Network Working Group                                        G. Huston
Request for Comments: 2990                                     Telstra
Category: Informational                                 November 2000


                Next Steps for the IP QoS Architecture

Abstract

   While there has been significant progress in the definition of
   Quality of Service (QoS) architectures for internet networks, there
   are a number of aspects of QoS that appear to need further
   elaboration as they relate to translating a set of tools into a
   coherent platform for end-to-end service delivery.  This document
   highlights the outstanding architectural issues relating to the
   deployment and use of QoS mechanisms within internet networks, noting
   those areas where further standards work may assist with the
   deployment of QoS internets.

   This document is the outcome of a collaborative exercise on the part
   of the Internet Architecture Board.

Table of Contents

1. Introduction

   The default service offering associated with the Internet is
   characterized as a best-effort variable service response.  Within
   this service profile the network makes no attempt to actively
   differentiate its service response between the traffic streams
   generated by concurrent users of the network.  As the load generated
   by the active traffic flows within the network varies, the network's
   best effort service response will also vary.

   The objective of various Internet Quality of Service (QoS) efforts is
   to augment this base service with a number of selectable service
   responses.  These service responses may be distinguished from the
   best-effort service by some form of superior service level, or they
   may be distinguished by providing a predictable service response
   which is unaffected by external conditions such as the number of
   concurrent traffic flows, or their generated traffic load.

   Any network service response is an outcome of the resources available
   to service a load, and the level of the load itself.  To offer such
   distinguished services there is not only a requirement to provide a
   differentiated service response within the network, there is also a
   requirement to control the service-qualified load admitted into the
   network, so that the resources allocated by the network to support a
   particular service response are capable of providing that response
   for the imposed load.  This combination of admission control agents
   and service management elements can be summarized as "rules plus
   behaviors". To use the terminology of the Differentiated Service
   architecture [4], this admission control function is undertaken by a
   traffic conditioner (an entity which performs traffic conditioning
   functions and which may contain meters, markers, droppers, and
   shapers), where the actions of the conditioner are governed by
   explicit or implicit admission control agents.

   As a general observation of QoS architectures, the service load
   control aspect of QoS is perhaps the most troubling component of the
   architecture.  While there are a wide array of well understood
   service response mechanisms that are available to IP networks,

matching a set of such mechanisms within a controlled environment to
respond to a set of service loads to achieve a completely consistent
service response remains an area of weakness within existing IP QoS
architectures.  The control elements span a number of generic
requirements, including end-to-end application signaling, end-to-
network service signaling and resource management signaling to allow
policy-based control of network resources.  This control may also
span a particular scope, and use 'edge to edge' signaling, intended
to support particular service responses within a defined network
scope.

One way of implementing this control of imposed load to match the
level of available resources is through an application-driven process
of service level negotiation (also known as application signaled
QoS).  Here, the application first signals its service requirements
to the network, and the network responds to this request.  The
application will proceed if the network has indicated that it is able
to carry the additional load at the requested service level.  If the
network indicates that it cannot accommodate the service requirements
the application may proceed in any case, on the basis that the
network will service the application's data on a best effort basis.
This negotiation between the application and the network can take the
form of explicit negotiation and commitment, where there is a single
negotiation phase, followed by a commitment to the service level on
the part of the network.  This application-signaled approach can be
used within the Integrated Services architecture, where the
application frames its service request within the resource
reservation protocol (RSVP), and then passes this request into the
network.  The network can either respond positively in terms of its
agreement to commit to this service profile, or it can reject the
request.  If the network commits to the request with a resource
reservation, the application can then pass traffic into the network
with the expectation that as long as the traffic remains within the
traffic load profile that was originally associated with the request,
the network will meet the requested service levels.  There is no
requirement for the application to periodically reconfirm the service
reservation itself, as the interaction between RSVP and the network
constantly refreshes the reservation while it remains active.  The
reservation remains in force until the application explicitly
requests termination of the reservation, or the network signals to
the application that it is unable to continue with a service
commitment to the reservation [3].  There are variations to this
model, including an aggregation model where a proxy agent can fold a
number of application-signaled reservations into a common aggregate
reservation along a common sub-path, and a matching deaggregator can
reestablish the collection of individual resource reservations upon
leaving the aggregate region [5].  The essential feature of this
Integrated Services model is the "all or nothing" nature of the

model.  Either the network commits to the reservation, in which case
the requestor does not have to subsequently monitor the network's
level of response to the service, or the network indicates that it
cannot meet the resource reservation.

An alternative approach to load control is to decouple the network
load control function from the application.  This is the basis of the
Differentiated Services architecture.  Here, a network implements a
load control function as part of the function of admission of traffic
into the network, admitting no more traffic within each service
category as there are assumed to be resources in the network to
deliver the intended service response.  Necessarily there is some
element of imprecision in this function given that traffic may take
an arbitrary path through the network.  In terms of the interaction
between the network and the application, this takes the form of a
service request without prior negotiation, where the application
requests a particular service response by simply marking each packet
with a code to indicate the desired service.  Architecturally, this
approach decouples the end systems and the network, allowing a
network to implement an active admission function in order to
moderate the workload that is placed upon the network's resources
without specific reference to individual resource requests from end
systems.  While this decoupling of control allows a network's
operator greater ability to manage its resources and a greater
ability to ensure the integrity of its services, there is a greater
potential level of imprecision in attempting to match applications'
service requirements to the network's service capabilities.

