.19, it is assumed that *Prm* traffic has been marked either at the source or at the edge of the network to one of the DSCP markings within the set D_p where $D_p = \{D_t, \ldots\}$, or one of the EXP markings within the set E_p.

The SLA for the *Prm* class is assured by treating the class with an AF PHB, with a minimum bandwidth allocation of ~80–90% of the remaining bandwidth after the *RT* and *Ctrl* classes have been serviced, inline with the prioritized Diffserv model described in Section 3.3.2.2. The RED congestion control mechanism is used within the *Prm* class queue rather than tail drop to maximize TCP throughput. RED tuning is described in detail in Section 3.4.

Capacity planning is used to ensure that the *Prm* load is below class thresholds, such that the required OP factor is achieved, ensuring that it receives zero packet loss due to congestion, with the only actual packet loss experienced by the class being due to layer 1 bit errors or network element failures. This also ensures that the delay

for the class, while not as low as for the *RT* class, may be statistically bounded.

3.3.2.3.3 Control Class (*Ctrl*)

Considering the *Ctrl* class Diffserv configuration in Figure 3.19, traffic is classified into the *Ctrl* class by matching on the DSCP which is assumed to have been pre-marked at the source: either to DSCP D_c, by network management end-systems, or to DSCP 48 by routing protocol end-systems or one of the EXP markings within the set E_c.

The *Ctrl* class is assured a minimal share of the access-link bandwidth, e.g. 1% is typically sufficient to support routing protocol and management traffic on high bandwidth core links. The class also has the ability to re-use bandwidth from the other classes that may be idle, up to the available link bandwidth. As a number of applications used for network control and management use TCP (e.g. BGP, SNMP, Telnet) RED is used within the *Ctrl* class queue to ensure TCP throughput within the class is maximized when congestion occurs. RED tuning is described in Section 3.4.

3.3.2.3.4 Standard Data Class (*Std*)

There are two ways that the standard class could be classified:

- either explicitly, based upon DSCP, assuming that *Std* traffic has been marked either at the source or at the edge of the network to one of the DSCPs in the set D_s or one of the EXP markings within the set E_s.
- Or implicitly; as per Figure 3.19, assuming the ordering of classes within the Diffserv policy defines a first match order for classification criteria, the wildcard classification criteria used for the *Std* class ensures that all traffic that has not been classified into the *Real-time*, *Bus*, or *Ctrl* classes is classified into the *Std* class.

The *Standard* class SLA is assured by treating the class with an AF PHB, with a minimum bandwidth allocation of ~10–20% of the remaining bandwidth after the *RT* and *Ctrl* classes have been serviced, in line

with the prioritized Diffserv model described in Section 3.3.2.2. RED is used within the *Std* class queue to maximize TCP throughput.

Capacity planning is used to ensure that the required over-provisioning factor for the class is achieved and hence that the loss and delay SLAs for the class are met.

3.3.2.4 Design Variations

In this section, we consider design variations to the basic core design described in the preceding sections, which are needed to support the variations in the edge design described in Section 3.2.2.4. It is noted that the variations to the edge Diffserv design described in Sections 3.2.2.4.1–3.2.2.4.3 (Low-speed Edge Design, Hierarchical Shaping and Scheduling, and Unmanaged Access Router Services) have no impact on the core network design; these variations affect the edge only. Enhancements to the core Diffserv design are required in order to support the edge Diffserv design described in Sections 3.2.2.4.4–3.2.2.4.6:

- *Premium data delay-optimized class (Prm-delay).* There are potentially a number of designs that could be applied to provide support for a *Prm-delay* class across the core:
  - The *Prm-delay* class could potentially be mapped to the real-time class. It is noted that the impact of large data packets on the worst-case delay experienced by other traffic in the same class, is very much less significant on higher speed links, such as those in the backbone. Nonetheless, the delay targets for the *Prm-delay* class are typically less stringent than for VoIP traffic, for example. Hence, for reasons of SLA differentiation and service isolation, it may be required to service the *Prm-delay* service from another class.
  - The *Prm-delay* class could potentially be mapped to the core *Prm* class, defined in Section 3.3.2.3.1, assuming that the class capacity planning thresholds are set to ensure that the *Prm-delay* SLAs for delay can be met.
  - Another core AF class could be added to provide support for the *Prm-delay* class. The approach taken for the class would be similar to that for the *Prm* class; however, this provides the capability to offer service isolation, and SLA differentiation with different

overprovisioning factors and hence capacity planning thresholds for the new class.

