Network Working Group
Request for Comments: 2751
Category: Standards Track

S. Herzog
IPHighway
January 2000

Signaled Preemption Priority Policy Element

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document describes a preemption priority policy element for use by signaled policy based admission protocols (such as [RSVP] and [COPS]).

Preemption priority defines a relative importance (rank) within the set of flows competing to be admitted into the network. Rather than admitting flows by order of arrival (First Come First Admitted) network nodes may consider priorities to preempt some previously admitted low priority flows in order to make room for a newer, high-priority flow.

Table of Contents

1 Introduction

Traditional Capacity based Admission Control (CAC) indiscriminately admits new flows until capacity is exhausted (First Come First Admitted). Policy based Admission Control (PAC) on the other hand attempts to minimize the significance of order of arrival and use policy based admission criteria instead.

One of the more popular policy criteria is the rank of importance of a flow relative to the others competing for admission into a network node. Preemption Priority takes effect only when a set of flows attempting admission through a node represents overbooking of resources such that based on CAC some would have to be rejected. Preemption priority criteria help the node select the most important flows (highest priority) for admission, while rejecting the low priority ones.

Network nodes which support preemption should consider priorities to preempt some previously admitted low-priority flows in order to make room for a newer, high-priority flow.

This document describes the format and applicability of the preemption priority represented as a policy element in [RSVP-EXT].

2 Scope and Applicability

The Framework document for policy-based admission control [RAP] describes the various components that participate in policy decision making (i.e., PDP, PEP and LDP). The emphasis of PREEMPTION_PRI elements is to be simple, stateless, and light-weight such that they could be implemented internally within a node’s LDP (Local Decision Point).

Certain base assumptions are made in the usage model for PREEMPTION_PRI elements:

- They are created by PDPs

  In a model where PDPs control PEPs at the periphery of the policy domain (e.g., in border routers), PDPs reduce sets of relevant policy rules into a single priority criterion. This priority as expressed in the PREEMPTION_PRI element can then be communicated to downstream PEPs of the same policy domain, which have LDPs but no controlling PDP.

- They can be processed by LDPs

  PREEMPTION_PRI elements are processed by LDPs of nodes that do not have a controlling PDP. LDPs may interpret these objects, forward them as is, or perform local merging to forward an equivalent merged PREEMPTION_PRI policy element. LDPs must follow the merging strategy that was encoded by PDPs in the PREEMPTION_PRI objects. (Clearly, a PDP, being a superset of LDP, may act as an LDP as well).

- They are enforced by PEPs

  PREEMPTION_PRI elements interact with a node’s traffic control module (and capacity admission control) to enforce priorities, and preempt previously admitted flows when the need arises.

3 Stateless Policy

Signaled Preemption Priority is stateless (does not require past history or external information to be interpreted). Therefore, when carried in COPS messages for the outsourcing of policy decisions, these objects are included as COPS Stateless Policy Data Decision objects (see [COSP, COPS-RSVP]).

4 Policy Element Format

The format of Policy Data objects is defined in [RSVP-EXT]. A single Policy Data object may contain one or more policy elements, each representing a different (and perhaps orthogonal) policy.

The format of preemption priority policy element is as follows:

```
+-------------+-------------+-------------+-------------+
| Length (12)               | P-Type = PREEMPTION_PRI   |
+------+------+-------------+-------------+-------------+
| Flags       | M. Strategy | Error Code  | Reserved(0) |
+------+------+-------------+-------------+-------------+
| Preemption Priority       | Defending Priority        |
+------+------+-------------+-------------+-------------+
```

Length: 16 bits
Always 12. The overall length of the policy element, in bytes.

P-Type: 16 bits
PREEMPTION_PRI = 3

This value is registered with IANA, see Section 7.

Flags: 8 bits
Reserved (always 0).

Merge Strategy: 8 bit

1	Take priority of highest QoS: recommended
2	Take highest priority: aggressive
3	Force Error on heterogeneous merge

Reserved: 8 bits
Error code: 8 bits

0	NO_ERROR	Value used for regular PREEMPTION_PRI elements
1	PREEMPTION	This previously admitted flow was preempted
2	HETEROGENEOUS	This element encountered heterogeneous merge

Reserved: 8 bits
Always 0.

Preemption Priority: 16 bit (unsigned)
The priority of the new flow compared with the defending priority of previously admitted flows. Higher values represent higher Priority.

