# QBone Bandwidth Broker Architecture

STATE OF THE ART

Work in Progress


## Abstract

This document is a rewrite of the QBBAC Bandwidth Broker Requirements document (version 0.7) and an attempt to harmonize different ideas and proposals (e.g. see [1] and [10]) that have been made over the past few months within the QBBAC. The goal is to recommend a simple, but adequately capable bandwidth broker architecture for the QBone.


# Introduction

The purpose of this document is to establish a minimal set of requirements for network clouds wishing to participate in inter-domain QoS signaling trials across the QBone. In the QBone test bed, each participating network is a differentiated service (DiffServ) domain supporting one or more globally well known forwarding services built from fundamental DiffServ building blocks.

The primary strength of the differentiated service architecture [7,11] is the ability to achieve end-to-end QoS assurances while: 1) allowing for aggregation into a small number of DS behavior aggregates in the core; 2) requiring only bilateral service level agreements (SLAs) between all participating domains; and 3) allowing for maximal flexibility in local resource management decisions.

Any inter-domain DiffServ reservation signaling protocol must not break this model. Only the signaling interfaces between peering QBone domains should be specified and not the details of service level agreements or the underlying means by which individual QBone domains manage their network resources. Indeed, it is anticipated that within the QBone there will be significant variation in the implementations and resource management strategies behind the uniform signaling interface. Finally, because it is important to bootstrap non-trivial QoS deployments, any such protocol must mesh well with the end-to-end signaling capabilities of hosts and must be simple enough to facilitate rapid deployment, while remaining flexible enough to support future performance optimizations and protocol extensions.

### Goals

The goals of this document are as follows:

- Define a model of the "bandwidth broker" resource managers to be deployed in the QBone.
- Recommend a deployment phasing for the QBone bandwidth broker work.
- Specify a common interdomain interface for the QBone bandwidth broker.

The technology being discussed here is too new for a complete and definitive analysis of the requirements for the bandwidth broker to take place. Therefore, the best approach is to discuss some of the basic

requirements and basic models and to suggest some candidates for the inter-domain protocol that are likely to prove robust and extendible. This is a stage for experimentation and trying out ideas.

The over-riding principles are:

- Leverage DiffServ model for aggregation and scalability in the forwarding path and keep peering arrangements simple and bilateral
- To keep the design simple to allow for the construction of prototypes and to make inter-operation simpler
- To leave room for future growth. This means building the inter-domain protocol with the flexibility that some developers may include some functions (optionally) which other developers may not want. There should be facilities to pass through or ignore information that a particular bandwidth broker implementation does not support.
- To leave space for different ideas and designs wherever possible. This implies that it is, at this point, better to ***under*** specify rather than over-specify functions and interfaces. That is, specify the bare minimum needed to inter-operate for the minimum functions, and leave the rest to the discretion of the group(s) building the prototype.

## Scope

As just discussed, the scope of this document is limited to the inter-domain protocol. It provides neither a full bandwidth broker design nor a complete requirements analysis. In particular, the details of Service Level Specification (SLS) and SLA negotiation are left to a later time. This is discussed further later on in this document.

It is also generally recognized that policy control, policy-based admission control, accounting, authorization and authentication functions, network management functions and both inter- and intra-domain routing either affect or are (or can be) affected by the bandwidth broker. These are all important issues and should be explored, but are beyond the scope of this document. However, QBone participants should be able to experiment with these issues, and so if there is interest, experimental extensions may be specified in the minimal inter-domain BB protocol to allow for this. The addition of specific experimental TLVs should be discussed within the BBAC.

Further, some of these important issues can be worked out through a combination of additional companion documents generated by the BBAC of QBone and IETF internet drafts in the appropriate workgroups.

Although this document assumes a pure DiffServ environment, where every set of network elements inside a trust domain is considered to be a DS domain, it may be desirable in the future to extend this work to support end-to-end signaling along paths that include non-DiffServ capable domains or elements.

There will be a phased introduction of bandwidth broker functions in QBone:

Phase 0
: Initial prototypes without inter-bandwidth broker communication. Note that it is possible in Phase 0 to have only a human "bandwidth broker".

Phase 1
: First prototype of the inter-domain bandwidth broker protocol

Phase 2
: Later prototypes with possibly "improved" inter-domain bandwidth broker protocol and additional functions

# Basic Concepts

In this section, there is an outline of some basic concepts that are the starting point for the bandwidth broker within the QBone project(s).

## Services

### Summary of QBone Premium Service

QBone Premium Service (QPS) [5] is itself an instance of the Premium Service described in [4]. The fundamental idea is to provide a service with quantitative, absolute bandwidth assurance. The service may be provided entirely within a domain, from domain-edge to domain-edge (within the same domain) or across a number of domains.

An instantiation of QPS requires a number of parameters to be specified (and agreed) between the service provider(s) and the customer(s). These parameters are [5]:

- Start and end times
- Source and destination
- MTU Size
- Peak Rate

The following guarantees are given by the service:

- No loss due to congestion
- No latency guarantees
- Worst-case jitter bounds (ipdv) except in the case of IP route changes.

QPS is unidirectional and "out of profile" traffic is dropped.

This discussion is only about the technical aspects of QPS. A discussion of any financial and legal aspects of the service **is intentionally omitted**. It is important to note that there is *no* specification of how all this is accomplished.

### More abstract concept of service

Note that while the initial phases of the BB work concentrate on QPS, the inter-domain BB protocol needs to be flexible enough to handle other services. The design of the inter-domain BB protocol should take this into account, and therefore a slightly more abstract view of service is discussed here.

One can abstract from the above description of QPS to the elements that must be specified, either implicitly or explicitly, for any and every service (in this context).

These elements must first of all fix the service in space and time, i.e. it must be specified between what times the service will be delivered (or can be requested) and the points (in space) at which the service will be delivered (or can be requested). It is assumed, of course, that this specification can be left open-ended, or it can be implied that the service can be requested at all times and at all places where the provider has a

presence.

Likewise, from the customer side, there must be some specification of what the input is. Exactly what must be specified is dependent on the service being requested. Note that although different providers may offer the same service in different ways, for a service that is intended to be global, the same specification should be used at the protocol level. One can expect in general that stricter service requires more specification (as in QPS) whereas a service with fewer guarantees requires much less specification (or none, e.g. Best-effort).

Finally, there has to be a specification of what the service provides (or what it consists of). This may be quantitative (as in the case of QPS) or qualitative, absolute or relative. By qualitative is meant statements like "low loss". By relative is meant statements like "Gold service has delay no worse than Silver service". Note that both absolute and relative may be quantitative or qualitative (this is somewhat different from the terminology in [12]).

The concept of service is end-to-end as fixed by the space coordinates, but the endpoints themselves may be networks and need not only be hosts. Further, in general, the endpoints may be left implicit.

## The Diffserv idea

Diffserv is described in [7] in some detail. This is a brief summary for the purposes of understanding the relationship between services and mechanisms, and consequently the relationship between signalling resource reservations and bandwidth broker actions.

The DiffServ architectural model improves the scalability of QoS provisioning by pushing state and complexity to the edges of the network and keeping classification and packet handling functions in the core network as simple as possible. Briefly put, flows are classified, policed, marked and shaped at the edges of a DS domain. The nodes at the core of the network handle packets according to a Per Hop Behavior (PHB) that is selected on the basis of the contents of the DS field in the packet header. The number of DS code points and the number of PHBs is limited and consequently this mechanism allows for a large number of individual (micro-)flows to be aggregated from the point of view of the core router.

A PHB is defined in [7] as a "description of the externally observable forwarding behavior of a DS node applied to a particular DS behavior aggregate". The actual mechanisms causing this behavior are not strictly part of the PHB description. From the description of the behavior supplied by a PHB, it is intended that one can make a service description; at least that part of the service description that says what effect a service has.

The other part of the service description, namely that related to the customer's traffic, is related to the traffic conditioning concepts described in the DiffServ architecture. Traffic conditioning mechanisms include:

- Classification
- Marking
- Metering
- Shaping
- Dropping

and together they make up a traffic conditioning specification. These mechanisms can be set on the basis of the traffic profile usually specified in terms of classification parameters (how to recognize the specific flow

or set of flows), and metering mechanism and parameters (what are the characteristics asserted for the specific flow or set of flows).

QPS is based on the Expedited Forwarding PHB defined by Nichols, Jacobson and Poduri [6] which provides the necessary characteristics (configurable rate allocated to an aggregate independent of any other traffic on the link). With traffic-conditioned input and links in each DS domain configured at or above the specified rate, the service characteristics of QPS can be achieved.

Assuming statically configured SLAs and SLSs between adjacent domains, the service is then realized by the bandwidth broker receiving a resource allocation request and configuring the routers at the edges of (and internal to) its domain with the set of parameters for the PHB mechanisms and the traffic conditioning mechanisms derived from:

- The resource allocation request (RAR)
- The service definition
- The SLA/SLS in place with the peer domains (if applicable)

The further handling of the RAR is the subject of the differences between the Phase 0 and Phase 1 bandwidth brokers.

