Internet Engineering Task Force Integrated Services WG
INTERNET-DRAFT S. Shenker/C. Partridge/B. Davie/L. Breslau
draft-ietf-intserv-predictive-svc-01.txt Xerox/BBN/Bellcore/Xerox
? 1995
Expires: ?/?/96

Specification of Predictive Quality of Service

Status of this Memo

Abstract

This memo describes the network element behavior required to deliver Predictive service in the Internet. Predictive service is a real-time service that provides low packet loss and a fairly reliable delay bound. This service is intended for applications that are tolerant of occasional late arriving packets, but require substantial and quantified levels of delay control from the network. Predictive service is very similar to Controlled Delay service, and the two specifications have a fair amount of shared language. The main salient different between the two services is that Predictive service offers a delay bound and Controlled Delay does not. If no characterizations are provided, then Predictive service is, from an application's perspective, almost indistinguishable from Controlled

Delay; the delay bounds are of little use if the endpoints are not aware of them. Thus, the distinction between Predictive and Controlled Delay is important only in contexts where characterizations are made available to endpoints. This specification follows the service specification template described in [1].

Introduction

This document defines the requirements for network elements that support Predictive service. This memo is one of a series of documents that specify the network element behavior required to support various qualities of service in IP internetworks. Services described in these documents are useful both in the global Internet and private IP networks.

This document is based on the service specification template given in [1]. Please refer to that document for definitions and additional information about the specification of qualities of service within the IP protocol family.

This memo describes the specification for Predictive service in the Internet. Predictive service is a real-time service that provides a fairly reliable delay bound. That is, the large majority of packets are delivered within the delay bound. This is in contrast to Guaranteed service [2], which provides an absolute bound on packet delay, and Controlled Delay service [3], which provides no quantitative assurance about end-to-end delays. Predictive service is intended for use by applications that require an upper bound on end-to-end delay, but that can be tolerant of occasional violations of that bound.

This document is one of a series of documents specifying network element behavior in IP internetworks that provide multiple qualities of service to their clients. Services described in these documents are useful both in the global Internet and private IP networks.

This document follows the service specification template given in [1]. Please refer to that document for definitions and additional information about the specification of qualities of service within the IP protocol family.

End-to-End Behavior

The end-to-end behavior provided by a series of network elements that conform to this document provides three levels of delay control. Each service level is associated with a fairly reliable delay bound, and almost all packets are delivered within this delay bound.

Moreover, all three levels of predictive service will have average delays that are no worse than best effort service, and the maximal delays should be significantly better than best effort service when there is significant load on the network. Packet losses are rare as long as the offered traffic conforms to the specified traffic characterization (see Invocation Information). This characterization of the end-to-end behavior assumes that there are no hard failures in the network elements or packet routing changes within the lifetime of an application using the service.

NOTE: While the per-hop delay bounds are exported by the service module (see Exported Information below), the mechanisms needed to collect per-hop bounds and make these end-to-end bounds known to the applications are not described in this specification. These functions, which can be provided by reservation setup protocols, routing protocols or by other network management functions, are outside the scope of this document.

The delay bounds are not absolute firm. Some packets may arrive after their delay bound, or they may be lost in transit. At the same time, packets may often arrive well before the bound provided by the service. No attempt to control jitter, beyond providing an upper bound on delay, is required by network elements implementing this service. It is expected that most packets will experience delays well below the actual delay bound and that only the tail of the delay distribution will approach (or occasionally exceed) the bound. Consequently, the average delay will also be well below the delay bound.

This service is designed for use by playback applications that desire a bound on end-to-end delay. Such applications may or may not be delay adaptive. The delay bound is useful for those applications that do not wish to adapt their playback point or that require an upper bound on end-to-end delay. Note that the delay bound provided along an end-to-end path should be stable. That is, it should not change as long as the end-to-end path does not change.

This service is subject to admission control.

Motivation

Predictive service is designed for playback applications that desire a reserved rate with low packet loss and a maximum bound on end-to-end packet delay, and that are tolerant of occasional dropped or late packets. The presence of delay bounds serves two functions. First, they provide some characterization of the service so that a non-service-adaptive application (that is, an application that does not want to continually change its service request in response to current conditions) can know beforehand the maximum delays its packets will

experience in a given service class. These bounds will allow such applications to choose an appropriate service class. Second, such delay bounds can help applications that are not interested in adapting to current delays set their playback point. For many noninteractive "playback" applications, fidelity is of more importance than reducing the playback delay; the delay bound allows the application to achieve high fidelity by having a stable playback point with a very few late packets.

