Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-intserv-ctrl-load-svc-00.txt

Integrated Services WG
J. Wroclawski
MIT LCS

November, 1995
Expires: 5/96

Specification of the Controlled-Load Network Element Service

Status of this Memo

Abstract

This memo specifies the network element behavior required to deliver Controlled-Load service in the Internet. The controlled-load service provides the client data flow with a quality of service closely approximating the QoS that same flow would receive from an unloaded network element, but uses capacity (admission) control to assure that this service is received even when the network element is overloaded.

Introduction

This document defines the requirements for network elements that support the Controlled-Load 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 to an application by a series of network elements conforming to this specification tightly approximates the behavior visible to applications receiving best-effort service *under unloaded conditions* from the same series of network elements. Assuming the network is functioning correctly, these applications may assume that:

- A very high percentage of transmitted packets will be successfully delivered by the network to the receiving end-nodes. (The percentage of packets not successfully delivered must closely approximate the basic packet error rate of the transmission medium).

- The transit delay experienced by a very high percentage of the delivered packets will not greatly exceed the minimum transmit delay experienced by any successfully delivered packet (the "speed-of-light delay").

NOTE: the term "unloaded" above is used in the sense of "not heavily loaded or congested" rather than in the sense of "no other network traffic whatsoever".

To ensure that these conditions are met, clients requesting

controlled-load service provide the intermediate network elements with a estimation of the data traffic they will generate; the TSpec. In return, the service ensures that network element resources adequate to process traffic falling within this descriptive envelope will be available to the client. Should the client's traffic generation properties fall outside of the region described by the TSpec parameters, the QoS provided to the client may exhibit characteristics indicative of overload, including large numbers of delayed or dropped packets. The service definition does not require that the precise characteristics of this overload behavior match those which would be received by a best-effort data flow traversing the same path under overloaded conditions.

Motivation

The controlled load service is intended to support a broad class of applications which have been developed for use in today's Internet, but are highly sensitive to overloaded conditions. Important examples of this class are the "adaptive real-time applications" currently offered by a number of vendors and researchers. These applications have been shown to work well on unloaded nets, but poorly on much of todays overloaded Internet. A service which mimics unloaded nets serves these applications well.

The controlled-load service is intentionally minimal, in that there are no optional functions or capabilities in the specification. The service offers only a single function but system and application designers can assume that all implementations will be indentical in this respect.

Internally, the controlled-load service is suited to a wide range of implementation techniques; including evolving scheduling and admission control algorithms which allow sophisticated implementations to be highly efficient in the use of network resources. It is equally amenable to extremely simple implementation in circumstances where maximum utilization of network resources is not the only concern.

Network Element Data Handling Requirements

Each network element accepting a request for controlled-load service must ensure that adequate bandwidth and packet processing resources are available to handle the requested level of traffic, as given by the requestor's TSpec. This must be accomplished through active

admission control. All resources important to the operation of the network element must be considered when admitting a request. Common examples of such resources include link bandwidth, router or switch port buffer space, and computational capacity of the packet forwarding engine.

The controlled-load service does not accept or make use of specific target values for control parameters such as delay or loss. Instead, acceptance of a request for controlled-load service is defined to imply a commitment by the network element to provide the requestor with service closely equivalent to that provided to uncontrolled (best-effort) traffic under unloaded conditions. This definition may be taken to include:

- Little or no average packet queueing delay over all timescales significantly larger than the "burst time". The burst time is defined as the time required for the flow's maximum size data burst to be transmitted at the flow's requested transmission rate, where the burst size and rate are given by the flow's TSpec, as described below.

- A very low level of congestion loss. In this context, congestion loss includes packet losses due to shortage of any required processing resource, such as buffer space or link bandwidth. Although occasional congestion losses may occur, any substantial sustained loss represents a failure of the admission control algorithm.

NOTE:

Implementations of controlled-load service are not required to provide any control of short-term packet delay jitter beyond that described above. However, the use of packet scheduling algorithms that provide additional jitter control is not prohibited by this specification.

Packet losses due to non-congestion-related causes, such as link errors, are not bounded by this service.

A network element may employ statistical approaches to decide whether adequate capacity is available to accept a service request. For example, a network element processing a number of flows with long-

term characteristics predicted through measurement may be able to overallocate its resources to some extent without reducing the level of service delivered to the flows.

