Network Working Group
Request for Comments: 2997
Category: Standards Track

Y. Bernet
Microsoft
A. Smith
Allegro Networks
B. Davie
Cisco Systems

November 2000

Specification of the Null Service Type

Status of this Memo

Copyright Notice

Abstract

In the typical Resource Reservation Protocol (RSVP)/Intserv model, applications request a specific Intserv service type and quantify the resources required for that service. For certain applications, the determination of service parameters is best left to the discretion of the network administrator. For example, ERP applications are often mission critical and require some form of prioritized service, but cannot readily specify their resource requirements. To serve such applications, we introduce the notion of the 'Null Service'. The Null Service allows applications to identify themselves to network Quality of Service (QoS) policy agents, using RSVP signaling. However, it does not require them to specify resource requirements. QoS policy agents in the network respond by applying QoS policies appropriate for the application (as determined by the network administrator). This mode of RSVP usage is particularly applicable to networks that combine differentiated service (diffserv) QoS mechanisms with RSVP signaling [intdiff]. In this environment, QoS policy agents may direct the signaled application's traffic to a particular diffserv class of service.

1. Motivation

Using standard RSVP/Intserv signaling, applications running on hosts issue requests for network resources by communicating the following information to network devices:

1. A requested service level (Guaranteed or Controlled Load).
2. The quantity of resources required at that service level.
3. Classification information by which the network can recognize specific traffic (filterspec).
4. Policy/identity information indicating the user and/or the application for which resources are required.

In response, standard RSVP aware network nodes choose to admit or deny a resource request. The decision is based on the availability of resources along the relevant path and on policies. Policies define the resources that may be granted to specific users and/or applications. When a resource request is admitted, network nodes install classifiers that are used to recognize the admitted traffic and policers that are used to assure that the traffic remains within the limits of the resources requested.

The Guaranteed and Controlled Load Intserv services are not suitable for certain applications that are unable to (or choose not to)specify the resources they require from the network. Diffserv services are better suited for this type of application. Nodes in a diffserv network are typically provisioned to classify arriving packets to some small number of behavior aggregates (BAs) [diffarch]. Traffic is handled on a per-BA basis. This provisioning tends to be 'top-down' with respect to end-user traffic flows in the sense that there is no signaling between hosts and the network. Instead, the network administrator uses a combination of heuristics, measurement and experience to provision the network devices to handle aggregated traffic, with no deterministic knowledge of the volume of traffic that will arrive at any specific node.

In applying diffserv mechanisms to manage application traffic, network administrators are faced with two challenges:

1. Provisioning - network administrators need to anticipate the volume of traffic likely to arrive at each network node for each diffserv BA. If the volume of traffic arriving is likely to exceed the capacity available for the BA claimed, the network administrator has the choice of increasing the capacity for the BA, reducing the volume of traffic claiming the BA, or compromising service to all traffic arriving for the BA.

2. Classification - diffserv nodes classify traffic to user and/or application, based on the diff-serv codepoint (DSCP) in each packet's IP header or based on other fields in the packet's IP header (source/destination address/port and protocol). The latter method of classification is referred to as MF classification. This method of classification may be unreliable and imposes a management burden.

By using RSVP signaling, the management of application traffic in diffserv networks can be significantly facilitated. (Note that RSVP/diffserv interoperability has been discussed previously in the context of the Guaranteed and Controlled Load Intserv services.) This document focuses on RSVP/diffserv interoperability in the context of the Null Service.

2. Operational Overview

In the proposed mechanism, the RSVP sender offers the new service type, 'Null Service Type' in the ADSPEC that is included with the PATH message. A new Tspec corresponding to the new service type is added to the SENDER_TSPEC. In addition, the RSVP sender will typically include with the PATH message policy objects identifying the user, application and sub application ID [identity, application].

(Note that at this time, the new Tspec is defined only to carry the maximum packet size parameter (M), for the purpose of avoiding fragmentation. No other parameters are defined.)

Network nodes receiving these PATH messages interpret the service type to indicate that the application is requesting no specific service type or quantifiable resources. Instead, network nodes manage the traffic flow based on the requesting user, the requesting application and the type of application sub-flow.

This mechanism offers significant advantages over a pure diffserv network. At the very least, it informs each network node which would be affected by the traffic flow (and which is interested in intercepting the signaling) of:

1. The demand for resources in terms of number of flows of a particular type traversing the node.
2. The binding between classification information and user, application and sub-application.

This information is particularly useful to policy enforcement points and policy decision points (PEPs and PDPs). The network administrator can configure these elements of the policy management system to apply appropriate policy based on the identity of the user, the application and the specific sub application ID.

PEPs and PDPs may be configured to return an RSVP RESV message to the sender. The returned RESV message may include a DCLASS object [dclass]. The DCLASS object instructs the sender to mark packets on the corresponding flow with a specific DSCP (or set of DSCPs). This mechanism allows PEP/PDPs to affect the volume of traffic arriving at a node for any given BA. It enables the PEP/PDP to do so based on sophisticated policies.

