Network Working Group
Request for Comments: 3006
Category: Standards Track

B. Davie
C. Iturralde
D. Oran
Cisco Systems, Inc.
S. Casner
Packet Design
J. Wroclawski
MIT LCS

November 2000

Integrated Services in the Presence of Compressible Flows

Status of this Memo

Copyright Notice

Abstract

An Integrated Services (int-serv) router performs admission control and resource allocation based on the information contained in a TSpec (among other things). As currently defined, TSpecs convey information about the data rate (using a token bucket) and range of packet sizes of the flow in question. However, the TSpec may not be an accurate representation of the resources needed to support the reservation if the router is able to compress the data at the link level. This specification describes an extension to the TSpec which enables a sender of potentially compressible data to provide hints to int-serv routers about the compressibility they may obtain. Routers which support appropriate compression take advantage of the hint in their admission control decisions and resource allocation procedures; other routers ignore the hint. An initial application of this approach is to notify routers performing real-time transport protocol (RTP) header compression that they may allocate fewer resources to RTP flows.

Table of Contents

1. Introduction

In an Integrated Services network, RSVP [RFC 2205] may be used as a signalling protocol by which end nodes and network elements exchange information about resource requirements, resource availability, and the establishment and removal of resource reservations. The Integrated Services architecture currently defines two services, Controlled-Load [RFC 2211] and Guaranteed [RFC 2212]. When establishing a reservation using either service, RSVP requires a variety of information to be provided by the sender(s) and receiver(s) for a particular reservation which is used for the purposes of admission control and allocation of resources to the reservation. Some of this information is provided by the receiver in a FLOWSPEC object; some is provided by the sender in a SENDER_TSPEC object [RFC 2210].

A situation that is not handled well by the current specs arises when a router that is making an admission control decision is able to perform some sort of compression on the flow for which a reservation is requested. For example, suppose a router is able to perform IP/UDP/RTP header compression on one of its interfaces [RFC 2508]. The bandwidth needed to accommodate a compressible flow on that interface would be less than the amount contained in the SENDER_TSPEC. Thus the router might erroneously reject a reservation that could in fact have been accommodated. At the same time, the sender is not at liberty to reduce its TSpec to account for the compression of the data, since it does not know if the routers along the path are in fact able to perform compression. Furthermore, it is probable that only a subset of the routers on the path (e.g., those connected to low-speed serial links) will perform compression.

This specification describes a mechanism by which the sender can provide a hint to network elements regarding the compressibility of the data stream that it will generate. Network elements may use this hint as an additional piece of data when making admission control and resource allocation decisions.

This specification is restricted to the case where compression is performed only on a link-by-link basis, as with header compression. Other cases (e.g., transcoding, audio silence detection) which would affect the bandwidth consumed at all downstream nodes are for further study. In these latter cases, it would be necessary to modify a sender TSpec as it is passed through a compressing node. In the approach presented here, the sender TSpec that appears on the wire is never modified, just as specified in [RFC 2210].

2. Addition of a Hint to the Sender TSpec

The appropriate place for a `compressibility hint' is the Sender TSpec. The reasons for this choice are:

- The sender is the party who knows best what the data will look like.
- Unlike the Adspec, the Sender TSpec is not modified in transit
- From the perspective of RSVP, the Sender TSpec is a set of opaque parameters that are passed to `traffic control' (admission control and resource allocation); the compressibility hint is just such a parameter.

An alternative to putting this information in the TSpec would be to use an additional object in the RSVP PATH message. While this could be made to work for RSVP, it does not address the issue of how to get the same information to an intserv router when mechanisms other than RSVP are used to reserve resources. It would also imply a change to RSVP message processing just for the purposes of getting more information to entities that are logically not part of RSVP (admission control and resource allocation). The inclusion of the information in the TSpec seems preferable and more consistent with the Integrated Services architecture.

The contents of the hint are likely to vary depending on the exact scenario. The hint needs to tell the routers that receive it:

- the type of compression that is possible on this flow (e.g. IP/UDP/RTP);

- enough information to enable a router to determine the likely compression ratio that may be achieved.

