Internet Engineering Task Force
Internet Draft
Expires in April, 2006

K. Nichols
Pollere LLC
L. Sampson
R. Barrios
K. Adams
J. Pulliam
J. Kim
Lockheed Martin

draft-nichols-dcpel-strawman-arch-00
October 2005

A Strawman Architecture for Diffserv Control Plane Elements

<draft-nichols-dcpel-strawman-arch-00>

Status of this Memo

Copyright Notice

[Note: this is an informal pdf version of the draft]

Abstract

Diffserv (RFC 2474, 2475, and 3086) made explicit that
IP QoS can be separated into the differentiated
treatment given to packets in the forwarding path and

the task of configuring these forwarding path
components to allocate QoS according to policy and
availability. The IETF Diffserv WG described the
forwarding path architecture in detail and specified
some specific forwarding path elements. This draft
attempts a similar approach of specifying the elements
of a diffserv control plane, gives a general
architecture and example solution that fits into this
architecture, and lays out some issues. An example of a
diffserv control plane architecture is presented,
derived from the control plane described in RFC2638,
and an example implementation of that approach is
briefly described. The authors hope to stimulate a
discussion of the architectural model and its elements
and to elicit more example solutions that may fit,
change, or extend the model.

A control plane must be configurable, secure, and
monitorable. The authors believe the operations and
management issues of a diffserv control plane must be
made explicit and the approach to solving them properly
constrained. Resolving operational and management
issues is key to moving to availability of IP QoS.

A pdf version of this document is available at:
www.pollere.com at Resources: Current Work.

Table of Contents

1 Introduction

1.1 Background

This document describes a general model for a diffserv control plane which has evolved from the control plane described in RFC2638 and is consistent with RFCs 2474, 2475, and 3086. In addition, this control plane can interact with the router model of RFC3290's figure 1 through specification of the QoS Agent and control messages and some of the configuration and management interface. A key point in the Diffserv approach is that IP QoS can be divided into two functions, much like IP end-to-end connectivity. In IP forwarding, connectivity is achieved by the interaction of two components; the packet forwarding part and the routing part. Diffserv can be separated into the differentiated treatment given to packets in the forwarding path and the task of configuring the parameters of the forwarding path components to allocate QoS according to policy and availablility. The Diffserv model is based on the separation of forwarding path and control plane and on the notion that a small number of forwarding path primitives can be composed to create a wide range of QoS features. A diffserv control plane should be decoupled from the forwarding path and be part of the

network infrastructure of a network domain.

Per-Domain Behaviors (PDBs) [5] give a technical service description on a domain. The Traffic Aggregate is the forwarding path portion of a PDB which is defined on an entire DS domain, but the control plane must configure the QoS tables to produce the correct TA which will experience the expected metrics as it crosses the DS domain. The control plane consists of entities that can produce configuration messages based on information about policy and the state of the network. This information can be detailed or simple and can be obtained in a range of ways. Diffserv configuration does not happen at forwarding speeds and may, indeed, take place over very long time scales. Even an extremely dynamic configuration will not cause updates at forwarding rates.

1.2 Goal of this document

The goal of this document is to describe a path-decoupled diffserv control plane that is composed

of functional elements. The intent is analogous to the diffserv forwarding path development where the individual elements can vary in complexity (or not be present at all) so that customization and innovation can occur in an environment where the interfaces and functionalities are well understood. This is intended to provide a common framework in which both intra- and interdomain Diffserv QoS can be developed, respecting the varied needs of networks while encouraging common interfaces and specific functionalities that permit diverse products to interoperate.

The model defined here is shown to apply to a prototype Diffserv control plane, Domain-Managed QoS, and some of those components and interfaces are described. While the model is intended as a strawman for discussion, DMQoS is a particular implementation that the authors have worked on, presented for illustration.

We see the following issues: aggregation, security, performance and mangement, an operationally useful and constrained policy componenet. A goal is to have discussion on these issues and to bring out any others.

1.3 Definitions

The following definitions are stated in RFCs 2474, 2475, and 3086 and are repeated here for easy reference:

Behavior Aggregate: a collection of packets with the same codepoint crossing a link in a particular direction.

Differentiated Services Domain: a contiguous portion of the Internet over which a consistent set of differentiated services policies are administered in a coordinated fashion. A differentiated services domain can represent different administrative domains or autonomous systems, different trust regions, different network technologies (e.g., cell/frame), hosts and routers, etc. Also DS domain.

Differentiated Services Boundary: the edge of a DS domain, where classifiers and traffic conditioners are likely to be deployed. A differentiated services boundary can be further sub-divided into ingress and egress nodes, where the ingress/egress nodes are the downstream/upstream nodes of a boundary link in a given traffic direction. A differentiated services boundary typically is found at the ingress to the first-hop differentiated services-compliant router (or network node) that a host's packets traverse, or at the egress of the last-hop differentiated services-compliant router or network node that packets traverse before

arriving at a host. This is sometimes referred to as the boundary at a leaf router. A differentiated services boundary may be co-located with a host, subject to local policy. Also DS boundary.

Traffic stream: an administratively significant set of one or more microflows which traverse a path segment. A traffic stream may consist of the set of active microflows which are selected by a particular classifier.

Traffic Aggregate: a collection of packets with a codepoint that maps to the same PHB, usually in a DS domain or some subset of a DS domain. A traffic aggregate marked for the foo PHB is referred to as the " foo traffic aggregate" or the "foo aggregate"

interchangeably. This generalizes the concept of Behavior Aggregate from a link to a network.

Per-Domain Behavior: the expected treatment that an identifiable or target group of packets will receive from "edge-to-edge" of a DS domain. (Also PDB.) A particular PHB (or, if applicable, list of PHBs) and traffic conditioning requirements are associated with each PDB.

A Service Level Specfication (SLS) is a set of parameters and their values which together define the service offered to a traffic stream by a DS domain. It is expected to include specific values or bounds for PDB parameters.