# Bandwidth broker as Oracle

To meet these requirements, it is recommended that each QBone domain be represented by an "oracle" that responds to admissions requests for network resources. Such oracles have become colloquially known as "bandwidth brokers" [8].

The oracle model is as follows: In general, a bandwidth broker may receive a resource allocation request (RAR) from one of two sources: Either a request from an element in the domain that the bandwidth broker controls (or represents), or a request from a peer (adjacent) bandwidth broker. This document does not specify the form of the intra-domain protocol or messages, only the inter-domain protocol.

In any case, the bandwidth broker responds to this request with a confirmation of service or denial of service. This response is known as a Resource Allocation Answer (RAA). The request may have certain side effects also, such as altering the router configurations at the access, at the inter-domain borders, and/or internally within the domain, and possibly generating additional RAR messages requesting downstream resources. These side effects are local to the domain and are not specified here. The mechanism for triggering the response is defined in the protocol specification.

The basic input to the bandwidth broker oracle is what is described in a previous section as necessary for an abstract service; namely, the space-time coordinates of the service, the kind of service (and possibly parameters of the service) and possibly the characteristics of the input. There may, of course, be other input, but this document is only concerned with the minimum necessary input.

# Service Level Agreements and Service Level Specifications

## Description of SLAs

Service level agreements are concluded between peer domains, presumably (logically) adjacent, where one domain is the service provider and the other domain is the customer. It is possible for the client to be an

individual.

SLAs are assumed to be bilateral, between peer domains, and Bandwidth Brokers are the agents whose (functional) responsibilities include the implementation of the technical aspects of the agreements.

An SLA provides a guarantee that traffic offered by the (peer) customer domain, that meets certain stated conditions, will be carried by the service provider domain to one or more appropriate egress points with one or more particular service levels. The guarantees may be hard or soft, may carry certain tariffs, and may also carry certain monetary or legal consequences if they are not met. They may also include certain non-technical guarantees and issues that do not bear directly on packet handling, which is our main concern here.

The technical conditions and service levels may include policing, shaping and DS PHBs, but in fact may be larger than that in the sense of including matters of the various policies applicable, availability guarantees given, access guarantees given, trouble ticket procedures and response times and so forth.

An SLA, then is a partially technical document that is determined by network administrators, lawyers, and others, and is communicated via means ordinarily appropriate to that sort of agreement. In a sense, it contains the larger context for, and possibly limits to, the technical agreements assumed to be included in the SLA. "Inclusion of technical agreements" should not be taken to mean that all the details must be included in the SLA. What is required is that enough information is included to determine an SLS in sufficient detail, including (but not limited to)

1. PHBs to be applied
2. Traffic conditioners, policers, markers, shapers and their parameters
3. Any applicable policies

An SLA is changed by the (human) parties involved in the agreement. Bandwidth brokers do not involve themselves in SLA negotiation and do not communicate SLAs between peers. Thus SLA (re-)negotiation is not one of the tasks of a bandwidth broker.

This view of the SLA is that it is a human agreement and in fact sets the *context* and *parameters* of the behavior of the bandwidth broker with respect to the packet handling service. It may include also bandwidth broker behavior with respect to the application of policies, and other issues which may influence routing, recovery behavior, authorization, authentication and accounting, along with other network management functions.

It is likely that a wide variety of SLAs will flourish to meet a wide variety of technical and contractual requirements. As interesting as the space of potential SLAs (and their components) may be, it is unnecessary for a reservation signaling protocol to refer explicitly to established SLAs.

## Description of SLSs

The SLS contains the technical details of the agreement specified by the SLA. An SLS has, as its scope, the acceptance and treatment of traffic meeting certain conditions and arriving from a peer domain on a certain link. More specifically, the SLS asserts that traffic of a given class, meeting specific policing conditions, entering the domain on a given link, will be treated according to a particular (set of) PHB(s) and if the destination of the traffic is not in the receiving domain, then the traffic will be passed on to another domain (which is on the path toward the destination according to the current routing table state) with which a similar (compatible and comparable) SLS exists specifying an equivalent (set of) PHB(s).

A traffic conditioning specificiation (TCS) specifies classifier rules and any corresponding traffic profiles and metering, marking, discarding and/or shaping rules which are to be applied to traffic aggregates selected by a classifier. The Internet Draft "A Framework for Differentiated Service" [FRAME] gives the following examples of parameters that may be specified by a TCS:

1. Detailed service performance parameters such as expected throughput, drop probability, latency;
2. Constraints on the ingress and egress points at which the service is provided, indicating the 'scope' of the service;
3. Traffic profiles which must be adhered to for the requested service to be provided, such as token bucket parameters;
4. Disposition of traffic submitted in excess of the specified profile;
5. Marking services provided;
6. Shaping services provided;
7. Mapping of globally well-known services DSCP values (not from [FRAME])

It is the responsibility of the service-providing domain (i.e. the receiver of the traffic specified in the SLS) to treat the traffic as specified in the SLS until those packets leave the domain. The SLS represents a commitment to consider certain classes of RARs and to treat the traffic conforming to the parameters of the admitted RARs in a manner consistent with a globally well-known service specification (GWSS). Since sservices are built from PHBs and the concatenation of PHBs, this is equivalent to handling conforming packets with the appropriate PHB within the domain. If the destination of the traffic is not within the domain itself, then there must be (at least one, but perhaps several) SLS(s) with an adjacent downstream DS domain at an egress point for the traffic that provide(s) a total committment, over all the egress SLSs that can be used to carry traffic toward that destination, at least as great as that of the SLS on the ingress(es). This can be made precise with requirements on inequalities between the traffic conditioning specifications of the SLSs.

The intent is that for any given SLS on the ingress side, that there is sufficient capacity on the egress side to service it. Suppose that you have an SLS on the ingress with a single destination domain for e.q. capacity 10. If you only have one egress in your network that can reach that destination domain then you must have an SLS with the next downstream domain through that router on that interface with capacity at least 10. If you have multiple possible egresses, and you know that the SLS will be realized by reservations for multiple (aggregates of) flows, then you can spread that capacity 10 over those several egresses and no single SLS has to have that capacity by itself (though severally, they have to be able to handle that capacity). If you know that there is a single flow associated with that SLS, then it is questionable whether you can distribute it among several SLSs with downstream domains on the way to the destination because then you will almost certainly cause packets to arrive out of order.

So, *the scope of the SLS is through the domain, from ingress point to egress point or destination (if traffic sink is within the domain)*.

Because full parameterization of SLSs is complex and is currently poorly understood, an SLS establishment and renegotiation protocol should be very minimal and highly extensible. This issue is left for Phase 2 or later. Instead, for Phase 0 and Phase 1, the terms of bilateral SLSs are propagated out-of-band (either through another protocol or manually), so that any two peering bandwidth brokers have a shared understanding of the SLS that exists between them.

## Reservations

At this point, we should distinguish a number of concepts. We have already discussed SLAs and SLSs briefly. The SLS is itself not a reservation, but rather a commitment to allow reservations (or a potential for reservations). An analogy can be found in stock options: A stock option is a promise to allow an individual to buy X shares of stock at a given (fixed) price, no matter what the current price of the shares is. When the individual exercises the option, the shares are purchased at the given price and potential profit is realized. In a similar way, an SLS is a promise to allow a certain amount of resource usage and this "option" is exercised by sending an (inter-domain) RAR.

An interdomain reservation depends on sequences of interlocking SLAs and SLSs between DS domains. As pointed out earlier, for an interdomain reservation to succeed, the SLSs and policy requirements of the domains must be compatible and "ripple through" the sequence of agreements between physically adjacent domains. Further, the sequence of agreements must fulfill the service expectations (performance) of the requester.

Actual reservations are accomplished via the protocols described in this document. A reservation represents actually committed resources but not necessarily used resources. As traffic flows, the resource is actually used. How much can be used depends on the type of reservation of course.

Every bandwidth broker must, therefore, track: the SLSs between its DS domain and peering DS domains, the set of established reservations consuming resources in its domain and the availability of all reservable resources in its domain. The SLSs (which we are assuming at this point are not dynamic) are tracked by the bandwidth broker and (shared with) the policy decision and enforcement points. The reservations are tracked by the bandwidth broker and (shared with) the network management system. The actual resource use is tracked by the routers themselves and (possibly) monitored by the bandwidth broker.