- An alternative design is possible using advanced scheduler implementations, which provide support for more than one priority queue as described in Chapter 2, Section 2.2.4.1.4. This would allow the possibility of using the highest priority queue for RT traffic, and the next priority queue for *Prm-delay* traffic, enabling the RT traffic to receive the lowest delay and jitter, while the *Prm-delay* traffic would have a bounded delay and jitter, independent of the scheduler implementation and the configuration and load of the other AF queues.

- *Business data throughput-optimized class with loss commitment (Prm-loss).* An edge *Prm-loss* class, with in- and out-of-contract capabilities, could be supporting by being mapped to the core *Prm* class, defined in Section 3.3.2.3.1. This would require that WRED be applied to the *Prm* class queue such that if there is congestion in the queue, then out-of-contract traffic will be preferentially discarded over in-contract traffic. This is achieved by having a more aggressive WRED profile (lower minth and maxth settings) for the out-of-contract traffic, as described in Chapter 2, Section 2.2.4.2.4. The WRED profile for in-contract traffic should be set such that the in-contract traffic should not be dropped (see Section 3.4); hence, the only actual packet loss that in-contract traffic within the class can experience is due to layer 1 bit errors or network element failures, which account for the loss rate for the class offered by the SP.

- *Video class.* The possible design options for supporting a video class across the core are ostensibly the same as those for supporting a *Prm-delay* class described above. However, the latency requirements may be tighter for a video class, and the requirements for service isolation may be different, hence the final resulting choice made for supporting a video class may be different from the *Prm-delay* class. A common approach is to service VoIP and video from the same class in the core.

By way of example, we consider how the design would change if support for these three additional classes were provided in the core: we

assume that the edge video class is mapped to the core real-time class, and that *Prm-delay* and *Prm-loss* classes are both mapped to the *Prm* class, as shown in Figure 3.20.

The Diffserv policy in Figure 3.21 augments the design from Section 3.2.3 with the modifications required to support these additional edge service classes highlighted in bold.

The changes from the previous design are as follows:

- The classification criteria for the real-time class are changed to classify the DSCP and EXP values corresponding to both VoIP and video classes.
- The classification criteria for the *Prm* class is changed to classify the DSCPs corresponding to the *Prm-loss*, and *Prm-delay* classes.
- WRED, rather than RED, is configured within the *Prm* class queue, to enable different RED profiles for in- and out-of-contract traffic for the *Prm-loss* class.

Further discussion on considerations of mapping and aggregating edge classes to core classes is provided in [BABIARZ].

3.3.2.5 Core-marking Scheme

There is a choice as to what marking scheme is used in the core, and that choice depends upon the type of network technology used in the core, IP or MPLS, and the type of network service that is used, either a private network or a virtual private network (VPN) service built on a public network. We consider the typical combinations:

- *IP core/private network*. If IP is used in the core of a private network, then a consistent marking scheme can be used end-to-end as the full 6-bits of the DS field are available for use both at the edge of the network and within the core. This is typical of private networks, e.g. networks serving the needs of a single enterprise organization.

  Even though a consistent marking scheme is used, there may be fewer classes in the core than at the edge of the network and hence several edge classes may be aggregated into a single core class, with several DSCP values being classified into a single class.