2 Diffserv Control Plane Model

2.1 Overview

Before the Diffserv WG was started, one of the proposals (later published in RFC 2638) proposed an approach to the control plane. Once the working group was chartered, the emphasis was on forwarding path mechanisms. There is a subsequent body of work on diffserv control planes using bandwidth brokers (BBs), bandwidth managers and resource managers including [9, 12, 11, 16, 10, 13, 15, 19].
In addition, RFC 3086 and [14] give some description of what a Diffserv control plane needs to do. These works on the Diffserv control plane (DCP hereafter, DSCP would be confusing) have some commonality. The intent of this section is to both reflect this commonality and spell out general features for a DCP.

A DCP operates over a particular trust region, usually a single DS domain or an AS, and can be implemented as a single agent or by a collection or hierarchy of

agents and may contain its own databases or have access to information that is part of the network infrastructure. Externally, the DCP should appear as a service available at a particular (well-known) address, regardless of how it is implemented. DCPs should handle messages external to their domain of control, from authorized attached hosts or adjacent domains as well

as messages from their domain's network management and should be capable of monitoring selected network control information. A DCP configures the network edge in response to these messages, network performance measures, and in accordance with the policies of the domain as expressed in statements of rules. To deploy a secure and policy-controlled system, only a DCP can configure the edge routers which it does via a secure association.

Provisioning sets up static membership and limits on PDBs and their traffic aggregates while allocation sets up a portion of a PDB to an identifiable traffic stream for a specified duration. Packet schedulers in the forwarding path should be configured so that the metrics for each PDB's traffic aggregate are met and not changed in response to signaling. Those settings are part of the infrastructure of the network, determining its capacity for the various behaviors it commits to supplying. While the results of provisioning may be pushed out to the network through the DCP, provisioning is not a task of the DCP.

Allocation refers to the process of making traffic commitments anywhere along the continuum from strictly preallocated to dynamic call set-up and requires an allocation architecture capable of encompassing this entire spectrum in any mix. Allocation only results in changes to the configuration of the domain edge routers while the interior configuration remains the same. Static levels may be provisioned with time-of-day specifications, but cannot be changed in response to a dynamic message. Dynamic covers the range from a telephoned or e-mailed request to a signaled model. In cases where differential QoS is allocated in a strictly static way on the connection to an attached network, the attached network may control entry into that fixed aggregate in a number of ways, including per-flow or per-session admission. Where a dynamic QoS entity (possibly a DCP) exists in an attached network, a secure connection with that DCP can be established and used to request additional QoS beyond the basic allocation.

Signalling can indirectly (through the DCP) change classifier settings, thus changing some combination of

who is admitted to a traffic aggregate and whether the resources are committed. Whether an allocation is statically pre-allocated, signalled for in advance, or signalled for upon immediate need, the DCP should not handle them differently. Policy can and should be applied to appropriately prioritize the different approaches. For example, a priority could be attached to allocations that are requested in advance while still keeping the overall priorities of the network in place.

2.2 Diffserv Control Plane Elements

Figure 1 shows the basic functions needed by a diffserv control plane. The model comprises an Allocation Engine, a Request Manager, a Network State Manager, and databases of Network State History, Allocation State History, Policy rules, and Authentication rules. The control plane will need to configure edge routers, receive network alerts, handle messages from entities external to the network as well as internal network management. A DCP is expected to include the allocation engine, request manager, at least some part of the network state manager, and access to the required databases.

Fig. 1: Elements of a Diffserv control plane

The Request Manager handles dynamic messages that concern allocation, either from attached users or networks or from network management. These may use the same protocol or may differ. During allocation, the resources of three localities come into play: ingress, egress, and transit of the domain. Policy rules for each ingress and egress link reflect the SLA with the network on the other end of the link (e.g., I tell you what I will accept) and the Network State gives the physical capabilities and topology.

2.2.1 Request Manager

DCPs of adjacent networks communicate with each other through secure associations across network trust boundaries. DCPs must also communicate with other entities that may request allocations such as users, network administrators, and entities that request allocations on behalf of users such as call control agents. Requests may be internal or external and all requests must be authenticated. Might use some RSVP variant, custom messages over SCTP, or something from the NSIS WG. SIP might be used to contact a Call

Control agent (or SIP server) or even to communicate directly to the DCP. The Request Manager handles all messages related to requests, including any messages

that might be sent to indicate that a request is expiring or being pre-empted.

2.2.2 Allocation Engine

Allocation records are organized by PDB and will contain information about the PDB, its implementation (DSCP and edge restrictions), and applicable policies (could be restrictions on largest size allocable, number of simultaneous users, authorization levels). In addition, a list of available bandwidth for that PDB is kept, perhaps with ingress and egress routers specified. A list of allocated bandwidth for the PDB must also be kept, along with the identity of the user network and other information about the commitment. For a more sophisticated DCP (one that utilizes signaling and/or monitors network traffic instrumentation), it will be necessary to track committed allocations.

Policy data specific to the requester, topology restrictions, and perhaps general policies for the PDB must also be consulted to determine if the request can be granted. If the commitments are pre-emptable, it may be efficient to attach the authorization level to the committed bandwidth record.

Allocation records are generally expected to have an ingress and an egress port in order to permit the control of the ingress and egress allocations where resources are expected to be the most limited and where policies must match with the policies of the source or destination at the links. This prevents the overloading of an egress link to an attached network and ensures all policies are checked. Allocations are characterized by ingress, egress, PDB type, DSCP(s) used, time and duration, and information about the identity of the current user of the allocation that can be used to consult policy. It is possible to wildcard any of these fields should the particular network and/or resources allocated warrant it.

Through the Network State Manager, a DCP knows its own

network's topology and should not allocate more resources than it can handle. This may be done conservatively by assuming that all allocated flows will traverse the most limited link or the DCP may exploit its knowledge of the topology and routing used to be less conservative. For a meshier network, a reasonable allocation should permit functioning in the case of single link failures. This property of the DCP makes it path-aware, that is able to determine which links packets of any particular source/destination pair will take. This information can be used in making

traffic allocations and in making changes to allocations should the network topology change. The DCP will be informed of such changes as quickly as the routing infrastructure, then can consult policy to determine whether any link allocations have been affected. If so, the separate source/destination aggregates sharing that link allocation can be checked to see which has lower priority by policy or whose removal or change would be most efficacious by whatever other measures have been specified by policy. The policers for these aggregates can be immediately reconfigured to reduce or eliminate that traffic and a message about this should be sent to the affected attached network at the source (and possibly the destination) with a source field of the DCP service address.

