

simple
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2008

B. Rosen
NeuStar
S. Loreto
Ericsson
K. Kiss
Nokia

February 18, 2008

Optimizing Notifications for Presence Network Agents
draft-rosen-simple-watcher-count-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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 and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 21, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

In large presence systems deployed in multiservice networks, presence information is often known by the network in addition to, or instead of the presentity's devices (endpoints). Examples of such information include location and availability for various kinds of session establishment. Even if devices know the information, the

Internet-Draft

Optimizing Notifications for PNAs

February 2008

network often has more bandwidth and better scale to keep the presence server up to date. A Presence Network Agent (PNA) can publish presence information to a Presence Server(PS). When done large scale, the basic publish operation can be inefficient. When the network has millions of subscribers, only some of which have watchers, blind Publish operations are unnecessary. WINFO can be used to determine watchers, but the efficiency of maintaining WINFO per subscriber, and the size of the messages involved, make that solution unattractive. The PNA would prefer to have the Presence Server simply tell it when there was at least one watcher.

This document describes an XML document stored on the PS by which the PNA maintains a list of subscribers it can provide presence for as a SIP event package that tells the PNA when the number of watchers for a presentity on the list (or a specific presence element for a presentity) goes from 0 to at least 1 or from 1 to 0.

Internet-Draft

Optimizing Notifications for PNAs

February 2008

Table of Contents

1.	Terminology	4
2.	Background	4
3.	PNA Presentity List	5
3.1.	Application Unique ID (AUID)	5
3.2.	XML Schema	5
3.3.	Default Document Namespace	6
3.4.	MIME Type	6
3.5.	Validation Constraints	6
3.6.	Data Semantics	7
3.6.1.	Naming Conventions	7
3.6.2.	Resource Interdependencies	7
3.6.3.	Authorization Policies	7
4.	Watcher-Count Event Package	7
4.1.	Event Package Name	7
4.2.	Event Package Parameters	7
4.3.	SUBSCRIBE Bodies	8
4.4.	Subscription Duration	8
4.5.	NOTIFY Bodies	8
4.6.	Notifier Processing of SUBSCRIBE Requests	9
4.7.	Notifier Generation of NOTIFY Requests	9
4.8.	Subscriber Processing of NOTIFY Requests	10
4.9.	Handling of Forked Requests	10
4.10.	Rate of Notifications	10
4.11.	State Agents	10
5.	Watcher-count XML Document	11
5.1.	Structure of the watcher-count document	11
5.2.	Computing Watcher Counts from the Document	12
5.3.	Example	12
5.4.	XML Schema	13
6.	IANA Considerations	14
6.1.	application/watcher-count+xml MIME Registration	14
6.2.	URN Sub-Namespace Registration for urn:ietf:params:xml:ns:watcher-count	14
6.3.	Package Registration	15

7.	Security Considerations	15
7.1.	Denial of Service Attacks	15
7.2.	Divulging Sensitive Information	16
8.	Acknowledgements	16
9.	Normative References	16
	Authors' Addresses	17
	Intellectual Property and Copyright Statements	19

[1.](#) Terminology

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 [[RFC2119](#)].

[2.](#) Background

Presence systems [[RFC3856](#)] are being deployed in networks where the network itself has presence information. This information may also be known to the endpoint, but the network is often in a better position to publish [[RFC3903](#)] the information to the presence server. Mobile networks are an example of a network where a presence system can have a very large number of presentities and providers, and where bandwidth from the device to the presence server is restricted such that volatile presence data would be much more efficient if the network published some data elements to the presence server rather than the user agent itself. A "Presence Network Agent" (PNA) is a server operated by an entity in the network that has some presence information and publishes this information to the presence server on behalf of the presentity. Optimization techniques are available which make the actual Publish operation reasonably efficient. However, in large networks, there could be millions of presentities, and if interconnected with other networks, even more watchers, but at any point in time, there may not be any watchers for a particular presentity. If there are no watchers, and presence information that changes relatively often is published to the presence server anyway, there can be significant wasted effort and bandwidth in both the

presence server and the presence network agent.