3.1 Operational Notes

3.1.1 Scalability Issues

In any network in which per-flow signaling is used, it is wise to consider scalability concerns. The Null Service encourages signaling for a broader set of applications than that which would otherwise be signaled for. However, RSVP signaling does not, in general, generate a significant amount of traffic relative to the actual data traffic associated with the session. In addition, the Null Service does not encourage every application to signal. It should be used by applications that are considered mission critical or needing QoS management by network administrators.

Perhaps of more concern is the impact on processing resources at network nodes that process the signaling messages. When considering this issue, it's important to point out that it is not necessary to process the signaling messages at each network node. In fact, the combination of RSVP signaling with diff-serv networks may afford significant benefits even when the RSVP messages are processed only at certain key nodes (such as boundaries between network domains, first-hop routers, PEPs or any subset of these). Individual nodes should be enabled or disabled for RSVP processing at the discretion of the network administrator. See [intdiff] for a discussion of the impact of RSVP signaling on diff-serv networks.

In any case, per-flow state is not necessarily required, even in nodes that apply per-flow processing.

2.1.2 Policy Enforcement in Legacy Networks

Network nodes that adhere to the RSVP spec should transparently pass signaling messages for the Null Service. As such, it is possible to introduce a small number of PEPs that are enabled for Null Service into a legacy network and to realize the benefits described in this document.

2.1.3 Combining Existing Intserv Services with the Null Service

This document does not preclude applications from offering both a quantitative Intserv service (Guaranteed or Controlled Load)and the Null Service, at the same time. An example of such an application would be a telephony application that benefits from the Guaranteed Service but is able to adapt to a less strict service. By advertising its support for both, the application enables network policy to decide which service type to provide.

3. Signaling Details

3.1 ADSPEC Generation

The RSVP sender constructs an initial RSVP ADSPEC object specifying the Null Service Type. Since there are no service specific parameters associated with this service type, the associated ADSPEC fragment is empty and contains only the header word. Network nodes may or may not supply valid values for bandwidth and latency general parameters. As such, they may use the unknown values defined in [RFC2216].

The ADSPEC is added to the RSVP PATH message created at the sender.

3.2 RSVP SENDER_TSPEC Object

An additional Tspec is defined to correspond to the Null Service. If only the Null Service is offered in the ADSPEC, then this is the only Tspec included in the SENDER_TSPEC object. If guaranteed or controlled load services are also offered in the ADSPEC, then the new Tspec is appended following the standard Intserv token-bucket Tspec [RFC2210].

3.3 RSVP FLOWSPEC Object

Receivers may respond to PATH messages by generating an RSVP RESV message including a FLOWSPEC object. The FLOWSPEC object should specify that it is requesting the Null Service. It is possible that, in the future, a specific Rspec may be defined to correspond to the new service type.

4. Detailed Message Formats

4.1 Standard ADSPEC Format

A standard RSVP ADSPEC object is described in [RFC2210]. It includes a message header and a default general parameters fragment. Following the default general parameters fragment are fragments for each supported service type.

4.2 ADSPEC for Null Service Type

The Null Service ADSPEC includes the message header and the default general parameters fragment, followed by a single fragment denoting the Null Service. The new fragment introduced for the Null Service is formatted as follows:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    6 (a)      |x| Reserved    |           0 (b)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

a - indicates Null Service (6).
x - is the break-bit.
b - indicates zero words in addition to the header.

A complete ADSPEC supporting only the Null Service is illustrated below:

```
      31            24 23           16 15            8 7
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    1 | 0 (a) |    Reserved           |  Msg length -1 (b)           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    2 | 1 (c)         |x| Reserved    |           8 (d)               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    3 |    4 (e)      |    (f)        |           1 (g)               |
    + +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    4 |                    IS hop cnt (32-bit unsigned)                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    5 |    6 (h)      |    (i)        |           1 (j)               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    6 |   Path b/w estimate  (32-bit IEEE floating point number)      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    7 |    8 (k)      |    (l)        |           1 (m)               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    8 |        Minimum path latency (32-bit integer)                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    9 |   10 (n)      |    (o)        |           1 (p)               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   10 |        Composed MTU (32-bit unsigned)                         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   11 |    6 (q)     |x| Reserved    |           0 (r)               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Word 1: Message Header:
(a) - Message header and version number
(b) - Message length (10 words not including header)

Words 2-10: Default general characterization parameters
(c) - Per-Service header, service number 1 (Default General Parameters)
(x) - Global Break bit (NON_IS_HOP general parameter 2)
(d) - Length of General Parameters data block (8 words)
(e) - Parameter ID, parameter 4 (NUMBER_OF_IS_HOPS general parameter)
(f) - Parameter 4 flag byte
(g) - Parameter 4 length, 1 word not including header
(h) - Parameter ID, parameter 6 (AVAILABLE_PATH_BANDWIDTH general parameter)
(i) - Parameter 6 flag byte
(j) - Parameter 6 length, 1 word not including header
(k) - Parameter ID, parameter 8 (MINIMUM_PATH_LATENCY general parameter)
(l) - Parameter 8 flag byte

(m) - Parameter 8 length, 1 word not including header
(n) - Parameter ID, parameter 10 (PATH_MTU general parameter)
(o) - Parameter 10 flag byte
(p) - Parameter 10 length, 1 word not including header

Word 11: Null Service parameters

(q) - Per-Service header, service number 6 (Null)
(x) - Break bit for Null Service
(r) - Length (0) of per-service data not including header word.

Note that the standard rules for parsing ADSPEC service fragments ensure that the ADSPEC will not be rejected by legacy network elements. Specifically, these rules state that a network element encountering a per-service data header which it does not understand should set bit 23 (the break-bit) to indicate that the service is not supported and should use the length field from the header to skip over the rest of the fragment.

Also note that it is likely that it will not be possible for hosts or network nodes to generate meaningful values for words 5 and/or 7 (bandwidth estimates and path latency), due to the nature of the service. In this case, the unknown values from [RFC2216] should be used.

4.3 SENDER_TSPEC Object Format

The following Tspec is defined to correspond to the Null Service:

```
     31            24 23           16 15            8 7             0
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   1 |   6 (a)       |0|  Reserved    |             2 (b)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   2 | 128 (c)       |    0 (d)      |             1 (e)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   3 | Maximum Packet Size [M] (32-bit integer)                      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Word 1: Service header
(a) - Service number 6 (Null Service)
(b) - Length of per-service data, 2 words not including per-service header

Word 2-3: Parameter
(c) - Parameter ID, parameter 128 (Null Service TSpec)
(d) - Parameter 128 flags (none set)
(e) - Parameter 128 length, 1 words not including parameter header

Note that the illustration above does not include the standard RSVP SENDER_TSPEC object header, nor does it include the sub-object header (which indicates the message format version number and length), defined in RFC 2205 and RFC 2210, respectively.

In the case that only the Null Service is advertised in the ADSPEC, the Tspec above would be appended immediately after the SENDER_TSPEC object header and sub-object header. In the case that additional service types are advertised, requiring the token bucket specific Tspec defined in RFC2210, the Tspec above would be appended following the token bucket Tspec, which would in turn follow the object header and sub-object header.

4.4 FLOWSPEC Object Format

The format of an RSVP FLOWSPEC object originating at a receiver requesting the Null Service is shown below. The value of 6 in the per-service header (field (c), below) indicates that Null Service is being requested.

```
     31            24 23           16 15            8 7
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 1 | 0 (a)         |    reserved   |         3 (b)                 |
 + +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 2 |   6 (c)       |0|  Reserved   |             2 (d)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 3 | 128 (e)       |    0 (f)      |             1 (g)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 4 | Maximum Packet Size [M] (32-bit integer)                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

(a) - Message format version number (0)
(b) - Overall length (3 words not including header)
(c) - Service header, service number 6 (Null)
(d) - Length of data, 2 words not including per-service header
(e) - Parameter ID, parameter 128 (Null Service TSpec)
(f) - Parameter 128 flags (none set)
(g) - Parameter 128 length, 1 words not including parameter header

4.5 DCLASS Object Format

DCLASS objects may be included in RESV messages. For details regarding the DCLASS object format, see [dclass].

5. Security Considerations

The message formatting and usage rules described in this note raise no new security issues beyond standard RSVP.

6. References

[RFC2205] Braden, R., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.

[RFC2216] Shenker, S. and J. Wroclawski, "Network Element QoS Control Service Specification Template", RFC 2216, September 1997.

[RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997.

[intdiff] Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Nichols, K., Speer, M., Braden, B. and B. Davie, "A Framework for Integrated Services Operation over Diffserv Networks", RFC 2998, November 2000.

[diffarch] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[identity] Yadav, S., Yavatkar, R., Pabbati, R., Ford, P., Moore, T., Herzog, S., "Identity Representation for RSVP", RFC 2752, January 2000.

[application] Bernet, Y., "Application and Sub Application Identity Policy Objects for Use with RSVP", RFC 2872, June 2000.

[dclass] Bernet, Y., "Format of the RSVP DCLASS Object", RFC 2996, November 2000.

7. Acknowledgments

We thank Fred Baker, Dinesh Dutt, Nitsan Elfassy, John Schnizlein, Ramesh Pabbati and Sanjay Kaniyar for their comments on this memo.

8. Authors' Addresses

Yoram Bernet
Microsoft
One Microsoft Way
Redmond, WA 98052

Phone: +1 (425) 936-9568
EMail: Yoramb@microsoft.com

Andrew Smith
Allegro Networks
6399 San Ignacio Ave.
San Jose, CA 95119, USA

FAX: +1 415 345 1827
Email: andrew@allegronetworks.com

Bruce Davie
Cisco Systems
250 Apollo Drive
Chelmsford, MA 01824

Phone: +1 (978)-244-8000
EMail: bsd@cisco.com

9. Full Copyright Statement

Acknowledgement

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