If metric-based TE is used, the DCP will be aware of the metrics used by the SPF algorithm. (It may be possible to affect TE metrics from a DCP, but this requires further research to determine if necessary and desirable.) The additional network information might be communicated by use of the IS-IS TE extensions; SNMP traps may also be employed.

2.2.3 Policy Rules Database

The DCP must have a policy database that can be consulted to determine if each authenticated requester is authorized to have a particular resource at a particular time and, in the case of pre-emption, the priority level of each requester. A network administrator configures the policy database. The Allocation Engine should probably be notified of

changes to the policy database.

2.2.4 Network State Manager

In general, a change in allocation means network edge routers must be reconfigured. The network elements that can be configured are classifiers and traffic conditioners. The DCP must be able to reconfigure these in response to a change in allocation. For dynamic allocations or the dynamic portion of allocations, edge devices should be configured with soft state that will expire if not refreshed periodically by the DCP. This may not be required for static allocations. Configuration can be done through SNMP, CLI, COPS or other approaches.

Network topology can be maintained by monitoring routing or other network information and other state may come from SNMP traps or other alerts.

As allocations depend in some degree on the topology and capabilities of the network, the DCP must be able to receive and respond to network alarms that may be generated in response to changes (e.g., topology changes that affect available bandwidth) or from a measurement infrastructure (unmet SLAs, etc.).

2.2.5 Authentication Rules Database

Authentication is either part of the DCP or an external function it can access. Requests that come through some intermediary such as a call control agent that are on behalf of an end-user (principal) must have a way to authenticate with identity of principal without principal revealing too much of its own information. Network management configures this database and the Allocation Engine consults it.

Figure 2 shows how a request might be handled. The request can come from a user, an associated DCP within the same trust domain, a DCP in a different trust domain, or some other entity. The possibility of having a DCP hierarchy is addressed later.

Fig. 2: Responding-to-a>Responding to a request

2.2.6 Diffserv Router QoS agents

The informal diffserv router model of RFC 3290 indicates that a QoS agent may be present in the edge routers. QoS agents also appear in RFC 3175 and the RSVP RFCs. Spelling out the functionality of these agents is an important task for a diffserv control plane as they can be a critical part of the infrastructure. Certainly, a diffserv control plane could be realized without an agent in the edge routers, but they can be usefully employed in some architectures. Our model explicitly includes the QoS agent, or DCPA. All QoS resource requests are passed from the router forwarding plane to the DCPA; if an ARSVP agent is present, it would also pass requests to the DCPA. Reqests are passed to the DCPA for authentication, authorization, and approval. The DCPA may make some accept/reject decisions locally and may message the DCP service address for other decisions. This resembles a COPS model where the DCPA becomes the PEP and the DCP is the PDP and perhaps that would be a suitable communication protocol. Configuration messages should be passed through the DCPA which can resolve conflicts and authenticate configurations.

This is illustrated in figure 3. There is one known

address for the DS domain's DCP and QoS requests are sent to this address. The domain-wide DCP is shown with a dotted outline to indicate it may represent a real or virtual entity and options for its distribution are covered in the next section. All resource requests may be referred to the domain-level DCP or allocations of local interest may be parceled out among DCPAs. Parceling out allocations of local interest follows some simple rules so that resource managment decisions are delgated not distributed. Only one entity has control of a resource at a time, eliminating race conditions. Local control over the access links may be efficient.

Fig. 3: DCP service using agents in edge routers

DCPA-local allocations may be further tagged with information about whether more bandwidth may be requested of the domain-level DCP for this allocation

or other information that might be useful in local commitment of resources or in requesting changes from the DCP. As portions of allocations are committed, the available rates are reduced. As committed resources are released, the available rates are increased. High and low water marks may be used to request more resources from the DCP or offer to return resources to a general pool. Resources should be tracked pending the set up of the entire path. May use a timeout to return to allocation pool if no confirmation returned. DCPAs will have some operational rules, for example, these might include rules like:

* the DCPA can allocate from the allocation pool in response to authenticated requests

* the DCPA can bump lower priority committments in favor of higher priority requests

* rules about asking the DCP to rearrange the allocation pool

* how to treat high or low water marks on committed allocations

* respond to DCP requests for status

* how to handle configuraton requests

2.3 Admission Control Model

In figure 4, the requesting entity in a user network can be a host. A host asks for permission to send and sends (and receives) packets marked for a particular PDB defined on the network. The permission asking and granting might be dynamic or preconfigured or implicit.

When an application aims a request at a network domain, its physical path includes the edge router. The router should send QoS requests to the local DCP agent. The host always sends its data to the edge router, but it may perform some additional conditioning functions on the data (like marking and shaping). The DCP has the responsibility for allocating its DS domain(s). It receives requests, determines if the resources are available (to that particular requestor), grants or refuses permission, and generates a response and/or

configuration.

Fig. 4: Admission-Control,-Host>Admission Control Components

Whenever a network boundary is crossed, it is important to ensure that trust is not violated or to pass trust in a controlled way (e.g. authentication). An attached network is responsible for ensuring that the data packets it sends conform to all appropriate SLSs at risk of having packets dropped. Use of mutually agreed DSCPs can be used to distinguish packets for different SLSs. Once admitted, the DS domain has the responsibility for delivering packets reliably and queuing them consistently with their DSCPs. These characteristics are determined by the PDB provisioned on the domain and SLSs can use those charcteristics.