Some real-time applications may want a service providing end-to-end delay bounds. However, they may be willing to forgo the absolute bound on delay provided by Guaranteed service [2]. By relaxing the service commitment from a firm to a fairly reliable delay bound, network elements will in many environments be able to accommodate more flows using Predictive service while meeting the service requirement. Thus, Predictive service relaxes the service commitment in favor of higher utilization, when compared to Guaranteed service.

At the same time, these applications may require a higher level of assurance, in the form of a quantitative delay bound, than Controlled Delay service [3] provides. The use of Predictive service, rather than Controlled Delay, may also allow applications to avoid adapting their service requests to changing network performance.

In order to accommodate the requirements of different applications, Predictive service provides multiple levels of service. Applications can choose the level of service providing the most appropriate delay bound.

For additional discussion of Predictive service, see [4,6].

Associated with this service are characterization parameters which describe the delay bound and the current delays experienced in the three services levels. If the characterizations are provided to the endpoints, these will provide some hint about the likely end-to-end delays that might result from requesting a particular level of service, as well as providing information about the end-to-end delay bound. This is intended to aid applications in choosing the appropriate service level. The delay bound information can also be used by applications not wishing to adapt to current delays.

Predictive service is very similar to controlled delay service. The only salient difference is the predictive service provides a fairly reliable delay bound, whereas controlled delay does not have any quantified service assurance. Note that if no characterizations are provided, then this service is, from an application's perspective, almost indistinguishable from controlled delay; the delay bounds are of little use if the endpoints are not aware of them. Thus, the

distinction between predictive and controlled delay is important only in contexts where characterizations are made available to endpoints.

Network Element Data Handling Requirements

The network element must ensure that packet delays are below a specified delay bound. There can be occasional violations of the delay bound, but these violations should be very rare. Similarly, Predictive service must maintain a very low level of packet loss. Although packets may be lost or experience delays in excess of the delay bound, any substantial loss or delay bound violations represents a "failure" of the admission control algorithm. However, vendors may employ admission control algorithms with different levels of conservativeness, resulting in very different levels of delay violations and/or loss (delay bound violations might, for instance, vary from 1 in 10^4 to 1 in 10^8).

This service must use admission control. Overprovisioning alone is not sufficient to deliver predictive service; the network element must be able to turn flows away if accepting them would cause the network element to experience queueing delays in excess of the delay bound.

There are three different logical levels of predictive service. A network element may internally implement fewer actual levels of service, but must map them into three levels at the predictive service invocation interface. Each level of service is associated with a delay bound, with level 1 having the smallest delay and level 3 the largest. If the network element implements different levels of service internally, the delay bounds of the different service levels should differ substantially. The actual choice of delays is left to the network element, and it is expected that different network elements will select different delay bounds for the same level of service.

All three levels of service should be given better service, i.e., more tightly controlled delay, than best effort traffic. The average delays experienced by packets receiving different levels of predictive service and best-effort service may not differ significantly. However, the tails of the delay distributions, i.e., the maximum packet delays seen, for the levels of Predictive service that are implemented and for best-effort service should be significantly different when the network has substantial load.

Predictive service does not require any control of delay jitter

(variation in network element transit delay between different packets in the flow) beyond the limit imposed by the per-service level delay bound. Network element implementors who find it advantageous to do so may use resource scheduling algorithms that exercise some jitter control. See the guidelines for implementors section for more discussion of this issue.

Links are not permitted to fragment packets as part of predictive service. Packets larger than the MTU of the link must be policed as nonconformant which means that they will be policed according to the rules described in the Policing section below.

Invocation Information

The Predictive service is invoked by specifying the traffic (TSpec) and the desired service (RSpec) to the network element. A service request for an existing flow that has a new TSpec and/or RSpec should be treated as a new invocation, in the sense that admission control must be reapplied to the flow. Flows that reduce their TSpec and/or their RSpec (i.e., their new TSpec/RSpec is strictly smaller than the old TSpec/RSpec according to the ordering rules described in the section on Ordering below) should never be denied service.

