Network Working Group
Request for Comments: 2216
Category: Informational

S. Shenker
J. Wroclawski
Xerox PARC/MIT LCS

September 1997

# Network Element Service Specification Template

## Status of this Memo



## Abstract

This document defines a framework for specifying services provided by network elements, and available to applications, in an internetwork which offers multiple qualities of service. The document first provides some necessary context -- including relevant definitions and suggested data formats -- and then specifies a "template" which service specification documents should follow. The specification template includes per-element requirements such as the service's packet handling behavior, parameters required and made available by the service, traffic specification and policing requirements, and traffic ordering relationships. It also includes evaluation criteria for elements providing the service, and examples of how the service might be implemented (by network elements) and used (by applications).

## Introduction

This document defines the framework used to specify the functionality of internetwork system components which support the the ability to provide multiple, dynamically selectable qualities of service to applications using an internetwork. The behavior of individual routers and subnetworks is captured as a set of "services", some or all of which may be offered by each element. The concatenation of these services along the end-to-end data paths used by an application provides overall quality of service control.

The definition of a service states what is required of a router (or, more generally, any network element; a router, switch, subnet, etc.) which supports a particular service. The service definition also

specifies parameters used to invoke the service, the relationship between those parameters and the service delivered, and the end-to-end behavior obtained by concatenating several instances of the service.

Each service definition also specifies the interface between that service and the environment. This includes the parameters needed to invoke the service, informational parameters which the service must make available for use by setup, routing, and management mechanisms, and information which should be carried between end-nodes and network elements by those mechanisms in order to achieve the desired end-to-end behavior. However, a service definition does not describe the specific protocols or mechanisms used to establish state in the network elements for flows that use the described service.

Services defined following the guidelines of this document are intended for use both within the global Internet and private IP networks. In certain cases a concatenation of network element services may be used to provide a range of end-to-end behaviors, some more suited to a decentralized internet and some more appropriate for a tightly managed private network. This document points out places where such distinction may be appropriate.

This document is comprised of three parts. The first defines some terms used both in this document and in the various service specification documents. The second discusses data formats and representations. The third portion of the document describes the various components of the service specification template.

## Definitions

The following terms are used throughout this document. Service specification documents should employ the same terms to express these concepts.

- Quality of Service (QoS)

In the context of this document, quality of service refers to the nature of the packet delivery service provided, as described by parameters such as achieved bandwidth, packet delay, and packet loss rates. Traditionally, the Internet has offered a single quality of service, best-effort delivery, with available bandwidth and delay characteristics dependent on instantaneous load. Control over the quality of service seen by applications is exercised by adequate provisioning of the network infrastructure. In contrast, a network with dynamically controllable quality of service allows individual application sessions to request network packet delivery characteristics according to their perceived needs, and may provide

different qualities of service to different applications. It should be understood that there is a range of useful possibilities between the two endpoints of providing no dynamic QoS control at all and providing extremely precise and accurate control of QoS parameters.

o Network Element

A "Network Element" (or the equivalent shorter form "Element"), is any component of an internetwork which directly handles data packets and thus is potentially capable of exercising QoS control over data flowing through it. Network elements include routers, subnetworks, and end-node operating systems. A QoS-capable network element is one which offers one or more of the services defined according to the rules given in this document. Note that this definition, by itself, preclude QoS-capable network elements that meet performance goals purely through adequate provisioning rather than active admission and traffic control mechanisms. A "QoS-aware" network element is one which supports the interfaces (described below) required by the service definitions. Thus, a QoS-aware network element need not actually offer any of the services defined according to the format of this document; it merely needs to know how to deny service requests.

o Flow

For the purposes of this document a flow is a set of packets traversing a network element all of which are covered by the same request for control of quality of service. At a given network element a flow may consist of the packets from a single application session, or it may be an aggregation comprising the combined data traffic from a number of application sessions.

> NOTE: this definition of a flow is different from that used in IPv6, where a flow is defined as those packets with the same source address and FlowID.