There are a number of ways edge router configuration can be done, some of which do not require that the allocator know the specific address/location of the edge router. For example, the allocator might send a "cookie" to host (sender or receiver) to use in signalling, thus passing the trust across the boundary in a limited use cookie. The allocator might multicast the configuration information to all the boundary routers. If the allocator knows the specific router to be configured, the information can be unicast (using SNMP, COPS-PR, some other signalling). Finally, the allocator might "do nothing" in the case of a preconfigured allocation. As some packets might be encrypted, the available packet fields for edge filtering might be only the three fields of tunnel source, tunnel destination, and DSCP. This architecture does not require more information. Edge routers generally follow the informal model of RFC 3290.

Requests can come from many sources, including hosts, applications, users, system/network admins, and trusted signals. Requests need to include the requestor's identifying information as well as information that identifies the flow, microflow, or behavior aggregate for which the request is intended. The requestor need not be either the source or destination of the request.

A request may contain such information as PDB, rate and burst of the target packet flow and the time period

when the request is to be serviced.

A DCP is agnostic about what signaling method, if any, is used. It might be standard or proprietary. An NSIS protocol, modified versions of RSVP, or even SIP might be used.

2.4 Distributing a DCP

Although a DCP appears as a single service at a single address, it can be implemented by a single entity, a fully distributed set of entities, a hierarchy of entities, or some hybrid. Distribution might be for reliability, to avoid a control bottleneck, or to reduce latency of responses. Each peer DCP entity can be given a (revocable) pool of allocations for controlled PDBs. A peer DCP entity should be able to request additional allocation, either from a central entity at the top of the hierarchy or from the collection of peer entities when it is approaching capacity. Distributed peer entities must communicate about the current state of the allocation database, i.e. whether a resource is committed and which entity currently has control over commitment. If the intent is to decrease response time and increase local autonomy, the model is one of delegating control over some resources among the entities, not one of cooperative decision-making among the entities; only one entity has control of a resource at a time. For reliability a cooperative decision-making model might be used. The DCP service address is advertised by each of the peer entities, thus SPF routing will ensure that messages go to the closest one.

Each peer entity may have a full copy of policies with respect to authorization, pre-emption, etc, or some subset. Where there are clear policy boundaries the policy rules can be localized. If not all information is available at a peer entity, it must have the capability to request it from within association of DCP entities. Discrete units of allocation may be parceled out among a distributed hierarchy. Committed (in-use) resources cannot be moved to another DCP entity or user without an explicit pre-emption step.

DCP entities must communicate about the status of their allocation databases. Thus, each needs a functional block for coordination. Figure 5 shows coordination. There should be a logging facility for the coordinated whole of the DCP information.

Fig. 5: Updating Allocations and Coordination between entities

2.5 Example DCP: Bandwidth Broker

This model derives from the approach of RFC 2638 [3, 4] which defines "agents called bandwidth brokers (BB)... that can be configured with organizational policies, keep track of the currrent allocation of marked traffic, and interpret new requests to mark traffic in light of the policies and current allocation." Further, "BBs only need to establish relationships of limited trust with their peers in adjacent domains, unlike schemes that require the setting of flow specifications in routers throughout an end-to-end path. In practical technical terms, the BB architecture makes it possible to keep state on an administrative domain basis, rather than at every router and ... make[s] it possible to confine per flow state to just the leaf routers." Figure 6 shows a BB in an ISP network with signaling crossing domain boudaries to the DCPs in attached customer networks. As we will see, it is not necessary that a dynamic DCP be available in the attached networks.

Fig. 6: BB operating in and between DS domains

In the following example, we assume a SIP-based call control structure, though any signaling structure will interact with users and BBs in a similar fashion. However, SIP is a standard protocol and is easily adapted to a variety of QoS control planes. Any sesssion might be admission controlled by the user network. The focus for this example is a voice or video call, from Fluffy, in network Neighbor1, to Fido, in network Neighbor2 across an ISP. The steps that are taken:

1. Fluffy's sends a SIP Invite message which is held at the local SIP server while local policy and QoS level availability are consulted through the BB. The SIP server would use a QoS precondition as described in RFC 3312. The BB uses such information as source, dest, amount, type, time. (A request might also come directly from a user, but the BB should receive the same type of information.) The Neighbor1 BB is sent a request from Fluffy asking that a flow with source address Fy:4 and destination address Fo:8 (SIP

supplies the detailed address) be configured for the VW PDB at 128 kbps from 10 am till noon and is signed by Fluffy in a secure, verifiable way.

2. From the address information, BB.N1 determines that one end of the call lies outside its borders across the ISP link and checks allocation (if requestor Fluffy is authorized to use VW on the link at this

time). The Border Router has a total allocation of 256 kbps which is currently unused.

3. If allocation exists, BB.N1 may commit it or may hold it until an indication is returned that the session is accepted by Fido. (This might involve more complex signaling of intermediate transit networks or pre-emption.) The QoS precondition for Red1 is met and the call set up information, a SIP Invite, is sent to Fido's network where the session may be refused (busy, no resources, not interested, etc.) or accepted. In either case the response returns to Neighbor1 where the SIP server handles the call/session action (busy signal, sends OK, etc) and informs BB.N1 which commits or releases the resources and keeps records. The call set up messages that cross the ISP look like any other packet bound from N1 to N2. If the QoS precondition cannot be met at N1, the SIP server will notify Fluffy the call has failed.

4. At Neighbor2, the SIP server holds the Invite while the precondition is cleared by BB.N2. This may simply involve consulting local policy and checking only local allocations controlled by BB.N2. It may involve more complex QoS control messages transiting between the intermediate network (adding bandwidth, pre-emption). Once the precondition is cleared by BB.N2, the Invite can be passed to Fido and a SIP OK returned in the case of success.

5. Once BB.N1 commits the allocation, it does any necessary configuration. This might include configuring QoS tables in edge (host-facing) routers to properly send and receive the call and configuring border (cross trust domain) routers to send and admit the call. If the call is just added to an aggregate allocation at the border router and there are

edge/host mechanisms for assuring the identity and use level of the call, then it's unlikely a border router change will be made. The Border Router policer might be pre-set to the entire feasible allocation, 256 kbps, whether or not it is all presently committed or not.

