

Internet Draft
Expiration: Apr. 1999
File: [draft-ietf-rap-signaled-priority-00.txt](#)

Shai Herzog
IPHighway

Preemption Priority Policy Element

November 18, 1998

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress".

To learn the current status of any Internet-Draft, please check the `l1d-abstracts.txt` listing contained in the Internet-Drafts Shadow Directories on `ftp.ietf.org`, `nic.nordu.net`, `ftp.isi.edu`, or `munni.oz.au`.

A revised version of this draft document will be submitted to the RFC editor as a Proposed Standard for the Internet Community. Discussion and suggestions for improvement are requested. This document will expire at the expiration date listed above. Distribution of this draft is unlimited.

Abstract

This document describes a preemption priority policy element for use by 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

Abstract.....	1
Table of Contents.....	2
1.Introduction.....	3
2.Scope and Applicability.....	3
3.Policy Element Format:.....	4
4.Priority Merging Issues.....	5
5.Priority Merging Strategies.....	6
5.1.Take priority of highest QoS.....	6
5.2.Take highest priority.....	7
5.3.Force error on heterogeneous merge.....	7
6.Error Processing.....	7
7.Security Considerations.....	8
8.Example.....	8
8.1.Computing Merged Priority.....	8
8.2.Translation (Compression) of Priority Elements.....	9
9.References.....	9
10 . Author Information.....	9

1. Introduction

Traditional Capacity based Admission Control (CAC) indiscriminately flows until capacity is exhausted. Policy based Admission Control (PAC) on the other hand attempts to minimize the significance of order of arrival and use policy based 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/through 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 describe 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 [[Fwk](#)] describes the various components that participate in policy decision making (i.e., PDP, PEP and LPD). The emphasis of PREEMPTION_PRI elements is to be simple and get processed quick enough 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 the periphery (boarder routers), PDPs reduce their entire set of relevant policy rules into a single priority criteria. This priority as expressed in the PREEMPTION_PRI element can then be communicated to in-cloud core nodes, which have LPDs but no controlling PDP.

- They are processed by LDPs

PREEMPTION_PRI elements are interpreted and are forwarded locally within in-cloud core nodes (in their LDP modules). Policy ignorant nodes (PINs) may interpret these objects, and forward them as is, or

they can perform local merging and forward an equivalent merged

Herzog et. al.

Expires June 1998

[Page 3]

PREEMPTION_PRI policy element. In both cases, whether for interpretation only or for merging, LDPs must follow the merging strategy as specified in the policy element itself.

- They are enforced by PEPs

PREEMPTION_PRI elements interact with a node's traffic control module (and capacity admission control) to enforce priorities, and to perform preemption of previously admitted flows if the need arises.

3. 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 Preemption Priority policy element, as registered with IANA.

Flags: 8 bits

Reserved (always 0).

Merge Strategy: 8 bit

- 1 Take priority of highest QoS: recommended
- 2 Take highest priority: aggressive
- 3 Error (fail) on heterogeneous merge

Reserved: 8 bits

Error code: 8 bits

- | | | |
|---|---------------|---|
| 0 | No error. | Value used for all regular PP elements. |
| 1 | Preemption | This previously admitted flow was preempted |
| 2 | Heterogeneous | This PP element encountered heterogeneous merge |

Reserved: 8 bits

Always 0.

Preemption Priority: 16 bit ($0..2^{16}$)

The priority of the new flow compared with the defending priority of previously admitted flows. Higher values represent higher Priority.

Defending Priority: 16 bits

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 or equal to the defending priority. A wide gap between preemption and defending priority provides leeway for 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 escape preemption itself. This mechanism provides some order dependency, although it is not absolute, as is the case for CAC.

4. Priority Merging Issues

Consider the case where two reservations merge:

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

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

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

Free-Riders:

If F3 assumes Priority=High, the result is that F1 managed to get a free ride for his high QoS, 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:

Herzog et. al.