2. State and Stateless QoS

These two approaches to load control can be characterized as state-
based and stateless approaches respectively.

The architecture of the Integrated Services model equates the
cumulative sum of honored service requests to the current reserved
resource levels of the network.  In order for a resource reservation
to be honored by the network, the network must maintain some form of
remembered state to describe the resources that have been reserved,
and the network path over which the reserved service will operate.
This is to ensure integrity of the reservation.  In addition, each
active network element within the network path must maintain a local
state that allows incoming IP packets to be correctly classified into
a reservation class.  This classification allows the packet to be
placed into a packet flow context that is associated with an
appropriate service response consistent with the original end-to-end
service reservation.  This local state also extends to the function

of metering packets for conformance on a flow-by-flow basis, and the
additional overheads associated with maintenance of the state of each
of these meters.

In the second approach, that of a Differentiated Services model, the
packet is marked with a code to trigger the appropriate service
response from the network elements that handles the packet, so that
there is no strict requirement to install a per-reservation state on
these network elements.  Also, the end application or the service
requestor is not required to provide the network with advance notice
relating to the destination of the traffic, nor any indication of the
intended traffic profile or the associated service profile.  In the
absence of such information any form of per-application or per-path
resource reservation is not feasible.  In this model there is no
maintained per-flow state within the network.

The state-based Integrated Services architectural model admits the
potential to support greater level of accuracy, and a finer level of
granularity on the part of the network to respond to service
requests.  Each individual application's service request can be used
to generate a reservation state within the network that is intended
to prevent the resources associated with the reservation to be
reassigned or otherwise preempted to service other reservations or to
service best effort traffic loads.  The state-based model is intended
to be exclusionary, where other traffic is displaced in order to meet
the reservation's service targets.

As noted in RFC2208 [2], there are several areas of concern about the
deployment of this form of service architecture.  With regard to
concerns of per-flow service scalability, the resource requirements
(computational processing and memory consumption) for running per-
flow resource reservations on routers increase in direct proportion
to the number of separate reservations that need to be accommodated.
By the same token, router forwarding performance may be impacted
adversely by the packet-classification and scheduling mechanisms
intended to provide differentiated services for these resource-
reserved flows.  This service architecture also poses some challenges
to the queuing mechanisms, where there is the requirement to allocate
absolute levels of egress bandwidth to individual flows, while still
supporting an unmanaged low priority best effort traffic class.

The stateless approach to service management is more approximate in
the nature of its outcomes.  Here there is no explicit negotiation
between the application's signaling of the service request and the
network's capability to deliver a particular service response.  If
the network is incapable of meeting the service request, then the
request simply will not be honored.  In such a situation there is no
requirement for the network to inform the application that the

request cannot be honored, and it is left to the application to
determine if the service has not been delivered.  The major attribute
of this approach is that it can possess excellent scaling properties
from the perspective of the network.  If the network is capable of
supporting a limited number of discrete service responses, and the
routers uses per-packet marking to trigger the service response, then
the processor and memory requirements in each router do not increase
in proportion to the level of traffic passed through the router.  Of
course this approach does introduce some degree of compromise in that
the service response is more approximate as seen by the end client,
and scaling the number of clients and applications in such an
environment may not necessarily result in a highly accurate service
response to every client's application.

It is not intended to describe these service architectures in further
detail within this document.  The reader is referred to RFC1633 [3]
for an overview of the Integrated Services Architecture (IntServ) and
RFC2475 [4] for an overview of the Differentiated Services
architecture (DiffServ).

These two approaches are the endpoints of what can be seen as a
continuum of control models, where the fine-grained precision of the
per application invocation reservation model can be aggregated into
larger, more general and potentially more approximate aggregate
reservation states, and the end-to-end element-by-element reservation
control can be progressively approximated by treating a collection of
subnetworks or an entire transit network as an aggregate service
element.  There are a number of work in progress efforts which are
directed towards these aggregated control models, including
aggregation of RSVP [5], the RSVP DCLASS Object [6] to allow
Differentiated Services Code Points (DSCPs) to be carried in RSVP
message objects, and operation of Integrated Services over
Differentiated Services networks [7].

3. Next Steps for QoS Architectures

Both the Integrated Services architecture and the Differentiated
Services architecture have some critical elements in terms of their
current definition which appear to be acting as deterrents to
widespread deployment.  Some of these issues will probably be
addressed within the efforts to introduce aggregated control and
response models into these QoS architectures, while others may
require further refinement through standards-related activities.