A diagram of this process is shown in figure 7. Dotted lines show control paths and dashed lines show associated configuration information that is instantiated in the device.

Fig. 7: Set up of special handling for Fluffy calling Fido

When the individual session flows exit N1's BR, the ISP border router can police on the DSCP alone or on the

source and/or destination with the DSCP. The ISP Border Router is shown as policing on the source (N1) and DSCP only, but it might also be set to police on source (N1), destination (N2), and DSCP, if desired. The N1 BB would then need to track its use of the allocation by destination.

A simple intradomain Bandwidth Broker can be configured by a network administrator with information about what flows should be admitted to which PDBs and what sort of traffic conditioning, if any, should be applied to these. The BB pushes this information out to the appropriate edge router(s) using CLI, SNMP, COPS+, or other proprietary interfaces. Only edge/border routers keep state information, keeping track of the appropriate DSCP packets should be marked with and the configuration information for any traffic conditioners needed. Interior routers only inspect DSCP for QoS decisions. More complex models don't change the packet forwarding path, just the way the BB gets its information. When packets with unmatched header fields arrive, it's possible for edge/border routers to query the BB, but this can result in problems with denial-of-service attacks or in the fact of misconfiguration of an attached network. More likely is that unmatched packets will be dropped and network management/alert signals will be generated giving drops by DSCP, alerts for exceeding rate allocations.

3 InterDomain QoS Issues for the BB model

When independently adminstered domains connect, they may not be using the same PDB internally to supply quality of service characteristics that are essentially the same. The SLA between the two domains will need to include the mutually agreed identifier for each type of transit service. For example, ISP1 may have a PDB called "Virtual Wire" that is used to implement an SLS it offers attached networks and an attached network might have a PDB called "Guaranteed Service" that it wishes to be carried across the ISP with the attributes that are in ISP's VW PDB. Either the ISP might make the VW name and attributes available in the SLA, in which case the attached network asks for allocations of VW (into which it places its GS packets), or there is some name for the service that both agree to use, such as Premium. Regardless of the implementation, it's important to keep in mind that the PDB is a tool each domain uses in creating an externally visible service. The SLS covers packet treatments once they arrive at the network boundaries.

3.1 Static configuration

The link between two networks may be configured for allocations of jointly accepted PDBs that are both realizable and expected to handle requirements. For example, there may be some number of telephony users that one network wishes to support across the link. Erlang formulas can be used to arrive at the amount of bandwidth needed. If this amount of bandwidth is supportable with the appropriate characteristics across the link, then this should be allocated. Otherwise some compromise is needed.

The allocations that each domain agrees to for upstream must be supportable to anywhere or to a list of destinations that is specified to the upstream domain at the particular agreed service level. The downstream domain reserves some portion of the appropriate PDB that supplies this service for use by the upstream domain's traffic. In most cases, it makes sense for the downstream domain to expect the upstream domain to use the DSCP it specifies for this service. Then the downstream domain merely polices this DSCP from the

upstream domain. It is possible that the downstream domain may want to hide which DSCP it is using in which case, it will need to remark the packets on ingress. If there is a direct connection between the two, the policers at each domain's ingress are set to police only on DSCP, otherwise it may be necessary to inspect the source field.

3.2 Static allocation with requests

Tthe total ingress aggregates are pre-allocated and all policers and shapers set accordingly but some agreements specify an "ask before use" message for all or part of the allocation. One application of this is where resources are somewhat overallocated in the ISP, so there is some possibility of the additional bandwidth not being available. Another might be for the case where topology changes make more or less bandwidth available. The "don't ask" amount could be a minimum that is expected to be available if the link is at all functional.

Consider an example where D1's agreement with ISP is for a static allocation of Premium traffic with rules on when requests must be made of ISP. ISP uses the VW PDB for this traffic and wants to carefully account for use. Referring to figure 8, assume that D1's Border Router can handle up to 256 kbps of type VW traffic, but only 128 kbps is committed. Fluffy's request means that the D1 network must ask the next network, ISP, for permission to use the remaining 128 kbps. BB.D1 aims a request at ISP's BB for 128 kbps of type P traffic from 10am till noon. The destination (D2) may be given. A

secure association between BB.D1 and BB.ISP must be ensured. BB.D1 waits for ISP's BB to return a reply.

Fig. 8: Static allocation with request policies

BB.ISP consults its policies with regards to the requestor, BB.D1, and the two session endpoints, D1 and D2. Since R2 has a policy not to be asked about commitments below 128 kbps, BB.ISP increases the committed amount to 128 kbps, increases its committed amount from D1 to 256 kbps and returns an okay to BB.D1. The policy "accept commit" means that the network is configured to accept whatever the committed amount

is. Here, we show the Border Router policer for P traffic from D1 preconfigured at its 256kbps maximum which means BB.ISP does not need to change configuration information. It is also possible to set the policer for only the current committed amount, with a floor of the "don't ask" amount. In that case, BB.ISP must reconfigure the BR policer with each signaling transaction.

Both directions of a session could be configured through the same messages or they might be done with independent messages. "Ask" commitments should have a limited lifetime and/or time out if not refreshed.

This covers the resource set up for a session that requires special service. The call control messages, if needed, are carried by ISP transparently, the call information being instantiated at each end network, D1 and D2. A discussion of this, using SIP, was covered in the last section.

3.3 End-to-end request with signaling

In general, for a multi-network connection, requests get passed along from each BB to the next hop BB until either the final okay is received or a no (see RFC2638 examples). A no might return information about which domain or direction said no. If a domain has the resources but needs to pass the request on to the next domain, it should put a hold on the resources that is released by the return of a no or committed by the return of an okay or timed out.

The example shown in figure 9 is for Fluffy in N1 calling Fido in N2, where N1 is connected to ISP1 and N2 to ISP2. In this case, we do not assume a BB for the ISP2, but merely an agreement between ISP2 and ISP1 to mutually accept 512kbps of P-marked traffic which must conform to a specific profile (e.g., the 512 kbps is burst-limited by agreement one packet per millisecond). Assume N2 polices its incoming traffic from ISP2 to

128kbps, in agreement with ISP2. Assume N1 and N2 are using SIP, similarly to the previous example. When Fluffy initiates the call,

1. SIP.N1 gets the session invite and sends a request

for the 128kbps necessary to BB.N1.