The TSpec takes the form of a token bucket plus a minimum policed unit (m) and a maximum packet size (M).

The token bucket has a bucket depth, b, and a bucket rate, r. Both b and r must be positive. The rate, r, is measured in bytes of IP datagrams per second, and can range from 1 byte per second to as large as 40 terabytes per second (or about what is believed to be the maximum theoretical bandwidth of a single strand of fiber). Clearly, particularly for large bandwidths, only the first few digits are significant and so the use of floating point representations, accurate to at least 0.1% is encouraged.

The bucket depth, b, is also measured in bytes and can range from 1 byte to 250 gigabytes. Again, floating point representations accurate to at least 0.1% are encouraged.

The range of values is intentionally large to allow for the future bandwidths. The range is not intended to imply that a network element must support the entire range.

The minimum policed unit, m, is an integer measured in bytes. All IP datagrams less than size m will be counted against the token bucket as being of size m. The maximum packet size, M, is the biggest

packet that will conform to the traffic specification; it is also measured in bytes. The flow must be rejected if the requested maximum packet size is larger than the MTU of the link. Both m and M must be positive, and m must be less then or equal to M.

The RSpec is a service level. The service level is specified by one of the integers 1, 2, or 3. Implementations should internally choose representations that leave a range of at least 256 service levels undefined, for possible extension in the future.

The TSpec can be represented by two floating point numbers in single-precision IEEE floating point format followed by two 32-bit integers in network byte order. The first value is the rate (r), the second value is the bucket size (b), the third is the minimum policed unit (m), and the fourth is the maximum packet size (M).

The RSpec may be represented as an unsigned 16-bit integer carried in network byte order.

For all IEEE floating point values, the sign bit must be zero. (All values must be positive). Exponents less than 127 (i.e., 0) are prohibited. Exponents greater than 162 (i.e., positive 35) are discouraged.

Exported Information

Each predictive service module must export the following information. All of the data elements described below are characterization parameters.

For each logical level of service, the network element exports the delay bound as well as three measurements of delay (thus making twelve quantities in total). Each of the measured characterization parameters is based on the maximal packet transit delay experienced over some set of previous time intervals of length T; these delays do not include discarded packets. The three time intervals T are 1 second, 60 seconds, and 3600 seconds. The exported parameters are averages over some set of these previous time intervals.

There is no requirement that these characterization parameters be based on exact measurements. In particular, these delay measurements can be based on estimates of packet delays or aggregate measurements of queue loading. This looseness is intended to avoid placing undue burdens on network element designs in which obtaining precise delay measurements is difficult.

These delay parameters (both the measured values and the bound) have

an additive composition rule. For each parameter the composition function computes the sum, enabling a setup protocol to deliver the cumulative sum along the path to the end nodes.

The characterization parameters are measured in units of one microsecond. An individual element can advertise a delay value between 1 and 2**28 (somewhat over two minutes) and the total delay added across all elements can range as high as 2**32-1. Should the sum of the values of individual network elements along a path exceed 2**32-1, the end-to-end advertised value should be 2**32-1.

Note that while the delay measurements are expressed in microseconds, a network element is free to measure delays more loosely. The minimum requirement is that the element estimate its delay accurately to the nearest 100 microseconds. Elements that can measure more accurately are encouraged to do so.

> NOTE: Measuring delays in milliseconds is not acceptable, as it may lead to composed delay values with unacceptably large errors along paths that are several hops long.

The characterization parameters may be represented as a sequence of twelve 32-bit unsigned integers in network byte order. The first four integers are the parameters for the delay bound and for the measurement values for T=1, T=60 and T=3600 for level 1. The next four integers are the parameters for the delay bound and for the measurement values for T=1, T=60 and T=3600 for level 2. The last four integers are the parameters for the delay bound and for the measurement values for T=1, T=60 and T=3600 for level 3.

The following values are assigned from the characterization parameter name space.

Predictive service is service_name 3.