Mechanisms used to associate a request for quality of service control with the packets covered by that request are beyond the scope of this document.

o Service

The phrase "service" or "QoS Control Service" describes a named, coordinated set of QoS control capabilities provided by a single network element. The definition of a service includes a specification of the functions to be performed by the network

element, the information required by the element to perform these functions, and the information made available by the element to other elements of the system. A service is conceptually implemented within the "service module" contained within the network element.

NOTE: The above defines a precise meaning for the word "service". Service is a word which has a variety of meanings throughout the networking community; the definition of "service" given here refers specifically to the actions and responses of a single network element such as a router or subnet. This contrasts with the more end-to-end oriented definition of the same word seen in some other networking contexts.

o Behavior

A "behavior" is the QoS-related end-to-end performance seen by an application session. This behavior is the end result of composing the services offered by each network element along the path of the application's data flow.

When each network element along a data flow path offers the same service, it is frequently possible to explain the resulting end-to-end behavior in a straightforward fashion. The behavior of a data flow path comprised of elements using different services is more complicated, and may in fact be undefined. A future version of this document may impose additional requirements on the service specification relating to multi-service concatenation.

o Characterization

A characterization is a computed approximation of the actual end-to-end behavior which would be seen by a flow requesting specific QoS services from the network. By providing additional information to the end-nodes before a flow is established, characterizations assist the end-nodes in choosing the services to be requested from the network.

o Characterization Parameters

Characterizations are computed from a set of characterization parameters provided by each network element on the flow's path, and a composition function which computes the end-to-end characterization from those parameters. The composition function may in practice be executed in a distributed fashion by the setup or routing protocol, or the characterization parameters may be gathered to a single point and the characterization computed at that point.

Several characterizations may be computed for a single candidate data flow. Conversely, a service may provide no characterizations, and under some conditions no characterizations may be available to the end-nodes requesting QoS services.

o Composition Function

A composition function accepts characterization parameters as input and computes a characterization, as described above.

o Traffic Specification (TSpec)

A Traffic Specification, or TSpec, is a description of the traffic pattern for which service is being requested. In general, the TSpec forms one side of a "contract" between the data flow and the service. Once a service request is accepted, the service module has agreed to provide a specific QoS as long as the flow's data traffic continues to be accurately described by the TSpec.

As examples, this specification might take the form of a token bucket filter (defined below) or an upper bound on the peak rate. Note that the traffic specification specifies the flow's *allowed* traffic pattern, not the flows *actual* traffic pattern. The behavior of a service when a flow's actual traffic does not conform to the traffic specification must be defined by the service (see "Policing" below).

o Service Request Specification (RSpec)

A Service Request Specification, or RSpec, is a specification of the quality of service a flow wishes to request from a network element. The contents of a service request specification is highly specific to a particular service. As examples, these specifications might contain information about bandwidth allocated to the flow, maximum delays, or packet loss rates.

o Setup Protocol

A setup protocol is used to carry QoS-related information from the end-nodes requesting QoS control to network elements which must exercise that control, and to install and maintain to required QoS control state in those network elements. A setup protocol may also be used to collect QoS-related information from interior network elements along an application's data flow path for ultimate delivery to end nodes. Examples of protocols which perform setup functions are RSVP [RFC 2205], ST-II [RFC 1819], and Q.2931.

Note that other mechanisms, such as network management protocols, may also perform this function. The phrase "setup protocol" conventionally refers to a protocol with this function as its primary purpose.

o Token Bucket

A Token Bucket is a particular form of traffic specification consisting of a "token rate" r and a "bucket size" b. Essentially, the r parameter specifies the continually sustainable data rate, while the b parameter specifies the extent to which the data rate can exceed the sustainable level for short periods of time. More specifically, the traffic must obey the rule that over all time periods, the amount of data sent cannot exceed rT+b, where T is the length of the time period.