2. BB.N1 determines that Fluffy is authorized and that the 128 kbps can be carried to N1's border. BB.N1 asks BB.ISP1 for 128kbps of type P to the destination address (tunnel endpoint gets put in through the Message Guard/HAIPE device) for Fido [does SIP server coordinate with the MG or does MG "fix up" the request?].

3. BB.TC determines that the destination address is one that is routed through ISP2, notes that there are sufficient uncommitted resources both on the TC-ISP2 link and on the path between N1's ingress and the egress to ISP2, so returns an okay to BB.N1 as well as configuring the Border Router where N1 is attached to police for 128kbps of P from N1 (possibly also including destination). BB.TC updates its entries.

4. BB.N1 lets SIP.N1 know that the quality level for the call has been set up (as far as it can) and may configure its own Border and Edge Routers at this time. (Steps 1-3 may include a bidirectional reservation of 128 kbps or that direction could be handled during the response from Fido in N2.)

5. SIP.N1 sends a message to SIP.N2 asking if Fido can/will accept a call from Fluffy. This is sent as an ordinary data message.

6. SIP.N2 checks with BB.N2 for allocation of P type traffic for this session. BB.N2 may reject the session for either policy or resource reasons. BB.N2 has only a static agreement with ISP2, so no further signaling is required to check QoS availability at N2. If okay, BB.N2 puts a hold on the resource while it signals Fido. If Fido accepts the call, the resource is committed and an okay is returned to SIP.N1.

7. SIP.N1 messages BB.N1 to commit all resources and the call starts (in one direction) or set up begins in the other direction with BB.N1 signalling for incoming allocation now.

Fig 9: Interdomain QoS with signaling

4 Domain Managed QoS and Prototype

This section describes a large scale prototype the authors are implementing. The architecture is called Domain-Managed QoS and is a BB-based Diffserv control plane that can interoperate with ARSVP. We adapted a diffserv control plane to a network with some challenges that would not appear in most networks, specifically, a satellite network with long delays, functioning with encryption, and paying particular attention to precedence and pre-emption issues.

4.1 Overview of DMQ and components

We made an early commitment to a service oriented architecture, with specific functional optimizations where needed for performance (see figure 10). The service bus provides a basis for integration which is quite flexible, scales very well, and allows us to build very generalized components for use across the enterprise. XML based syntax is expected to replace other syntax for encapsulation of network and service data. In addition, UDDI is robust and performs well for registry and discovery.

Fig 10 Domain Managed QoS Bandwidth Broker Executive Component (as implemented)

After evaluating several approaches to management of policy, XACML was found to make an excellent basis for exchange of authentication and resource allocation policy.

The service architecture provided the basis for us to integrate both XML based messaging (examples shown in this draft), and ARSVP components that our team built, to provide a choice of capability for user equipment. Other messaging formats are easily integrated to provide additional interoperation flexibility.

We sought out commercial off-the-shelf products where possible. It was a real help to find some products we could use, but a great deal of customization was also required to integrate all functions and to implement a working prototype. A benefit of IETF work on the Diffserv control plane could be more available product options and easier integration.

For management and execution of node, element, agent, and service configuration, we selected and integrated Intelliden R Series. The key feature we were looking

for was information model driven architectures for configuration management. We evaluated several other packages and architectures for this function, but found the flexibility provided by the information model centric approach made our life easier.

Large scale network management provided FCAPS (fault, configuration, accounting/cost, performance, security) generated and correlated network state, focused on individual boxes and links. For this implementation we integrated with Telcordia Surveillance Manager for Fault Events, and Telcordia Service Director for performance management generated events.

To provide network topology, we used Packet Design's Route Explorer. Routing protocol exchanges proved to be quite valuable for deriving current network state relevant to resource allocation.

4.2 Resource allocation in DMQ

In DMQ, the acquisition of resource allocations for services requiring quality of service treatment is initiated by a requesting entity submitting a Traffic Profile Request (TPR) and receiving a Traffic Conditioning Agreement (TCA). The TPR describes the specific traffic stream requirements of the service being requested by the requestor for treatment across the DS domain. It includes the source and destination IP addresses (or source/destination tunnel pairs when communicating over encrypted tunnels), the amount of bandwidth required, the DSCP, the priority of the particular service for use in precedence and preemption scenarios, when the service is to start and how long the service will be needed (complete TPR schema follows). The TCA is the response to a TPR (schema can be reviewed in a subsequent section). It provides status, a unique identifier the user can use to check status on the requested services as well as summary information extracted from the TPR. Both of these messages are based on an XML schema.

Once the TPR is received by the Bandwidth Broker QoS service, the Request Handler Component of the Request Manager acts as a controller for the traffic request.

The Authentication Component controlled by the Request Manager performs an authentications step by consulting its policy database to make sure the requestor is a known user and in good standing i.e. currently logged into the system. The TPR is then archived into persistent storage. The Policy Component controlled by the Request Handler then performs an authorization step by consulting its policy database to validate that the requestor is allowed to make the request described in the TPR. This validation step has been designed to be very flexible. Examples of policy checks are to determine if the size of traffic being requested fits the amount allowed for a requestor, and if the total

amount allowed for the community of interest the requestor belongs to has not been exceeded. Other examples of policy checks include determining whether a requestor is authorized to request a particular service at the requested QoS during a particular time or day, or a precedence and preemption policy which determines which traffic streams need to be preempted due to topology changes that affect bandwidth which is determined by the Allocation Engine of the BB by consulting its policy database and Network State Manager.

Once the policy check is complete the Resource Controller of the Request Manager routes the message to the Resource Allocation Component of the Allocation Engine. The Resource Allocator determines the path the message would travel. Two paths are determined, source-to-dest and dest-to-source. The reason for two paths is that the TPR contains two bandwidth amounts, one for the source and one for the return. This allows even tighter control of the bandwidth allocation. For example, if a video feed was being requested then the amount being pulled from the source would be very high but the amount needed from the destination to the source would be low. By allowing such taxonomy in the bandwidth request allows for finer management of the network's bandwidth.