## Resource Allocation Requests

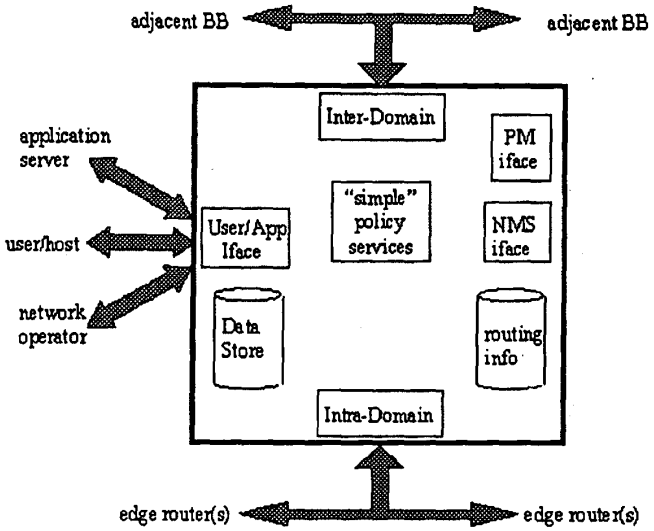
Resource allocation requests (RARs) may succeed or fail depending on the details of an established SLS, details of SLSs along the path, as well as the current state of resource availability along the path. For example (and assume here that all requests are for a specific well-known service), an SLS between an ISP and a customer may specify that all customer RARs for less than 1Mbps will be rejected; or an SLS may specify that the ISP will always make available on one day's notice at least 10Mbps to a specified destination; or an SLS may specify that the customer may request "destination-independent" reservations. There is tremendous flexibility here that is unnecessary to capture or even reveal in the reservation protocol. RARs and their subsequent acknowledgement or rejection are implicitly understood to conform or violate the terms of an existing SLS.

In response to admitted RARs, policers must be reconfigured to admit new DS traffic according to the TCSs in place. An affirmative RAA implicitly acknowledges that this reconfiguration has occurred in whatever manner is consistent with the SLSs and TCSs in place. The space of possible TCSs will inevitably be constrained by the underlying traffic conditioning technologies available on the relevant unidirectional interface. Simple conditioners may only support policing simple behavior aggregates, while more complex conditioners may actually consult route tables to determine classification (e.g. to police according to a profile specific to an ingress-egress pair).

Also unnecessary to the inter-DS-domain signaling protocol are the details behind the admissions control decisions and subsequent traffic conditioner configuration of individual DS domains. These decisions will be based on local resource availability and policy. There will likely be a wide variety of technologies and algorithms for managing the network resources of individual DS domains, but again, this complexity can be obscured behind a uniform admissions control interface.

# Nodal Model

A functional decomposition of the bandwidth broker is shown here.




Not all the components will be used by every implementation. It is important to note that since a bandwidth broker touches on a number of functions in the network, including network management, policy control and configuration management, that these functions may in fact be obtained as services from other nodes implementing them, rather than these functions being implemented in the bandwidth broker itself.

The main functional blocks that concern us here are the user/application protocol, the intra-domain communication protocol and the inter-domain peer protocol, and this last is described in some detail later. In this section, we give a short description of the components.

**Key Protocols**

User/application protocol
: This is an interface provided for resource allocation requests from within the bandwidth broker's domain. These requests may be manual (e.g. via a web interface) or they may consist of messages from one or another setup protocol (for example RSVP messages).

Intra-domain protocol
: The purpose of this protocol is to communicate BB decisions to routers within the bandwidth broker's domain in the form of router configuration parameters for QoS operation and (possibly) communication with the policy enforcement agent within the router. Current bandwidth broker implementations have a number of different protocols for communicating with routers, including COPS, DIAMETER, SNMP, and vendor command line interface commands.

Inter-domain protocol
: The purpose of this protocol is to provide a mechanism for peering BBs to ask for and answer with admission control decisions for aggregates and exchange traffic.

**Data Interfaces**

Routing Tables

A bandwidth broker may require access to inter-domain routing information in order to determine the egress router(s) and downstream DS domains whose resources must be committed before incoming RARs may be accepted. Additionally, a bandwidth broker may require access to intra-domain routing information in order to determine the paths and therefore resource allocation information within the domain.

Data Repository

This respository contains common information for all the bandwidth broker components. The repository includes some or all of the following information and may be shared with other network components such as policy control and network management.

- SLS information for all ingress/egress routers
- Current reservations/resource allocations
- Configurations of routers
- Service mappings/DSCP mappings
- Policy information
- Network management information
- Monitoring information from routers
- Authorization and authentication databases (for users and peers)

**Interfaces to other entities**

The bandwidth broker may have interfaces to other functional entities in the network. Alternately, these functions may be implemented or packaged with the bandwidth broker. It is also to be noted that how configuration management functions are split between policy control and network management is the object of some discussion and debate in the IETF.

# Phase 0 BB Definition

The Phase 0 bandwidth broker definition does not have an inter-domain peer bandwidth broker protocol. It assumes a globally well-known service specification (QPS) which is statically provisioned and agreed upon by all DiffServ domains involved. This sevice is provided by statically negotiated bilateral SLSs which are set up via out-of-band protocols (phone or fax, for example). These are concatenated to provide the service. This is possible because the SLSs stretch from ingress router to egress router(s) of a domain. The concatenation runs then from the egress router of the source domain to the ingress router of the destination domain. The reservations for flows that use this service are also set up out-of-band between domains. It should be noted, finally, that the SLSs and reservations are unidirectional.

Within the source and destination domains, there is assumed to exist a protocol which effectively conveys the resource requests to the bandwidth broker in their respective domains. Note that this protocol can be a telephone call to the human "bandwidth broker" for a particular domain.

The bandwidth broker behaves as an oracle with side effects and returns a confirmation or denial of service to the requester. The protocol needed to do this, and the protocol needed to produce the appropriate side effects (if any) is not specified. The current Phase 0 bandwidth broker implementations use various protocols to accomplish this but since they are not communicated between DiffServ domains, they are not

the subject of this document. See the results of the BB operability event.

Inter-domain reservations work as follows [5]: There is a human "bandwidth broker" designated for each DiffServ domain. These bandwidth brokers communicate with a QBone "bandwidth czar" (also human) who maintains centrally a traffic demand matrix collected from bandwidth brokers in the individual domains. The traffic demand matrix is communicated to the QBone transit domains and the czar will request admission control decisions from the affected domains. When the admission control decisions have been coordinated, the reservations are made and the traffic can flow. If sources do not stay within their traffic parameters, border and/or edge routers will automatically condition the incoming traffic by dropping the excess. The bandwidth broker of a DiffServ domain reports this fact to the czar.

There may in addition be a protocol (for example, RSVP) which flows between hosts. This is assumed *NOT* to affect the transit domains lying between the source and the destination systems (see, for example [2]).

# Phase 1 BB Definition

The Phase 1 BB definition can be seen as a working-out of the scenario in [8] relating to "Statically defined SLSs with bandwidth broker messages exchanged". The Phase 1 BB specification is attempting to solve two problems: First, how should peer bandwidth brokers communicate with each other ? Second, is solving the so-called "last-mile" problem which deals with how to set up reservations end-to-end. While the complete protocol between endpoints and the bandwidth broker is not specified here, the contents of the RAR and RAA messages are specified.

In specifying the Phase 1 bandwidth broker functions, we expressly omit a number of interesting functions and leave them for future development. Among these are dynamic SLS negotiation, most AAA functions and policy functions. The idea is that people can experiment with these in the current framework.

In this phase, RARs flow inter-domain between peer (adjacent) bandwidth brokers, much as described in [8]. The protocol consists of a simple request-response protocol between the bandwidth broker peers, that carries the essential information outlined above for requesting a service in general.

A basic assumption of Phase 1 is that of a pure DiffServ environment, in which heterogeneous networks interoperate at layer 3 and, specifically, achieve QoS interoperability through DiffServ. We make no attempt to solve the intserv/DiffServ integration problem (though there is room to experiment with proposed solutions.) We assume that SLSs are already established (pairwise) between peer bandwidth brokers "out-of-band", that is, without a SLS negotiation protocol. We assume that there are globally well-known services and service IDs referring to those services. The SLSs refer also to these services and in addition, resource allocation requests use the well-known IDs. Further we assume that the BB handles end system requests for its domain, and that BBs may peer directly with non-adjacent BBs. This last is to facilitate the aggregation of service requests and will be explained more fully below.

Lastly,we assume that bandwidth brokers communicate with one another via long-running TCP sessions and that the reliability and flow control provided by TCP are sufficient for this application.

## System Design

We describe here how the protocol works end-to-end and discuss some issues that arise in this design.

Following sections contain the definition of the messages.

We assume first, for purposes of description, that the bandwidth broker for a domain is a single entity and accessible to all end systems in the domain. (This is not meant to preclude distributed implementations). Assume that the end systems have implemented the protocol to communicate with the bandwidth broker.

