Network Working Group A. Niemi
Internet-Draft Nokia
Expires: December 29, 2003 June 30, 2003

Requirements for Limiting the Rate of Event Notifications
draft-niemi-sipping-event-throttle-reqs-02

Status of this Memo

Copyright Notice

Abstract

All event packages are required to specify a maximum rate at which event notifications are generated by a single notifier. Such a limit is provided in order to reduce network congestion. In addition to the fixed limits introduced by specific event packages, further mechanisms for limiting the rate of event notification are also allowed to be defined by event package specifications but none have been specified so far. This memo discusses the requirements for a throttle mechanism that allows a subscriber to further limit the rate of event notification.

Table of Contents

1. Introduction

The SIP events framework described in RFC 3265 [2] mandates that each event package specification defines an absolute maximum on the rate at which notifications are allowed to be generated by a single notifier. Such a limit is provided in order to reduce network congestion.

All of the existing event package specifications include a maximum notification rate recommendation, ranging from once in every five seconds [3], [4], [5] to once per second [6].

Per the SIP events framework, each event package specification is also allowed to define additional throttling mechanisms which allow the subscriber to further limit the rate of event notification. So far none of the event package specifications have defined such throttling mechanisms.

This memo discusses the requirements for a generic throttling mechanism, which allows the subscriber to limit the rate of event notifications. It is intended that the throttle mechanism is not event package specific, but commonly available to be used with all event subscriptions.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [1], and indicate requirement priorities.

3. Event Throttle Model

A throttle is defined as a protocol element that establishes a throttling policy at the notifier. This policy simply indicates the minimum time period allowed between two notifications. In practice, this throttling policy only extends the default policy of each event package, making it subscriber-configurable.

Using notations from traffic theory, we can model the notifier as a statistical multiplexer with an input rate of Ci (i = 1,...,n), and an output rate of C <= C1 + ... + Cn. Typically, the statistical multiplexer is lossy, with a finite buffer size. The loss probability of the statistical multiplexer can be decreased by enlarging this buffer. Figure 1 illustrates the model.


```
              C1 |\
       1 ---------|  \
         ---------|    \        C
         ---------| ||||O---------- 1
         ---------|    /
       n ---------|  /
              Cn |/
```

Figure 1: Notifier modeled as a statistical multiplexer

In event notification, there is typically only a single input connection, characterized by the event package, and consisting of a stream of event notification packets. Properties of the buffer, such as buffer size, policy (e.g., FIFO, LIFO), and packet treatment in lossy conditions, are all implementation and event package specific.

> A valid buffer model is a LIFO (Last In First Out) buffer with a size of one notification. Out of all buffered notifications, only the latest one is ever sent to the subscriber. Another equally valid buffer model might be one that has a near infinite buffer size. In that case, it is enough that the output rate C exceeds the aggregate average rate of all the inputs. Under lossy conditions, notifications might be dropped or their state merged, depending on the event package.

The main implication of this model for event throttles is that they are lossy. Either some state changes are lost, or some level of accuracy in notifications is lost. The former will affect state changes that occur more frequent than what the throttling policy allows; and the latter will affect notifications of "stateless" nature, e.g., accuracy of buffered location updates decreases.

4. Example Use Case

There are many applications that potentially would make use of a throttle mechanism. This chapter only illustrates one possible use case, in which a device uses the event throttling mechanism to limit the amount of traffic it may receive.

4.1 Pre-conditions

A presence application in Lisa's device contains a list of 100 presentities. In order to decrease the processing and network load of watching 100 presentities, Lisa's presence application has included an event throttle to each of the subscriptions, to limit the maximum rate at which notifications are to be generated to once per 20 seconds.

4.2 Normal Flow

- o Heikki is one of the presentities Lisa is wathcing. Heikki's presence agent conforms to the throttling policy requested by Lisa's presence application.

- o Heikki changes his location, which results in a presence notification to be sent to Lisa.

- o Heikki's location changes again, and now very fast. His presence agent receives outgoing presence notification much more frequently than what the throttling policy allows it to generate notifications out to Lisa. The notifications are buffered.

- o Lisa receives presence updates conforming to the set throttling policy.

- o Now Heikki's movements stabilize, and his location remains stable.

4.3 Post-conditions

The throttled subscriptions even out rapid changes in presence status. Lisa still receives all of the buffered presence notifications. Her understanding of Heikki's presence status is temporarily different from Heikki's current real-time status, but as the buffered notifications get exhausted, will eventually converge to the real-time status.

5. Requirements

REQ1: The subscriber MUST be able to set using a throttle mechanism the minimum time period between two notifications in a specific subscription.

REQ2: The subscriber MUST be able to indicate that it requires the notifier to comply with the suggested throttling policy in a specific subscription.

REQ3: The notifier MUST be able to indicate that it does not support the use of a throttle mechanism in the subscription.

REQ4: It MUST be possible to use the throttle mechanism in subscriptions to all events.

REQ5: It MUST be possible to use the throttle mechanism together with any event filtering mechanism.

REQ6: The notifier MUST be allowed to use a throttling policy in which the minimum time period between two notifications is longer than the one given by the subscriber.

For example, due to congestion reasons, local policy at the notifier could temporarily dictate a throttling policy that in effect increases the subscriber-configured minimum time period between two notifications.

REQ7: The throttle mechanism MUST provide a reasonable resolution for setting the minimum period between two notifications. At a minimum, the throttling mechanism MUST include discussion of the situation resulting from a minimum time period which exceeds the subscription duration, and SHOULD provide mechanisms for avoiding this situation."

REQ8: A throttle mechanism MUST allow for the application of authentication and integrity protection mechanisms to subscriptions invoking that mechanism.

Note that Section 6 contains further discussion on the security implications of the throttle mechanism.

6. Security Considerations

Naturally all of the security considerations for event subscriptions and notifications also apply to subscriptions and notifications that use the throttle mechanism. In addition, using the event throttle mechanism may introduce some new security issues to consider. However, there are no additional requirements regarding security at this stage.

7. Open Issues

This chapter lists the main open issues within this document.

- o Is the model comprehensive?
- o Is this work mature enough to be handed a WG work item status?

8. Changes to "draft-niemi-sipping-event-throttle-reqs-01"

Changes from the last version were:

- o Refined the model based on feedback.

- o Clarified language and terminology used in the requirements, based on feedback.

9. Changes to "draft-niemi-sipping-event-throttle-reqs-00"

Changes from the previous version include:

o Added the chapter describing the model for event throttles.

o Reworded the requirements to reflect the model discussion

o Added acknowledgements, changelog, and open issues sections

10. Acknowledgements

The author would like to thank Tim Moran, Jonathan Rosenberg, Hisham Khartabil, Juha Kalliokulju, Paul Kyzivat, Henning Schulzrinne and Dean Willis for their valuable comments.

Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Informative References

[2] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[3] Rosenberg, J., "A Presence Event Package for the Session Initiation Protocol (SIP)", draft-ietf-simple-presence-10 (work in progress), January 2003.

[4] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Registrations", draft-ietf-sipping-reg-event-00 (work in progress), October 2002.

[5] Rosenberg, J., "A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)", draft-ietf-simple-winfo-package-05 (work in progress), January 2003.

[6] Mahy, R., "A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)", draft-ietf-sipping-mwi-02 (work in progress), March 2003.

Author's Address

Aki Niemi
Nokia
P.O. Box 321
NOKIA GROUP, FIN 00045
Finland

Phone: +358 50 389 1644
EMail: aki.niemi@nokia.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Acknowledgement

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