

SIPCORE Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 27, 2018

C. Holmberg
Ericsson
November 23, 2017

Push Notification with the Session Initiation Protocol (SIP)
draft-holmberg-sipcore-sip-push-03

Abstract

This document describes how push notification mechanisms can be used to wake up suspended Session Initiation Protocol (SIP) User Agents (UAs), in order to be able to receive and generate SIP requests. The document defines new SIP URI parameters, that can be used in a SIP REGISTER request to provide push notification information from the SIP User Agent (UA) to the SIP entity (realized as a SIP proxy in this document) that will send a push request to the push server in order to trigger a push notification towards the SIP UA.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 27, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

Internet-Draft

SIP PUSH

November 2017

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Conventions	4
3.	Push Resource ID (PRID)	5
4.	SIP User Agent (UA) Behavior	5
5.	SIP Proxy Behavior	6
5.1.	Push Notification Provider Information	6
5.2.	Trigger Periodic Re-registration	6
5.3.	SIP Request	6
6.	Network Address Translator (NAT) Considerations	7
7.	Grammar	7
8.	PNS Registration Requirements	8
9.	pn-prid and pn-type URI parameters for Apple Push Notification service	8
10.	pn-prid and pn-type URI parameters for Google Firebase Cloud Messaging (FCM) push notification service	9
11.	Security considerations	9
12.	IANA considerations	10
12.1.	pn-prid	10
12.2.	pn-type	10
12.3.	pn-enckey	10
12.4.	pn-encode	10
12.5.	PNS Sub-registry Establishment	11
13.	References	11
13.1.	Normative references	11
13.2.	Informative references	12
	Author's Address	12

[1.](#) Introduction

In order to save resources (e.g, battery life) some devices and operating systems require suspended Session Initiation Protocol (SIP) User Agents (UAs) [[RFC3261](#)] to be woken up using a push notification service. Typically each operating system uses a dedicated push notification service. For example, Apple iOS devices use the Apple Push Notification service (APNs).

Due to the restriction above, applications can not be woken up by

non-push notification traffic. This means that a suspended SIP UA will not be able to receive an incoming SIP request (e.g., a SIP INVITE request), or to send periodic re-registration requests.

This document describes how push notification mechanisms can be used to wake up suspended SIP UAs, in order to be able to receive and generate SIP requests. The document defines new SIP URI parameters, that can be used in a SIP REGISTER request to provide push notification information from the SIP UA to the SIP entity (realized as a SIP proxy in this document) that will send a push request to the push server in order to trigger a push notification towards the SIP UA.

When a SIP UA registers to a push service, it will receive a unique Push Resource ID (PRID) associated to that registration. The SIP UA will provide the PRID to the SIP network in a SIP REGISTER request. A SIP proxy (e.g., the SIP registrar) will store a mapping between the registered contact and the PRID.