Token buckets are further discussed in [PARTRIDGE].

o Token Bucket Filter

A Token Bucket Filter is a filtering or policing function which differentiates those packets in a traffic flow which conform to a particular token bucket specification from those packets which do not. The specific treatment accorded nonconforming packets is not specified in this definition; common actions are relegating the packet to best effort service, discarding the packet, or marking the packet in some fashion.

o Admission Control

Admission control is the process of deciding whether a newly arriving request for service from a network element can be granted. This action must be performed by any service which wishes to offer absolute quantitative bounds on overall performance. It is not necessary for services which provide only relative statements about performance, such as the Internet’s current best-effort service. The precise criteria for making the admission control decision are a specific to each particular service.

o Policing

Policing is the set of actions triggered when a flow’s actual data traffic characteristics exceed the expected values given in the flow’s traffic specification. Services which require policing functions to operate correctly must specify both the action to be

taken when such discrepancies occur and the locations in the network where discrepancies are to be detected. Examples of such actions might include relegating the packet to best effort service, dropping packets, reshaping the traffic, or marking non-conforming traffic in some fashion.

o Interfaces

The service module conceptually interacts with other portions of the network element through a number of interfaces. The service specification document should clearly define the specific data, including formats, which moves across each conceptual interface, and ensure that the mapping between conceptual interfaces and the specific mechanisms of the service being defined are clear.

Data Format and Representation

A service module will import and export a variety of data according to the specific requirements of the services the network element supports. Each service definition MUST specify the format of each such data item in an abstract manner. The information specified must be sufficient for the designer of a setup protocol to correctly select an appropriate concrete (packet) format for variables containing this data. At minimum, the following information must be given:

- Type: whether the quantity is an enumeration, a numerical value, etc.

- Range: for numerical quantities, the minimum and maximum values the quantity must be able to represent. For enumerated quantities, an estimate of the maximum number of items which may need be enumerated in the future, even if many of the values are currently unused.

- Precision: the precision with which a numerical quantity must be represented, and whether that precision is absolute (calling for an integer quantity) or a percentage of the value (allowing for a floating point quantity).

The service definition SHOULD additionally specify a preferred concrete format for each data field, in the usual packet-layout format used in current Internet Standard documents or in some other accepted specification format. If the service definition contains these concrete definitions, they should be sufficiently complete and detailed to allow the service definition to be incorporated by reference into the specifications for setup protocols and other users of the specified data.

NOTE: The wording above is intended to encourage the use of common data formats by all protocols carrying data related to a specific service, while not mandating this common format or infringing on the freedom of protocol specification designers to define data representations using alternative mechanisms such as ASN.1 or XDR.

## Service and Data Element Naming

End-nodes, network elements, setup protocols, and management entities within an integrated services internetwork need to exchange information about services, service invocation parameters, characterization parameters, and the intermediate variables and end results of composition functions. To support this requirement, a single uniform namespace is established for services and their parameters.

The namespace is a two-level hierarchy:

<service_name>.<parameter_name>.

Each of these elements is a integer numerical quantity.

<Service Name> is an integer in the range 1 to 254. The number space is broken into three regions.

Service number 1 is used to indicate that the associated parameter is generic", and is not associated with a specific service. This use of generic parameters is described more fully in [RFC 2215].

The range from 2 to 127 used to name services defined by the IETF. Procedures for allocating service numbers in this region will be established by the IETF INT-SERV WG and the IANA. Services designed for public use should obtain a number from this space. The minimum requirement for doing so is a published RFC following the format described in this note.

Service numbers in the region above 127 are reserved for experimental or private services. Service designers may allocate numbers from this space at random for local experimental use. A policy for global but temporary allocation of these numbers may be established in the future if necessary.

The value 0 is left unused to allow the direct mapping of parameter names to MIB object names, as described below.

The value 255 is reserved to facilitate future expansion of the service number space, if required.

<Parameter_name> is a number in the range 1 to 254, allocated on a per-service basis. Within this range, the values 1 to 127 are reserved for assignment to parameters with a common, shared meaning across all services. These parameters are defined in [RFC 2215].

Numbers for parameters specific to a service are assigned from the range 128-254 by the author of the service specification document.

The value 0 is left unused to allow the direct mapping of parameter names to MIB object names, as described below.