We distinguish several different cases here:

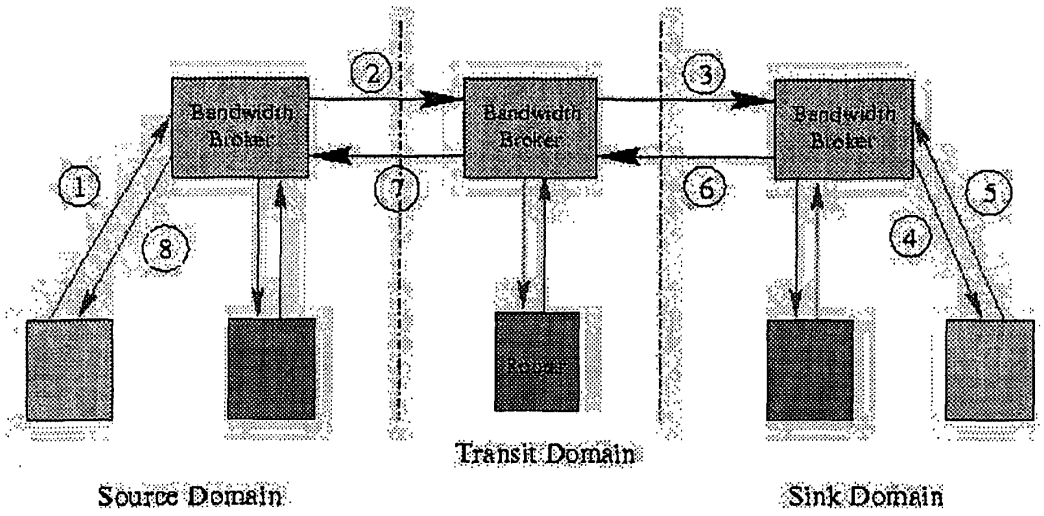
- An end system initiates a request for service with a fully-specified destination address (e.g. /32 for IPv4). The request is thus for service to another end system.
- An end system or a bandwidth broker request service to another domain (that is with a destination prefix that is not fully specified). This is, in effect, a pipe to another domain where the destination is not fully specified.
- A bandwidth broker receives a request for service with fully-specified destination prefix but uses a pipe ("core tunnel") to satisfy the request.

The first scenario shows the basics of inter-domain bandwidth broker communication. We do not expect that the entire mechanism will be used for every request in the network. This would not be especially scalable. The variations in the following scenarios can be used to support aggregation and increase scalability.

The fundamental problem is conveying the knowledge of flows to individual end systems (which might not be in a state to accept the flow) and the need for confirmation that the flow will indeed be accepted.

## Case 1: End system initiates a request for service to an end system

The figure below gives an overview of the communication involved in this scenario. It is important to note that the messages are ***pairwise***. That is, the request proceeds hop-by-hop and is sent only between "adjacent" entities. In the text that follows, numbers in parentheses, e.g. (1) are keyed to the flows in the figure.




End system request with fully specified destination

- Behaviour of the bandwidth broker in the originating domain

  The end system sends an RAR to the bandwidth broker (**1**). This message includes a globally well-known service ID and an IP destination IP address, a source IP address, an authentication field, times for which the service is requested and the other parameters of the service.

  The bandwidth broker makes a number of decisions at this point, including the following:
  a. Whether the requester is authorized for this service
  b. The egress router to which the flow may be assigned
  c. The route through the domain to the egress router
  d. Whether the flow fits in the SLS of the egress router with the net domain in the path to the destination
  e. Whether the flow (possibly according to the policies of the domain) may be accepted for the specified service.

  If these decisions all have a positive outcome, the bandwidth broker will modify the RAR by including the ID for the domain (e.g. for IPv4 a /x prefix where x <= 32) and sign the request with its own signature (**2**).

  In case these decisions have negative outcomes, then the bandwidth broker returns a Resource Allocation Answer (RAA) to the end system (**8**). There can be additional information included, such as a reason code for the rejection and hints about what parameters might be acceptable at the moment that the answer is sent.
- Transit domain handling of the request

  In this case, the bandwidth broker receives an RAR from an adjacent bandwidth broker with a fully-specified destination address specification (**2**). The transit bandwidth broker must perform a number of functions:

a. Authenticate that the request is indeed from a peer bandwidth broker.
b. Determine egress router (interface) from its (inter-domain) routing tables.
c. Check that the requested resources fall within the SLS with the sending domain connecting via one of the ingress routers of this domain.
d. Check that the requested resources fall within the SLS connecting to a successor domain en route to the destination.
e. Ensure that there are sufficient resources within the domain to support the flow from the ingress border router and (possibly) determine the intra-domain route. This determination may involve the domain's resource allocation strategy.
f. Determine whether the flow may be accepted (possibly according to the policies of the domain).

In case that all these decisions have positive outcomes, the transit bandwidth broker modifies the RAR as appropriate (e.g. putting its own ID in the sender's ID field and authentication string in the message) and sends it to the bandwidth broker of the following domain en route to the destination IP address (**3**).

In case that these decisions have negative outcomes, the BB returns an RAA to the sending domain (**7**). Additional information, such as rejection reason code and hints about acceptable parameters may be returned along with the RAA.

- Behaviour of the bandwdth broker in the destination domain.

In this scenario, the bandwidth broker of the destination domain knows the address of the end system which is to receive the flow. As in the behaviour just described, on the reception of the RAR (**3**), it makes the following decisions:

a. Authenticate that the request is indeed from a peer bandwidth broker.
b. Determine the intra-domain route from the ingress router to the end system and decides whether the resources are available to support the flow.
c. Check that the requested resources fall within any possible SLS with the end system.
d. Determine whether the flow may be accepted (possibly according to the policies of the domain).

In case these decisions have negative outcomes, an RAA is sent back (**6**), possibly with a reason code and hints about acceptable parameters.

In case all these decisions have positive outcomes, the bandwidth broker sends the RAR to the end system with appropriate changes (**4**). In this case, the end system makes the determination whether it can receive the flow. This is signalled with an RAA to the bandwidth broker of the destination domain (**5**). The RAA contains authentication of the end system, and parameters for the flow which the end system is willing to accept (which may be different from those received). In case the flow is rejected, the RAA contains a reason code and possibly hints about the set of service parameters that would be acceptable.

Upon receiving the RAA from the end system (**5**), the bandwidth broker authenticates the answer and forwards the RAA, with appropriate changes to the peer bandwidth broker that sent the RAR (**6**). At the same time, the bandwidth broker may configure traffic conditioners at the ingress router and possibly at other routers along the intra-domain path to the destination. Note: these are indicated by green arrows in the figure.

- Transit domain processing of the RAA

The RAA received from the peer bandwidth broker (**6**) is authenticated and the appropriate fields are modified and the RAA is sent to the next bandwidth broker in the chain back to the originating domain (**7**). Internally to the domain, the bandwidth broker may modify traffic conditioners and PHB parameters in the ingress and egress border routers in the path of the flow (indicated by the green arrows in the figure). In addition, resource allocation internal to the domain may be initiated by the bandwidth broker. This would consist of modifying PHB parameters and traffic conditioners in internal routers.

- Originating domain processing of the RAA

  When the bandwidth broker of the originating domain receives the RAA (**7**) and authenticates it, the bandwidth broker completes any resource allocation actions within the domain, modifies PHB and traffic conditioner parameters at the egress router for the flow and forwards the RAA to the requesting end system (**8**). This may include setting the marking functions for the flow in the access router serving the requesting end system (indicated by the green arrows in the figure).

  The end system receives the RAA and is able to send the flow. Note that there is nothing to prevent the end system from sending the flow earlier; however, the flow will not receive the requested service until the RAA is received and the DSCP of packets sent earlier than this will not be marked consistent with the service.

## Case 2: Resource Request for Core Tunnel Services

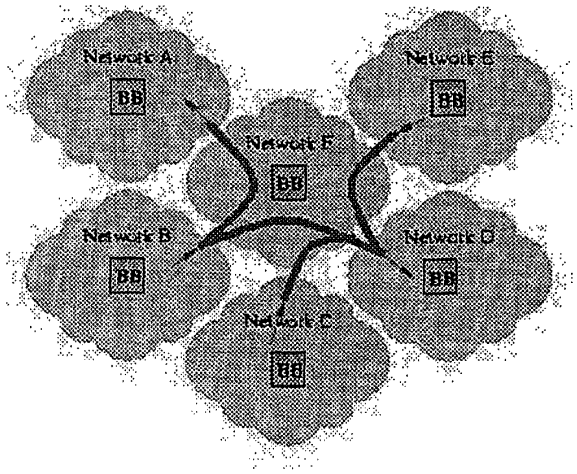
In this section, we handle the setup of a pipe between an origin domain and a destination domain. In this case, the destination prefix is not fully specified (i.e. for IPv4 /X where X < 32). In this document, we call such a pipe a *core tunnel.* The following explains this idea.

### Tunnel Concept

*Tunnel* is a term used in this document for an inter-domain reservation where one or both ends of the reservation is not fully specified (i.e. doesn't have a fully specified IP address), not to be confused with IP tunnels or MPLS tunnels. It is a vehicle for aggregating reservations. A tunnel can extend from DS domain to DS domain (i.e. a *core tunnel* or one or the other end can be fully specified. Here we discuss mostly core tunnels, but all teh variations are possible.