If the PNA knew which presentities that it had presence information for had active watchers, then it could optimize its publishing operations. The presence server knows that, but the only way for the PNA to get that information is the WINFO package [[RFC3857](#)]. WINFO provides the requisite data, but inefficiently. All the PNA wants to know is when the number of watchers goes from zero (no watchers) to at least 1, or from at least one to zero. There is no efficient way to get that information.

PNAs provide information for a set of presentities, and the set of presentities the PNA serves may not match the set of presentities a particular PS serves. The PS has to know which presentities the PNA serves in order to send it the right watcher state information. This is simply a list of presentities that changes over time. The list can be very long, so sending it in its entirety when something changes on the list is not desirable.

Editor Note: There is an opportunity for further optimization if the PS knows which elements the PNA can publish. Because of filtering and presence rules, just because there is at least one watcher, it doesn't mean that the elements the PNA publishes are visible to any watchers. The PS could optimize notification of watcher counts to only show when at least one watcher was receiving at least one element the PNA publishes. This could further be extended to have the actual watcher count for each element sent by the PS to the PNA.

[3.](#) PNA Presentity List

This document defines a Presentity List document intended to be maintained on the PS by the PNS using XCAP [[RFC4825](#)].

[3.1.](#) Application Unique ID (AUID)

This specification defines the "pna-presentity-list" AUID within the IETF tree, via the IANA registration in [Section 6](#).