3.1 QoS-Enabled Applications

   One of the basic areas of uncertainty with QoS architectures is
   whether QoS is a per-application service, whether QoS is a
   transport-layer option, or both.  Per-application services have
   obvious implications of extending the QoS architecture into some form
   of Application Protocol Interface (API), so that applications could
   negotiate a QoS response from the network and alter their behavior
   according to the outcome of the response.  Examples of this approach
   include GQOS [8], and RAPI [9].  As a transport layer option, it
   could be envisaged that any application could have its traffic
   carried by some form of QoS-enabled network services by changing the
   host configuration, or by changing the configuration at some other
   network control point, without making any explicit changes to the
   application itself.  The strength of the transport layer approach is
   that there is no requirement to substantially alter application
   behavior, as the application is itself unaware of the
   administratively assigned QoS.  The weakness of this approach is that
   the application is unable to communicate what may be useful
   information to the network or to the policy systems that are managing
   the network's service responses.  In the absence of such information
   the network may provide a service response that is far superior than
   the application's true requirements, or far inferior than what is
   required for the application to function correctly.  An additional
   weakness of a transport level approach refers to those class of
   applications that can adapt their traffic profile to meet the
   available resources within the network.  As a transport level
   mechanism, such network availability information as may be available
   to the transport level is not passed back to the application.

   In the case of the Integrated Services architecture, this transport
   layer approach does not appear to be an available option, as the
   application does require some alteration to function correctly in
   this environment.  The application must be able to provide to the
   service reservation module a profile of its anticipated traffic, or
   in other words the application must be able to predict its traffic
   load.  In addition, the application must be able to share the
   reservation state with the network, so that if the network state
   fails, the application can be informed of the failure.  The more
   general observation is that a network can only formulate an accurate
   response to an application's requirements if the application is
   willing to offer precise statement of its traffic profile, and is
   willing to be policed in order to have its traffic fit within this
   profile.

   In the case of the Differentiated Services architecture there is no
   explicit provision for the application to communicate with the
   network regarding service levels.  This does allow the use of a

transport level option within the end system that does not require
explicit alteration of the application to mark its generated traffic
with one of the available Differentiated Services service profiles.
However, whether the application is aware of such service profiles or
not, there is no level of service assurance to the application in
such a model.  If the Differentiated Services boundary traffic
conditioners enter a load shedding state, the application is not
signaled of this condition, and is not explicitly aware that the
requested service response is not being provided by the network.  If
the network itself changes state and is unable to meet the cumulative
traffic loads admitted by the ingress traffic conditioners, neither
the ingress traffic conditioners, nor the client applications, are
informed of this failure to maintain the associated service quality.
While there is no explicit need to alter application behavior in this
architecture, as the basic DiffServ mechanism is one that is managed
within the network itself, the consequence is that an application may
not be aware whether a particular service state is being delivered to
the application.

There is potential in using an explicit signaling model, such as used
by IntServ, but carrying a signal which allows the network to manage
the application's traffic within an aggregated service class [6].
Here the application does not pass a complete picture of its intended
service profile to the network, but instead is providing some level
of additional information to the network to assist in managing its
resources, both in terms of the generic service class that the
network can associate with the application's traffic, and the
intended path of the traffic through the network.

An additional factor for QoS enabled applications is that of receiver
capability negotiation.  There is no value in the sender establishing
a QoS-enabled path across a network to the receiver if the receiver
is incapable of absorbing the consequent data flow.  This implies
that QoS enabled applications also require some form of end-to-end
capability negotiation, possibly through a generic protocol to allow
the sender to match its QoS requirements to the minimum of the flow
resources that can be provided by the network and the flow resources
that can be processed by the receiver.  In the case of the Integrated
services architecture the application end-to-end interaction can be
integrated into the RSVP negotiation.  In the case of the
Differentiated Services architecture there is no clear path of
integrating such receiver control into the signaling model of the
architecture as it stands.

If high quality services are to be provided, where 'high quality' is
implied as being 'high precision with a fine level of granularity',
then the implication is that all parts of the network that may be
involved with servicing the request either have to be over-

   provisioned such that no load state can compromise the service
   quality, or the network element must undertake explicit allocation of
   resources to each flow that is associated with each service request.

   For end-to-end service delivery it does appear that QoS architectures
   will need to extend to the level of the application requesting the
   service profile.  It appears that further refinement of the QoS
   architecture is required to integrate DiffServ network services into
   an end-to-end service delivery model, as noted in [7].

3.2 The Service Environment

   The outcome of the considerations of these two approaches to QoS
   architecture within the network is that there appears to be no single
   comprehensive service environment that possesses both service
   accuracy and scaling properties.

   The maintained reservation state of the Integrated Services
   architecture and the end-to-end signaling function of RSVP are part
   of a service management architecture, but it is not cost effective,
   or even feasible, to operate a per-application reservation and
   classification state across the high speed core of a network [2].

   While the aggregated behavior state of the Differentiated Services
   architecture does offer excellent scaling properties, the lack of
   end-to-end signaling facilities makes such an approach one that
   cannot operate in isolation within any environment.  The
   Differentiated Services architecture can be characterized as a
   boundary-centric operational model.  With this boundary-centric
   architecture, the signaling of resource availability from the
   interior of the network to the boundary traffic conditioners is not
   defined, nor is the signaling from the traffic conditioners to the
   application that is resident on the end system.  This has been noted
   as an additional work item in the IntServ operations over DiffServ
   work, concerning "definition of mechanisms to efficiently and
   dynamically provision resources in a DiffServ network region". This
   might include protocols by which an "oracle" (...) conveys
   information about resource availability within a DiffServ region to
   border routers." [7]

   What appears to be required within the Differentiated Services
   service model is both resource availability signaling from the core
   of the network to the DiffServ boundary and some form of signaling
   from the boundary to the client application.