The delay characterization parameters are parameter_number's one through twelve, in the order given above. That is,

parameter_name	definition
1	Service Level = 1, Delay Bound
2	Service Level = 1, Delay Measure, T = 1
3	Service Level = 1, Delay Measure, T = 60
4	Service Level = 1, Delay Measure, T = 3600
5	Service Level = 2, Delay Bound
6	Service Level = 2, Delay Measure, T = 1
7	Service Level = 2, Delay Measure, T = 60
8	Service Level = 2, Delay Measure, T = 3600
9	Service Level = 3, Delay Bound

```
        10                  Service Level = 3, Delay Measure, T = 1
        11                  Service Level = 3, Delay Measure, T = 60
        12                  Service Level = 3, Delay Measure, T = 3600
```

The end-to-end composed results are assigned parameter_names N+12, where N is the value of the per-hop name given above.

No other exported data is required by this specification.

Policing

Policing is done at the edge of the network, at all heterogeneous source branch points and at all source merge points. A heterogeneous source branch point is a spot where the multicast distribution tree from a source branches to multiple distinct paths, and the TSpec's of the reservations on the various outgoing links are not all the same. Policing need only be done if the TSpec on the outgoing link is "less than" (in the sense described in the Ordering section) the TSpec reserved on the immediately upstream link. A source merge point is where the multicast distribution trees from two different sources (sharing the same reservation) merge. It is the responsibility of the invoker of the service (a setup protocol, local configuration tool, or similar mechanism) to identify points where policing is required. Policing is allowed at points other than those mentioned above.

The token bucket parameters require that traffic must obey the rule that over all time periods, the amount of data sent cannot exceed rT+b, where r and b are the token bucket parameters and T is the length of the time period. For the purposes of this accounting, links must count packets that are smaller than the minimal policing unit to be of size m. Packets that arrive at an element and cause a violation of the the rT+b bound are considered nonconformant. Policing to conformance with this token bucket is done in two different ways. At all policing point, non conforming packets are treated as best-effort datagrams. [If and when a marking ability becomes available, these nonconformant packets should be ``marked'' as being noncompliant and then treated as best effort packets at all subsequent routers.] Other actions, such as delaying packets until they are compliant, are not allowed.

NOTE: This point is open to discussion. The requirement given above may be too strict; it may be better to permit some delaying of a packet if that delay would allow it to pass the policing function. Intuitively, a plausible approach is to allow a delay of (roughly) up to the maximum queueing delay experienced by

completely conforming packets before declaring that a packet has failed to pass the policing function and dropping it. The merit of this approach, and the precise wording of the specification that describes it, require further study.

A related issue is that at all network elements, packets bigger than the MTU of the link must be considered nonconformant and should be classified as best effort (and will then either be fragmented or dropped according to the element's handling of best effort traffic). [Again, if marking is available, these reclassified packets should be marked.]

Ordering

TSpec's are ordered according to the following rule: TSpec A is a substitute ("as good or better than") for TSpec B if (1) both the token bucket depth and rate for TSpec A are greater than or equal to those of TSpec B, (2) the minimum policed unit m is at least as small for TSpec A as it is for TSpec B, and (3) the maximum packet size M is at least as large for TSpec A as it is for TSpec B.

A merged TSpec may be calculated over a set of TSpecs by taking the largest token bucket rate, largest bucket size, smallest minimal policed unit, and largest maximum packet size across all members of the set. This use of the word "merging" is similar to that in the RSVP protocol; a merged TSpec is one that is adequate to describe the traffic from any one of a number of flows.

Service request specifications (RSpecs) are ordered by their numerical values (in inverse order); service level 1 is substitutable for service level 2, and service level 2 is substitutable for service level 3.

In addition, predictive service is related to controlled delay service in the sense that a given level of predictive service is considered at least as good as the same level of controlled delay service. That is, predictive level 1 is substitutable for controlled delay level 1, and so on. See additional comments in the guidelines section.

Guidelines for Implementors

It is expected that the service levels implemented at a particular element will offer significantly different levels of delay bounds. There seems little advantage in offering levels whose delay bounds

differ only slightly. So, while a particular element may offer less than three levels of service, the levels of service it does offer should have notably different delay bounds. For example, appropriate delay bounds for three levels of predictive service are 1, 10 and 100 milliseconds.

