Network Working Group
Internet-Draft
Expires: July 23, 2003

A. Niemi
Nokia
January 22, 2003

Requirements for Limiting the Rate of Event Notifications
draft-niemi-sipping-event-throttle-reqs-00

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].

3. 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 mobile device uses the event throttling mechanism to limit the amount of traffic it may receive.

A mobile application is watching the state of 100 presentities each generating notifications at a maximum rate of once per five seconds. Assuming that the arrival times of notifications are evenly distributed, this will result in a maximum notification frequency of:

$$f = 100 * (1 / 5s) = 100 / 5 \text{ Hz} = 20 \text{ Hz}$$

experienced by the mobile. The same watcher subscribing using a throttle mechanism to limit the maximum rate at which notifications are to be generated to once per 20 seconds can expect a maximum notification frequency of:

f = 100 * (1 / 20s) = 100 / 20 Hz = 5 Hz

thus resulting in 75% reduction in the maximum rate of incoming presence notifications.

Note that the actual rate of notification is the sum of many factors, and this example only makes a very broad assumption on the absolute maximum rate at which the notifications might be generated.

4. Requirements

REQ1: The subscriber MUST be able to limit using a throttle mechanism the maximum rate at which the notifier is allowed to generate notifications in a subscription.

REQ2: The subscriber MUST be able to indicate that it requires the use of a throttle mechanism in the subscription.

REQ3: The subscriber SHOULD be allowed to indicate support for the throttle mechanism without requiring it.

REQ4: The notifier MUST be able to indicate that it does not support the use of a throttle mechanism in the subscription.

REQ5: If the throttle mechanism isn't required by the subscriber, the notifier SHOULD be able to ignore it.

REQ6: It MUST be possible to use the throttle mechanism in subscriptions to all events.

REQ7: It MUST be possible to use the throttle mechanism together with any event filtering mechanism.

REQ8: The notifier MUST be allowed to use a maximum rate lower than the one given by the subscriber.

REQ9: Authentication and integrity protection SHOULD be applied to subscriptions that apply the throttle mechanism.

Note that Section 5 contains further discussion on the security implications of the throttle mechanism.

5. 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 introduces some new security issues to consider:

The throttle mechanism might allow a subscriber to set a very low maximum notification rate - one that possibly exceeds the subscription expiration. Such a limit inserted by a malicious third party would result in very few if any notifications to be generated, which could be perceived as theft of service to the subscriber.

Similarly, the throttle mechanism might allow the subscriber to set a very high maximum rate of notification that possibly is higher than the default recommended rate of notification. Such a high rate inserted by a malicious third party could result in denial of service of the notifier due to performance issues.

Using the throttle mechanism potentially allows a subscriber to increase the number of active subscriptions due to the decrease in the maximum rate of notifications generated by a single notifier. If a malicious third party is able to remove the throttle from the subscriptions, the subscriber might be flooded with notifications.

All of the above problems can be avoided by ensuring that the integrity and authenticity of subscriptions is protected by applying relevant security measures.

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-09 (work in progress), December 2002.

[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-04 (work in progress), December

2002.

[6] Mahy, R., "A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)", draft-ietf-sipping-mwi-01 (work in progress), November 2002.

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

Full Copyright Statement

Acknowledgement

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