[3.2.](#) XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:pna-presentity-list"
  xmlns="urn:ietf:params:xml:ns:watcher-count-presentity-list"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="watcher-count-presentity-list">
    <xs:annotation>
      <xs:documentation>Root element for pna-presentity-list
        </xs:documentation>
    </xs:annotation>
    <xs:simpleType name="PNA">
      <xs:annotation>
        <xs:documentation>PNA</xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:anyURI"/>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

```

</xs:simpleType>
<xs:complexType>
  <xs:sequence>
    <xs:element name="PresententityList">
      <xs:annotation>
        <xs:documentation>List of Presentities</xs:documentation>
      </xs:annotation>
    <xs:simpleType name="Presententity">
      <xs:annotation>
        <xs:documentation>One Presentity on the list</xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:anyURI"/>
    </xs:simpleType>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

[3.3.](#) Default Document Namespace

The default document namespace used in evaluating a URI is urn:ietf:params:xml:ns:pna-presentity-list.

[3.4.](#) MIME Type

Documents conformant to this schema are known by the MIME type "application/pna-presentity-list+xml", registered in [Section 6](#).

[3.5.](#) Validation Constraints

The Presence Network Agent must be authorized to provide presence data for the presentities on the list.

[3.6.](#) Data Semantics

The PNA element defines the URI of the PNA maintaining the list. This URI must match the URI from which the PNA subscribes to the watcher-count event package on the PS.

[3.6.1.](#) Naming Conventions

The PS must maintain a document with the file name containing the PNA identity. Provisioning may be used to connect the file name to the PNA URI.

[3.6.2.](#) Resource Interdependencies

There are no resource interdependencies in this application usage beyond those defined by the schema.

[3.6.3.](#) Authorization Policies

This application usage does not change the default authorization policy defined by XCAP.

[4.](#) Watcher-Count Event Package

This section fills in the details needed to specify an event package as defined in [Section 4.4 of \[RFC3265\]](#).

[4.1.](#) Event Package Name

[RFC3265] requires package definitions to specify the name of their package or template-package.

The name of this template-package is "watcher-count". It cannot be applied to any other package. Recursive template-packaging is not allowed.

[4.2.](#) Event Package Parameters

[RFC3265] requires package and template-package definitions to specify any package specific parameters of the Event header field.

A single parameter "PNA" is defined. The parameter indicates pna-presententity-list URI.

[4.3.](#) SUBSCRIBE Bodies

[RFC3265] requires package or template-package definitions to define the usage, if any, of bodies in SUBSCRIBE requests.

No bodies are defined for the watcher-count package.

[4.4.](#) Subscription Duration

[RFC3265] requires package definitions to define a default value for subscription durations, and to discuss reasonable choices for durations when they are explicitly specified.

The PNA typically supports a large number of presentities, which typically have watchers come and go. The PNA wants notifications for any of the presentities on its list. Therefore, the duration of the subscription is typically long. The default value for the subscription duration is one day. However, clients SHOULD use an Expires header field to specify their preferred duration.

[4.5.](#) NOTIFY Bodies

[RFC3265] requires package definitions to describe the allowed set of body types in NOTIFY requests, and to specify the default value to be used when there is no Accept header field in the SUBSCRIBE request.

The body of the watcher-count notification contains a watcher-count document. This document contains a list of presentities in the pna-presentitivity list whose watcher count went from 0 to 1 or 1 to 0 and the current watcher count (which will always be zero or one). All watcher-count subscribers and notifiers MUST support the application/watcher-count+xml format described in herein, and MUST list its MIME type, application/watcher-count+xml, in any Accept header field present in the SUBSCRIBE request.

Other watcher count formats might be defined in the future. In that case, the watcher-count subscriptions MAY indicate support of other formats. However, they MUST always support and list application/watcher-count+xml as an allowed format.

Of course, the watcher-count notifications generated by the server MUST be in one of the formats specified in the Accept header field in the SUBSCRIBE request. If no Accept header field was present, the notifications MUST use the application/watcher-count+xml format described herein.

[4.6.](#) Notifier Processing of SUBSCRIBE Requests

[RFC3265] specifies that packages should define any package-specific processing of SUBSCRIBE requests at a notifier, specifically with regards to authentication and authorization.

Although the number of watchers is less sensitive than identification of a watcher, watcher information is personal information. Therefore, all watcher-count subscriptions MUST be authenticated and then authorized before approval. Authentication MAY be performed using any of the techniques available through SIP, including digest, S/MIME, TLS or other transport specific mechanisms [[RFC3261](#)]. Authorization policy is at the discretion of the administrator.

Editor Note: There is a timing problem. When a PS gets a SUBSCRIBE, it should reply immediately with the presence state. However, if this causes watcher-count to go from 0 to 1, the PS doesn't have good state. It has to NOTIFY the PNA and wait for a response. That needs to be described here. There may be further complications with a one time or short subscription.

[4.7.](#) Notifier Generation of NOTIFY Requests

The SIP Event framework requests that packages specify the conditions under which notifications are sent for that package, and how such notifications are constructed.

Each watcher-count subscription is associated with a set of "inner" subscriptions being watched. This set is defined by the URI in the pna-presentity-list. A watcher-count notifier MUST generate a notification any time the count of watchers in any of the watched subscriptions goes from zero to at least one, or from one to zero. To optimize the notification, the PS may delay the issuance of the NOTIFY for a provisioned period of time. 5 seconds is the suggested default time. The delay gives the PS an opportunity to gather additional watcher count changes and send one NOTIFY with all of them received in the delay period. The PS MUST make certain that no watcher count change from zero to at least one or one to zero is delayed by more than this period.