In a simple case such as IP/UDP/RTP header compression, it may be sufficient to tell the routers nothing more than the fact that IP/UDP/RTP data is being sent. Knowing this fact, the maximum packet size of the flow (from the TSpec), and the local conditions at the router, may be sufficient to allow the router to determine the reduction in bandwidth that compression will allow. In other cases, it may be helpful or necessary for the sender to include additional quantitative information to assist in the calculation of the compression ratio. To handle these cases, additional parameters containing various amounts of information may be added to the sender TSpec. Details of the encoding of these parameters, following the approach originally described in [RFC 2210] are described below.

3. Admission Control and Resource Allocation

Integrated Services routers make admission control and resource allocation decisions based on, among other things, information in the sender TSpec. If a router receives a sender TSpec which contains a compressibility hint, it may use the hint to calculate a `compressed TSpec' which can be used as input to the admission control and resource allocation processes in place of the TSpec provided by the sender. To make this concrete, consider the following simple example. A router receives a reservation request for controlled load service where:

- The Sender TSpec and Receiver TSpec contain identical token bucket parameters;
- The rate parameter in the token bucket (r) is 48 kbps;
- The token bucket depth (b) is 120 bytes;
- The maximum packet size (M) in the TSpecs is 120 bytes;
- The minimum policed unit (m) is 64 bytes;
- The Sender TSpec contains a compressibility hint indicating that the data is IP/UDP/RTP;
- The compressibility hint includes a compression factor of 70%, meaning that IP/UDP/RTP header compression will cause a reduction in bandwidth consumed at the link level by a factor of 0.7 (the result of compressing 40 bytes of IP/UDP/RTP header to 4 bytes on a 120 byte packet)

- The interface on which the reservation is to be installed is able to perform IP/UDP/RTP header compression.

The router may thus conclude that it can scale down the token bucket parameters r and b by a factor of 0.7, i.e., to 33.6 kbps and 84 bytes respectively. M may be scaled down by the same factor (to 84 bytes), but a different calculation should be used for m. If the sender actually sends a packet of size m, its header may be compressed from 40 bytes to 4, thus reducing the packet to 28 bytes; this value should be used for m.

Note that if the source always sends packets of the same size and IP/UDP/RTP always works perfectly, the compression factor is not strictly needed. The router can independently determine that it can compress the 40 bytes of IP/UDP/RTP header to 4 bytes (with high probability). To determine the worst-case (smallest) gain provided by compression, it can assume that the sender always sends maximum sized packets at 48 kbps, i.e., a 120 byte packet every 20 milliseconds. The router can conclude that these packets would be compressed to 84 bytes, yielding a token bucket rate of 33.6 kbps and a token bucket depth of 84 bytes as before. If the sender is willing to allow an independent calculation of compression gain by the router, the explicit compression factor may be omitted from the TSpec. Details of the TSpec encoding are provided below.

To generalize the above discussion, assume that the Sender TSpec consists of values (r, b, p, M, m), that the explicit compression factor provided by the sender is f percent, and that the number of bytes saved by compression is N, independent of packet size. The parameters in the compressed TSpec would be:

```
r' = r * f/100
b' = b * f/100
p' = p
M' = M-N
m' = m-N
```

The calculations for r' and b' reflect that fact that f is expressed as a percentage and must therefore be divided by 100. The calculations for M' and m' hold only in the case where the compression algorithm reduces packets by a certain number of bytes independent of content or length of the packet, as is true for header compression. Other compression algorithms may not have this property. In determining the value of N, the router may need to make worst case assumptions about the number of bytes that may be removed by compression, which depends on such factors as the presence of UDP checksums and the linearity of RTP timestamps.

All these adjusted values are used in the compressed TSpec. The router's admission control and resource allocation algorithms should behave as if the sender TSpec contained those values. [RFC 2205] provides a set of rules by which sender and receiver TSpecs are combined to calculate a single `effective' TSpec that is passed to admission control. When a reservation covering multiple senders is to be installed, it is necessary to reduce each sender TSpec by its appropriate compression factor. The set of sender TSpecs that apply to a single reservation on an interface are added together to form the effective sender TSpec, which is passed to traffic control. The effective receiver TSpec need not be modified; traffic control takes the greatest lower bound of these two TSpecs when making its admission control and resource allocation decisions.