The value 255 is reserved to facilitate future expansion of the parameter number space, if required.

In addition to their uses within the integrated services framework, these <service_number>.<parameter_number> pairs should be used as last two levels of the MIB name when the corresponding values are made available to network management protocols.

## Specification Document Format

The following portion of this document describes the layout and contents of a service specification. Each service specification document MUST contain the sections marked [required] below, in the order listed. Each document SHOULD contain each of the remaining sections in the list below, unless there is a compelling argument that the presence of the section is not beneficial. Additional material, including references, should be included at the end of the document.

Some of these sections are normative, in that they describe specific requirements to which conformant implementations must adhere. Other sections are informational in nature, in that they describe necessary context and technical considerations important to the implementor of a service. The sections, and their nature (required or optional, and informational or normative) are listed below:

o Components

The body of a service specification document incorporates the following sections:

- End-to-End Behavior [required] [informational]

- Motivation [required] [informational]

- Network Element Data Handling Requirements [required] [normative]

- Invocation Information [required] [normative]
- Exported Information [required] [normative]
- Policing [required] [normative]
- Ordering and Merging [required] [normative]
- Guidelines for Implementors [optional] [informational]
- Evaluation Criteria [required] [informational]
- Examples of Implementation [optional] [informational]
- Examples of Use [optional] [informational]

o End-to-end Behavior

This is a description of the behavior that results if all network elements along the path offer the same service, invoked with a defined set of parameters.

In private networks it will generally be the case that the required end-to-end behavior is obtained by concatenation of network elements utilizing the same service and making significant use of characterizations.

In the global Internet, this will not always be true. End-to-end behaviors will frequently be obtained through a concatenation of network elements supporting different services, including in some cases elements which exercise no QoS control at all. Mechanisms to characterize end-to-end behavior in this circumstance are not fully established at this time. Future versions of this document may impose additional requirements on service specifications to facilitate inter-service composition.

This section is for informational purposes only.

o Motivation

This section discusses why this service is being defined. It describes what kinds of applications might make use of this service, and why this service might be more appropriate for those applications than other possible choices. This section is for informational purposes only.

o Network Element Data Handling Requirements

This section contains a description of the QoS properties seen by data packets processed by a network element using this service. The description must clearly explain what variables are controlled, the degree of control exercised, and what aspects of the service's handling model are fixed or assumed. Examples of degree of control information include "this property must be mathematically assured" and "this property should be met under most conditions". An example of a stated assumption is "this service is assumed to have extremely low packet loss; delay targets must be met using admission control rather than by discarding packets when overloaded".

Requirements on packet handling SHOULD, when at all possible, be expressed as performance requirements rather than by specifying a a particular packet scheduling algorithm. The performance requirements might, for example, be a specification of the maximal packet delays or the minimal bandwidth share given to a flow.

This section also specifies actions which the packet handling path is required to take to actively provide feedback to end-nodes about conditions at the network element. Such actions might include explicitly generated congestion feedback, indicated either as bits set in the header of data packets or separate control messages sent.

When writing this section of the service specification document, the authors' goal should be to specify the required behavior as precisely as necessary while still leaving adequate room for the implementation and architectural tradeoffs appropriate to different circumstances and classes of network elements. Successfully achieving this balance may require some care.

o Invocation Information

This section describes the set of parameters required by a service module to invoke the service, and a description of how the parameter values are used by the service module. For example, a hypothetical "bounded delay" service might be described as accepting a request indicating a delay target for the network element and the set of packets subject to that delay target, and processing packets in the given set with a delay of the target value or less.

Necessary invocation information for most services can be broken into two parts, the Traffic Specification (TSpec) and the Service Request Specification (RSpec). The TSpec gives characteristics of the data

traffic to be handled, while the Rspec specifies the properties desired from the service. For example, a service offering a mathematical bound on delay might accept a TSpec giving the traffic flow's bandwidth and burstiness specified as a Token Bucket, and an RSpec giving the maximum tolerable queueing delay.