It is RECOMMENDED that a given notification only mention a particular presentity once. The PNA MUST NOT depend on this behavior. When the same presentity URI is encountered in more than one wc element's "r" value, the "c" value in the last wc element determines the watcher count of the presentity following processing in the PNA. That means that the order of wc elements may matter. The recommended behavior

would filter multiple watcher changes from growing the size of the NOTIFY body.

Upon a successful SUBSCRIBE where no subscription for a particular pna-presentity-list was extant at the time of the subscription, the initial NOTIFY from the PS to the PNA MUST have all of the presentities for which there is at least one watcher. That is, the PNA and PS behave as if just before the subscription, all presentities on the list had no watchers. Presentities that actually do have at least one watcher will be listed in the initial NOTIFY. If at any time the PNA is concerned that it has lost track of watcher count, it can reSUBSCRIBE, triggering a complete notification of watcher count. Note that the size of this initial NOTIFY can be quite large.

[4.8.](#) Subscriber Processing of NOTIFY Requests

[RFC3265] expects packages to specify how a subscriber processes NOTIFY requests in any package specific ways, and in particular, how it uses the NOTIFY requests to construct a coherent view of the state of the subscribed resource. The watcher-count NOTIFY will only contain information about those presentities whose watcher count changed from zero to at least one, or from one to zero. To construct a coherent view of the total state of all presentities on the pna-presentity-list, a watcher-count subscriber will need to combine NOTIFYS received over time. See [Section 4.7](#) for a discussion of how the PNA can trigger a complete state update from the PS.

[4.9.](#) Handling of Forked Requests

The behavior of forked requests for watcher-count is not defined and implementations MUST NOT fork SUBSCRIBE or NOTIFY requests.

[4.10.](#) Rate of Notifications

[RFC3265] mandates that packages define a maximum rate of notifications for their package.

For reasons of congestion control, it is important that the rate of notifications not become excessive. As a result, it is RECOMMENDED that the server not generate watcher-count notifications for a single presentity at a rate faster than once every 5 seconds. See

[Section 4.8](#) for a discussion of pacing NOTIFY operations for changes to multiple presentity's watcher count. It is RECOMMENDED that such a pacing mechanism be used.

[4.11.](#) State Agents

[RFC3265] asks packages to consider the role of state agents in their design.

Rosen, et al.

Expires August 21, 2008

[Page 10]

Internet-Draft

Optimizing Notifications for PNAs

February 2008

State agents are not required for watcher-count.

[5.](#) Watcher-count XML Document

[5.1.](#) Structure of the watcher-count document

Watcher-count is an XML [ref] document that MUST be well-formed and SHOULD be valid. Watcher-count documents MUST be based on XML 1.0 and MUST be encoded using UTF-8. This specification makes use of XML namespaces for identifying watcher-count documents and document fragments. The namespace URI for elements defined by this specification is a URN [[RFC2141](#)], using the namespace identifier 'ietf' defined by [[RFC2648](#)] and extended by [[RFC3688](#)]. This URN is: urn:ietf:params:xml:ns:watcher-count

A watcher-count document begins with the root element tag "watcher-count-list". It consists of any number of "wc" (for "watcher-count") sub- elements, each of which is a presentity URI and a count of watchers for that presentity. The count is either zero or one, where zero means no watchers are presently receiving any form of presence updates for the presentity and one means there is at least one active watcher. Other elements from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored. There are two attributes associated with the "watcher-count-list" element, both of which MUST be present:

PNA: This element contains the URI of a pna-presence-list maintained on the PS. The presentities in the watcher-count document MUST be on that list

version: This attribute allows the recipient of watcher-count documents to properly order them. Versions start at 0, and increment by one for each new document sent to a watcher-count

subscriber. Versions are scoped within a watcher-count subscription. Versions MUST be representable using a 32 bit integer. However, versions do not wrap.

Each "wc" element contains a list of presentities whose count of watchers has changed from zero to at least one or from one to zero. Other elements from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored. There are two attributes associated with the "wc" element, both of which MUST be present:

r: This attribute contains a URI for the resource being watched. It is mandatory.

c: This attribute contains an integer value of 0 or 1. Zero means that there are presently no watchers for this resource. One means there is at least one watcher.

The names "wc", "r" and "c" are deliberately short, as the document is expected to be long and contain a great many such elements.

[5.2.](#) Computing Watcher Counts from the Document

Typically, the watcher-count NOTIFY will only contain information about those presentities whose state has changed. To construct a coherent view of the total state of all presentities on the pna-presentity-list, a watcher-count subscriber will need to combine NOTIFYS received over time. The watcher-count subscriber conceptually maintains a table for each presentity on the pna-presentity-list. Each pna-presentity-list is uniquely identified by the URI in the "PNA" attribute of the "watcher-count-list" element. Each table contains a row for each presentity in that list. Each row is indexed by the URI for that presentity. It is conveyed in the "r" attribute of the "wc" element. The contents of each row contain the count of watchers of that presentity as conveyed in the "wc" element. Other implementations are possible. For example, the PNA could maintain a list of presentities whose watcher-count is >1 and add/delete presentities to its list based on notifications it receives from the PS.

The tables are also associated with a version number. The version number MUST be initialized with the value of the "version" attribute from the "watcherinfo" element in the first document received. Each time a new document is received, the value of the local version number, and the "version" attribute in the new document, are compared. If the value in the new document is one higher than the local version number, the local version number is increased by one, and the document is processed. If the value in the document is more than one higher than the local version number, the local version number is set to the value in the new document, the document is processed, and the watcherinfo subscriber SHOULD generate a refresh request to trigger a full state notification. If the value in the document is less than the local version, the document is discarded without processing.

[5.3.](#) Example

The following is an example of watcher-count information.