Defending Priority: 16 bits (unsigned)
Once a flow was admitted, the preemption priority becomes irrelevant. Instead, its defending priority is used to compare with the preemption priority of new flows.

For any specific flow, its preemption priority must always be less than or equal to the defending priority. A wide gap between preemption and defending priority provides added stability: moderate preemption priority makes it harder for a flow to preempt others, but once it succeeded, the higher defending priority makes it easier for the flow to avoid preemption itself. This provides a mechanism for balancing between order dependency and priority.

5 Priority Merging Issues

Consider the case where two RSVP reservations merge:

```
       F1: QoS=High,  Priority=Low
       F2: QoS=Low,   Priority=High

F1+F2= F3: QoS=High,  Priority=???
```

The merged reservation F3 should have QoS=Hi, but what Priority should it assume? Several negative side-effects have been identified that may affect such a merger:

Free-Riders:

If F3 assumes Priority=High, then F1 got a free ride, assuming high priority that was only intended to the low QoS F2. If one associates costs as a function of QoS and priority, F1 receives an "expensive" priority without having to "pay" for it.

Denial of Service:

If F3 assumes Priority=Low, the merged flow could be preempted or fail even though F2 presented high priority.

Denial of service is virtually the inverse of the free-rider problem. When flows compete for resources, if one flow receives undeserving high priority it may be able to preempt another deserving flow (hence one free-rider turns out to be another’s denial of service).

Instability:

The combination of preemption priority, killer reservation and blockade state [RSVP] may increase the instability of admitted flows where a reservation may be preempted, reinstated, and preempted again periodically.

5.1 Priority Merging Strategies

In merging situations LDPs may receive multiple preemption elements and must compute the priority of the merged flow according to the following rules:

a. Preemption priority and defending priority are merged and computed separately, irrespective of each other.

b. Participating priority elements are selected.

   All priority elements are examined according to their merging strategy to decide whether they should participate in the merged result (as specified bellow).

c. The highest priority of all participating priority elements is computed.

The remainder of this section describes the different merging strategies the can be specified in the PREEMPTION_PRI element.

5.1.1 Take priority of highest QoS

The PREEMPTION_PRI element would participate in the merged reservation only if it belongs to a flow that contributed to the merged QoS level (i.e., that its QoS requirement does not constitute a subset another reservation.) A simple way to determine whether a flow contributed to the merged QoS result is to compute the merged QoS with and without it and to compare the results (although this is clearly not the most efficient method).

The reasoning for this approach is that the highest QoS flow is the one dominating the merged reservation and as such its priority should dominate it as well. This approach is the most amiable to the prevention of priority distortions such as free-riders and denial of service.

This is a recommended merging strategy.

5.1.2 Take highest priority

All PREEMPTION_PRI elements participate in the merged reservation.

This strategy disassociates priority and QoS level, and therefore is highly subject to free-riders and its inverse image, denial of service.

This is not a recommended method, but may be simpler to implement.

5.1.3 Force error on heterogeneous merge

A PREEMPTION_PRI element may participate in a merged reservation only if all other flows in the merged reservation have the same QoS level (homogeneous flows).

The reasoning for this approach assumes that the heterogeneous case is relatively rare and too complicated to deal with, thus it better be prohibited.

This strategy lends itself to denial of service, when a single receiver specifying a non-compatible QoS level may cause denial of service for all other receivers of the merged reservation.

Note: The determination of heterogeneous flows applies to QoS level only (FLOWSPEC values), and is a matter for local (LDP) definition. Other types of heterogeneous reservations (e.g. conflicting reservation styles) are handled by RSVP and are unrelated to this PREEMPTION_PRI element.

This is a recommended merging strategy when reservation homogeneity is coordinated and enforced for the entire multicast tree. It is more restrictive than Section 5.1.1, but is easier to implement.

5.2 Modifying Priority Elements

When POLICY_DATA objects are protected by integrity, LDPs should not attempt to modify them. They must be forwarded as-is or else their security envelope would be invalidated. In other cases, LDPs may modify and merge incoming PREEMPTION_PRI elements to reduce their size and number according to the following rule:

Merging is performed for each merging strategy separately.

There is no known algorithm to merge PREEMPTION_PRI element of different merging strategies without loosing valuable information that may affect OTHER nodes.