A service accepting an invocation request may be thought of as entering into a "contract" to provide the service described by the RSpec as long as the flow's traffic continues to be described by the TSpec. If the flow's traffic pattern falls outside the bounds of the TSpec, the QoS provided to the flow may change. The precise nature of this change is also described by the service specification (see "Policing" below).

The RSPec and TSpec components of the invocation information should be specified separately and independently, as they will often be generated by different elements of the internetwork

All quantitative information specifications in this section should follow the guidelines given in the Data Formats section of this document, above.

o Exported Information and Characterization Parameters

This section describes information which must be collected and exported by the service module. Exported information is available to other modules of the network element, and by extension to setup protocols, routing protocols, network management tools, and the like.

Information exported by service modules may be used in several ways. For example, quantities such as the amount of link bandwidth dedicated to the service and the set of data flows currently receiving the service are appropriate pieces of information to make available as network management variables.

A service definition may identify a particular subset of the information exported by a service module as characterization parameters. These characterization parameters may be used to compute or estimate the end-to-end behavior of a data flow traversing a concatenation of network service elements. They may also be used to characterize portions of the path for use by network elements (e.g., in computing the buffer necessary, an element may need to know something about the service characteristics of the upstream portion of the path). A service which defines characterization parameters also specifies the characterizations they are used to generate and the composition functions used to generate the characterizations.

NOTE: Characterization parameters are identified as such by virtue of being the inputs to a service's defined composition functions. Because characterization parameters are part of a service's overall exported data set, they are also available to other functions, such as network management. The discussion below relates solely to their use as characterization parameters, and is not intended to limit other uses.

Characterization parameters may be relatively static quantities, such as the bandwidth available on a specific link, or relatively dynamic quantities, such as a running estimation of current packet delay.

Support for a service's defined characterization parameters is mandatory. Any network element offering this service must be able to measure, compute, or, if allowed by the specification, estimate the service's characterization parameters. Service designers are encouraged to understand the implications of specifying characterization parameters for a service, particularly with respect to not unduly restricting the choice of hardware and software architectures used to implement the network element.

Characterization parameters are used by composing the values exported by each network element along a data flow's path according to a composition rule. For each parameter or set of parameters used to develop a characterization, the service specification must specify the composition rule to be used. These composition rules should result in characterizations that are independent of the order in which the element are composed; commutativity and associativity are sufficient but not necessary conditions for this.

Characterization parameters are available through a general interface, and are provided in response to a request from some other module, such as a setup protocol or the routing protocol. The question of exactly how, or if, a specific protocol (e.g., RSVP) uses characterization parameters to generate characterizations is described in the specification of that specific protocol.

The correct use of characterization parameters supplied by service modules is a function of the setup, routing, or management protocol controlling the module. There is no absolute guarantee that characterizations will be available to end-nodes desiring to use a

QoS control service. Service designers targeting services for the global Internet may wish to ensure that a service is useful even in the absence of characterizations, and to exhibit such uses in the "Examples" sections of the service description document.

Conversely, the availability of characterizations may be mandatory in certain circumstances, particularly for private IP networks providing tightly controlled qualities of service for specific applications. Service designers targeting this environment should particularly ensure that the service provides adequate characterization parameters and composition functions to meet the needs of target audiences. It may be appropriate to specify the same basic service with additional characterizations for meeting specific requirements beyond those of the global Internet.

Some useful "general" characterization parameters and corresponding composition rules are not associated with any specific service. These include the speed-of-light latency of communication links and available link bandwidth. These general characterization parameters are defined in [RFC 2215].

Although every conformant implementation of a service is required to provide that service’s characterization parameters, it is still possible that the desired characterization parameters will not be available for composition at all network elements in a path. This situation may arise when different network element services are used at different points in the end-to-end path, as may be required in a heterogeneous internetworking environment. For this reason, characterization parameters and composition function results conceptually include a "validity flag". A network element which is unable to provide the characterization parameter must set this flag, and otherwise leave parameter or composed value unchanged. Once set, the flag is preserved by the composition function, and serves as an indicator of the validity of the data when the final composed result is delivered to its destination.