3.3 QoS Discovery

   There is no robust mechanism for network path discovery with specific
   service performance attributes.  The assumption within both IntServ
   and DiffServ architectures is that the best effort routing path is
   used, where the path is either capable of sustaining the service
   load, or not.

   Assuming that the deployment of service differentiating
   infrastructure will be piecemeal, even if only in the initial stages
   of a QoS rollout, such an assumption may be unwarranted.  If this is
   the case, then how can a host application determine if there is a
   distinguished service path to the destination?  No existing
   mechanisms exist within either of these architectures to query the
   network for the potential to support a specific service profile. Such
   a query would need to examine a number of candidate paths, rather
   than simply examining the lowest metric routing path, so that this
   discovery function is likely to be associated with some form of QoS
   routing functionality.

   From this perspective, there is still further refinement that may be
   required in the model of service discovery and the associated task of
   resource reservation.

3.4 QoS Routing and Resource Management

   To date QoS routing has been developed at some distance from the task
   of development of QoS architectures.  The implicit assumption within
   the current QoS architectural models is that the routing best effort
   path will be used for both best effort traffic and distinguished
   service traffic.

   There is no explicit architectural option to allow the network
   service path to be aligned along other than the single best routing
   metric path, so that available network resources can be efficiently
   applied to meet service requests.  Considerations of maximizing
   network efficiency would imply that some form of path selection is
   necessary within a QoS architecture, allowing the set of service
   requirements to be optimally supported within the network's aggregate
   resource capability.

   In addition to path selection, SPF-based interior routing protocols
   allow for the flooding of link metric information across all network
   elements.  This mechanism appears to be a productive direction to
   provide the control-level signaling between the interior of the
   network and the network admission elements, allowing the admission

systems to admit traffic based on current resource availability
rather than on necessarily conservative statically defined admission
criteria.

There is a more fundamental issue here concerning resource management
and traffic engineering.  The approach of single path selection with
static load characteristics does not match a networked environment
which contains a richer mesh of connectivity and dynamic load
characteristics.  In order to make efficient use of a rich
connectivity mesh, it is necessary to be able to direct traffic with
a common ingress and egress point across a set of available network
paths, spreading the load across a broader collection of network
links.  At its basic form this is essentially a traffic engineering
problem.  To support this function it is necessary to calculate per-
path dynamic load metrics, and allow the network's ingress system the
ability to distribute incoming traffic across these paths in
accordance with some model of desired traffic balance.  To apply this
approach to a QoS architecture would imply that each path has some
form of vector of quality attributes, and incoming traffic is
balanced across a subset of available paths where the quality
attribute of the traffic is matched with the quality vector of each
available path.  This augmentation to the semantics of the traffic
engineering is matched by a corresponding shift in the calculation
and interpretation of the path's quality vector.  In this approach
what needs to be measured is not the path's resource availability
level (or idle proportion), but the path's potential to carry
additional traffic at a certain level of quality. This potential
metric is one that allows existing lower priority traffic to be
displaced to alternative paths.  The path's quality metric can be
interpreted as a metric describing the displacement capability of the
path, rather than a resource availability metric.

This area of active network resource management, coupled with dynamic
network resource discovery, and the associated control level
signaling to network admission systems appears to be a topic for
further research at this point in time.

3.5 TCP and QoS

A congestion-managed rate-adaptive traffic flow (such as used by TCP)
uses the feedback from the ACK packet stream to time subsequent data
transmissions.  The resultant traffic flow rate is an outcome of the
service quality provided to both the forward data packets and the
reverse ACK packets.  If the ACK stream is treated by the network
with a different service profile to the outgoing data packets, it
remains an open question as to what extent will the data forwarding
service be compromised in terms of achievable throughput.  High rates
of jitter on the ACK stream can cause ACK compression, that in turn

will cause high burst rates on the subsequent data send.  Such bursts
will stress the service capacity of the network and will compromise
TCP throughput rates.

One way to address this is to use some form of symmetric service,
where the ACK packets are handled using the same service class as the
forward data packets.  If symmetric service profiles are important
for TCP sessions, how can this be structured in a fashion that does
not incorrectly account for service usage?  In other words, how can
both directions of a TCP flow be accurately accounted to one party?

Additionally, there is the interaction between the routing system and
the two TCP data flows.  The Internet routing architecture does not
intrinsically preserve TCP flow symmetry, and the network path taken
by the forward packets of a TCP session may not exactly correspond to
the path used by the reverse packet flow.

TCP also exposes an additional performance constraint in the manner
of the traffic conditioning elements in a QoS-enabled network.
Traffic conditioners within QoS architectures are typically specified
using a rate enforcement mechanism of token buckets.  Token bucket
traffic conditioners behave in a manner that is analogous to a First
In First Out queue.  Such traffic conditioning systems impose tail
drop behavior on TCP streams.  This tail drop behavior can produce
TCP timeout retransmission, unduly penalizing the average TCP goodput
rate to a level that may be well below the level specified by the
token bucket traffic conditioner.  Token buckets can be considered as
TCP-hostile network elements.