For each level of service, packet loss and violation of the delay bound are expected to be very rare. As a preliminary guideline, we suggest that over long term use (measured in hours or days), the aggregate rate of delay bound violation and packet loss should be less than 1 in 10,000 packets. Violations of the delay bound are likely to be correlated. On shorter time scales, delay bound violation rates should not exceed 1 in 1,000 during any 60 second interval.

An additional service currently being considered is the controlled delay service described in [3]. It is expected that if an element offers both predictive service and controlled delay service, it should not implement both but should use the predictive service as a controlled delay service. This is allowed since (1) the required behavior of predictive service meets all of the requirements of controlled delay service, (2) the invocations are compatible, and (3) the ordering relationships are such that a given level of predictive service is at least as good as the same level of controlled delay service.

Evaluation Criteria

Evaluating a network element's implementation of predictive service is somewhat difficult, since the quality of service depends on overall traffic load and the traffic pattern presented. In this section we sketch out a methodology for testing a network element's predictive service.

The idea is that one chooses a particular traffic mix (for instance, three parts level 1, one part level 2, two parts level 3 and one part best-effort traffic) and loads the network element with progressively higher levels of this traffic mix (i.e., 40% of capacity, then 50% of capacity, on beyond 100% capacity). For each load level, one measures the utilization, mean delays, packet loss rate, and delay bound violation rate for each level of service (including best effort). Each test run at a particular load should involve enough traffic that is a reasonable predictor of the performance a long-lived application such as a video conference would experience (e.g., an hour or more of traffic).

This memo does not specify particular traffic mixes to test.

However, we expect in the future that as the nature of real-time Internet traffic is better understood, the traffic used in these tests will be chosen to reflect the current and future Internet load.

Examples of Implementation

One implementation of predictive service would be to have a queueing mechanism with three priority levels, with level 1 packets being highest priority and level 3 packets being lowest priority. Maximum packet delays and link utilization would be measured for each class over some relatively short interval, such as 10,000 packet transmission times. The admission control algorithm would use these measurements to determine whether or not to admit a new flow. Specifically, a new flow would be admitted if the network element expects to be able to meet the delay bounds of the packets in each service class after admitting a new flow. For an example of an admission control algorithm for Predictive service, see [5].

Note that the viability of measurement based admission control for predictive service depends on link bandwidth and traffic patterns. Specifically, with bursty traffic sources, sufficient multiplexing is needed for measurements of existing traffic to be good predictors of future traffic behavior. In an environments where sufficient multiplexing is not possible, parameter based admission control may be necessary.

Examples of Use

We give two examples of use, both involving an interactive application.

In the first example, we assume that either the receiving application is ignoring characterizations or the setup protocol is not delivering the characterizations to the end-nodes. We further assume that the application's data transmission units are timestamped. The receiver, by inspecting the timestamps, can determine the end-to-end delays and determine if they are excessive. If so, then the application asks for a better level of service. If the delays are well below the required level, the application can ask for a worse level of service.

In the second example, we assume that characterizations are delivered to the receiving application. The receiver chooses the worst service level whose characterization for the delay bound is less than the required level (once latencies are added in).

References

[1] S. Shenker and J. Wroclawski. "Network Element Service Specification Template", Internet Draft, June 1995, <draft-ietf-intserv-svc-template-01.txt>

[2] S. Shenker and C. Partridge. "Specification of Guaranteed Quality of Service", Internet Draft, July 1995, <draft-ietf-intserv-guaranteed-svc-01.txt>

[3] S. Shenker and C. Partridge and J. Wroclawski. "Specification of Controlled Delay Quality of Service",Internet Draft, June 1995, <draft-ietf-intserv-control-del-svc-01.txt>

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

[5] S. Jamin, P. Danzig, S. Shenker and L. Zhang, "A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks", Sigcomm '95, September 1995.

[6] D. Clark, S. Shenker and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", Sigcomm '92, October 1992.

Security Considerations

Security considerations are not discussed in this memo.

Authors' Address:

Scott Shenker
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304-1314
shenker@parc.xerox.com
415-812-4840
415-812-4471 (FAX)

Craig Partridge
BBN
2370 Amherst St
Palo Alto CA 94306
craig@bbn.com

Bruce Davie
Bellcore
445 South St

Morristown, NJ, 07960
bsd@bellcore.com

Lee Breslau
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304-1314
breslau@parc.xerox.com
415-812-4402
415-812-4471 (FAX)