A network element may employ any appropriate means to ensure that admitted flows receive appropriate service.

Links are not permitted to fragment packets which receive the controlled-load service. Packets larger than the MTU of the link must be treated as nonconformant to the TSpec. This implies that they will be policed according to the rules described in the Policing section below.

The controlled-load service is invoked by specifying the data flow's desired traffic parameters (TSpec) to the network element. Requests placed for a new flow will be accepted if the network element has the capacity to forward the flow's packets as described above. Requests to change the TSpec for an existing flow should be treated as a new invocation, in the sense that admission control must be reapplied to the flow. Requests that reduce the TSpec for an existing flow (in the sense that the new TSpec is strictly smaller than the old TSpec according to the ordering rules given below) should never be denied service.

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

The token bucket specification includes a bucket rate r and a bucket depth, b. Both r and b must be positive. The rate, r, is measured in bytes of IP datagrams per second. Values of this parameter may range from 1 byte per second to 40 terabytes per second. Network elements MUST return an error for requests containing values outside this range. Network elements MUST return an error for any request containing a value within this range which cannot be supported by the element. In practice, only the first few digits of the r parameter are significant, so the use of floating point representations, accurate to at least 0.1% is encouraged.

The bucket depth, b, is measured in bytes. Values of this parameter may range from 1 byte to 250 gigabytes. Network elements MUST return an error for requests containing values outside this range. Network elements MUST return an error for any request containing a value within this range which cannot be supported by the element. In practice, only the first few digits of the b parameter are significant, so the use of floating point representations, accurate to at least 0.1% is encouraged.

The range of values allowed for these parameters is intentionally

large to allow for future network technologies. Any given network element is not expected to support the full range of values.

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. Network elements MUST reject a service request 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 preferred concrete representation for the TSpec is 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).

Exported Information

The controlled-load service is assigned service_name 5.

The controlled-load service has no required characterization parameters. Specific implementations may export appropriate measurement and monitoring information.

Policing

The controlled-load service is suitable for use with multicast as well as unicast data flows. This capability introduces some complexity into the policing requirements.

Controlled-load traffic must be policed for conformance to its TSpec at every network element. The TSpec's token bucket parameters require that traffic must obey the rule that over all time periods, the amount of data sent does not 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.

At all policing points, non-conforming packets are treated as BEST-EFFORT datagrams. (See the NOTEs below for further discussion of this

issue).

If resources are available, it is desirable for the policing function at points within the interior of the network (but *not* at edge traffic entry points) to enforce slightly "relaxed" traffic parameters to accommodate packet bursts somewhat larger than the actual TSpec.

Other actions, such as reshaping the traffic stream (delaying packets until they are compliant), are not allowed.

NOTE: RESHAPING. The prohibition on delaying packets is one of many possible design choices. 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.

NOTE: INTERACTION WITH BEST-EFFORT TRAFFIC. Implementors of this service should clearly understand that in certain circumstances (routers acting as the "split points" of a multicast distribution tree supporting a shared reservation) large numbers of packets may fail the policing test *as a matter of normal operation*. According to the definition above, these packets should be processed as best-effort packets.

If the network element's best-effort queueing algorithm does not distinguish between these packets and elastic best-effort traffic such as TCP flows, THESE PACKETS WILL "BACK OFF" THE ELASTIC TRAFFIC AND DOMINATE THE BEST-EFFORT BANDWIDTH USAGE. The integrated services framework does not currently address this issue. However, several possible solutions to the problem are known [RED, xFQ]. Network elements supporting the controlled load service should also implement some mechanism in their best-effort queueing path to discriminate between classes of best-effort traffic and provide elastic traffic with protection from inelastic best-effort flows.

NOTE: EDGE POLICING. The text above specifys that the policing

function treats non-conformant packets as best-effort at all points. A possible alternative is to replace this with language reading:

> At points where traffic first enters the network (end-nodes), non-conforming packets are DROPPED. At these points, the reservation setup mechanism must ensure that the TSpec used is *no smaller* than the TSpec specified by the source for the traffic it is generating.
>
> At all other policing points, non-conforming packets are treated as BEST-EFFORT datagrams.

The effect of this change is significant. Under the non-dropping model, it is possible for a source to vastly over-send its TSpec, with the excess packets being delivered if conditions permit. The service offered in this case has been described as "best-effort-with-floor"; essentially a best-effort delivery service with enough resources reserved for a certain minimum traffic level.