This kind of request may originate in an end system that knows, for example, that it has a large number of requests for service of a certain kind to send to a destination domain and is prepared to aggregate the resource requests to intermediate domains. The request may also originate with a bandwidth broker, as a result of aggregation algorithms (which may be adminstratively triggered or could be triggered based on historical data, for example). It is this latter case that we will discuss here, though the same procedures hold for both cases. Also, the same procedures hold where there are **no** transit domains.

The nature of the trigger is not specified in this document and indeed is a research question. The key trade-off here is reserving (possibly idle) bandwidth vs. the number of signalling messages. The research questions include: How large a pipe to request; how much in advance to request a pipe (and on the basis of what?); when to reduce or remove a pipe (and how much to reduce ?); and how often to adjust the reservation (negotiation).

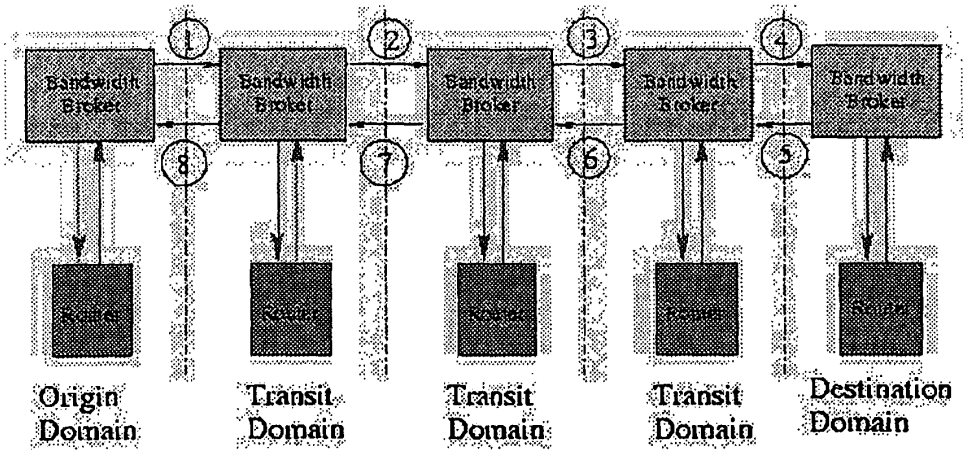



Core tunnels extend from the egress interface of the originating domain to the ingress interface of the destination domain. Note that tunnels as well as reservations are unidirectional. The setting up of a core tunnel involves the intermediate bandwidth brokers, but the use of it for aggregating individual flows does not.

The figure above shows core tunnels extending across several domains. Note the difference between the tunnels and the reservations. The tunnels have origin and destination pairs, while the reservations for several tunnels may be merged at the border router interfaces (shown by the merging of the thick red lines in the figure).

**Establishment of the tunnel**

Assuming that the establishment of a core tunnel is triggered in the origin bandwidth broker, we have the sequence of the above figure. Note that in the text below, numbers in parentheses are keyed to the circled numbers in the figure.



- Behaviour in the originating domain.

  The bandwidth broker in the origin domain creates an RAR which includes the IP prefix of the destination domain along with the normal information required in an RAR (where, extent, when) and an indication that a core tunnel is being requested. This RAR is sent to the banwidth broker in the next domain (**1**) in the path on the way to the destination domain.
- Transit domain processing of the RAR

  In all transit domains, except for the penultimate domain, the bandwidth brokers behave in exactly the same way as for an RAR with a fully specified destination address. Each transit-domain bandwidth broker performs a number of functions on reception of an RAR from a peer bandwidth broker in an adjacent domain ((**1**),(**2**),(**3**)) among which are the following:
  a. Authenticate request from peer bandwidth broker
  b. Determine the egress router from its inter-domain routing table
  c. Check that the RAR falls within the SLS with the sending domain connecting to one of the ingress routers (interfaces) of this domain.
  d. Check that the RAR falls within the SLS with the successor domain (determined via the interdomain routing tables) en route to the destination via one of the egress routers (interfaces) of this domain.
  e. Ensure that there are sufficient resources to support the RAR from the ingress router to the egress router and (possibly) determine the intra-domain route.
  f. Determine whether the flow may be accepted (possibly according to the policies of this domain).

  In case these decisions have positive outcomes, the transit bandwidth broker modifies the RAR by replacing the sender ID and authentication field with its own ID and authentication string. The modified RAR is then sent to the next domain en route to the destination ((**2**),(**3**)).

  In case these decisions have negative outcomes, the bandwidth broker returns an RAA to the sender indicating failure ((**6**),(**7**),(**8**)). Additional information such as a reason code and hints about acceptable parameters may be included.
- Penultimate domain processing of the RAR

  In addition to all the checks outlined in the previous step, the bandwidth broker in the penultimate domain creates, on acceptance of the RAR, a *core tunnel voucher* which contains information about the reservation, ensuring that it fits within the SLS between the penultimate domain and the destination domain. This voucher is added to the RAR and sent to the destination domain (**4**). It is used later by the origin domain bandwidth broker to refer to the reservation (see next section).

  If the reservation is not accepted, the bandwidth broker returns an RAA (**6**) as above.
- Behaviour in the destination domain

  When the bandwidth broker in the destination domain receives the RAR (**4**), it performs the following functions:
  a. Authentication that the request is indeed from a peer bandwidth broker.
  b. Checks that the RAR falls under the SLS with the sending domain connecting via the specified ingress router (interface).
  c. Checks that there are sufficient resources in the domain to support the RAR. (*Note: this is a research issue.*)

d. Determination of whether the RAR can be accepted (possibly according to domain policies).

If the outcomes of these decisions are positive, the destination domain bandwidth broker stores the voucher from the penultimate domain and stores also the identifier of the origin domain. It then returns an RAA (with the voucher) **(5)** to the penultimate domain.

If the outcomes are negative, then it returns an RAA possibly with a reason code and hints about acceptable parameters **(5)**.

- Transit domain processing of the RAA

In all transit domains (including the penultimate domain) the bandwidth broker authenticates the RAA from the sender **((5),(6),(7)** and replaces the sender ID and authentication strings with its own ID and authentication string and then sends the RAA on to the following domain in the direction of the origin domain **((6),(7),(8))**.

At the same time, the bandwidth broker may make adjustments to traffic conditioning (shaping, policing, marking, metering) and PHB functions in its affected border routers and (possibly) in the internal routers of the domain. This is indicated by the green arrows in the figure.

- Origin domain processing of the RAA

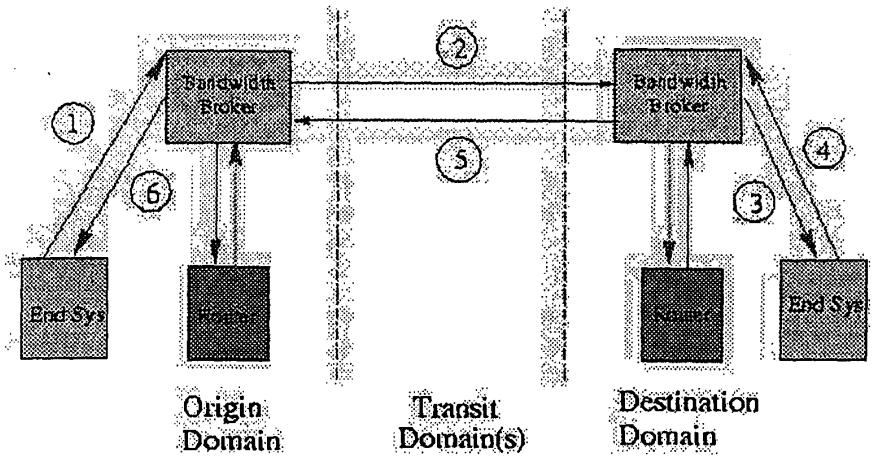
On receiving the RAA for its request **(8)**, the origin bandwidth broker authenticates the RAA and checks the information in it to see whether the request was accepted or not. If the RAR was accepted, the bandwidth broker stores the voucher created in the penultimate domain in the path. At this time, the bandwidth broker may also make adjustments to traffic conditioning and PHB functions in its border router, and it may at this time establish a TCP session with the bandwidth broker in the destination domain (if it has not already done so).

**Other Tunnels**

In addition to core tunnels, other configurations are possible, for example, where the source address is fully-specified (is an end system) but the destination address is not (*head tunnels*), or where the source address is not fully-specified but the destination address is (*tail tunnels*). Both of these cases can be handled with some minor modifications to this protocol (in the origin and destination domain BBs).

## Case 3: Core tunnel handling of a request with fully-specified destination

In this case, the service request has a fully specified destination address, but a seperate reservation in the core network(s) is not made. Instead this service request is aggregated into a *core tunnel* assumed in this case to be previously set up. Note that only the origin and destination bandwidth brokers and the end systems are involved in this communication.