```
<?xml version="1.0"?>
<watcher-count-list xmlns="urn:ietf:params:xml:ns:watcher-count"
    pna="http://www.example.com/west-pna-presence-list.xml"
    version="23">
  <wc r="sip:+12125551212@example.net" c="1">
  <wc r="sip:+12125553333@example.net" c="0">
</watcher-count-list>
```

[5.4.](#) XML Schema

The following is the schema definition of the watcher-count document format:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:ietf:params:xml:ns:watcher-count"
    xmlns:tns="urn:ietf:params:xml:ns:watcher-count" >
<!-- This import brings in the XML language attribute xml:lang-->
```

```

<xs:import namespace="http://www.w3.org/XML/1998/namespace"
            schemaLocation="http://www.w3.org/2001/03/xml.xsd" />
<xs:element name="watcher-count">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:watcher-count-list" minOccurs="0"
                  maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
              maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="resource" type="xs:anyURI" use="required"/>
    <xs:attribute name="version" type="xs:nonNegativeInteger"
                  use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="wc">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:wc" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax"
              minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="r" type="xs:anyURI" use="required"/>
    <xs:attribute name="c" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>

```

[6.](#) IANA Considerations

This document registers a new MIME type, application/watcher-count+xml, registers a new XML namespace and registers a new event package.

[6.1.](#) application/watcher-count+xml MIME Registration

MIME media type name: application
 MIME subtype name: watcher-count+xml
 Mandatory parameters: none

Optional parameters: Same as charset parameter application/xml as specified in [RFC3023].
Encoding considerations: Same as encoding considerations of application/xml as specified in [RFC3023].
Security considerations: See [Section 10 of \[RFC3023\]](#) and [Section 7](#) of this specification.
Interoperability considerations: none.
Published specification: This document.
Applications which use this media type: This document type is used to support optimization of publishing operations from a Presence Network Agent to a Presence Server.
Additional Information:
 Magic Number: None
 File Extension: .wif or .xml
 Macintosh file type code: "TEXT"
Personal and email address for further information: Brian Rosen
<br@brianrosen.net>
Intended usage: COMMON
Author/Change controller: The IETF.

[6.2.](#) URN Sub-Namespace Registration for urn:ietf:params:xml:ns:watcher-count

This section registers a new XML namespace, as per the guidelines in [?].