Protocols which transport characterization parameters and composition data must define and support a concrete representation for this validity flag, as well as for the characterization parameters themselves.

NOTE: This service specification template does not allow a service definition to *require* that a setup or invocation mechanism used with the service perform any function other than transport of invocation parameters to the network elements and signalling of errors generated by the network elements to the end nodes. A notable example of this is that service specification documents may not require or assume that characterizations defined in the specification are actually computed or presented to the end nodes.

That point notwithstanding, the practical usefulness of a specific service may be highly dependent on the presence of some additional behavior in the networked system, such as the computation and

presentation of characterizations to end-nodes or the reliable assurance that every network element in the path from sender to receivers supports the given service. Service specification authors are strongly encouraged to clearly explain the situation of their service in this regard. Statements such as:

> The characterizations defined by this service serve as useful hints to the application. However, the service is specifically intended to be useful even if characterizations are not available.

or

> The usefulness of this service depends strongly on the delivery of both characterizations and the knowledge that all network elements on the path support the service. Requests for this service when characterizations are not available are likely to lead to incorrect or misleading results.

are appropriate. It may also be useful to consider this point in the "Examples of Use" section described below.

NOTE: The possibility of modifying the overall architecture to provide information about the invoking protocol in a service request, and to allow a service to require that the invocation protocol support specific additional functionality, is an area of active study.

o Policing

This portion of the service description describes the nature of policing used to enforce adherence to a flow's Traffic Specification. The specification document must specify the following points

- Expected policing action. This is the action taken when packets not conforming to the TSpec are detected. Example actions include relegating nonconforming packets to best effort, immediately dropping nonconforming packets, delaying these packets until they once again "fit" into the TSpec, or "marking" nonconforming packets in some way.

- Legality of alternative policing actions. The section must specify whether actions not specifically mentioned in specification's description of policing behavior are legal. For example, a service description which specifies that nonconforming packets are to be dropped should state whether an alternate action, such as delaying these packets, is acceptable.

- Location of policing actions in the internetwork. The description of policing must specify where that policing is done. Possibilities include "at the edges of the network only", "at every hop", "heterogeneous branch points" (points where the branches of a multicast tree converge and have different TSpecs reserved downstream), and "source merge points" (points where multiple data streams covered by a single resource reservation converge). The specification should clearly state requirements about topology information (for example "this is an edge node" or "this is a source merge point") which must be available from the setup protocol or another source.

  In this section the specification should also specify the legality of policing at additional points in the network, beyond those listed above. This is important due to technical effects such as are described in the next paragraph.

  Applicable additional technical considerations. If policing of data flows is required or legal at points other than the flow's first entry into the network, the service definition should describe any additional technical considerations which affect the design of such policing. For example, many potential services will allow a data flow to become more bursty as it progresses through the network. If such a service allows policing at points other than the network edge, the traffic specification describing the flow will have to be modified from that given by the application to the network to account for this growing burstiness. Otherwise, it is likely that the flow will be overpoliced, with packets being penalized unnecessarily.

o Ordering and Merging

  Ordering and merging come into play when a network element receives several invocation requests covering the same data flow. As examples, this could occur if several receivers of a multicast data flow requested QoS services for that flow using the RSVP setup protocol, or if a flow was subject to both a statically installed permanent invocation request and a dynamic request from a resource setup protocol.

  In this situation the service module must be able to answer questions about the ordering between different invocation requests, and must be able to generate a single new invocation request which meets the semantics of the setup protocol and the requirements of all the original requesters. Operationally, this is achieved by having the invoking protocol ask the service module, given a set of invocation requests I1...In, to compute a new request which results in the desired behavior.

Five operations must be defined in this section. These are:

- Ordering. The section must define an ordering relationship between the service's TSpecs and RSpecs. This may be a partial ordering, in that some TSpecs or RSpecs may be unordered with respect to each other.

- Summation. This function computes an invocation request which represents the sum of N input invocation requests. Typically this function is used to compute the size of a service request adequate for a shared reservation for N different flows. It is desirable but not required that this function compute the "least possible sum".