Note that in the text below, numbers in parentheses are keyed to the circled numbers in the figure.

- Originating domain processing the RAR

  The bandwidth broker in the origin domain receives an RAR (**1**) from an end system in its control. According to its own algorithms, it chooses to aggregate this request with others in an existing core tunnel. The bandwidth broker checks the following:
  a. Whether the requester is authorized for this service
  b. The route through the domain to the egress router. It was assumed that in setting up the core tunnel, the bandwidth broker would check to ensure that the resources to support it were available in the domain. However, that check could be delayed to, or repeated at this point.
  c. Whether the flow fits in the core tunnel
  d. Whether the flow (possibly according to the policies of the domain) may be accepted for the specified service.

  If the outcomes of these decisions are positive, the bandwidth broker replaces the sender ID and authentication string in the RAR with its own ID and authentication string, and places the Core Tunnel Voucher TLV for the core tunnel into the message and sends the RAR directly to the bandwidth broker of the destination domain (**2**).

  If the outcomes are negative, then the bandwidth broker returns an RAA to the end system (**6**) indicating failure along with a reason code and possibly hints about acceptable parameter values.
- Destination Domain processing the RAR

  When the destination bandwidth broker receives the RAR (**2**), it checks the following:
  a. Authenticate that the request is indeed from a peer bandwidth broker.
  b. Authenticate the Core Tunnel TLV
  c. Check that the requested resources fit in the core tunnel
  d. Determine the intra-domain route from the ingress router to the end system and decides whether the resources are available to support the flow.
  e. Determine whether the flow may be accepted (possibly according to the policies of the domain).

In case these decisions have negative outcomes, an RAA is sent back (**5**), possibly with a reason code and hints about acceptable parameters.

In case all these decisions have positive outcomes, the bandwidth broker sends the RAR to the end system with appropriate changes (**3**). In this case, the end system makes the determination whether it can receive the flow. This is signalled with an RAA to the bandwidth broker of the destination domain (**4**). The RAA contains authentication of the end system, and parameters for the flow which the end system is willing to accept (which may be different from those received). In case the flow is rejected, the RAA contains a reason code and possibly hints about the set of service parameters that would be acceptable.

Upon receiving the RAA from the end system (**4**), the bandwidth broker authenticates the answer and forwards the RAA, with appropriate changes to the origin bandwidth broker (**5**). At the same time, the destination bandwidth broker may configure traffic conditioners at the ingress router and possibly at other routers along the intra-domain path to the destination Note: these are indicated by green arrows in the figure.

- Origin processing of the RAA

When the bandwidth broker of the originating domain receives the RAA (**5**) and authenticates it, the bandwidth broker completes any resource allocation actions within the domain, modifies PHB and traffic conditioner parameters at the egress router for the flow and forwards the RAA to the requesting end system (**6**). This may include setting the marking functions for the flow in the access router serving the requesting end system (indicated by the green arrows in the figure).

The end system receives the RAA and is able to send the flow.

## Takedown

Either of the endpoints of a QBone reservation may release the reservation, or the BBs in the endpoint domains (if they are not holders of the endpoint of the reservation) may do so. It is assumed that intermediate bandwidth brokers who are aware of a reservation (i.e. one representing a tunnel, not made within a tunnel) also know their peer bandwidth brokers both upstream and downstream with respect to the reservation.

Note that a QBone reservation set up by the SIBBS protocol may have an exact end time specified. In this case the reservation is removed automatically by all parties involved without the need for a takedown message to be sent.

We propose a semi-soft state mechanism for backup of the takedown procedure. This is a refresh of the reservation RAR with a fairly long time constant (on the order of minutes) that is there in case a number of unlikely events cause the takedown messages and retries to be lost. More details of this mechanism will be in the next version of this document.

Takedown is accomplished via the RAR/RAA pair. A node wishing to release the reservation sends an RAR indicating a release of the reservation (or part of it). A complete release should result in a 0 reservation. A negative adjustment that is not a complete release may only be sent by the initiator of the reservation (or its bandwidth broker).

The following conditions and behaviours are defined for reservation takedown:

1. Unless there is some error or internal inconsistency in the RAR, a reduction/takedown always succeeds.
2. A release of a reservation is indicated in one of the following ways:
   a. *decrease* and *delta* are indicated in the flag field of the RAR and the reservation parameter values are equal to the currently held reservation. If the absolute value of the values in the RAR are greater than the reservation currently held by the bandwidth broker, the entire reservation is released and a notification TLV is included in the RAR/RAA.
   b. *decrease* and *absolute* are indicated in the flag field of the RAR. A release is indicated by 0 values in the RAR. A reduction is indicated by non-zero values in the RAR (but "less than" the value of the currently held reservation, where "less than" has a special meaning when applied to a multi-valued object like a reservation). An inconsistent condition occurs if *decrease* and *absolute* are indicated but the values in the RAR are "greater than" the currently held reservation. In this case, the reservation is retained and the RAR should be treated as an error.
3. Since either end can send a takedown, messages may cross. If a takedown arrives at a BB for a reservation that no longer exists, it is by definition successful and receives a positive RAA. It is not, however, forwarded since there may be multiple paths to the origin or destination domains and it would not be known to which peer BB the message should be forwarded.
4. Since in general (except as noted above) a release will always succeed, an RAA can be sent immediately to the sender of the RAR. In this case, however, the BB sending the RAA is responsible for forwarding the RAR to its peer in the next downstream domain.

## Failure Conditions

## Message Formats

The following subsections give the message formats for the Phase 1 BB protocol.

### RAR

The following table outlines the RAR message format. Note that not all of the fields are used in an RAR sent between end systems and bandwidth brokers (i.e. intra-domain).

| Field | Explanation |
|---|---|
| Version | Bandwidth broker protocol version ID (current version is 1) |
| RAR ID | Unique RAR ID (perhaps IP address + sequence number) generated by initial RAR sender and propagated forward; may be used for bookkeeping purposes by any intermediate BB; must be returned in matching RAA message |
| Sender ID | Identifier of the DS domain that sent the RAR; rewritten by intermediate domains; used to authenticate the RAR.For RARs sent to or from end systems, this field is not used. |
| Sender Signature | Each RAR message should be signed with the public key of the sending DS domain; this field in conjunction with the Sender ID allows the RAR receiver to authenticate that the RAR is from a peer DS domain and to reference internal state on the SLS in place with that domain |
| Source Prefix | IP address prefix for source terminus of the service request |
| Destination Prefix | IP address prefix for destination terminus of service request |
| Ingress Router ID | IP address of the interface between two domains for which the sending domain is requesting service. This field is replaced in the message by each sending bandwidth broker. When sent by an end-system, this field contains the IP address of the access router interface through which the flow will pass (for example, the default router) en route to the destination. When sent from a bandwidth broker to an end system, it contains the IP address of the access router interface over which the flow will be forwarded. |
| Start Time | now \| specific future time |
| Stop Time | indefinite \| as long as possible \| specific future time |
| Flags | The following flags are defined:<br>• receiver pays (collect call)<br>• probe (determine parameters/acceptance but do not commit resources)<br>• establish Core Tunnel<br>• renegotiation<br>• delta/absolute values for Service Parameterization Object (SPO)<br>• increment/decrement values for SPO |
| GSID | Globally well-known service ID |
| Service Parameterization Object (SPO) | Service specification parameters dependent on the particular GWS indicated by the GSID. |
| Additional TLVs | Core Tunnel Voucher |

**RAA**

Corresponding to each RAR generated, is an RAA message, each having the following format:

| Field | Explanation |
|---|---|
| Version | Bandwidth broker protocol version ID (current version is 1) |
| RAR ID | Unique RAR ID (perhaps IP address + sequence number) generated by initial RAR sender and propagated forward; may be used for bookkeeping purposes by any intermediate BB; must be returned in matching RAA message |
| Sender ID | Identifier of the DS domain that sent the RAR; rewritten by intermediate domains; used to authenticate the RAR.For RARs sent to or from end systems, this field is not used. |
| Sender Signature | Each RAR message should be signed with the public key of the sending DS domain; this field in conjunction with the Sender ID allows the RAR receiver to authenticate that the RAR is from a peer DS domain and to reference internal state on the SLS in place with that domain |
| Source Prefix | Copied from RAR |
| Destination Prefix | Copied from RAR |
| Ingress Router ID | Copied from RAR as received by this bandwidth broker |
| Start Time | Copied from RAR |
| Stop Time | Copied from RAR. If 'as long as possible' was specified in the RAR, then this may be set to a specific future time. |
| Flags | The following flags are defined:<br>• RAR Accepted<br>If this bit is set (on) then the RAR has been accepted and the learned service parameters may be found in the SPO; if this bit is off, the RAR was rejected and the SPO may optionally be rewritten to reflect the "nearest match" reservation that would have been accepted.Additionally, a reason code TLV may be included following the SPO.<br>• Core tunnel set up<br>This bit indicates that a core tunnel was set up as a result of the associated RAR and that there is a 'voucher' TLV contained in this message. |
| GSID | Copied from the RAR |
| Service Parameterization Object (SPO) | Service specification parameters dependent on the particular GWS indicated by the GSID; parameters that were left blank in the RAR may be completed in the RAA or rewritten to reflect a renegotiation hint as described in the "Flags" field above. |
| Additional TLVs | • Reason Code TLV<br>• Core Tunnel Voucher TLV |

## Additional Objects (TLVs)

### The SPO

The final parameter of both message types, the Service Parameterization Object (SPO), merits further discussion. This parameter is intended to be a service-specific specification of requested or learned service parameters. Depending on the service in question, this may be a simple parameter (e.g. bits-per-second of bandwidth) or may be quite complex (full TSpec, trTCM configuration, etc.).