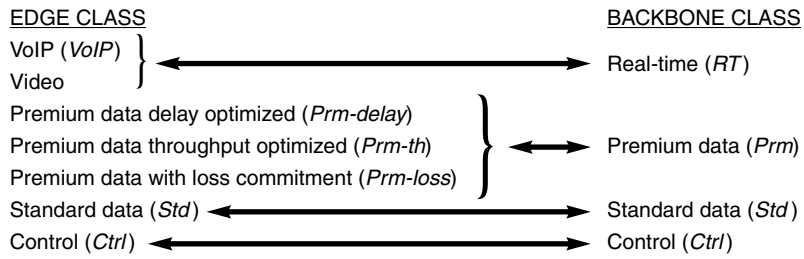

Figure 3.20 Edge to core class mapping: design variations

```
policy core-policy
  class Real-time
    classify DSCP (D_RT)
    classify EXP (E_RT)
    SR-TCM(R_RT, B_RT, 0)
      green-action transmit
      red-action drop
    EF
  class Prm
    classify DSCP (D_d, D_lin, D_lout)
    classify EXP (E_d, E_lin, E_lout)
    AF (R_p)
    RED (D_lin, <minth>, <maxth>, <w>, <pmax>)
    RED (D_lout, <minth>, <maxth>, <w>, <pmax>)
    RED (E_lin, <minth>, <maxth>, <w>, <pmax>)
    RED (E_lout, <minth>, <maxth>, <w>, <pmax>)
  class Ctrl
    classify DSCP (D_c, 48)
    classify EXP (E_c)
    AF (R_c)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
  class Std
    classify *
    AF (R_s)
    RED (*, <minth>, <maxth>, <w>, <pmax>)
```

Figure 3.21 Backbone Diffserv Policy: design variation

- *MPLS core/private network*. If MPLS is used in the core, a mapping will be required from the edge-marking scheme to the core-marking scheme; this mapping requires the edge label switched router (E-LSR) to set the MPLS experimental (EXP) field as a function of the received DSCP for upstream traffic. If more than eight class markings are used at the edge, then as the MPLS EXP field is only 3 bits, and hence can represent only 8 values, there must be a many to one mapping of some of the edge-markings to a single core-marking.

  Careful selection of the edge class marking scheme can make the mapping from edge to core classes easier, by making use of the typical default behavior applied at E-LSRs, which is to copy the first three bits of the DSCP (i.e. equivalent to the class selector codepoint values), into the MPLS EXP field when a label is imposed (see Chapter 2, Section 2.3.6.2.1). If there were less than eight edge class markings used, then by ensuring that the first 3 bits of the DSCP are unique for each edge class, there would be a unique MPLS EXP field value assigned to each edge class by default. A possible mapping of the edge DSCP markings from Figure 3.14 to a core MPLS marking scheme is given in Figure 3.22.

  From Figure 3.22, we note that there are 7 discrete DSCP values used for marking at the edge, which allows a one-to-one mapping

Edge class	Edge DSCP marking	Core class	Core EXP marking	Explicit mapping required?
VoIP	D_v = 46 (EF)	*Real-time*	E_{rt} = 5	No
Prm-th	D_p = 10 (AF11)	*Prm*	E_p = 1	No
Std	D_s = 0	*Std*	E_s = 0	No
Ctrl: Routing protocols	= 48 (CS6)	*Ctrl*: Routing protocols	= 6	No
OA&M	D_c = 16 (CS2)	OA&M	E_c = 7	Yes
Prm-delay	D_d = 18 (AF21)	*Prm*	E_d = 2	No
Prm-loss: In-contract	D_{lin} = 10 (AF11)	*Prm*: In-contract	E_{lin} = 1	No
Out-of-contract	D_{lout} = 12 (AF12)	Out-of-contract	E_{lout} = 3	Yes

Figure 3.22 Mapping of edge to core marking schemes

of edge class markings to core class markings. Further, we note with this particular edge scheme the first 3 bits of the DSCP are not unique for each edge class. Therefore, applying the typical default E-LSR behavior, which is to copy the first three bits of the DSCP into the MPLS EXP field, would result in some discrete edge classes incorrectly receiving the same marking within the core. The *Ctrl* class uses CS2 at the edge and the *Prm-delay* class uses AF21; the first three bits of both CS2 and AF21 are 010, hence applying the default E-LSR copying behavior, packets with these markings would both be marked as EXP 2 within the core and would therefore be indistinguishable. Similarly, the *Prm-loss* class uses AF11 and AF12 to represent in- (D_{lin}) and out-of-contract (D_{lout}) traffic at the edge of the network, with DSCP values of AF11 and AF12 respectively. The first three bits of both AF11 and AF12 are 001, hence applying the default E-LSR copying behavior, packets with these markings would both be marked as EXP 1 within the core and would therefore be indistinguishable. There are two possible ways of overcoming this issue:

- Either explicit mappings can be applied at the ingress E-LSR, rather than the default behavior of copying the first three bits into the EXP field, such that classes can be appropriately distinguished within the core. This is shown in Figure 3.22, where CS2 is explicitly mapped to an EXP marking of 7, so that it is distinguishable from EXP 2, and hence the *Ctrl* class can be distinguished from the *Prm* class within the core. Similarly, AF12 is explicitly mapped to EXP 3, so that in- and out-of-contract traffic within the *Prm* class can be correctly distinguished.
- Or the edge-marking scheme could be changed such that, when applying the default E-LSR copying behavior, the resulting EXP markings allow the core classes to be appropriately distinguished. In this example, this could be achieved if the edge-markings used for the *Ctrl* class (D_c) and for the out-of-contract traffic within the *Prm-loss* class (D_{lout}) were changed to DSCP 56 (CS7) and DSCP 28 (AF32) respectively.

If there are more than eight edge classes, then ensuring that the first three bits of the DSCP of each edge class represents the core

class that it will be mapped to, would allow the default E-LSR copying behavior to be used, rather than requiring any explicit class mappings.

Alternatively, explicit mappings may be used between edge and core classes. This places less constraints on the markings used at the edge, but requires explicit configuration at each E-LSR, to map each edge class to a core class.

- *MPLS core/virtual private network.* When MPLS is used by a network service provider to deliver VPN services, providing network services to a number of enterprises, for example, there are two possible approaches to the core-marking scheme that the network provide can take:
  - *Universal edge-marking scheme.* The network service provider could impose a single edge-marking scheme on all VPN customer networks. This would then allow a single mapping to the core-marking scheme to be used and the options are effectively no different than the "*MPLS core/private network*" model described above, other than the fact that the network service provider defines the marking scheme. This model may represent the simplest solution for the network service provider, but may not be attractive to enterprise organizations, which may have already adopted their own marking scheme, which would have to be changed in migrating to the VPN service.
  - *Per-VPN edge-marking scheme.* Alternatively, the network service provider could support different edge-marking schemes per edge customer. This would require mappings between edge and core-marking schemes on a per VPN customer basis. Use of the Pipe MPLS Diffserv tunneling model described on Chapter 2, Section 2.3.6.2.3.2 would allow the different VPN customer's edge-marking schemes to be preserved transparently across the core. This represents a more complicated model for the network service provider, but can simplify the migration to a VPN for the customer.
- Some deployments may use both IP transport and MPLS transport within the core; in these cases, classification by both DSCP and MPLS EXP classification may be required.

3.4 Tuning (W)RED

The random early detection (RED) congestion avoidance algorithm was originally designed as an algorithm aimed at improving throughput for TCP-based applications, by preventing "global synchronization" between TCP sessions; the operation of RED described in detail in Chapter 2, Section 2.2.4.2.3. Global synchronization is where congestion occurs and queue-limits are exceeded causing drops in multiple sessions, which each back-off, such that the effective aggregate throughput drops below line rate. The sessions then all increase their rate of sending until congestion occurs again and the cycle repeats creating a sawtooth aggregate throughput characteristic. RED aims to prevent this by randomly discarding packets from individual sessions as the average-queue depth increases, such that a higher aggregate throughput is maintained.

