Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-intserv-control-del-svc-02.txt

Integrated Services WG
Shenker/Partridge/Wroclawski
Xerox/BBN/MIT

14 November 1995
Expires: ?/?/96

Specification of Controlled Delay Quality of Service

Status of this Memo

This document is a product of the Integrated Services working group of the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at int-serv@isi.edu and/or the author(s).

This draft reflects changes from the IETF meeting in Stockholm.

Abstract

This memo describes the network element behavior required to deliver Controlled Delay service in the Internet. Controlled delay service provides three levels of delay control; network elements, when overloaded, are required to control delay by denying service requests. However, there are no quantitative assurances about the absolute level of delay provided. The controlled delay service is designed for service-adaptive and

delay-adaptive applications; i.e., applications that are prepared to dynamically adapt to changing packet transmission delays and to dynamically change the level of packet delivery delay control they request from the network when their current level of service is not adequate. The controlled delay service imposes relatively minimal requirements on network components that implement it, and is intended to be usable in situations ranging from small centrally managed private IP networks to the global Internet. This specification follows the service specification template described in [1].

Introduction

This document defines the requirements for network elements that support Controlled Delay 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.

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. This service ensures that the levels of experienced delays and losses will be controlled, in that additional service requests will be turned away when the element is overloaded. In particular, the bandwidth available to the flow will be, on average, at least as great as specified in its service request. Criteria for determining when a resource is overloaded are not specified in this definition, but are left to the individual vendor. This service makes no assurances about the absolute levels of delay or jitter the receiving application will experience. However, all three levels of controlled delay 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 service is subject to admission control.

Motivation

Controlled delay service is designed for service-adaptive and delay-adaptive applications. These applications are sensitive to packet delivery delay, but are prepared to adapt to dynamically changing delays by varying their playback point. In addition, they may be prepared to change their requested level of service at any time if the current level of service received from the network is not adequate. This flexibility allows such applications to operate successfully and efficiently over a wide range of network conditions.

Many applications that transmit interactive data, such as audio and video conferencing sessions, are well suited to operation with the controlled delay service. Applications that desire proven guarantees on packet delivery time, such as real-time control and servoing systems or playback applications that are intolerant of late-arriving packets, are generally not in this category.

The end-to-end behavior obtained with controlled delay service provides a middle ground between the employment of adaptive applications in a pure best-effort network and the employment of a network that rigidly controls delay. Strengths of this middle ground are that applications can obtain some load control and delivery preference for their packets while still benefiting from their adaptive behavior; that the service can be usefully deployed in large, unstructured internetworks; and that the specification is amenable to highly efficient implementation and use of network resources.

Associated with this service are characterization parameters which describe 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. This is intended to aid applications in choosing the appropriate service level. However, this service is still quite usable without these characterizations.

Network Element Data Handling Requirements

The network element must ensure that the packet loss and delays are controlled. This must be accomplished through active admission control. In particular, overprovisioning is not sufficient to deliver controlled delay service; the element must be able to turn flows away if accepting them would cause the element to have excessive queueing delays. However, no quantitative specification of

average, statistical, or maximal delays is required.

There are three different logical levels of service. A network element may internally implement fewer (or more) actual levels of service, but must map them into three logical levels at the controlled delay service invocation interface. The levels have different degrees of delay control, with level 1 having the most tightly controlled delay, and level 3 having the least tightly controlled delay. The different levels do not have to give strictly ordered delays for each packet; that is, the network element need not ensure that every packet given level 1 service experiences less delay than if it were given level 2 service. The element need only ensure that the typical delays are no greater in level 1 than in level 2 (and similarly for levels 2 and 3).

All three levels of service should be given better service, i.e. more tightly controlled delay, than uncontrolled best effort traffic. The average delays experienced by packets receiving different levels of controlled delay 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 controlled delay service that are implemented and for best-effort service should be significantly different when the network has substantial load.

The controlled delay service must maintain a very low level of packet loss. Although packet losses may occur, any substantial loss 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 loss (varying, for instance, from 1 in 10^4 to 1 in 10^8).

The controlled delay service definition does not require any control of short-term packet jitter (variation in network element transit delay between different packets in the flow) beyond the control already exercised on delay. Network element implementors who find it advantageous to do so may use resource scheduling algorithms that exercise some jitter control.

Links are not permitted to fragment packets as part of controlled delay 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 controlled delay 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 controlled delay service module exports at least the following information. All of the parameters described below are characterization parameters.