- Minimum. This function computes the minimum of two TSpecs. Typically this function is used to compute the TSpec for an actual service invocation given a target TSpec for the service request and a TSpec for the flow's actual traffic pattern. The minimum function must compute the smallest TSpec adequate to describe the minimum of the requested TSpec and the flow's actual traffic.

- RSVP-Merge function. This function computes the invocation request used to request service at an RSVP [RFC 2205] merge point. The function must a) compute an appropriate invocation request for a set of downstream reservations being merged, and b) generate appropriate reservation parameters to be passed upstream by RSVP. This function is described further below and in [RFC 2210].

- Least Common Request function. This function computes an invocation request sufficient to provide service at least equivalent to any one of the original requests passed to the function. This function differs from the RSVP-merge function in that it simply computes an upper bound. It does not need to compute new invocation parameters to be passed upstream by RSVP and cannot utilize the second option discussed in "Notes on RSVP Merging" below.

oo Notes on Ordering

Typically the ordering relation will be described separately for the service's TSpec and RSpec. An invocation request is ordered with respect to another if and only if both its TSpec and its RSpec are similarly ordered with respect to each other.

For TSpecs, the basic ordering relation is well defined. TSpec A is substitutable for TSpec B if and only any flow that is compliant with TSpec B is also compliant with TSpec A. The service specification must explain how to compare two TSpecs to determine whether this is true.

For RSpecs, the ordering relation is dependent on the service. RSpec A is substitutable for RSpec B if the quality of service invoked by RSpec A is at least as good as the quality of service invoked by RSpec B. Since there is no precise mathematical description of "goodness" of quality of service, these ordering relations must be spelled out explicitly in the service description.

oo Notes on RSVP Merging

The purpose of the RSVP merging function is to compute an invocation request which will provide service to the merged flow at least equivalent to that which any of the original requests would obtain for its corresponding unmerged flow. This equivalence may be obtained in two ways

1) The merged request may be computed as an upper bound on the set of original (unmerged) invocation requests. In this case, the service offered by the merged request to any particular traffic flow is identical to that offered by the largest unmerged request, by definition.

2) The merged request may be computed as a value smaller than the upper bound on the set of original requests, but the results passed upstream may restrict the traffic sources to behavior which makes the merged and unmerged requests behave identically.

Note that the merging rules for a particular service may apply either option 1 or option 2 to the different components of a TSpec, as appropriate. The decision is typically made as follows:

When a downstream service module instance can tolerate a flow which exceeds the parameter, the upper bound should be used. For example, if the service supports policing to protect itself against excess traffic, the traffic rate supported by a merged reservation might be an upper bound across the traffic rates supported by each unmerged reservation. The effect of this will be to install the merged reservation at the local node and to inform each traffic source of the largest traffic rate protected by reservation along any *one* distribution path from the source to a receiver.

When a downstream service module instance will not function properly if the parameter is exceeded, the merged function should select the least agressive value of the parameter to install and pass upstream. In this case, the traffic sources will be informed of a parameter value which is appropriate for *all* distribution paths traversed by the traffic flow. For example, services which can handle packets of only limited size can incorporate packet size in the TSpec, and treat the parmeter as described in option 2. The

effect of this will be to limit packet sizes in the flow to those which can be handled by every instance of the service along the flow's path.

This merging calculation must be performed by the service module because it is specific to a particular service.

oo Notes on Calculating Upper Bounds

Both the RSVP-Merge function and the Least Common Request function may make use of calculated upper bounds on TSpec and RSpec parameters.