Under the dropping model, the service loses its "best-effort-with-floor" characteristics, and becomes essentially a fixed-traffic-level service. In return, it offers significantly more protection against overload of the network resources and degradation of other flows' QoS.

NOTE: ARCHITECTURAL OPTIONS. The text above specifies a functional and consistant model for policing of controlled-load data which can be implemented within the current IP protocols.

In this model, it is necessary to police at every network element because the policing function does not actually drop traffic which exceeds the TSpec, but instead carries it as best-effort. Since there is no end-to-end mechanism in place to limit a controlled-load flow's traffic to the TSpec value, every network element must perform this function for itself. Since excess controlled-load traffic (traffic above the TSpec) is not dropped, every network element should also perform the best-effort service discrimination function described above.

The alternative option of "marking" packets which have failed the policing test at some node is not available within the current IP protocol. If marking were available, it would be necessary to police only at certain points within the network. In this case, the relevant language above might be replaced with a paragraph

reading:

> Policing is performed 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 occurs when 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.

Note that the best-effort traffic discrimination function described above must still be performed at every network element. In this case, the discrimination might be based in part on the mark bit.

At all network elements, packets bigger than the outgoing link MTU must be considered nonconformant and classified as best effort (and will then either be fragmented or dropped according to the element's handling of best effort traffic). It is expected that this situation will not arise with any frequency, because flow setup mechanisms are expected to notify the sending application of the appropriate path MTU.

Ordering and Merging

The controlled-load service TSpec is ordered according to the following rule: TSpec A is a substitute for ("as good or better than") TSpec B if and only 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.

The sum of n controlled-load service TSpecs is used when computing the TSpec for a shared reservation of n flows. It is computed by taking:

- The minimum across all TSpecs of the minimum policed unit parameter m.

- The maximum across all TSpecs of the maximum packet size parameter M.

- The sum across all TSpecs of the token bucket rate parameter r.

- The sum across all TSpecs of the token bucket size parameter b.

The perfect minimum of two TSpecs is defined as a TSpec which would view as compliant any traffic flow that complied with both of the original TSpecs, but would reject any flow that was non-compliant with at least one of the original TSpecs. This perfect minimum can be computed only when the two original TSpecs are ordered, in the sense described above.

A definition for computing the minimum of two unordered TSpecs is:

- The minimum of the minimum policed units m.

- The maximum of the maximum packet sizes M.

- The minimum of the token bucket rates r.

- The maximum of the token bucket sizes b.

NOTE: The proper definition the minimum TSpec function is a topic of current discussion. The definition above is provisional and subject to change.

Guidelines for Implementors

The intention of this service specification is that network elements deliver a level of service closely approximating best-effort service under unloaded conditions. As with best-effort service under these conditions, it is not required that every single packet must be successfully delivered with zero queueing delay. Network elements providing controlled-load service are permitted to oversubscribe the available resources to some extent, in the sense that the bandwidth and buffer requirements indicated by summing the TSpec token buckets of all controlled-load flows may exceed the maximum capabilities of the network element. However, this oversubscription may 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. The most conservative approach, rejection of new flows whenever the addition of their traffic would cause the sums of the token buckets to exceed the capacity of the network element, may be appropriate in other circumstances.

Specific issues related to this subject are discussed in the "Evaluation Criteria" and "Examples of Implementation" sections below.

Implementors are encouraged (but not required) to implement policing behavior (the behavior seen when a flow's actual traffic exceeds its TSpec) which closely approximates the behavior of well-designed best-effort services under overload. In particular, it is undesirable to employ queueing models which lead to heavily bi-modal delay distributions or large numbers of mis-ordered packet arrivals.

Evaluation Criteria

The basic requirement placed on an implementation of controlled-load service is that, under all conditions, it provide accepted data flows with service closely similar to the service that same flow would receive using best-effort service under unloaded conditions.

This suggests a simple two-step evaluation strategy. Step one is to compare the service given best-effort traffic and controlled-load traffic under underloaded conditions.

- Measure the packet loss rate and delay characteristics of a test flow using best-effort service and with no load on the network element.

- Compare those measurements with measurements of the same flow receiving controlled-load service with no load on the network element.

