Network Working Group
Request for Comments: 4354
Category: Informational

M. Garcia-Martin
Nokia

January 2006

A Session Initiation Protocol (SIP) Event Package and Data Format for Various Settings in Support for the Push-to-Talk over Cellular (PoC) Service

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

The Open Mobile Alliance (OMA) is defining the Push-to-talk over Cellular (PoC) service where SIP is the protocol used to establish half-duplex media sessions across different participants, to send instant messages, etc. This document defines a SIP event package to support publication, subscription, and notification of additional capabilities required by the PoC service. This SIP event package is applicable to the PoC service and may not be applicable to the general Internet.

Table of Contents

1. Introduction

The Open Mobile Alliance (OMA) (http://www.openmobilealliance.org) is currently specifying the Push-to-talk over Cellular (PoC) service. This service allows a SIP User Agent (PoC terminal) to establish a session to one or more SIP User Agents (UAs) simultaneously, usually initiated when the initiating user pushes a button.

OMA has defined a collection of very stringent requirements in support of the PoC service. In order to provide the user with a satisfactory experience, the initial session establishment (from the time the user presses the button to the time they get an indication to speak) must be minimized.

The PoC terminal may support hardware capabilities such as a speakerphone and/or headset and software that provide the capability for the user to configure the PoC terminal to accept session initiations immediately and play out the media as soon as it is received without requiring the intervention of the called user. This mode of operation is known as Auto-Answer mode or automatic mode. The user may alternatively configure the PoC terminal to first alert the user and require the user to accept the session invitation manually before media is accepted. This mode of operation is known as Manual-Answer mode. The PoC terminal may support both or only one of these modes of operation. The user may change the Answer Mode (AM) configuration of the PoC terminal frequently based on their current circumstances and preference (perhaps because the user is busy or in a public area where she cannot use a speaker phone, etc.).

SIP PoC terminals can support various SIP-based communication services in addition to Push-to-talk (e.g., VoIP telephony, presence services, messaging services, etc.). The user may at times wish to disable the acceptance of Push-to-talk sessions whilst still remaining SIP registered for one or more other SIP-based services. When the PoC terminal is configured to not accept any incoming Push-to-talk sessions, this is known as Incoming Session Barring (ISB).

A user may wish to contact another user who has a PoC terminal with Incoming Session Barring enabled. A user may send an Instant Personal Alert to another user to inform him that he wishes to engage him in a PoC Session. This Instant Personal Alert is received even when the destination PoC terminal has enabled Incoming Session Barring. If a user wishes to disable the acceptance of Instant Personal Alerts, he can configure his PoC terminal not accept any incoming Instant Personal Alerts. This is known as Instant Personal Alert Barring (IPAB).

Some PoC terminals may provide support for handling multiple PoC sessions simultaneously whereas other terminals are only able to handle one PoC session at time. Or, even if the terminal is able to handle multiple PoC sessions simultaneously, the user may desire to have just one single PoC session at a time. This indication of support for multiple PoC sessions simultaneously is known as Simultaneous PoC Sessions Support (SSS).

The OMA PoC Architecture utilizes SIP servers within the network that may perform roles such as a conference focus [12], an RTP translator, or a policy server. A possible optimization to minimize the delay in providing the caller with an indication to speak consist of the SIP network server to perform buffering of media packets in order to provide an early or unconfirmed indication to the caller and allow the caller to start speaking before the called PoC terminal has answered. This optimization only is appropriate when the called PoC terminal is currently accepting PoC sessions and its Answer Mode is set to Auto-Answer. This optimization therefore requires the network SIP server to have knowledge of the current ISB and AM settings of the called PoC terminal.

Similarly, in order to avoid unnecessary transmission of Instant Personal Alerts across the radio interface, the network SIP server needs to have knowledge of the current IPAB setting at the terminal.