The handling of the receiver RSpec depends on whether controlled load or guaranteed service is used. In the case of controlled load, no additional processing of RSpec is needed. However, a guaranteed service RSpec contains a rate term R which does need to be adjusted downwards to account for compression. To determine how R should be adjusted, we note that the receiver has chosen R to meet a certain delay goal, and that the terms in the delay equation that depend on R are b/R and C/R (when the peak rate is large). The burstsize b in this case is the sum of the burstsizes of all the senders for this reservation, and each of these numbers has been scaled down by the appropriate compression factor. Thus, R should be scaled down using an average compression factor

$$f\_avg = (b1*f1 + b2*f2 + ... + bn*fn)/(b1 + b2 + ... bn)$$

where bk is the burstsize of sender k and fk is the corresponding compression factor for this sender. Note that f_avg, like the individual fi's, is a percentage. Note also that this results in a compression factor of f in the case where all senders use the same compression factor f.

To prevent an increase in delay caused by the C/R term when the reduced value of R is used for the reservation, it is necessary for this hop to `inflate' its value of C by dividing it by (f_avg/100). This will cause the contribution to delay made by this hop's C term to be what the receiver would expect when it chooses its value of R.

There are certain risks in adjusting the resource requirements downwards for the purposes of admission control and resource allocation. Most compression algorithms are not completely deterministic, and thus there is a risk that a flow will turn out to be less compressible than had been assumed by admission control. This risk is reduced by the use of the explicit compression factor provided by the sender, and may be minimized if the router makes

worst case assumptions about the amount of compression that may be achieved. This is somewhat analogous to the tradeoff between making worst case assumptions when performing admission control or making more optimistic assumptions, as in the case of measurement-based admission control. If a flow turns out to be less compressible that had been assumed when performing admission control, any extra traffic will need to be policed according to normal intserv rules. For example, if the router assumed that the 48 kbps stream above could be compressed to 33.6 kbps and it was ultimately possible to compress it to 35 kbps, the extra 1.4 kbps would be treated as excess. The exact treatment of such excess is service dependent.

A similar scenario may arise if a sender claims that data for a certain session is compressible when in fact it is not, or overstates the extent of its compressibility. This might cause the flow to be erroneously admitted, and would cause insufficient resources to be allocated to it. To prevent such behavior from adversely affecting other reserved flows, any flow that sends a compressibility hint should be policed (in any router that has made use of the hint for its admission control) on the assumption that it is indeed compressible, i.e., using the compressed TSpec. That is, if the flow is found to be less compressible than advertised, the extra traffic that must be forwarded by the router above the compressed TSpec will be policed according to intserv rules appropriate for the service. Note that services that use the maximum datagram size M for policing purposes (e.g. guaranteed service [RFC 2210]) should continue to use the uncompressed value of M to allow for the possibility that some packets may not be successfully compressed.