The larger issue exposed in this consideration is that provision of
some form of assured service to congestion-managed traffic flows
requires traffic conditioning elements that operate using weighted
RED-like control behaviors within the network, with less
deterministic traffic patterns as an outcome.  A requirement to
manage TCP burst behavior through token bucket control mechanisms is
most appropriately managed in the sender's TCP stack.

There are a number of open areas in this topic that would benefit
from further research.  The nature of the interaction between the
end-to-end TCP control system and a collection of service
differentiation mechanisms with a network is has a large number of
variables.  The issues concern the time constants of the control
systems, the amplitude of feedback loops, and the extent to which
each control system assumes an operating model of other active
control systems that are applied to the same traffic flow, and the
mode of convergence to a stable operational state for each control
system.

3.6 Per-Flow States and Per-Packet classifiers

   Both the IntServ and DiffServ architectures use packet classifiers as
   an intrinsic part of their architecture.  These classifiers can be
   considered as coarse or fine level classifiers.  Fine-grained
   classifiers can be considered as classifiers that attempt to isolate
   elements of traffic from an invocation of an application (a 'micro-
   flow') and use a number of fields in the IP packet header to assist
   in this, typically including the source and destination IP addresses
   and source and source and destination port addresses.  Coarse-grained
   classifiers attempt to isolate traffic that belongs to an aggregated
   service state, and typically use the DiffServ code field as the
   classifying field.  In the case of DiffServ there is the potential to
   use fine-grained classifiers as part of the network ingress element,
   and coarse-gained classifiers within the interior of the network.

   Within flow-sensitive IntServ deployments, every active network
   element that undertakes active service discrimination is requirement
   to operate fine-grained packet classifiers.  The granularity of the
   classifiers can be relaxed with the specification of aggregate
   classifiers [5], but at the expense of the precision and accuracy of
   the service response.

   Within the IntServ architecture the fine-grained classifiers are
   defined to the level of granularity of an individual traffic flow,
   using the packet's 5-tuple of (source address, destination address,
   source port, destination port, protocol) as the means to identify an
   individual traffic flow.  The DiffServ Multi-Field (MF) classifiers
   are also able to use this 5-tuple to map individual traffic flows
   into supported behavior aggregates.

   The use of IPSEC, NAT and various forms of IP tunnels result in a
   occlusion of the flow identification within the IP packet header,
   combining individual flows into a larger aggregate state that may be
   too coarse for the network's service policies.  The issue with such
   mechanisms is that they may occur within the network path in a
   fashion that is not visible to the end application, compromising the
   ability for the application to determine whether the requested
   service profile is being delivered by the network.  In the case of
   IPSEC there is a proposal to carry the IPSEC Security Parameter Index
   (SPI) in the RSVP object [10], as a surrogate for the port addresses.
   In the case of NAT and various forms of IP tunnels, there appears to
   be no coherent way to preserve fine-grained classification
   characteristics across NAT devices, or across tunnel encapsulation.

   IP packet fragmentation also affects the ability of the network to
   identify individual flows, as the trailing fragments of the IP packet
   will not include the TCP or UDP port address information. This admits

the possibility of trailing fragments of a packet within a
distinguished service class being classified into the base best
effort service category, and delaying the ultimate delivery of the IP
packet to the destination until the trailing best effort delivered
fragments have arrived.

The observation made here is that QoS services do have a number of
caveats that should be placed on both the application and the
network.  Applications should perform path MTU discovery in order to
avoid packet fragmentation.  Deployment of various forms of payload
encryption, header address translation and header encapsulation
should be undertaken with due attention to their potential impacts on
service delivery packet classifiers.

3.7 The Service Set

The underlying question posed here is how many distinguished service
responses are adequate to provide a functionally adequate range of
service responses?

The Differentiated Services architecture does not make any limiting
restrictions on the number of potential services that a network
operator can offer.  The network operator may be limited to a choice
of up to 64 discrete services in terms of the 6 bit service code
point in the IP header but as the mapping from service to code point
can be defined by each network operator, there can be any number of
potential services.

As always, there is such a thing as too much of a good thing, and a
large number of potential services leads to a set of issues around
end-to-end service coherency when spanning multiple network domains.
A small set of distinguished services can be supported across a large
set of service providers by equipment vendors and by application
designers alike.  An ill-defined large set of potential services
often serves little productive purpose.  This does point to a
potential refinement of the QoS architecture to define a small core
set of service profiles as "well-known" service profiles, and place
all other profiles within a "private use" category.

3.8 Measuring Service Delivery

There is a strong requirement within any QoS architecture for network
management approaches that provide a coherent view of the operating
state of the network.  This differs from a conventional element-by-
element management view of the network in that the desire here is to
be able to provide a view of the available resources along a

   particular path within a network, and map this view to an admission
   control function which can determine whether to admit a service
   differentiated flow along the nominated network path.

   As well as managing the admission systems through resource
   availability measurement, there is a requirement to be able to
   measure the operating parameters of the delivered service.  Such
   measurement methodologies are required in order to answer the
   question of how the network operator provides objective measurements
   to substantiate the claim that the delivered service quality
   conformed to the service specifications.  Equally, there is a
   requirement for a measurement methodology to allow the client to
   measure the delivered service quality so that any additional expense
   that may be associated with the use of premium services can be
   justified in terms of superior application performance.