When the UA supports multiple PoC sessions simultaneously the server needs to act as a B2BUA in order to multiplex media and floor control signaling between multiple sessions using a single bandwidth limited radio bearer. When handling of multiple PoC sessions simultaneously is not needed the server can act as a SIP proxy. It is therefore advantageous for the server to be informed whether the UA currently intends to support multiple PoC sessions simultaneously.

This document proposes additional SIP capabilities to enable the communication of the ISB, AM, IPAB, and SSS settings between the SIP PoC terminal and the SIP network server.

We define a SIP event package that allows a SIP Event Publication Agent (EPA) to publish the user’s settings at that particular EPA which may impact some specific session attempts. This allows subscribers to subscribe to the Event State Compositor to this event package to gather this information, and anticipate to the user’s needs when a session is attempted to that user. It is believed that the SIP event package defined here is not applicable to the general Internet: it has been designed to serve the architecture of the PoC service. In particular, and in the context defined by RFC 3903 [8], it is the intention of OMA to make PoC terminals behave as Event Publication Agents (EPA), and network servers behave as Event State

Compositors (ESC). It is possible that PoC terminals and network servers may also subscribe to the user's PoC related settings, so that changes in this state made in one terminal are kept in synchronization across all different terminals or with the network server for a particular user.

This document defines a PoC-settings document that allows an EPA to convey its ISB, AM, IPAB, and SSS settings to an ESC. The EPA sends a PoC-settings document in PUBLISH requests [8]. The PoC-settings document contain represents the settings view at that particular EPA. The ESC can collect PoC-settings document for the same user at different EPAs, apply a composition policy, and provide notifications. Notifications can contain a composed view of the settings or a list of settings per EPA, depending on whether the ESC is able to resolve conflicts. A subscriber can receive notifications of changes in this document according to the procedures specified in RFC 3265 [5]. The aim of this memo is to follow the procedure indicated in RFC 3427 [6] and to register a new poc-settings event package with IANA.

2. Terminology

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

3. Applicability Statement

The event package defined in this document is intended for use with network-based application servers that provide a Push-to-Talk over Cellular service.

4. Requirements

A comprehensive description of all the requirements that affect the Push-to-Talk over Cellular service developed by the Open Mobile Alliance can be found in the Open Mobile Alliance web page at http://www.openmobilealliance.org.

For the sake of simplicity, we briefly discuss here those requirements that affect the solution described in this document. These requirements can be summarized as follows:

1. There must be a mechanism that reduces the session setup time as much as possible.
2. In order to allow proper usage of scarce resources, there must be a mechanism that saves the air interface from being congested with unneeded or undesired traffic.
3. The mechanism should not involve the implementation of new protocols, unless strictly needed.

These requirements lead to a solution whereby the user can indicate to a network node his ability to accept or reject sessions or certain types of messages. Pushing these settings to a network node allows the network node to produce a faster response to the originator, perhaps even declining or filtering some SIP requests towards the destination. This approaches the goal of reducing the session setup time.

5. The "poc-settings" Event Package

RFC 3265 [5] defines a SIP extension for subscribing to remote nodes and receiving notifications of changes (events) in their states. It leaves the definition of many aspects of these events to concrete extensions, known as event packages. This document qualifies as an event package. This section fills in the information required for all event packages by RFC 3265 [5].

Additionally, RFC 3903 [8] defines an extension that allows SIP User Agents to publish event state. According to RFC 3903 [8], any event package intended to be used in conjunction with the SIP PUBLISH method has to include a considerations section. This section also fills the information for all event packages to be used with PUBLISH requests.

We define a new "poc-settings" event package. Event Publication Agents (EPA) use PUBLISH requests to inform an Event State Compositor (ESC) of changes in the poc-settings event package. Acting as a notifier, the ESC notifies subscribers to the user’s poc-settings information when changes occur.

5.1. Package Name

The name of this package is "poc-settings". As specified in RFC 3265 [5], this value appears in the Event header field present in SUBSCRIBE and NOTIFY requests. As specified in RFC 3903 [8], this value also appears in the Event header field present in PUBLISH requests.