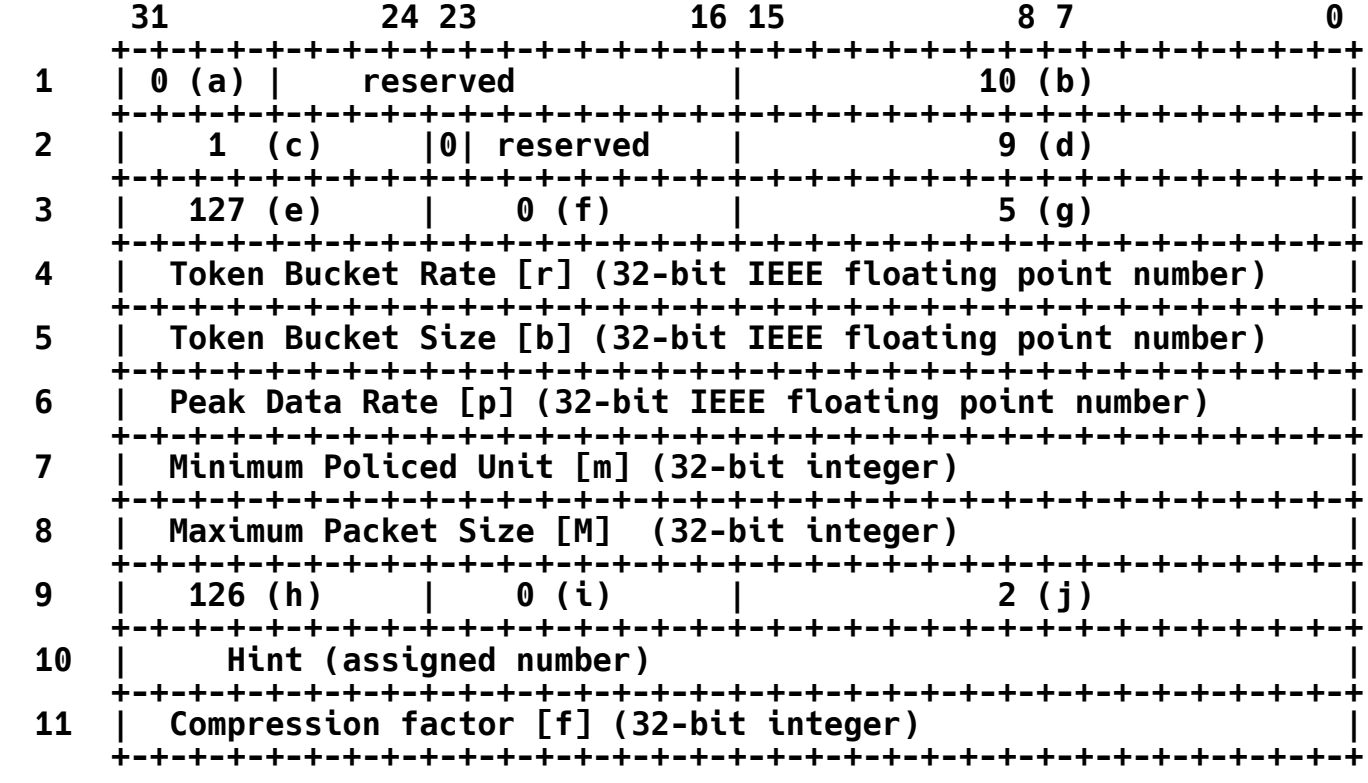
Note that RSVP does not generally require flows to be policed at every hop. To quote [RFC 2205]:

> Some QoS services may require traffic policing at some or all of (1) the edge of the network, (2) a merging point for data from multiple senders, and/or (3) a branch point where traffic flow from upstream may be greater than the downstream reservation being requested. RSVP knows where such points occur and must so indicate to the traffic control mechanism.

For the purposes of policing, a router which makes use of the compressibility hint in a sender TSpec should behave as if it is at the edge of the network, because it is in a position to receive traffic from a sender that, while it passed through policing at the real network edge, may still need to be policed if the amount of data sent exceeds the amount described by the compressed TSpec.

4. Object Format

The compressibility hint may be included in the sender TSpec using the encoding rules of Appendix A in [RFC 2210]. The complete sender TSpec is as follows:



(a) - Message format version number (0)
(b) - Overall length (10 words not including header)
(c) - Service header, service number 1 (default/global information)
(d) - Length of service 1 data, 9 words not including header
(e) - Parameter ID, parameter 127 (Token_Bucket_TSpec)
(f) - Parameter 127 flags (none set)
(g) - Parameter 127 length, 5 words not including header
(h) - Parameter ID, parameter 126 (Compression_Hint)
(i) - Parameter 126 flags (none set)
(j) - Parameter 126 length, 2 words not including header

The difference between this TSpec and the one described in [RFC 2210] is that the overall length contained in the first word is increased by 3, as is the length of the `service 1 data', and the original TSpec parameters are followed by a new parameter, the compressibility hint. This parameter contains the standard parameter header, and an

assigned number indicating the type of compression that is possible on this data. Different values of the hint would imply different compression algorithms may be applied to the data. Details of the numbering scheme for hints appear below.

Following the hint value is the compression factor f, expressed as a 32 bit integer representing the factor as a percentage value. The valid range for this factor is (0,100]. A sender that does not know what value to use here or wishes to leave the compression factor calculation to the routers' discretion may use the reserved value 0 to indicate this fact. Zero is reserved because it is not possible to compress a data stream to zero bits per second. The value 100 indicates that no compression is expected on this stream.

In some cases, additional quantitative information about the traffic may be required to enable a router to determine the amount of compression possible. In this case, a different encoding of the parameter would be required.