RED makes a drop decision prior to enqueuing a packet into a queue, based upon the current average queue depth of that queue and a set of configurable parameters, which hence define the characteristics of RED. A particular set of RED parameters specifies a RED profile, which is defined by a minimum queue threshold (*minth*), maximum queue threshold (*maxth*), and probability of discard at maxth (*maxp*). Weighted RED (WRED) simply extends the basic concept of WRED, by allowing a number of different RED profiles to be used for the same queue, where each profile may be applied to a particular subset of the traffic (normally identified by a specific DSCP or MPLS EXP marking or markings) destined for the queue. This results in different drop characteristics, and consequently probability of drop, per profile. WRED is used for differentiation in the drop probability between in- and out-of-contract traffic in support of classes such as those defined in Section 3.2.2.4.5.2.

The goal of tuning RED is to maximize the link utilization while minimizing the mean queue depth, hence minimizing delay. There are many factors which can impact the tuning of RED, including the number of active TCP sessions, whether those session are long-lived or short-lived, what the RTT is for those sessions, what TCP stacks are

used and the particulars of the specific RED implementation itself. These factors vary network-by-network, site-by-site, and probably hour-by-hour, hence optimal RED tuning is probably not achievable in practice. In this section, however, we propose some generic RED tuning guidelines, derived from a number of sources including [FLOYD, CHRISTIANSEN, HOLLOT] which can be fine-tuned as necessary based upon operational experience acquired in the specific deployment environment.

3.4.1 Tuning the Exponential Weighting Constant

The exponential weighting constant (*w*) determines how closely the average queue depth tracks the actual queue depth. The lower *w* the more closely the average queue depth tracks the actual queue depth; this has two consequences: the RED drop behavior is more sensitive to traffic bursts, and the closer the maximum instantaneous queue size is to *maxth*, hence the easier it is to limit the maximum latency via RED. For larger *w*, the converse is true. The following values for *w* are suggested $w = 1/B$ where B is the output rate expressed in MTU sized packets, i.e.:

$$w = (\mathrm{BW}/\mathrm{CMTU} * 8)$$

The MTU used in these calculations should be the maximum packet size, which is normally 1500 bytes (the MTU for Ethernet), rather than the MTU of the particular interface, which can often be greater than 1500 bytes. The bandwidth used in the calculations should be the effective servicing rate for the class; for a single class deployment, this would be the link rate.

Note that the enhancement to RED was proposed in "Red-Light" [JACOBSON1], which has been implemented by some vendors, does not have the concept of a configurable exponential weighting constant.

3.4.2 Tuning Minth and Maxth

The minth value should be set high enough to maximize the link utilization; if too low, packets may be dropped unnecessarily, and the link bandwidth will not be fully utilized. The difference between the maxth and the minth should be large enough to avoid global synchronization; if the difference is too small, many packets may be dropped at once, resulting in global synchronization.

Most TCP tuning focuses on the concept of the "pipesize," which is the amount of unacknowledged data in flight between the sender and the receiver, which is the bottleneck bandwidth * RTT product. Where a single RED profile is used in a queue (i.e. no in-/out-of-contract), the following settings (in packets) for minth and maxth are suggested:

For rates < 10 Mbps:

minth = MIN [5, 1 * pipesize]
maxth = 3 * pipesize
[where pipesize = *RT*T * BW/(MTU * 8)]
i.e. minth = (ROUND(1**RT*T*BW/(MTU*8),0))
maxth = (ROUND(3**RT*T*BW/(MTU*8),0))

For rates > 10 Mbps:

minth = MIN [5, 0.15 * pipesize]
maxth = 1 * pipesize
[where pipesize = RTT * bw/(MTU * 8)]
i.e. minth = (ROUND(0.15*RTT*BW/(MTU*8),0))
maxth = (ROUND(1*RTT*BW/(MTU*8),0))

The RTT used in these calculations is the round trip time between the TCP hosts. As this varies depending upon queuing delays, access rates, propagation delays, etc., in the absence of other guidance a value of 100–200 ms is suggested. The bandwidth used in the calculations should be the effective servicing rate for the class; for a single class deployment this would be the link rate.

3.4.3 Mark Probability Denominator

It is suggested that a mark probability denominator of 0.1 is used, i.e. the probability of drop at maxth is 0.1.