5.2. Event Package Parameters

RFC 3265 [5] allows event packages to define additional parameters carried in the Event header field. This event package, "poc-settings", does not define additional parameters.

5.3. SUBSCRIBE Bodies

According to RFC 3265 [5], a SUBSCRIBE request can contain a body. The purpose of the body depends on its type. Subscriptions to the poc-settings event package will normally not contain bodies.

The Request-URI of the SUBSCRIBE request identifies the user about whose poc-settings the subscriber wants to be informed.

5.4. Subscription Duration

The default expiration time for subscriptions within this package is 3600 seconds. As per RFC 3265 [5], the subscriber MAY specify an alternate expiration in the Expires header field.

5.5. NOTIFY Bodies

As described in RFC 3265 [5], the NOTIFY message will contain bodies describing the state of the subscribed resource. This body is in a format listed in the Accept header field of the SUBSCRIBE request, or a package-specific default format if the Accept header field was omitted from the SUBSCRIBE request.

In this event package, the body of the notification contains a PoC-settings document (see Section 6). The ESC has gathered PoC-settings documents for the user at different EPAs. The ESC applies a composition policy and composes a PoC-settings document with a common view of all these settings across different EPAs. In case the ESC is not able to resolve a conflict, due to contradictory information provided by two different EPAs, the ESC provides a PoC-settings document containing the settings at each terminal so that the subscriber can resolve the conflict.

All subscribers and notifiers of the "poc-settings" event package MUST support the "application/poc-settings+xml" data format described in Section 6. The SUBSCRIBE request MAY contain an Accept header field. If no such header field is present, it has a default value of "application/poc-settings+xml" (assuming that the Event header field contains a value of "poc-settings"). If the Accept header field is present, it MUST include "application/poc-settings+xml" and MAY include any other types capable of representing user settings for PoC.

5.6. Notifier Processing of SUBSCRIBE Requests

The contents of a PoC-settings document can contain sensitive information that can reveal some privacy information. Therefore, PoC-settings documents MUST only be sent to authorized subscribers. In order to determine if a subscription originates in an authorized user, the user MUST be authenticated as described in Section 5.6.1 and then he MUST be authorized to be a subscriber as described in Section 5.6.2.

5.6.1. Authentication

Notifiers MUST authenticate all subscription requests. This authentication can be done using any of the mechanisms defined in RFC 3261 [4] and other authentication extensions.

5.6.2. Authorization

Once authenticated, the notifier makes an authorization decision. A notifier MUST NOT accept a subscription unless authorization has been provided by the user. The means by which authorization are provided are outside the scope of this document. Authorization may have been provided ahead of time through access lists, perhaps specified in a web page. Authorization may have been provided by means of uploading some kind of standardized access control list document.

5.7. Notifier Generation of NOTIFY Requests

RFC 3265 [5] details the formatting and structure of NOTIFY messages. However, packages are mandated to provide detailed information on when to send a NOTIFY, how to compute the state of the resource, how to generate neutral or fake state information, and whether state information is complete or partial. This section describes those details for the poc-settings event package.

A notifier MAY send a NOTIFY at any time. Typically, it will send one when the poc-settings stage of a user changes. The NOTIFY request MAY contain a body containing a PoC-settings document. The times at which the NOTIFY is sent for a particular subscriber, and the contents of the body within that notification, are subject to any rules specified by the authorization policy that governs the subscription. However, typically the NOTIFY will contain an indication of those PoC-related services for which a change has occurred.

In the case of a pending subscription, when final authorization is determined, a NOTIFY can be sent. If the result of the authorization decision was success, a NOTIFY SHOULD be sent and SHOULD contain a

complete PoC-settings document with the current state of the user's PoC settings. If the subscription is rejected, a NOTIFY MAY be sent. As described in RFC 3265 [5], the Subscription-State header field indicates the state of the subscription.

The body of the NOTIFY MUST be sent using one of the types listed in the Accept header field in the most recent SUBSCRIBE request, or using the type "application/poc-settings+xml" if no Accept header field was present.