Once the path has been determined then it is reserved. Each segment of the path is represented in persistent storage and the amount being requested is deducted from each Network Element (NE) along the path. The configuration of each NE is derived and stored with the

request. The configuration consists of setting policers and shapers. Once the reservation has been modeled the Allocator returns to the controller. The Request controller places the allocation request on a work flow queue, generates a TCA and returns the TCA to the requestor.

The Configuration Component of the Allocation Engine is responsible for monitoring the reservation queue and retrieving reservations that are due to go active within the near future; the lead time is configurable, fifteen minutes was used in one implementation by Lockheed Martin. Before issuing the configuration commands, the request is sent to the Conflict & Contention Component of the Network State Manager. This component is responsible for receiving events from external sources that may affect available bandwidth. It also has access to Layer 2 and Layer 3 fault, performance and security events, MAC, and Inventory management information. The request is vetted against the current network state to validate that it can still be serviced. If it cannot then it is dropped and a

status indicating such is recorded in its record. Otherwise, a green light is given to the Configuration Component which then translates the configuration request into specific command sets for each NE to be configured along the source and destination paths.

The above described how to reserve bandwidth under the QoS mechanism. Another feature that was implemented was an active monitor that received external events and sent them to the Conflict & Contention component. When these events were received, the event was applied against active traffic to determine if reallocation was necessary. An example would be if network capacity suddenly decreased in one segment of the network by 50%. The event would be received and the Allocation Engine would consult its policy database to determine how to apply the bandwidth decrease against active traffic streams and determine which ones could be dropped so higher priority streams would not be affected by the drop. Notifications were sent to all requesting entities of the traffic streams that were dropped. Also, the bandwidth capacity of the affected NEs was modified in the Inventory Management system so that new requests would be applied against the updated

values to ensure compliance with the new capacity. When capacity is restored, the NEs bandwidth capacity is automatically updated via a received event indicating the network has been repaired.

Another feature that was implemented was the ability for the Policy Component of the BB to interoperate with an ARSVP-over-DiffServ implementation. This was enabled by providing ARSVP aggregator/deaggregator agents that converted the ARSVP requests into BB QoS requests. An ARSVP component was constructed that generated ARSVP signaling commands that were sent to ARSVP agents which then signaled the ARSVP requestors and receivers.

4.3 Schema of TPR

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 4 U
(http://www.xmlspy.com) by Kanvasi Tejasen (Lockheed
Martin) -->
<!--W3C Schema generated by XMLSPY v2004 rel. 4 U
(http://www.xmlspy.com)-->
<xs:schema
targetNamespace="http://lmco.com/anl/request/ws/
TrafficProfileRequest.xsd"
elementFormDefault="qualified"
attributeFormDefault="qualified" id="Filters"
xmlns:mstns="http://tempuri.org/TrafficProfileRequest.xsd"
xmlns="http://lmco.com/anl/request/ws/TrafficProfileRequest.xsd"
```



```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
<xs:element name="trafficProfileRequest">
<xs:complexType>
<xs:sequence>
<xs:element name="trackIDTPR" type="xs:string"/>
<xs:element name="COIName" type="xs:string"/>
<xs:element name="entityID" type="xs:string"/>
<xs:element name="entityPW" type="xs:string"/>
<xs:element name="authenticationMethod" type="xs:string"/>
<xs:element ref="serviceType"/>
<xs:element name="serviceTime" type="serviceTimeType"/>
<xs:element name="sourceInfo" type="sourceInfoType"/>
<xs:element name="destinationInfo" type="destinationInfoType"/>
</xs:sequence>
</xs:complexType>
```

```
</xs:element>
<xs:element name="precedenceLevel">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="routine"/>
<xs:enumeration value="priority"/>
<xs:enumeration value="immediate"/>
<xs:enumeration value="flash"/>
<xs:enumeration value="flashOverride"/>
<xs:enumeration value="flashFlashOverride"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="serviceType">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="virtual wire"/>
<xs:enumeration value="control effort"/>
<xs:enumeration value="best effort"/>
</xs:restriction>
<xs:complexType name="destinationInfoType">
<xs:sequence>
<xs:element name="addressIP" type="xs:string"/>
<xs:element name="requestedBW" type="requestedBWType"/>
<xs:element ref="precedenceLevel"/>
</xs:sequence>
</xs:complexType>
<xs:element name="durationTime" type="xs:integer"/>
<xs:element name="durationUnit" type="xs:string"/>
<xs:complexType name="durationofFlowType">
<xs:sequence>
<xs:element ref="durationTime"/>
<xs:element ref="durationUnit"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="serviceTimeType">
<xs:sequence>
```



```
<xs:element ref="startDateTime"/>
<xs:element name="durationofFlow" type="durationofFlowType"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="sourceInfoType">
<xs:sequence>
<xs:element name="addressIP" type="xs:string"/>
<xs:element name="requestedBW" type="requestedBWType"/>
```