Closer measurements indicate higher evaluation ratings. A substantial difference in the delay characteristics, such as the smoothing which would be seen in an implementation which scheduled the controlled-load flow using a fixed, constant-bitrate algorithm, should result in a somewhat lower rating.

Step two is to observe the change in service received by a controlled-load flow as the load increases.

- Increase the background traffic load on the network element, while continuing to measuring the loss and delay characteristics of the controlled-load flow. Characteristics which remain essentially constant as the element is driven into overload indicate a high evaluation rating. Minor changes in the delay distribution indicate a somewhat lower rating. Significant increases in delay or loss indicate a poor evaluation rating.

This simple model is not adequate to fully evaluate the performance of controlled-load service. Three additional variables affect the evaluation. The first is the short-term burstiness of the traffic

stream used to perform the tests outlined above. The second is the degree of long-term change in the controlled-load traffic within the bounds of its TSpec. (Changes in this characteristic will have great effect on the effectiveness of certain admission control algorithms.) The third is the ratio of controlled-load traffic to other traffic at the network element (either best effort or other controlled services).

The third variable should be specifically evaluated using the following procedure.

With no controlled-load flows in place, overload the network element with best-effort traffic (as indicated by substantial packet loss and queueing delay).

Execute requests for controlled-load service giving TSpecs with increasingly large rate and burst parameters. If the request is accepted, verify that traffic matching the TSpec is in fact handled with characteristics closely approximating the unloaded measurements taken above.

Repeat these experiments to determine the range of traffic parameter (rate, burst size) values successfully handled by the network element. The useful range of each parameter must be determined for several settings of the other parameter, to map out a two-dimensional "region" of successfully handled TSpecs. When compared with network elements providing similar capabilities, this region indicates the relative ability of the elements to provide controlled-load service under high load. A larger region indicates a higher evaluation rating.

Examples of Implementation

One possible implementation of controlled-load service is to provide a queueing mechanism with two priority levels; a high priority one for controlled-load and a lower priority one for best effort service. An admission control algorithm is used to limit the amount of traffic placed into the high-priority queue. This algorithm may be based either on the specified characteristics of the high-priority flows (using information provided by the TSpecs), or on the measured characteristics of the existing high-priority flows and the TSpec of the new request.

Another possible implementation of controlled-load service is based on the existing capabilities of network elements which support "traffic classes" based on mechanisms such as weighted fair queueing or class-based queueing [xxx]. In this case, it is sufficient to map data flows accepted for controlled-load service into an existing traffic class with adequate capacity to avoid overload. This requirement is enforced by an admission control algorithm which considers the characteristics of the traffic class, the characteristics of the traffic already admitted to the class, and the TSpec of the new flow requesting service. Again, the admission control algorithm may be based either on the TSpec-specified or the measured characteristics of the existing traffic.

Admission control algorithms based on specified characteristics are likely be appropriate when the number of flows in the high-priority class is small, or the traffic characteristics of the flows appear highly variable. In these situations the measured behavior of the aggregate controlled-load traffic stream may not serve as an effective predictor of future traffic, leading a measurement-based admission control algorithm to produce incorrect results. Conversely, in situations where the past behavior of the aggregate controlled-load traffic *is* a good predictor of future behavior, a measurement-based admission control algorithm may allow more traffic to be admitted to the controlled-load service class with no degradation in performance. An implementation may choose to switch between these two approaches depending on the nature of the traffic stream at a given time.

Examples of Use

The controlled-load service may be used by any application which can make use of best-effort service, but is best suited to those applications which can usefully characterize their traffic requirements. Applications based on the transport of "continuous media" data, such as digitized audio or video, are an important example of this class.

The controlled-load service is not isochronous and does not provide any explicit information about transmission delay. For this reason, applications with end-to-end timing requirements, including the continuous-media class mentioned above, provide an application-specific timing recovery mechanism, similar or identical to the mechanisms required when these applications use best-effort service. A protocol useful to applications requiring this capability is the IETF Real-Time Transport Protocol [2].

Load-sensitive applications may choose to request controlled-load service whenever they are run. Alternatively, these applications may monitor their own performance and request controlled-load service from the network only when best-effort service is not providing acceptable performance. The first strategy provides higher assurance that the level of quality delivered to the user will not change over the lifetime of an application session. The second strategy provides greated flexibility and offers cost savings in environments where levels of service above best-effort incur a charge.

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>

Authors' Address:

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