3.4.4 In- and Out-of-contract

Where WRED is used for differentiation between in- and out-of-contract traffic within the same queue it is suggested that the values for RED recommended in the previous sections are used for the WRED profile applied to the out-of-contract traffic that a "no-drop" RED profile is used for in-contract traffic, i.e. a profile set high enough that in-contract traffic will not be dropped (alternatively, no RED profile or a queue-limit, could be used for out-of-contract traffic). The intended characteristics of this approach are:

- If in-contract traffic is policed such that the arrival rate of the class is not greater than the servicing rate of the queue, then the maximum queue depth should not increase significantly beyond maxth for the out-of-contract profile.
- As there could be both in- and out-of-contract traffic from the same flow, dropping out-of-contract traffic first should reduce the possibility of multiple simultaneous drops from the same flow, and hence be more optimal for aggregate throughput, i.e. across the sum of in- and out-of-contract traffic.

What about using more than two drop WRED drop profiles within a queue, e.g. three drop thresholds as per Diffserv AF11, AF12, AF13? Expressing this in the context of in- and out-of-contract, in practice, it is difficult to understand what meaningful service benefit is offered by differentiating between in-, out-, and exceedingly-out-of-contract traffic. If the intent from such a configuration is to allocate different proportions of bandwidth within a queue to subclasses of that queue, then WRED is not a suitable mechanism to use. To quote [JACOBSON2]: "It

is almost always better to run one instance of RED over each of these *x* queues rather than multiple instances of RED over one big queue." Hence, it is not common practice to use more than two WRED drop profiles within a queue, although there may be exceptional cases, where this is warranted.

References

[BABIARZ2] J. Babiarz, K. Chan, F. Baker, Aggregation of DiffServ Service Classes, draft-ietf-tsvg-diffserv-class-aggr, October 2006

[CARTER] S. F. Carter, Quality of service in BT's MPLS-VPN platform, *BT Technology Journal*, Volume 23, Issue 2, pp. 61–72, April 2005

[CHRISTIANSEN] M. Christiansen, K. Jeffay, D. Ott, F. Donelson Smith, Tuning RED for web traffic, in Proceedings of ACM SIGCOMM 2000, pages 139–150, August 2000

[FILSFILS] Clarence Filsfils, John Evans, Engineering a multiservice IP backbone to support tight SLAs, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, v.40 n.1, pp. 131–148, 16 September 2002

[FLOYD] http://www.aciri.org/floyd/REDparameters.txt

[G.1010] ITU-T Recommendation G.1010, End-user multimedia QoS categories, International Telecommunication Union, Geneva, Switzerland, Oct. 2001

[HOLLOT] C. Hollot, V. Misra, D. Towsley, W. Gong

A Control theoretic analysis of RED. In Proceeding of IEEE INFOCOM '01, 2001

[JACOBSON1] V. Jacobson, K. Nichols, K. Poduri, RED in a different light. Technical Report, CISCO Systems, Sept 1999

[JACOBSON2] V. Jacobson. Notes on using RED for Queue Management and Congestion Avoidance, NANOG 13, June 1998

[MCCREARY] S. McCreary, and K. Claffy, Trends in wide area IP traffic patterns – A view from Ames Internet Exchange, *Proceedings of 13th ITC Specialist Seminar on Internet Traffic Measurement and Modeling*, Monterey, CA. 18–20 Sep, 2000

[RFC2581] W. Stevens, M. Allman, V. Paxson, TCP Congestion Control, RFC 2581, April 1999

[RFC4594] J. Babiarz, K. Chan, F. Baker, Configuration Guidelines for Diff Serv Service Classes, RFC 4594, August 2006

Notes

1. 1500 bytes is the MTU size for Ethernet, which is generally the constraining MTU in networks today.
2. This maxim, known as Ockham's razor is a principle attributed to the English logician William of Ockham (circa 1295–1349), which is generally expressed in Latin as "*entia non sunt multiplicanda praeter necessitatem,*" which translates as "*entities should not be multiplied beyond necessity*."