```
<xs:element ref="precedenceLevel"/>
</xs:sequence>
</xs:complexType>
<xs:element name="startDateTime" type="xs:dateTime"/>
<xs:complexType name="requestedBWType">
<xs:simpleContent>
<xs:extension base="xs:long">
<xs:attribute name="unit" type="xs:string" use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:schema>
```

4.4 Schema of TCA

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://lmco.com/anl/request/ws/
TrafficConditionAgreement.xsd"
elementFormDefault="qualified"
attributeFormDefault="qualified" id="Filters"
xmlns:mstns="http://tempuri.org/TrafficConditionAgreement.xsd"
xmlns="http://lmco.com/anl/request/ws/TrafficConditionAgreement.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
<xs:element name="trafficConditionAgreement">
<xs:complexType>
<xs:sequence>
<xs:element ref="requestEntityID"/>
<xs:element ref="approvalInfo"/>
<xs:element ref="usageTime"/>
<xs:element ref="trackIDTPR"/>
</xs:sequence>
<xs:attribute name="TCAID" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="trackIDTPR" type="xs:string"/>
<!-- to refer to the corresponding traffic profile
request -->
<xs:element name="requestEntityID" type="xs:string"/>
<xs:element name="approvalInfo">
<xs:complexType>
<xs:sequence>
<xs:element ref="approvalStatus"/>
<xs:element ref="approvalStatement"/>
```

```
<xs:element ref="denialCode"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="approvalStatus" type="xs:boolean"/>
<xs:element name="approvalStatement" type="xs:string"/>
<xs:element name="denialCode" type="xs:integer"/>
<xs:element name="startTimeConfirm" type="xs:dateTime"/>
<xs:element name="leaseTime">
<xs:complexType>
<xs:sequence>
<xs:element ref="leaseTimeLength"/>
<xs:element ref="leaseTimeUnit"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="usageTime">
<xs:complexType>
<xs:sequence>
<xs:element ref="startTimeConfirm"/>
<xs:element ref="leaseTime"/>
<xs:element ref="BWRequestRenewalTime"/>
<xs:element ref="bandwidthRate"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="leaseTimeLength" type="xs:integer"/>
<xs:element name="leaseTimeUnit" type="xs:string"/>
<xs:element name="BWRequestRenewalTime" type="xs:dateTime"/>
<xs:element name="bandwidthRate">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:integer">
<xs:attribute name="unit" type="xs:string" use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:schema>
```

5 Security Considerations

The general security considerations of [RFC2474] and [RFC2475] apply. Messaging protocols must be secured. Communication with the agent in the router must not become a vehicle for denial of service attacks.

6 Acknowlegements

Many folks helped on DMQ, including Yadu Zambre on policy and security issues and many others including Peter Paluzzi, Javier

References

[1] RFC2474, "Definition of the Differentiated Services
Field (DS Field) in the IPv4 and IPv6 Headers", K.
Nichols, S. Blake, F. Baker, D. Black,
www.ietf.org/rfc/rfc2474.txt, Dec 1998.

[2] RFC 2475, "An Architecture for Differentiated Services",
S. Blake et. al, www.ietf.org/rfc/rfc2475.txt, Dec 1998.

[3] RFC2638, "A Two-bit Differentiated Services Architecture
for the Internet", K. Nichols, V. Jacobson, and L.
Zhang, www.ietf.org/rfc/rfc2638.{txt,ps}

[4] RFC 2598, "An Expedited Forwarding PHB", V. Jacobson, K.
Nichols, K. Poduri, ftp://ftp.isi.edu/ in-notes/rfc2598.txt

[5] RFC3086, "Definition of Differentiated Services
Per-domain Behaviors and Rules for their Specification",
K.Nichols and B.Carpenter, RFC 3086,
www.ietf.org/rfc/rfc3086.txt, April, 2001.

[6] RFC3290, "An Informal Management Model for Diffserv Routers,"
Y. Bernet et. al, www.ietf.org/rfc/rfc3290.txt

[7] RFC3312, "Integration of Resource Management and Session
Initiation Protocol (SIP)," Camarillo et. al., RFC 3312.

[8] RFC3662 "A Lower Effort Per-Domain Behavior for
Differentiated Services,"
draft-bless-diffserv-pdb-le-01, R. Bless, K. Nichols,
K. Wehrle, /www.ietf.org/rfc/rfc3662.txt, December,
2003.

[9] "A Scalable Model for Interbandwidth Broker Resource
Reservation and Provisioning," IEEE Journal on Selected
Areas on Communiations, Vol 22, No 10, December 2004,
pp. 2019-2034.

[10] Keith Kim, Petros Mouchtaris, Sunil Samtani, Rajesh

Talpade, Larry Wong, "A Bandwidth Broker Architecture
for VoIP QoS", in Proceedings of SPIE's International
Symposium on Convergence of IT and Communications
(ITCom), Colorado, August 2001.

[11] "Operax Resource Manager in TITAAN", application note
from Operax available at:
operax.com/docs/operax_resource_manager_titaan_product_sheetD52.pdf

[12] "A Quality of Service Architecture that Combines

Resource Reservation and Application Adaptation", I.
Foster, A. Roy, V. Sander. 8th International Workshop
on Quality of Service, 2000.

[13] "PacketCable(TM) Dynamic Quality-of-Service Specification"
, PKT-SP-DQOS-1-8-040113, available from www.cablelabs.com

[14] "Differentiated Services in the Internet", B. Carpenter
and K. Nichols, Proceedings of the IEEE, vol 90 no 9,
September, 2002, pp. 1479-1494.

[15] Multiservice Forum White Papers at www.msforum.org

[16] "End-to-End Provision of Policy Information for Network QoS"
, V. Sander et. al., Proceedings of the Tenth IEEE
Symposium on High Performance Distributed Computing
(HPDC), August, 2001.

[17] "Differentiated Services in the Internet", B. Carpenter
and K. Nichols, Proceedings of the IEEE, vol 90 no 9,
September, 2002, pp. 1479-1494.

[18] "A Per-Domain Behavior for Circuit Emulation in IP Networks,"
K. Nichols, V. Jacobson, K. Poduri, ACM CCR, April 2004.

[19] "Differentiated Services for the Internet", V. Jacobson,
First Internet2 Joint Applications/Engineering QoS
Workshop Proceedings, May 21-22, 1998, Santa Clara CA, pp26-31.

[20] G. Hoo and W. Johnston, "QoS as Middleware: Bandwidth
Reservation System Design", LBNL tech report, 1999, at
http://www-itg.lbl.gov/QoS/

Authors' Addresses

Kathleen Nichols
Pollere LLC
325M Sharon Park Drive #214
Menlo Park, CA 94025
USA

email: nichols@pollere.com

J. Pulliam
R. Barrios
L. Sampson
K. Adams
Lockheed Martin
San Jose, CA 95161
USA
email: jeffrey.s.pulliam@lmco.com

IPR Disclosure