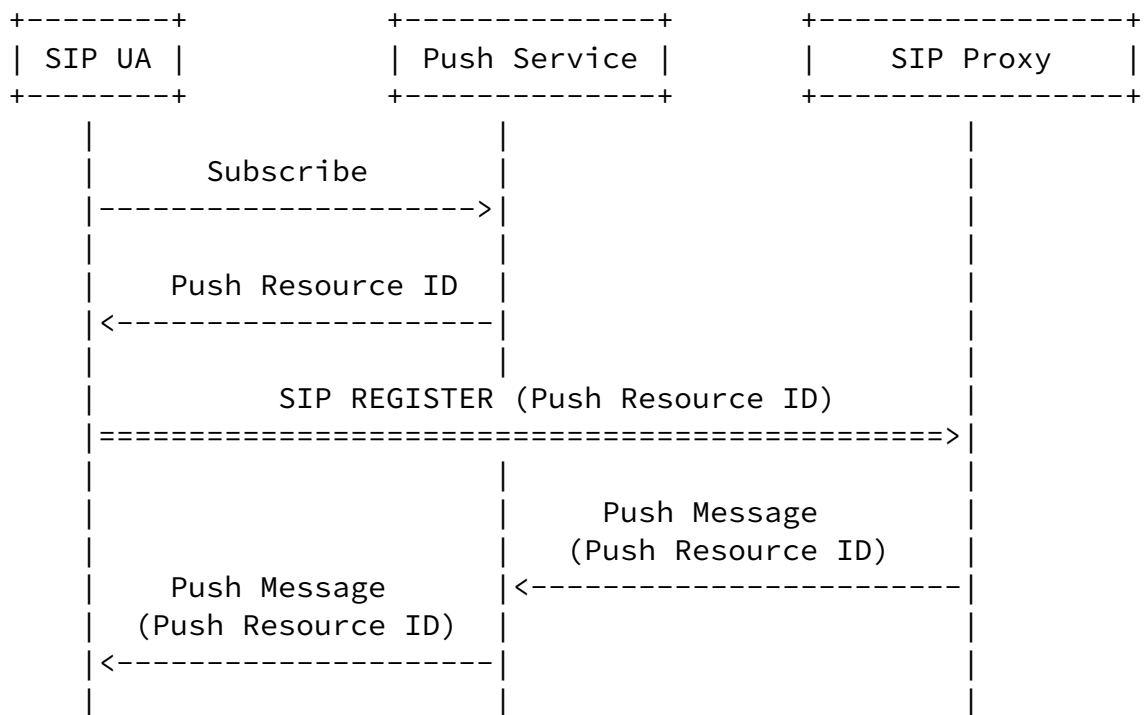
When the SIP proxy receives a SIP request for a new session, or a stand-alone SIP request, addressed towards a SIP UA, or when the SIP proxy determines that the SIP UA needs to perform a re-registration, the SIP proxy will send a push request to the push notification service used by the SIP UA, using the push resource ID associated with the registered contact of the SIP UA, in order to trigger a push notification towards the SIP UA. The SIP proxy will then forward the SIP request towards the SIP UA using normal SIP routing procedures. Once the SIP UA receives the push notification, it will be to receive the SIP request, and to generate a SIP request (e.g., a SIP REGISTER) itself.

Different push notification mechanisms exist today. Some are based on there standardized mechanism defined in [[RFC8030](#)], while others are proprietary (e.g., the Apple Push Notification service). Figure 1 shows the generic push notification architecture supported by the mechanism in this document.

Internet-Draft

SIP PUSH

November 2017



----- Push Notification API

===== SIP

```

REGISTER sip:alice@example.com SIP/2.0
Via: SIP/2.0/TCP alicemobile.example.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
To: Alice <sip:alice@example.com>
From: Alice <sip:alice@example.com>;tag=456248
  
```

```
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:alice@alicemobile.example.com;
  pn-type=acme:acme-param;
  pn-prid="ZTY4ZDJLMzODE1NmUgKi0K">
Expires: 7200
Content-Length: 0
```

Figure 1: SIP Push Notification Architecture

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 [[RFC2119](#)].

3. Push Resource ID (PRID)

When an entity registers with a Push Notification Server (PNS) it receives a unique Push Resource ID (PRID), which is a value associated with the registration.

The format of the PRID may vary depending on the PNS provider. The PRID may be part of a URI that can be used to retrieve the address and port of the PNS when sending push requests to the PNS. The PRID may also be a token value, in which case the address and port of the PNS needs to be provided using other means.

The details regarding discovery of the PNS, and the procedures for the push notification registration and maintenance are outside the scope of this document. The information needed to contact the PNS is typically pre-configured in the operating system (OS) of the device.