   Such measurement methodologies appear to fall within the realm of
   additional refinement to the QoS architecture.

3.9 QoS Accounting

   It is reasonable to anticipate that such forms of premium service and
   customized service will attract an increment on the service tariff.
   The provision of a distinguished service is undertaken with some
   level of additional network resources to support the service, and the
   tariff premium should reflect this altered resource allocation.  Not
   only does such an incremental tariff shift the added cost burden to
   those clients who are requesting a disproportionate level of
   resources, but it provides a means to control the level of demand for
   premium service levels.

   If there are to be incremental tariffs on the use of premium
   services, then some accounting of the use of the premium service
   would appear to be necessary relating use of the service to a
   particular client.  So far there is no definition of such an
   accounting model nor a definition as to how to gather the data to
   support the resource accounting function.

   The impact of this QoS service model may be quite profound to the
   models of Internet service provision.  The commonly adopted model in
   both the public internet and within enterprise networks is that of a
   model of access, where the clients service tariff is based on the
   characteristics of access to the services, rather than that of the
   actual use of the service.  The introduction of QoS services creates
   a strong impetus to move to usage-based tariffs, where the tariff is
   based on the level of use of the network's resources.  This, in turn,
   generates a requirement to meter resource use, which is a form of
   usage accounting.  This topic was been previously studied within the

IETF under the topic of "Internet Accounting" [11], and further
refinement of the concepts used in this model, as they apply to QoS
accounting may prove to be a productive initial step in formulating a
standards-based model for QoS accounting.

3.10 QoS Deployment Diversity

It is extremely improbable that any single form of service
differentiation technology will be rolled out across the Internet and
across all enterprise networks.

Some networks will deploy some form of service differentiation
technology while others will not.  Some of these service platforms
will interoperate seamlessly and other less so.  To expect all
applications, host systems, network routers, network policies, and
inter-provider arrangements to coalesce into a single homogeneous
service environment that can support a broad range of service
responses is an somewhat unlikely outcome given the diverse nature of
the available technologies and industry business models.  It is more
likely that we will see a number of small scale deployment of service
differentiation mechanisms and some efforts to bridge these
environments together in some way.

In this heterogeneous service environment the task of service
capability discovery is as critical as being able to invoke service
responses and measure the service outcomes.  QoS architectures will
need to include protocol capabilities in supporting service discovery
mechanisms.

In addition, such a heterogeneous deployment environment will create
further scaling pressure on the operational network as now there is
an additional dimension to the size of the network.  Each potential
path to each host is potentially qualified by the service
capabilities of the path.  While one path may be considered as a
candidate best effort path, another path may offer a more precise
match between the desired service attributes and the capabilities of
the path to sustain the service.  Inter-domain policy also impacts
upon this path choice, where inter-domain transit agreements may
specifically limit the types and total level of quality requests than
may be supported between the domains.  Much of the brunt of such
scaling pressures will be seen in the inter-domain and intra-domain
routing domain where there are pressures to increase the number of
attributes of a routing entry, and also to use the routing protocol
in some form of service signaling role.

3.11 QoS Inter-Domain signaling

   QoS Path selection is both an intra-domain (interior) and an inter-
   domain (exterior) issue.  Within the inter-domain space, the current
   routing technologies allow each domain to connect to a number of
   other domains, and to express its policies with respect to received
   traffic in terms of inter-domain route object attributes.
   Additionally, each domain may express its policies with respect to
   sending traffic through the use of boundary route object filters,
   allowing a domain to express its preference for selecting one
   domain's advertised routes over another.  The inter-domain routing
   space is a state of dynamic equilibrium between these various route
   policies.

   The introduction of differentiated services adds a further dimension
   to this policy space.  For example, while a providers may execute an
   interconnection agreement with one party to exchange best effort
   traffic, it may execute another agreement with a second party to
   exchange service qualified traffic.  The outcome of this form of
   interconnection is that the service provider will require external
   route advertisements to be qualified by the accepted service
   profiles.  Generalizing from this scenario, it is reasonable to
   suggest that we will require the qualification of routing
   advertisements with some form of service quality attributes.  This
   implies that we will require some form of quality vector-based
   forwarding function, at least in the inter-domain space, and some
   associated routing protocol can pass a quality of service vector in
   an operationally stable fashion.

   The implication of this requirement is that the number of objects
   being managed by routing systems must expand dramatically, as the
   size and number of objects managed within the routing domain
   increases, and the calculation of a dynamic equilibrium of import and
   export policies between interconnected providers will also be subject
   to the same level of scaling pressure.

   This has implications within the inter-domain forwarding space as
   well, as the forwarding decision in such a services differentiated
   environment is then qualified by some form of service quality vector.
   This is required in order to pass exterior traffic to the appropriate
   exterior interconnection gateway.

3.12 QoS Deployment Logistics

   How does the widespread deployment of service-aware networks
   commence?  Which gets built first - host applications or network
   infrastructure?