URI: The URI for this namespace is
urn:ietf:params:xml:ns:watcher-count.
Registrant Contact: IETF, SIMPLE working group, <simple@ietf.org>, Brian Rosen <br@brianrosen.net>.
XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
```



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
        content="text/html; charset=iso-8859-1"/>
  <title>Watcher Count Namespace</title>
</head>
<body>
  <h1>Namespace for Watcher Count</h1>
  <h2>urn:ietf:params:xml:ns:watcher-count</h2>
  <p>See <a href="ftp://ftp.rfc-editor.org/in-notes/rfc3858.txt">
      RFC3858</a>.</p>
</body>
</html>
END
```

[6.3.](#) Package Registration

This specification registers an event template package as specified in [Section 6.2 of \[RFC3265\]](#).

Package Name: watcher-count

Template Package: yes

Published Specification: This document

Person to Contact: Brian Rosen, br@brianrosen.net.

[7.](#) Security Considerations

[7.1.](#) Denial of Service Attacks

Watcher count generates notifications about changes in the state of watchers for a particular resource. A single notification could be extremely large, as in the initial state notification. This makes a watcher-count subscription a potential tool for denial of service attacks. Preventing these can be done through a combination of sensible authorization policies and good operating principles.

Watcher-count subscriptions to that resource should only be allowed from explicitly authorized entities, whose identity has been properly authenticated. That prevents a watcher-count NOTIFY stream from being generated from subscriptions made by an attacker.

[7.2.](#) Divulging Sensitive Information

Watcher count indicates how many users are interested in a particular resource. Depending on the package and the resource, this can be very sensitive information. One way in which this information can be revealed is eavesdropping. An attacker can observe watcher-count notifications, and learn this information. To prevent that, watchers MAY use the sips URI scheme when subscribing to a watcherinfo resource. Notifiers for watcher-count MUST support TLS and sips as if they were a proxy (see [Section 26.3.1 of \[RFC3261\]](#)).

SIP encryption, using S/MIME, MAY be used end-to-end for the transmission of both SUBSCRIBE and NOTIFY requests.

Another way in which this information can be revealed is through spoofed subscriptions. These attacks can be prevented by authenticating and authorizing all watcher-count subscriptions. In order for the notifier to authenticate the subscriber, it MAY use HTTP Digest ([Section 22 of \[RFC3261\]](#)). As a result, all watchers MUST support HTTP Digest. This is a redundant requirement, however, since all SIP user agents are mandated to support it by [\[RFC3261\]](#).

[8.](#) Acknowledgements

Guy Paradis and Fridy Sharon-Fridman of NeuStar contributed to the ideas in this document and reviewed the text.

[9.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2141] Moats, R., "URN Syntax", [RFC 2141](#), May 1997.
- [RFC2648] Moats, R., "A URN Namespace for IETF Documents", [RFC 2648](#), August 1999.
- [RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.
- [RFC3256] Jones, D. and R. Woundy, "The DOCSIS (Data-Over-Cable Service Interface Specifications) Device Class DHCP (Dynamic Host Configuration Protocol) Relay Agent Information Sub-option", [RFC 3256](#), April 2002.

Internet-Draft

Optimizing Notifications for PNAs

February 2008

A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.

[RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.

[RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), January 2004.

[RFC3856] Rosenberg, J., "A Presence Event Package for the Session Initiation Protocol (SIP)", [RFC 3856](#), August 2004.

[RFC3857] Rosenberg, J., "A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)", [RFC 3857](#), August 2004.

[RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", [RFC 3903](#), October 2004.

[RFC4825] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", [RFC 4825](#), May 2007.

Authors' Addresses

Brian Rosen
NeuStar, Inc.
470 Conrad Dr
Mars, PA 16046
US

Email: br@brianrosen.net

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: salvatore.loreto@ericsson.com

Rosen, et al.

Expires August 21, 2008

[Page 17]

Internet-Draft

Optimizing Notifications for PNAs

February 2008

Krisztian Kiss
Nokia
313 Fairchild Dr
Mountain View, CA 94043
US

Email: krisztian.kiss@nokia.com

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights 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; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be

found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat 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 implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).