In some cases it may be desirable to include more than one hint in a Tspec (e.g., because more than one compression scheme could be applied to the data.) In this case, multiple instances of parameter 126 may appear in the Tspec and the overall length of the Tspec and the length of the Service 1 data would be increased accordingly.

Note that the Compression_Hint is, like the Token_Bucket_Tspec, not specific to a single service, and thus has a parameter value less than 128. It is also included as part of the default/global information (service number 1).

4.1. Hint Numbering

Hints are represented by a 32 bit field, with the high order 16 bits being the IP-compression-protocol number as defined in [RFC 1332] and [RFC 2509]. The low order 16 bits are a sub-option for the cases where the IP-compression-protocol number alone is not sufficient for int-serv purposes. The following hint values are required at the time of writing:

- hint = 0x002d0000: IP/TCP data that may be compressed according to [RFC 1144]
- hint = 0x00610000: IP data that may be compressed according to [RFC 2507]
- hint = 0x00610100: IP/UDP/RTP data that may be compressed according to [RFC 2508]

5. Backward Compatibility

It is desirable that an intserv router which receives this new TSpec format and does not understand the compressibility hint should silently ignore the hint rather than rejecting the entire TSpec (or the message containing it) as malformed. While [RFC 2210] clearly specifies the format of TSpecs in a way that they can be parsed even when they contain unknown parameters, it does not specify what action should be taken when unknown objects are received. Thus it is quite possible that some RSVP implementations will discard PATH messages containing a TSpec with the compressibility hint. In such a case, the router should send a PathErr message to the sending host. The message should indicate a malformed TSpec (Error code 21, Sub-code 04). The host may conclude that the hint caused the problem and send a new PATH without the hint.

For the purposes of this specification, it would be preferable if unknown TSpec parameters could be silently ignored. In the case where a parameter is silently ignored, the node should behave as if that parameter were not present, but leave the unknown parameter intact in the object that it forwards. This should be the default for unknown parameters of the type described in [RFC 2210].

It is possible that some future modifications to [RFC 2210] will require unknown parameter types to cause an error response. This situation is analogous to RSVP's handling of unknown objects, which allows for three different response to an unknown object, based on the highest two bits of the Class-Num. One way to handle this would be to divide the parameter space further than already done in [RFC 2216]. For example, parameter numbers of the form x1xxxxxx could be silently ignored if unrecognized, while parameter numbers of the form x0xxxxxx could cause an error response if unrecognized. (The meaning of the highest order bit is already fixed by [RFC 2216].) A third possibility exists, which is to remove the unrecognized parameter before forwarding, but this does not seem to be useful.

6. Security Considerations

The extensions defined in this document pose essentially the same security risks as those of [RFC 2210]. The risk that a sender will falsely declare his data to be compressible is equivalent to the sender providing an insufficiently large TSpec and is dealt with in the same way.

7. IANA Considerations

This specification relies on IANA-assigned numbers for the compression scheme hint. Where possible the existing numbering scheme for compression algorithm identification in PPP has been used, but it may in the future be necessary for IANA to assign hint numbers purely for the purposes of int-serv.

8. Acknowledgments

Carsten Borman and Mike DiBiasio provided much helpful feedback on this document.

9. References

[RFC 1144] Jacobson, V., "Compressing TCP/IP Headers for Low-Speed Serial Links", RFC 1144, February 1990.

[RFC 1332] McGregor, G., "The PPP Internet Protocol Control Protocol (IPCP)", RFC 1332, May 1992.

[RFC 2205] Braden, R., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

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

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

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

[RFC 2216] Shenker, S. and J. Wroclawski, "Network Element Service Specification Template", RFC 2216, September 1997.

[RFC 2507] Degermark, M., Nordgren, B. and S. Pink,"Header Compression for IP", RFC 2507, February 1999.

[RFC 2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[RFC 2509] Engan, M., Casner, S. and C. Bormann, "IP Header Compression over PPP", RFC 2509, February 1999.

10. Authors' Addresses

Bruce Davie
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

EMail: bsd@cisco.com

Carol Iturralde
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

EMail: cei@cisco.com

Dave Oran
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134

EMail: oran@cisco.com

Stephen L. Casner
Packet Design
66 Willow Place
Menlo Park, CA 94025

EMail: casner@acm.org

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

EMail: jtw@lcs.mit.edu

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.