Notifiers will typically act as Event State Compositors (ESC) and thus will learn the poc-settings event state via PUBLISH requests sent from the user's Event Publication Agent (EPA) when the user changes one of those settings. It is possible that the notifier generates a NOTIFY request for a user for which no publication has taken place. In that case, the PoC-settings document will not contain any <entity> element (see Section 6.1 for a detailed description of the <entity> element).

For reasons of privacy, it will frequently be necessary to encrypt the contents of the notifications. This can be accomplished using S/MIME [9]. The encryption can be performed using the key of the subscriber as identified in the From field of the SUBSCRIBE request. Similarly, integrity of the notifications is important to subscribers. As such, the contents of the notifications MAY provide authentication and message integrity using S/MIME [9]. Since the NOTIFY is generated by the notifier, which may not have access to the key of the user represented by the poc-settings user, often the NOTIFY will be signed by a third party. The NOTIFY request SHOULD be signed by an authority over the domain of the user. In other words, for a user whose SIP URI is sip:user@example.com, the signator of the NOTIFY SHOULD be the authority for example.com.

5.8. Subscriber Processing of NOTIFY Requests

RFC 3265 [5] leaves it to event packages to describe the process followed by the subscriber upon receipt of a NOTIFY request, including any logic required to form a coherent resource state.

In this specification, each NOTIFY request contains either no PoC-settings document, or a document representing one or more PoC related settings for a given user. Within a dialog, the PoC-settings document in the NOTIFY request with the highest CSeq header field value is the current one. When no document is present in that NOTIFY, the PoC-settings document present in the NOTIFY with the next highest CSeq value is used.

5.9. Handling of Forked Requests

RFC 3265 [5] requires each package to describe handling of forked SUBSCRIBE requests.

This specification only allows a single dialog to be constructed as a result of emitting an initial SUBSCRIBE request. This guarantees that only a single subscriber is generating notifications for a particular subscription to a particular user. The result of this is that a user can have multiple SIP User Agents active, but these should be homogeneous, so that each can generate the same set of notifications for the user's poc-settings.

5.10. Rate of Notifications

RFC 3265 [5] requires each package to specify the maximum rate at which notifications can be sent.

Poc-settings notifiers SHOULD NOT generate notifications for a single user at a rate of more than once every five seconds.

5.11. State Agents

RFC 3265 [5] requires each package to consider the role of state agents in the package and, if they are used, to specify how authentication and authorization are done.

This specification allows state agents to be located in the network. Publication of PoC-settings document is linked to a user. However, a user may be simultaneously logged in at different PoC terminals. If a user changes her PoC settings from a terminal, it will send a PUBLISH request containing a PoC-settings document. These settings are applicable to the user independently of the terminal at which she is logged in. In other words, PoC settings changes done in a terminal affect all the PoC terminals where the user is logged. It is RECOMMENDED that each of the terminals where the user is logged in subscribes to its own PoC-settings document in order to keep a coherent state view with the state agent.

5.12. Examples

An example of a PoC-setting document is provided in Section 6.2.

5.13. Use of URIs to Retrieve State

RFC 3265 [5] allows packages to use URIs to retrieve large state documents.

PoC-settings documents are fairly small. This event package does not provide a mechanism to use URIs to retrieve large state documents.

5.14. PUBLISH Bodies

RFC 3903 [8] requires event packages to define the content types expected in PUBLISH requests.

In this event package, the body of a PUBLISH request contains a PoC-settings document (see Section 6). This PoC-settings document describes the PoC-related settings of a user at an EPA. EPAs SHOULD include their own information in a PoC-settings document; i.e., there SHOULD be a single <entity> element in the body of the PUBLISH request (See Section 6.1 for a detailed description of the <entity> element).

All EPAs and ESCs MUST support the "application/poc-settings+xml" data format described in Section 6 and MAY support other formats.

5.15. PUBLISH Response Bodies

This specification does not associate semantics to a body in a PUBLISH response.