The calculated upper bound need not be a least upper bound, nor do the various network elements along the path need to all use the same choice of upper bound. Any selection of invocation parameters Iu is compliant as long as it substitutable for each of the parameters I1...In from which it is calculated. Intuitively, one set of parameters is substitutable for another if the resulting quality of service is at least as desirable to all applications. A precise definition of this "substitutable for" function; the ordering relation, must be specified in the service definition. (It may be specified as the empty set, in which case merging of dissimilar requests will not be allowed). If the ordering function specified in this section gives a partial order (if it is possible for two RSpecs or TSpecs to be unordered), then a separate upper bound computation for the parmeter must be given as well.

oo Notes on Service Substitution

This portion of the service description may also note any relationships with other services which are strictly ordered with respect to the service being defined. Two services A and B are strictly ordered if it is always possible to substitute service B for the service A given a set of invocation parameters for service A. This ordering information may be used to allow network elements which provide service B to respond to requests for service A, even if the element does not provide service A directly. If the service specification describes such an inter-service ordering, it MUST also include a description of the invocation parameter mapping function for that ordering.

Substitution of of one service for another in cases where they are not strictly ordered is currently not supported. A future version of this document may augment the service specification format to support this capability.

o Guidelines for Implementors

Many services may be defined in a manner which allows the range of behavior of a compliant network element to be rather broad. This section should provide some guidance as to what range of behaviors the author of the service specification expects the community to desire in their implementations. Because these guidelines depend on such imprecise and undefinable notions at "typical loads", these guidelines cannot be incorporated as part of a strict compliance test. Instead, they are for informational purposes only.

o Evaluation Criteria

Specific functional behaviors required of an implementation for conformance to a service specification is detailed in the previous sections. However, the service specifications are intended to allow a wide range of implementations, and these implementations will differ in performance. This section describes tests that can be used to evaluate a network element's implementation of a given service.

Implementors of service modules face a number of tradeoffs, and it is unlikely that a single implementation would be considered "best" under all circumstances. For instance, given the same service specification, an implementation appropriate for a low-speed link might target extremely high link utilization, while a different implementation might attempt to reduce non-loaded packet forwarding delay to the minimum at the expense of somewhat lower utilization of the link. The intention of the tests specified in this section should be to probe the tradeoffs made by the implementation designer, and to provide metrics useful to guide the customer's choice of an appropriate implementation for her needs.

The tests specified in this section should be designed to operate on a single network element in isolation. This enables their use in a comparative rating system for QoS-aware network elements. In production networks, users will be more concerned with the end-to-end behavior obtained, which will depend not just on the particular network elements selected, but also on other factors such as the setup protocol and the bandwidth of the links. Some user-relevant performance factors are the rate of admission control rejections, the range of services offered, and the packet delay and drop rates in the various service classes. The form of any standardized end-to-end metrics and measurement tools for integrated service internetworks is not specified by this document or by service specification document which follow the format given here.

This section is for informational purposes only.

o Examples of Implementation

This section describes example instantiations of the service. Often these will just be references to the literature, or brief sketches of how the service could be implemented. The purposes of the section are to to provide a more concrete sense of the service being specified and to provide pointers and hints to aid the implementor. However, the descriptions in this section are specifically not intended to exclude other implementation strategies.

This section is for informational purposes only.

o Examples of Use

In order to provide more a more concrete sense of how this service might be used, this section describes some example uses of the service, for informational purposes only. The examples here are not meant to be exhaustive, and do not exclude in any way other uses of the service.

This section is for informational purposes only.

## Security Considerations

Security considerations are not discussed in this memo.

## References

[PARTRIDGE] C. Partridge, Gigabit Networking, Addison Wesley Publishers (1994).

[RFC 2215] Shenker, S., and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2215, September 1997.

[RFC 2205] Braden, R., Ed., et. al., "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.

[RFC 2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

[RFC 2211] Wroclawski, J., "Specification of the Controlled Load Quality of Service", RFC 2211, September 1997.

[RFC 1819] Delgrossi, L., and L. Berger, Editors, "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+", RFC 1819, August 1995.

[RFC 2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997.

Authors' Address:

Scott Shenker
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304-1314

Phone: 415-812-4840
Fax: 415-812-4471
EMail: shenker@parc.xerox.com

John Wroclawski
MIT Laboratory for Computer Science
545 Technology Sq.
Cambridge, MA 02139

Phone: 617-253-7885
Fax: 617-253-2673
EMail: jtw@lcs.mit.edu