- For each merging strategy, the highest QoS of all participating PREEMPTION_PRI elements is taken and is placed in an outgoing PREEMPTION_PRI element of this merging strategy.

- This approach effectively compresses the number of forwarded PREEMPTION_PRI elements to at most to the number of different merging strategies, regardless of the number of receivers (See the example in Appendix A.2).

6 Error Processing

A PREEMPTION_PRI error object is sent back toward the appropriate receivers when an error involving PREEMPTION_PRI elements occur.

PREEMPTION

When a previously admitted flow is preempted, a copy of the preempting flow’s PREEMPTION_PRI element is sent back toward the PDP that originated the preempted PREEMPTION_PRI object. This PDP, having information on both the preempting and the preempted priorities may construct a higher priority PREEMPTION_PRI element in an effort to re-instate the preempted flow.

Heterogeneity

When a flow F1 with Heterogeneous Error merging strategy set in its PREEMPTION_PRI element encounters heterogeneity the PREEMPTION_PRI element is sent back toward receivers with the Heterogeneity error code set.

7 IANA Considerations

Following the policies outlined in [IANA-CONSIDERATIONS], Standard RSVP Policy Elements (P-type values) are assigned by IETF Consensus action as described in [RSVP-EXT].

P-Type PREEMPTION_PRI is assigned the value 3.

8 Security Considerations

The integrity of PREEMPTION_PRI is guaranteed, as any other policy element, by the encapsulation into a Policy Data object [RSVP-EXT].

Further security mechanisms are not warranted, especially considering that preemption priority aims to provide simple and quick guidance to routers within a trusted zone or at least a single zone (no zone boundaries are crossed).

9 References

[RSVP-EXT] Herzog, S., "RSVP Extensions for Policy Control", RFC 2750, January 2000.

[COPS-RSVP] Boyle, J., Cohen, R., Durham, D., Herzog, S., Raja, R. and A. Sastry, "COPS usage for RSVP", RFC 2749, January 2000.

[RAP] Yavatkar, R., et al., "A Framework for Policy Based Admission Control", RFC 2753, January 2000.

[COPS] Boyle, J., Cohen, R., Durham, D., Herzog, S., Raja, R. and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[RSVP] Braden, R., ed., et al., "Resource ReSerVation Protocol (RSVP) - Functional Specification", RFC 2205, September 1997.

[IANA-CONSIDERATIONS] Alvestrand, H. and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

10 Author Information

Shai Herzog
IPHighway, Inc.
55 New York Avenue
Framingham, MA 01701

Phone: (508) 620-1141
EMail: herzog@iphighway.com

Appendix A: Example

The following examples describe the computation of merged priority elements as well as the translation (compression) of PREEMPTION_PRI elements.

A.1 Computing Merged Priority

```
                             r1
                            /   QoS=Hi (Pr=3, St=Highest QoS)
                           /
         s1-----A---------B--------r2  QoS=Low (Pr=4, St=Highest PP)
                 \        \
                  \        \   QoS=Low  (Pr=7, St=Highest QoS)
                   r4        r3

           QoS=Low (Pr=9, St=Error)
```

Example 1: Merging preemption priority elements

Example one describes a multicast scenario with one sender and four receivers each with each own PREEMPTION_PRI element definition.

r1, r2 and r3 merge in B. The resulting priority is 4.

Reason: The PREEMPTION_PRI of r3 doesn’t participate (since r3 is not contributing to the merged QoS) and the priority is the highest of the PREEMPTION_PRI from r1 and r2.

r1, r2, r3 and r4 merge in A. The resulting priority is again 4: r4 doesn’t participate because its own QoS=Low is incompatible with the other (r1) QoS=High. An error PREEMPTION_PRI should be sent back to r4 telling it that its PREEMPTION_PRI element encountered heterogeneity.

A.2 Translation (Compression) of Priority Elements

Given this set of participating PREEMPTION_PRI elements, the following compression can take place at the merging node:

```
From:
          (Pr=3, St=Highest QoS)
          (Pr=7, St=Highest QoS)
          (Pr=4, St=Highest PP)
          (Pr=9, St=Highest PP)
          (Pr=6, St=Highest PP)
To:
          (Pr=7, St=Highest QoS)
          (Pr=9, St=Highest PP)
```

Full Copyright Statement

Acknowledgement

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