5.16. Multiple Sources for Event State

RFC 3903 [8] requires event packages to specify whether multiple sources can contribute to the event state view at the ESC.

This event package allows different EPAs to publish the PoC settings for a particular user. Each EPA publishes its own settings grouped in an <entity> element. The EPA provides a globally unique identifier for a given address of record. This allows the ESC to differentiate EPAs and either compose a state resolving conflicts or provide the union of the states of all the EPAs that contributed to it. The composition policy at the ESC is outside the scope of this document.

5.17. Event State Segmentation

RFC 3903 [8] defines segments within a state document. Each segment is defined as one of potentially many identifiable sections in the published event state.

This event package defines, for a given EPA, four segments identified by the elements <isb-settings>, <am-settings>, <ipab-settings>, and <sss-settings>, respectively. Each of them refers to different states of the EPA.

5.18. Rate of Publication

RFC 3903 [8] allows event packages to define their own rate of publication.

There are no rate-limiting recommendations for poc-settings publication. Since changes in a PoC-settings document are typically triggered by interaction with a human user, there is not periodicity, nor a minimum or maximum rate of publication.

6. PoC-Settings Document

PoC-settings is an XML document [10] that MUST be well-formed and SHOULD be valid. PoC-settings documents MUST be based on XML 1.0 and MUST be encoded using UTF-8 [7]. This specification makes use of XML namespaces for identifying PoC-settings documents. The namespace URI for elements defined by this specification is a URN [2], using the namespace identifier 'oma'. This URN is:

urn:oma:params:xml:ns:poc:poc-settings

PoC-settings documents are identified with the MIME type "application/poc-settings+xml" and are instances of the XML schema defined in Section 6.1.

A PoC-settings document begins with the root element tag <poc-settings>. It consists of zero or more <entity> elements, each one including an 'id' attribute that contains a globally unique identifier for a given address of record that represents an EPA. An <entity> element represents an EPA, and it is uniquely identified by the 'id' attribute. EPAs SHOULD include a single <entity> element in a PoC-settings document. ESCs MAY include several <entity> elements in a PoC-settings document, typically when the ESC is unable to resolve conflicts due to incongruent publication from different sources.

A valid PoC-settings document can include zero <entity> elements if the ESC provides a notification for which no publication has occurred.

The <entity> element MAY contain other elements and attributes from different namespaces for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

The <entity> element consists of zero or one <isb-settings> elements, zero or one <am-settings> elements, zero or one <ipab-settings>, and zero or one <sss-settings> elements. Other elements and attributes

from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

An <isb-settings> element contains a single <incoming-session-barring> element that contains a boolean 'active' attribute. The 'active' attribute indicates whether incoming sessions are barred at the UA, depending on the user's preferences for this setting. Other elements and attributes from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

An <am-settings> element contains an <answer-mode> element, whose value can be set to either "automatic" or "manual". Other elements and attributes from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

A server such as a URI-list server [11] receives a SIP request addressed to one or more recipients. If the intended recipient set the <answer-mode> to "manual", the URI-list server proceeds with the session attempt. If she set it to "automatic", the URI-list server generates a 200-class response prior to contacting the intended recipient.

An <ipab-settings> element contains a single <incoming-personal-alert-barring> element that contains a boolean 'active' attribute. The 'active' attribute indicates whether incoming personal alert messages are barred at the UA, depending on the user's preferences for this setting. Other elements from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

An <sss-settings> element contains a single <simultaneous-sessions-support> element that contains a boolean 'active' attribute. The 'active' attribute indicates whether the SIP UA is willing to handle more than one PoC session simultaneously. If the 'active' attribute is set to "false" or "0", then when the SIP UA is engaged in a PoC session, and the SIP UA receives an second incoming request for a SIP PoC session, the UA will decline the invitation. If the 'active' attribute is set to "true" or "1", then when the SIP UA is engaged in a PoC session, and the SIP UA receives an second incoming request for a SIP PoC session, the UA will possibly accept the invitation. Other elements and attributes from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