No network operator will make the significant investment in
deployment and support of distinguished service infrastructure unless
there is a set of clients and applications available to make
immediate use of such facilities.  Clients will not make the
investment in enhanced services unless they see performance gains in
applications that are designed to take advantage of such enhanced
services.  No application designer will attempt to integrate service
quality features into the application unless there is a model of
operation supported by widespread deployment that makes the
additional investment in application complexity worthwhile and
clients who are willing to purchase such applications.  With all
parts of the deployment scenario waiting for the others to move,
widespread deployment of distinguished services may require some
other external impetus.

Further aspects of this deployment picture lie in the issues of
network provisioning and the associated task of traffic engineering.
Engineering a network to meet the demands of best effort flows
follows a well understood pattern of matching network points of user
concentrations to content delivery network points with best effort
paths.  Integrating QoS-mediated traffic engineering into the
provisioning model suggests a provisioning requirement that also
requires input from a QoS demand model.

4. The objective of the QoS architecture

What is the precise nature of the problem that QoS is attempting to
solve?  Perhaps this is one of the more fundamental questions
underlying the QoS effort, and the diversity of potential responses
is a pointer to the breadth of scope of the QoS effort.

All of the following responses form a part of the QoS intention:

  - To control the network service response such that the response
    to a specific service element is consistent and predictable.

  - To control the network service response such that a service
    element is provided with a level of response equal to or above a
    guaranteed minimum.

  - To allow a service element to establish in advance the service
    response that can or will be obtained from the network.

  - To control the contention for network resources such that a
    service element is provided with a superior level of network
    resource.

- To control the contention for network resources such that a
  service element does not obtain an unfair allocation of
  resources (to some definition of 'fairness').

- To allow for efficient total utilization of network resources
  while servicing a spectrum of directed network service outcomes.

Broadly speaking, the first three responses can be regarded as
'application-centric', and the latter as 'network-centric'.  It is
critical to bear in mind that none of these responses can be
addressed in isolation within any effective QoS architecture.  Within
the end-to-end architectural model of the Internet, applications make
minimal demands on the underlying IP network.  In the case of TCP,
the protocol uses an end-to-end control signal approach to
dynamically adjust to the prevailing network state.  QoS
architectures add a somewhat different constraint, in that the
network is placed in an active role within the task of resource
allocation and service delivery, rather than being a passive object
that requires end systems to adapt.

5. Towards an end-to-end QoS architecture

The challenge facing the QoS architecture lies in addressing the
weaknesses noted above, and in integrating the various elements of
the architecture into a cohesive whole that is capable of sustaining
end-to-end service models across a wide diversity of internet
platforms.  It should be noted that such an effort may not
necessarily result in a single resultant architecture, and that it is
possible to see a number of end-to-end approaches based on different
combinations of the existing components.

One approach is to attempt to combine both architectures into an
end-to-end model, using IntServ as the architecture which allows
applications to interact with the network, and DiffServ as the
architecture to manage admission the network's resources [7].  In
this approach, the basic tension that needs to be resolved lies in
difference between the per-application view of the IntServ
architecture and the network boundary-centric view of the DiffServ
architecture.

One building block for such an end-to-end service architecture is a
service signaling protocol.  The RSVP signaling protocol can address
the needs of applications that require a per-service end-to-end
service signaling environment.  The abstracted model of RSVP is that
of a discovery signaling protocol that allows an application to use a
single transaction to communicate its service requirements to both
the network and the remote party, and through the response mechanism,
to allow these network elements to commit to the service

requirements.  The barriers to deployment for this model lie in an
element-by element approach to service commitment, implying that each
network element must undertake some level of signaling and processing
as dictated by this imposed state.  For high precision services this
implies per-flow signaling and per-flow processing to support this
service model.  This fine-grained high precision approach to service
management is seen as imposing an unacceptable level of overhead on
the central core elements of large carrier networks.

The DiffServ approach uses a model of abstraction which attempts to
create an external view of a compound network as a single subnetwork.
From this external perspective the network can be perceived as two
boundary service points, ingress and egress.  The advantage of this
approach is that there exists the potential to eliminate the
requirement for per-flow state and per-flow processing on the
interior elements of such a network, and instead provide aggregate
service responses.

One approach is for applications to use RSVP to request that their
flows be admitted into the network.  If a request is accepted, it
would imply that there is a committed resource reservation within the
IntServ-capable components of the network, and that the service
requirements have been mapped into a compatible aggregate service
class within the DiffServ-capable network [7].  The DiffServ core
must be capable of carrying the RSVP messages across the DiffServ
network, so that further resource reservation is possible within the
IntServ network upon egress from the DiffServ environment.  The
approach calls for the DiffServ network to use per-flow multi-field
(MF) classifier, where the MF classification is based on the RSVP-
signaled flow specification.  The service specification of the RSVP-
signaled resource reservation is mapped into a compatible aggregate
DiffServ behavior aggregate and the MF classifier marks packets
according to the selected behavior.  Alternatively the boundary of
the IntServ and DiffServ networks can use the IntServ egress to mark
the flow packets with the appropriate DSCP, allowing the DiffServ
ingress element to use the BA classifier, and dispense with the per-
flow MF classifier.