For each level of service, the network element exports three measurements of delay (thus making nine quantities in total). Each of these 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 allowed to avoid placing undue burdens on network element designs in which obtaining precise delay measurements is difficult.

These delay parameters 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 delays 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 different elements delay exceed 2**32-1, the end-to-end advertised delay should be 2**32-1.

Note that while the granularity of measurement is microseconds, a conforming element is free to measure delays more loosely. The minimum requirement is that the element estimate its delay accurately to the nearest 100 microsecond granularity. Elements that can

measure more accurately are, of course, encouraged to do so.

NOTE: Measuring in milliseconds is not acceptable, because if the minimum delay value is a millisecond, a path with several hops will lead to a composed delay of at least several milliseconds, which is likely to be misleading.

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

The following values are assigned from the characterization parameter namespace.

The controlled delay service is service_name 1.

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

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

The end-to-end composed results are assigned parameter_names N+10, 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 non-compliant 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: The prohibition on delaying packets is open to discussion. It may be better to permit some delaying of a packet if that delay would allow it to pass the policing function. (In other words, to reshape the traffic). The challenge is to define a viable reshaping 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. 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 and Merging

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 3, and service level 2 is substitutable for service level 3.

Guidelines for Implementors

It is expected that the service levels implemented at a particular element will offer significantly different levels of delay control. There seems little advantage in offering levels that differ only slightly in the level of delay control. So, while a particular element may offer less than three levels of service, the levels of service it does offer should have notably different queueing delays.

> NOTE: An additional service currently being considered is the "predictive" service described in [3]. It is expected that if an element offers both predictive service and controlled delay service, that 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 defined in the predictive service specification document are such that a given level of predictive service is at least as good as the same level of controlled delay service. The inter-service mapping with predictive service, mentioned above, is omitted from the "Ordering and Merging" section of this draft of the controlled delay service specification because the exact definition of both services is still under discussion. Should the final definitions include an

inter-service mapping function, the Ordering and Merging sections of each document might contain words similar to the following:

"In addition, the controlled delay service is related to the predictive 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. See additional comments in the guidelines section."

Network elements are permitted to oversubscribe their traffic, where by oversubscribe, we mean that the sum of the token buckets of the controlled delay traffic exceeds the maximum throughput or buffer space of the router. However, given the requirement of low loss, this oversubscribing should only be done in cases where the element is quite sure that actual utilization is far less than the sum of the token buckets would suggest. A more conservative approach is to reject new flows, when the addition of their traffic would cause the sums of the token buckets to exceed the capacity of the network element.

Evaluation Criteria

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

The idea is that one chooses a particular traffic mix (for instance, 30 percent level 1, 10 percent level 2, 20 percent level 3 and 40 percent uncontrolled best-effort traffic) and loads the network element with progressively higher amounts 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, and the packet loss 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

A possible implementation of controlled delay 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. Each controlled delay service level would be associated with a target queue utilization level, say 20% for level 1, 50% for the combination of levels 1 and 2, and 70% for the combination of all three levels. The utilization of the link, by each of the three levels, would be measured over some relatively short time period (say, 5 seconds, or 10000 MTU packet transmission times). A new flow would be admitted to level 1 if the measured usage of level 1, plus the token bucket rate of the new flow, was below the target utilization of level 1. Similarly, a new flow would be admitted to level 2 if the measured usage of levels 1 and 2, plus the token bucket rate of the new flow, was below the target utilization of levels 1 and 2.

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 network is not delivering the characterizations to the end-nodes. We further assume that the application's data transmission units is timestamped. The receiver, by inspecting the timestamps, can determine the end-to-end delays and react 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. A protocol useful to applications providing this capability is the proposed IETF Real-Time Transport Protocol [2].

In the second example, we assume that characterization parameters are delivered to the receiving application. The receiver chooses the service level whose characterizations for the maximal delays for all intervals are under the required level after network latencies are considered. If the actual delays during the course of operation are worse than expected, the application can ask for a better level of service.

Security Considerations

Security considerations are not discussed in this memo.

References

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

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. "RTP: A Transport Protocol for Real-Time Applications", Internet Draft, March 1995, <draft-ietf-avt-svc-rtp-07.txt>

[3] S. Shenker, C. Partridge, B. Davie, and L. Breslau. "Specification of Predictive Quality of Service", Internet Draft, ?? 1995, <draft-ietf-intserv-predictive-svc-01.txt>

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

John Wroclawski
MIT Laboratory for Computer Science
545 Technology Sq.
Cambridge, MA 02139
jtw@lcs.mit.edu
617-253-7885
617-253-2673 (FAX)