Expires June 1998

[Page 5]

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

Denial of service is also a mirror image of the free-rider problem; given competition for resources among the flows, if one flow receives undeserving high priority it should 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. Priority Merging Strategies

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

- a. Preemption priority and defending priority are computed separately, irrespective of each other.
- b. All priority elements are examined according to their merging strategy to decide whether they should participate in the merged result (as specified below).
- c. Take the highest priority of all those who participate.
- d. A node may wish to reduce the number of PP elements it forwards by translating multiple PP elements into their equivalent one. In this case, it can only combine participating PP elements of the same merging strategy into one (by taking the highest priority amongst them). This effectively compresses the number of forwarded PP elements at most to the number of different merging strategies, regardless of the number of receivers (See the example in [Section 8.2](#)).

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

5.1. Take priority of highest QoS

The PP 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 of 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 thinking behind 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.

5.2. Take highest priority

All PP elements participate in the merged reservation.

This strategy allows all receivers to participate and contribute to the preemption priority even if they don't contribute to the merged reservation. It is therefore highly subject to free-riders and its mirror image, denial of service.

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

5.3. Force error on heterogeneous merge

A PP element with this strategy cannot participate in a merged reservation if any other flow in the merged reservation has a QoS level that is substantially different from its own. The determination of what is substantially different QoS level is up to the local node.

The thinking behind this approach is that most cases encounter homogenous receivers that use similar QoS levels. Furthermore, it assumes the use of PREEMPTION_PRI in the heterogeneous case is too complicated to deal with and should not be permitted.

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

6. Error Processing

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

Preemption

When a previously admitted flow is preempted, a copy of the PREEMPTING flow's PP element is sent downstream to the receiver (with the preemption error set). This allows receivers (or PDPs) to construct a higher priority PP element that may cause the flow to be re-instated.

Heterogeneity

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

7. Security Considerations

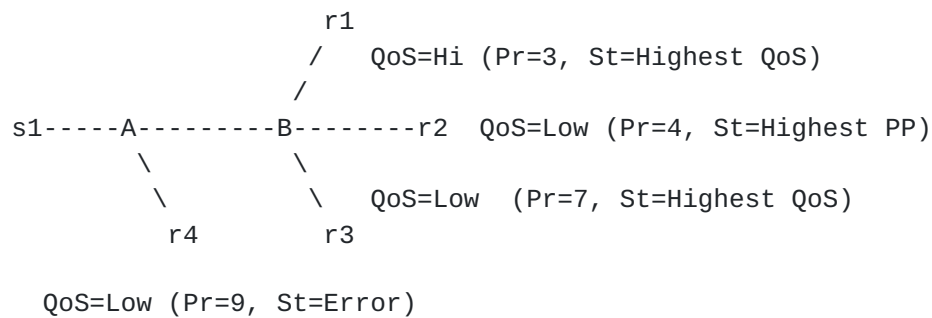
The integrity of PREEMPTION_PRI is guaranteed, as any other policy element, by the encapsulation into a Policy Data object [[RSVP-EXT](#)].

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

8. Example

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

8.1. Computing Merged Priority



Example 1: merging Preemption Priority elements

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

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

Reason: The PP of r3 doesn't participate (since r3 is not contributing to the merged QoS) and the priority is the highest of the PP 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 PP should be sent back to r4 telling it

that its PP element encountered heterogeneity.

Herzog et. al.

Expires June 1998

[Page 8]

8.2. Translation (Compression) of Priority Elements

Given this set of participating PP 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)

9. References

- [RSVP] Braden, R. ed., "Resource ReSerVation Protocol (RSVP) - Functional Specification." Internet-Draft, [draft-ietf-rsvp-spec-16.txt](#), June 1997.
- [COPS] Boyle, J., Cohen, R., Durham, D., Herzog, S., Raja, n R., Sastry, A., "The COPS (Common Open Policy Service) Protocol", Internet-Draft <[draft-ietf-rap-cops-02.txt](#)>, Aug. 1998.
- [RSVP-EXT] Herzog, S. "RSVP Extensions for Policy Control", Internet-Draft, [draft-ietf-rap-rsvp-ext-00.txt](#), Apr. 1998.
- [Fwk] R. Yavatkar, D. Pendarakis, R. Guerin. "A Framework for Policy Based Admission Control", Internet-Draft <[draft-ietf-rap-framework-00.txt](#)>, November, 1997.

10. Author Information

Shai Herzog, IPHighway
Parker Plaza, Suite 1500
400 Kelby St.
Fort-Lee, NJ 07024
(201) 585-0800
herzog@iphighway.com