A high precision end-to-end QoS model requires that any admission
failure within the DiffServ network be communicated to the end
application, presumably via RSVP.  This allows the application to
take some form of corrective action, either by modifying it's service
requirements or terminating the application.  If the service
agreement between the DiffServ network is statically provisioned,
then this static information can be loaded into the IntServ boundary
systems, and IntServ can manage the allocation of available DiffServ
behavior aggregate resources.  If the service agreement is

dynamically variable, some form of signaling is required between the
two networks to pass this resource availability information back into
the RSVP signaling environment.

6. Conclusions

   None of these observations are intended to be any reason to condemn
   the QoS architectures as completely impractical, nor are they
   intended to provide any reason to believe that the efforts of
   deploying QoS architectures will not come to fruition.

   What this document is intended to illustrate is that there are still
   a number of activities that are essential precursors to widespread
   deployment and use of such QoS networks, and that there is a need to
   fill in the missing sections with something substantial in terms of
   adoption of additional refinements to the existing QoS model.

   The architectural direction that appears to offer the most promising
   outcome for QoS is not one of universal adoption of a single
   architecture, but instead use a tailored approach where aggregated
   service elements are used in the core of a network where scalability
   is a major design objective and use per-flow service elements at the
   edge of the network where accuracy of the service response is a
   sustainable outcome.

   Architecturally, this points to no single QoS architecture, but
   rather to a set of QoS mechanisms and a number of ways these
   mechanisms can be configured to interoperate in a stable and
   consistent fashion.

7. Security Considerations

   The Internet is not an architecture that includes a strict
   implementation of fairness of access to the common transmission and
   switching resource.  The introduction of any form of fairness, and,
   in the case of QoS, weighted fairness, implies a requirement for
   transparency in the implementation of the fairness contract between
   the network provider and the network's users.  This requires some
   form of resource accounting and auditing, which, in turn, requires
   the use of authentication and access control.  The balancing factor
   is that a shared resource should not overtly expose the level of
   resource usage of any one user to any other, so that some level of
   secrecy is required in this environment

   The QoS environment also exposes the potential of theft of resources
   through the unauthorized admission of traffic with an associated
   service profile.  QoS signaling protocols which are intended to

undertake resource management and admission control require the use
of identity authentication and integrity protection in order to
mitigate this potential for theft of resources.

Both forms of QoS architecture require the internal elements of the
network to be able to undertake classification of traffic based on
some form of identification that is carried in the packet header in
the clear.  Classifications systems that use multi-field specifiers,
or per-flow specifiers rely on the carriage of end-to-end packet
header fields being carried in the clear.  This has conflicting
requirements for security architectures that attempt to mask such
end-to-end identifiers within an encrypted payload.

QoS architectures can be considered as a means of exerting control
over network resource allocation.  In the event of a rapid change in
resource availability (e.g. disaster) it is an undesirable outcome if
the remaining resources are completely allocated to a single class of
service to the exclusion of all other classes.  Such an outcome
constitutes a denial of service, where the traffic control system
(routing) selects paths that are incapable of carrying any traffic of
a particular service class.

8. References

   [1]   Bradner, S., "The Internet Standards Process- Revision 3", BCP
         9, RFC 2026, October 1996.

   [2]   Mankin, A., Baker, F., Braden, R., O'Dell, M., Romanow, A.,
         Weinrib, A. and L. Zhang, "Resource ReSerVation Protocol (RSVP)
         Version 1 Applicability Statement", RFC 2208, September 1997.

   [3]   Braden. R., Clark, D. and S. Shenker, "Integrated Services in
         the Internet Architecture: an Overview", RFC 1633, June 1994.

   [4]   Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W.
         Weiss, "An Architecture for Differentiated Services", RFC 2475,
         December 1998.

   [5]   Baker, F., Iturralde, C., Le Faucher, F., Davie, B.,
         "Aggregation of RSVP for IPv4 and IPv6 Reservations", Work in
         Progress.

   [6]   Bernet, Y., "Format of the RSVP DCLASS Object", RFC 2996,
         November 2000.

   [7]  Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Speer,
        M., Braden, R., Davie, B., Wroclawski, J. and E. Felstaine, "A
        Framework for Integrated Services Operation Over DiffServ
        Networks", RFC 2998, November 2000.

   [8]  "Quality of Service Technical Overview", Microsoft Technical
        Library, Microsoft Corporation, September 1999.

   [9]  "Resource Reservation Protocol API (RAPI)", Open Group Technical
        Standard, C809 ISBN 1-85912-226-4, The Open Group, December
        1998.

   [10] Berger, L. and T. O'Malley, "RSVP Extensions for IPSEC Data
        Flows", RFC 2007, September 1997.

   [11] Mills, C., Hirsh, D. and G. Ruth, "Internet Accounting:
        Background", RFC 1272, November 1991.

9.  Acknowledgments

   Valuable contributions to this document came from Yoram Bernet, Brian
   Carpenter, Jon Crowcroft, Tony Hain and Henning Schulzrinne.

10. Author's Address

   Geoff Huston
   Telstra
   5/490 Northbourne Ave
   Dickson ACT 2602
   AUSTRALIA

   EMail: gih@telstra.net

11.  Full Copyright Statement

Acknowledgement