6.1. XML Schema

Implementations according to this specification MUST comply to the following XML Schema, which defines the constraints of the PoC-settings document:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oma:params:xml:ns:poc:poc-settings"
    xmlns="urn:oma:params:xml:ns:poc:poc-settings"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
             schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xs:annotation>
    <xs:documentation xml:lang="en">
      XML Schema Definition in support of the Incoming Session
      Barring, Answer Mode, Incoming Personal Alert Barring,
      and Simultaneous Sessions Support in the Push-to-talk
      over Cellular (PoC) service.
    </xs:documentation>
  </xs:annotation>

  <xs:element name="poc-settings" type="poc-settingsType"/>

  <xs:complexType name="poc-settingsType">
    <xs:sequence>
      <xs:element name="entity" type="entityType"
                  minOccurs="0" maxOccurs="unbounded" />
      <xs:any namespace="##other" processContents="lax"
               minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="entityType">
    <xs:sequence>
      <xs:element name="isb-settings" type="isbSettingType"
                  minOccurs="0"/>
      <xs:element name="am-settings" type="amSettingType"
                  minOccurs="0"/>
      <xs:element name="ipab-settings" type="ipabSettingType"
                  minOccurs="0"/>
      <xs:element name="sss-settings" type="sssSettingType"
                  minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax"
              minOccurs="0" maxOccurs="unbounded"/>
```

```
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" use="required"/>
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="isbSettingType">
      <xs:sequence>
        <xs:element name="incoming-session-barring">
          <xs:complexType>
            <xs:attribute name="active" type="xs:boolean"
                          use="required" />
          </xs:complexType>
        </xs:element>
        <xs:any namespace="##any" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="amSettingType">
      <xs:sequence>
        <xs:element name="answer-mode">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="automatic"/>
              <xs:enumeration value="manual"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:any namespace="##any" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="ipabSettingType">
      <xs:sequence>
        <xs:element name="incoming-personal-alert-barring">
          <xs:complexType>
            <xs:attribute name="active" type="xs:boolean"
                          use="required" />
          </xs:complexType>
        </xs:element>
        <xs:any namespace="##any" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
```

```
    <xs:complexType name="sssSettingType">
      <xs:sequence>
        <xs:element name="simultaneous-sessions-support">
          <xs:complexType>
            <xs:attribute name="active" type="xs:boolean"
                          use="required"/>
          </xs:complexType>
        </xs:element>
       <xs:any namespace="##any" processContents="lax"
               minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>

  </xs:schema>
```

6.2. Example

The following is an example of a PoC-settings document:

```
<?xml version="1.0" encoding="UTF-8"?>

<poc-settings xmlns="urn:oma:params:xml:ns:poc:poc-settings">
  <entity id="do39s8zksn2d98x">
     <isb-settings>
       <incoming-session-barring active="true"/>
     </isb-settings>
     <am-settings>
       <answer-mode>automatic</answer-mode>
     </am-settings>
     <ipab-settings>
       <incoming-personal-alert-barring active="false"/>
     </ipab-settings>
     <sss-settings>
       <simultaneous-sessions-support active="true"/>
     </sss-settings>
  </entity>
</poc-settings>
```

7. Security Considerations

The "poc-settings" event package defined by this document is meant to be transported with SIP PUBLISH requests. Therefore, the Security Considerations (Section 14) in RFC 3903 [8] apply to this document. In particular, the settings contained in the "poc-settings" event package are applicable to the user that generated the SIP PUBLISH request. Therefore, servers that receive SIP PUBLISH requests containing a "poc-settings" event package SHOULD authenticate the user prior to authorizing the event publication (as required by RFC 3903 [8]).

Authentication and authorization of subscriptions have been discussed in Section 5.6. Lack of authentication or authorization may provide poc-settings information to unauthorized parties, who can use that information for creating attacks. For example, an unauthorized recipient of a PoC-settings document can learn that the publisher's terminal is set to answer PoC sessions in automatic answer mode and then create a malicious session containing inappropriate media that the UAS will play automatically. Or the attacker can learn that the terminal is willing to receive simultaneous PoC sessions and then try to exhaust resources in the SIP UA by creating bogus PoC sessions that leave hung states in the attacked SIP UA.

Integrity protection and confidentiality of notifications are also discussed in Section 5.7. If a notifier does not encrypt bodies of NOTIFY requests, an eavesdropper could learn the status of a SIP user agent and use it to create malicious PoC sessions. If the notifier does not integrity protect the bodies of NOTIFY requests, a man-in-the-middle attacker or malicious SIP proxy could modify the contents of the poc-settings event package notification. Although this does not cause harm, it can create annoyances (e.g., media clip due to lack of buffering) when PoC sessions are delivered to the user.

8. Acknowledgements

The author wants to thank Ilkka Westman, Andrew Allen, Chinmay Padhye, Gonzalo Camarillo, Paul Kyzivat, Haris Zisimopoulos, Joel M. Halpern, and Russ Housley for their comments.

9. IANA Considerations

9.1. Registration of the "poc-settings" Event Package

This specification registers an event package, based on the registration procedures defined in RFC 3265 [5]. The following is the information required for such a registration:

Package Name: poc-settings

Package or Template-Package: This is a package.

Published Document: RFC 4354

Person to Contact: Miguel A. Garcia-Martin, miguel.an.garcia@nokia.com

9.2. Registration of the "application/poc-settings+xml" MIME type

To: ietf-types@iana.org

Subject: Registration of MIME media type application/poc-settings+xml

MIME media type name: application

MIME subtype name: poc-settings+xml

Required parameters: (none)

Optional parameters: charset; Indicates the character encoding of enclosed XML. Default is UTF-8 [7].

Encoding considerations: Uses XML, which can employ 8-bit characters, depending on the character encoding used. See RFC 3023 [3], Section 3.2.

Security considerations: This content type is designed to carry information about current PoC user settings, which in some cases may be considered private information. Appropriate precautions should be adopted to limit disclosure of this information.

Interoperability considerations: This content type provides a common format for exchange of PoC settings information.

Published specification: RFC 4354 (this document).

Applications which use this media type: Push-to-talk over Cellular systems in compliance with the Open Mobile Alliance (OMA) PoC specifications.

Additional information: The Open Mobile Alliance publishes the Push-to-talk over Cellular specifications in the OMA web site at http://www.openmobilealliance.org

Person & email address to contact for further information: Miguel A. Garcia-Martin, miguel.an.garcia@nokia.com

Intended usage: Limited use, restricted to PoC terminals and servers.

Author/Change controller: Open Mobile Alliance (http://www.openmobilealliance.org), PoC working group.

Other information: This media type is a specialization of application/xml RFC 3023 [3], and many of the considerations described there also apply to application/poc-settings+xml.

10. References

10.1. Normative References

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

[2] Moats, R., "URN Syntax", RFC 2141, May 1997.

[3] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

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

[6] Mankin, A., Bradner, S., Mahy, R., Willis, D., Ott, J., and B. Rosen, "Change Process for the Session Initiation Protocol (SIP)", BCP 67, RFC 3427, December 2002.

[7] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

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

[9] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[10] Paoli, J., Sperberg-McQueen, C., Bray, T., and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C FirstEdition REC-xml-20001006, October 2000.

10.2. Informative References

[11] Camarillo, G. and A. Roach, "Requirements and Framework for Session Initiation Protocol (SIP) Uniform Resource Identifier (URI)-List Services", Work in Progress, April 2005.

[12] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, January 2006.

Author's Address

Miguel A. Garcia-Martin
Nokia
P.O.Box 407
NOKIA GROUP, FIN 00045
Finland

EMail: miguel.an.garcia@nokia.com

Full Copyright Statement

Intellectual Property

Acknowledgement

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