In the case of the QBone Premium Service (QPS) [5], QPS reservations are defined by the tuple: {source, dest, route, startTime, endTime, peakRate, MTU, jitter}. Analogously, the QPS SPO should have the following format:

| Field | Explanation |
|---|---|
| Route | TLV describing the per-DS-domain route along which service is requested. |
| PeakRate | QPS peakRate in bytes per second |
| MTU | QPS MTU in bytes |
| Jitter | QPS jitter bound in microseconds |

SPO formats must allow for a service to be "ramped up" as well as to be "ramped down" and downright "torn down". Therefore, there must exist at least one field that quantifies the service (e.g. PeakRate), rather than parameterizing the the assurance (e.g. Route, Jitter). The numerical SPO parameters are taken to be a delta if the "delta" flag in the Flags field of the RAR is on. Additionally, these parameters are added if the "increment" flag is on, and subtracted otherwise. So, for example, if the *renegotiation* flag is on, together with the *absolute* flag, then the value in the SPO replaces the entire current reservation.

### Reason Code TLV

The Reason Code TLV is sent anytime an RAR is rejected. It contains information allowing the receiver to diagnose the rejection. The format is as follows:

| Field | Explanation |
|---|---|
| Domain/System ID | TLV indicating a unique identifier (e.g. IP address) of the entity rejecting the RAR. |
| Reason Code | Among the possible reason codes are:<br>• Policy rejection: The RAR was rejected because of policy in the rejecting domain or system.<br>• Parameter rejection: The RAR was rejected because the parameters requested could not be honored. Appropriate parameters may be contained in the SPO returned with the RAA.<br>• Sender not authenticated: The sender of the RAR could not be authenticated.<br>• No SLS: The required SLS for the service did not exist. |
| RAR | TLV containing the offending RAR (or parts thereof) |

### Core Tunnel Voucher TLV

The Core Tunnel Voucher TLV is created by the last bandwidth broker in the chain making up the tunnel and is a permission or certificate that shows that the originator has a reservation for a specific service. The format of the Core Tunnel Voucher is as follows:

| Field | Explanation |
|---|---|
| Generator ID | Domain ID of the bandwidth broker creating the voucher. |
| Destination ID | Domain ID of the bandwidth broker in the destination domain of the tunnel. |
| Voucher | A field signed with the public key of the last bandwidth broker in the tunnel and consisting of the following fields:<br>• Global Well-known Service ID<br>• Ingress router ID (i.e. the ingress to the destination domain)<br>• Domain ID of the originating bandwidth broker<br>• SPO of the reservation requested |

**Unrecognized TLVs**

The TLVs defined in this document (and perhaps some others in revisions of it) are regarded as base. That is, all QBone BB implementations are required to recognize these TLVs. However, for future and experimental TLVs, we need to have a mechanism for nodes not recognizing non-required TLVs to handle them. Our design is the following: We define a base TLV named **Unrecognized TLV received**.

| Field | Explanation |
|---|---|
| Flags | • Found in RAR<br>• Found in RAA |
| Unrecognized TLV | The TL values that were not recognized by this node's message parser. |
| IP address | The IP address of the node reporting the condition |

The behaviour of a node receiving an unrecognized vector is as follows:

- The node creates a subfield consisting of the Flags, the unrecognized TL value and its own IP address.
- If the **Unrecognized TLV Received** is not present in the RAR/RAA, then it creates one and inserts it together with the relevant information into the RAR/RAA.
- If the **Unrecognized TLV Received** is present, the node inserts its information at the *end* of the vector and adjusts the length field.
- The node forwards all received vectors and the **Unrecognized TLV Received** onward in the RAR/RAA.
- The **Unrecognized TLV Received** from an RAR must be placed in the corresponding RAA.

**[Optimization]**An additional optimization we can make is to divide the code space of the TLVs into 2 parts so that one part of the space applies only to TLVs for functions relating to bandwidth brokers in the endpoint domains or the end systems and the other part applies only to functions that must (or should) be supported at intermediate nodes. With a slight change to the rules above, we can reduce the number of unrecognized TLVs reported.

# Contributors

The following people have contributed heavily to this and earlier versions of this document:

- Larry Dunn, Cisco

- Rüdiger Geib, Deutsche Telekom
- Susan Hares, Merit
- Rob Neilson, BCIT
- Francis Reichmeyer, IP Highway
- Dave Spence, Merit
- Andreas Terzis, UCLA
- Jeff Wheeler, Nortel

# Terminology

## Diffserv Terms

Downstream DS domain
: The DS domain downstream of traffic flow on a boundary link.

DS boundary node
: A DS node that connects one DS domain to a node in another DS domain or in a domain that is not DS-capable.

DS domain
: A DS-capable domain; a contiguous set of nodes which operate with a common set of service provisioning policies and PHB definitions.

DS egress node
: A DS boundary node in its role of handling traffic as it leaves a DS domain.

DS ingress node
: A DS boundary node in its role of handling traffic as it enters a DS domain.

Service
: The overall treatment of a defined subset of a customer's traffic within a DS domain.

Service Level Agreement (SLA)
: A service contract between a customer and a provider that specifies the forwarding service a customer should receive. A customer may be a user organization (source domain) or another DS domain (upstream domain).

Service Provisioning Policy
: A policy that defines how traffic conditioners are configured on DS boundary nodes and how traffic streams are mapped to DS behavior aggregates to achieve a range of services.

Upstream DS domain
: The DS domain upstream of traffic flow on a boundary link.

## "New" Terms

In addition to the terms form [RFC2475], we define the following:

Bandwidth Broker (BB)
: A bandwidth broker (BB) manages network resources for IP QoS services supported in the network and used by customers of the network services. A BB may be considered a type of policy manager (see Policy Manager definition below) in that it performs a subset of policy management functionality.

Connection Admission Control (CAC)

Connection admission control refers to the process, performed by the BB, of admitting connection requests to the network based on available resources in the network. The determination of available resources may be done on a static or dynamic basis.

Domain

A domain typically refers to DiffServ domain - see ***DS domain*** above, from [RFC2475].

Edge Router (or Edge Device)

We use the terms edge router, edge device, and boundary node interchangeably. See ***DS boundary node*** above, from [RFC2475].

Inter-Domain Communication

Inter-domain communication refers to the protocol messages and control data that are exchanged between BBs in adjacent domains.

Intra-Domain Communication

Intra-domain communication refers to the protocol messages and control data that are exchanged between a BB and the nodes (usually edge devices) within that BB's domain.

Peer Domains

Two domains are peer domains if they are adjacently connected.

Per Hop Behavior (PHB)

The externally observable forwarding behavior applied at a DS-compliant node to a DS behavior aggregate. Note that while each service is mapped to a PHB (and specific DS Code Point(s)), it is not possible to identify a service by it's PHB (e.g. AF).

Policy Manager (PM) or Policy Server (PS)

A policy manager (PM) or policy server (PS) typically manages the access of users to network policy services. As part of the process of admitting users to access policy services, a PM may employ a BB for CAC, as described above.

Premium Service

Premium Service refers to a quantitative differentiated service which provides a guaranteed low loss and jitter over a DS region. The Premium Service often is also described as "Virtual Leased Line (VLL)" Service. The exact service specification may be found in [QBONEARCH].

Resource Allocation Request (RAR)

A RAR refers to a request for network resources (or service) from an individual user to the BB of that user's domain. If the request includes network resources for outside of the user's local domain, the admission control may be performed based on the SLS(s) in place with adjacent domains. Accepted RARs may result in service provisioning policy (see above) installed in edge devices by a BB.

Service

Service is the overall treatment of a defined subset of a customer's traffic within a DS domain or end-to-end [RFC2475]. In this document, the [RFC2475] service definition will also be applied for traffic treatment between two domains. This leads to unilateral, bilateral and end-to-end service specifications. Whenever "service" is used as stand alone term in the following, bilateral and end-to-end services are meant.

Each "service" is mapped to a PHB identified by its DS Code-Point(s) (DSCPs). By this definition a "SERVICE" IS IDENTIFIED BY ITS "DSCPs" within a DS domain as well as between two adjacent DS domains in the following. The IETF does only standardize PHB's. IETF specifications usually DO NOT LINK DSCPs TO SPECIFIC SERVICES. While each service is mapped to a PHB (and specific DS Code Point(s)), it is not possible to identify a service by it's PHB (e.g. AF).

Unilateral Service

Unilateral service is used to refer to "service" as defined in DiffServ [RFC2475] (above).

Service Level Agreement (SLA)

See SLA in [rfc2475] which defines SLA as " a service contract between a customer and a service

provider that specifies the forwarding service a customer should receive. A customer may be a user organization (source domain) or another DS domain (upstream domain). A SLA may include traffic conditioning rules which constitute a Traffic Conditioning Agreement (TCA) in whole or in part."

Service Level Specification (SLS)

An SLS refers to the particular information relative to the BB and the network devices in order to support a SLA in that network. Information in an SLS is generally on the level of aggregate data flows and the resources/bandwidth provisioned for those flows. An SLS is typically applied at the endpoints of a link connecting adjacent domains and reflects traffic that will be sent from the upstream domain to the downstream domain.

Service Users

End systems users and other entities that can generate RARs. It could as well be an operator that does the RARs (e.g. after being contacted by end-users).

Subnet Bandwidth Manager (SBM)

A Subnet Bandwidth Manger (see [15]) is in charge of the resource allocation requests for a subnet. All users on a variety of hosts on a subnet would defer to the Subnet Bandwidth Manager to negotiate the bandwidth with the bandwidth broker within a domain. The communication path to request resources would be the host signaling the SBM that it needs premium service. The SBM will send an RAR to the BB within the domain. An SBM can also be pre-configured with the ability to requests certain bandwidth resources.

Virtual Leased Line (VLL)

See Premium Service.

# References


1. Multidomain Bandwidth Broker Model, Memo to the QBBAC, September 1999, D. Spence
2. Integrated Services Operation over Diffserv Networks, <draft-ietf-issll-diffserv-rsvp-03.txt> Internet Draft, Bernet, Yavatkar, Ford, Baker, Zhang, Speer, Braden, Davie, June 1999, Work in progress
3. A conceptual model for DiffServ routers, <draft-ietf-diffserv-model-00.txt>, Internet Draft, Bernet, Smith, Blake, June 1999, Work in progress
4. A Two-bit Differentiated Services Architecture for the Internet; K. Nichols, V. Jacobson, L. Zhang, 1998
5. QBone Architecture (v1.0), Ben Teitelbaum et al. Internet 2 QoS Working Group Draft, August 1999, Work-in-progress
6. An expedited forwarding PHB, V. Jacobson, K. Nichols, K. Poduri, RFC 2598, IETF proposed standard, June 1999
7. Architecture for Differentiated Services, S. Blake, D. Black, M Carlson, E. Davies, Z. Wang, W. Weiss, RFC 2475, December 1998
8. A Two-bit Differentiated Services Architecture for the Internet. K. Nichols, V. Jacobson, L. Zhang, July 1999, RFC 2638, Informational.
9. Aggregation of RSVP for IPv4 and IPv6 Reservations, <draft-ietf-issll-rsvp-aggr-00.txt>Fred Baker, Carol Iturralde, Francois Le Faucheur, Bruce Davie, work in progress.
10. SIBBS: Simple Interdomain Bandwidth Broker Signalling, Ben Teitelbaum, Note to the BBAC mailing list, September 1999.
11. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, K. Nichols, S. Black, F. Baker, D. Black, RFC 2474, Standards track, December 1998.
12. Y. Bernet, J. Binder, S. Blake, M. Carlson, B. Carpenter, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, W. Weiss, "A Framework for Differentiated Services", Internet Draft,

draft-ietf-diffserv-framework-02.txt, February 1999.
13. R. Guerin, S. Blake, S. Herzog, "Aggregating RSVP-based QoS Requests", Internet Draft, draft-guerin-aggreg-rsvp-00.txt, November 1997.
14. J. Wroclawski, "The Use of RSVP with IETF Integrated Services", Request for comments, rfc 2210, (proposed standard), Internet Engineering Task Force, September 1997.
15. Raj Yavatkar, Don Hoffman, Yoram Bernet,Fred Baker, Michael Speer "Subnet Banndwidth Manager: A protocol for RSVP-based Admission Control over 802-style networks" Internet Draft (work in progress) draft-ietf-issll-is802-sbm-09.txt

# Appendix 1: Alternative System Model

## System Model

This model follows along the lines of [2] and is shown in Figure X. (It is not exactly the model of [2], though.)

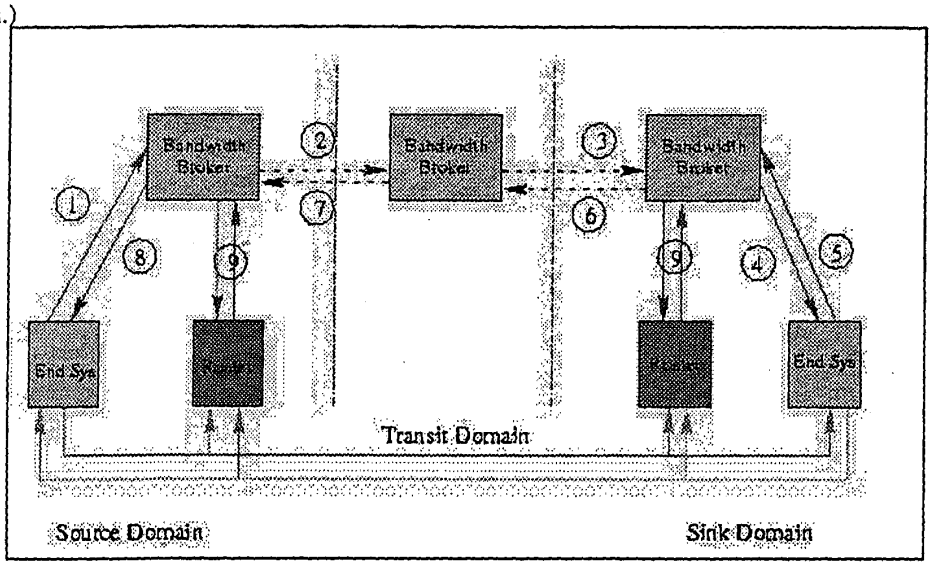



Figure X: System model 2

In this model, bandwidth broker communication takes place in addition to end-to-end communication between the end systems via RSVP. The RSVP protocol between the end systems could be tunneled through the transit domains and the PDUs re-appear at the endpoint domains. There are several designs possible in this case, some of them outlined in [2]. Figure X shows one of them.

The general approach is that the BB is concerned with edge-to-edge resource reservation, but not necessarily with the reservations in the source and sink domains. The RSVP messages sent by the end systems cause resource reservations (intserv) to be made both in the end systems themselves and in the path from the border router of the domain to the end system.

Here we describe in overview, the operation of the system, assuming that the source and sink domains are RSVP-aware, and that the transit domain(s) are not aware of the RSVP messages flowing through them.

(As noted in [2] there are several different ways to handle this, but we will stay with the simplest case.) This implies that the PATH and RESV messages originating in the source and sink domains are tunneled or otherwise masked from the transit domains (which may also have RSVP-aware routers for other purposes).

**System operation**

1. The end system, A in the source domain sends a reuqest (RAR) to its bandwidth broker A whose job it is to do resource allocation and make admission control decisions. Bandwidth broker A then checks whether the request fits into the SLSs that it currently has with adjacent domain(s) in the direction of domain C. Note that this implies a sufficiently long prefix to enable BB A to determine this.
2. Assuming that the request can be handled, BB A sends an inter-domain RAR to BB B. Note that BB A may aggregate this with other requests. It is not necessarily the case that each request received by a BB results in an inter-domain request.
3. BB B receives the inter-domain request from BB A, may again perform some level of aggregation and sends the request further on to BB C. Further, if domain B is multi-homed, then enough has to be known of the routing of the requests through domain B to determine the egress interface and SLS with domain C.
4. BB C makes the determination, again based on existing SLSs, whether to admit the reservation and responds to BB B. NOTE: This may involve further communication with the end system -- flows (4) and (5) in the diagram -- but this is not strictly necessary.
5. BB B notes the success or failure of the reservation and forwards the information back to BB A.
6. BB A notes the success of failure of the reservation and forwards the results back to the hosts.
7. The BBs in the source and sink domains may adjust the parameters in the (border) routers in their domains as a result of the reservation.

The end systems can then send the RSVP messages end-to-end which nails up the reservation.

---

*Ben Teitelbaum*
*ben@internet2.edu*
*Phil Chimento*
*chimento@ctit.utwente.nl*
*The work of Phil Chimento was supported by SURFnet contract Number 3365*
Last modified: Mon Feb 28 14:14:27 MET 2000