4. SIP User Agent (UA) Behavior

Once the SIP UA has registered with the PNS and received the PRID, and when the UA wants to receive push notifications triggered by the SIP proxy, the UA MUST send a SIP REGISTER using normal SIP

registration procedures. The UA MUST add a pn-prid URI parameter and a pn-type URI parameter to the SIP Contact header field URI of the request. The pn-prid URI parameter contains the PRID value. The pn-type contains additional, PNS-specific, information.

When the SIP UA receives a push notification, it MUST perform a SIP re-registration [[RFC3261](#)] by sending a SIP REGISTER request. If there are Network Address Translators (NATs) between the SIP UA and the SIP proxy, the SIP REGISTER request will create NAT bindings allowing incoming SIP requests to reach the SIP UA. If the SIP proxy triggered the push notification because it wants to forward a SIP request towards the SIP UA, the receipt of the SIP REGISTER request can be used by the SIP proxy as a trigger to forward the SIP request.

As long as the UA wants the SIP proxy to continue sending push requests, the UA MUST include the pn-prid and pn-type URI parameters in every re-registration SIP REGISTER request sent towards the SIP proxy. Note that, in some cases, the PNS might update the PRID value, in which case the re-registration SIP REGISTER request will contain the new value.

If the SIP UA at some point wants to stop the SIP proxy from sending push requests, the SIP UA MUST send a SIP REGISTER request without the pn-prid and pn-type URI parameters.

If the SIP UA expects to receive payload in the push notification, the SIP UA MAY add a pn-enckey and a pn-encsec Contact header field URI parameter, in order to allow encryption of the data using the mechanism in [[I-D.ietf-webpush-encryption](#)]. The pn-enckey URI parameter contains the public key, and the pn-encsec URI parameter contains the authentication secret [[I-D.ietf-webpush-encryption](#)].

NOTE: End-to-end encryption of the payload between the SIP proxy and the SIP UA cannot be used if the push notification request payload contains information that needs to be accessible by the push notification server.

[5.](#) SIP Proxy Behavior

[5.1.](#) Push Notification Provider Information

In some cases the push notification provider can be retrieved from the pn-prid URI parameter. In other cases the pn-type URI parameter is used to identify the push notification provider.

The protocol and format used for the push request depends on the push notification provider, and the details for constructing and sending the messages are outside the scope of this specification.

[5.2.](#) Trigger Periodic Re-registration

If the SIP UA needs to perform periodic re-registrations, the SIP proxy needs to have information about when those re-registrations are to be performed. The SIP proxy either needs to contain the SIP registrar functionality, or the SIP proxy needs to retrieve the information from the registrar using some other mechanism.

When the SIP proxy receives an indication that the SIP UA needs to perform a re-registration, the SIP proxy triggers a push request towards the push notification server associated with the PRID.

[5.3.](#) SIP Request

When the SIP proxy receives a SIP request for a new dialog (e.g., a SIP INVITE request) or a non-dialog SIP request (e.g., a SIP MESSAGE request) aimed for a SIP UA, if the Request-URI of the request contains a pn-prid URI parameter, the SIP proxy triggers a push request towards the push notification server associated with the PRID. After that, the SIP proxy forwards the SIP request towards the SIP UA using normal SIP procedures.

As the push notification will trigger the SIP UA to perform a re-registration, the SIP proxy can use the receipt of the SIP REGISTER request as a trigger to forward SIP request towards the SIP UA.

The SIP proxy **MUST NOT** transport the SIP request as push request payload, instead of forwarding the request using normal SIP procedures.

If the proxy is not able to contact the push notification provider, or even determine which push notification provider to contact, it SHOULD reject the SIP request.

If the SIP proxy is able to assume that the SIP UA is awake, and that the SIP UA is able to receive the SIP request, the SIP proxy MAY choose to not trigger a push notification request before trying to forward the SIP request towards the SIP UA. The SIP proxy can make such assumption e.g., if a TLS connection previously established by the SIP UA is still open.

6. Network Address Translator (NAT) Considerations

Whenever the UA receives a push notification, if the SIP UA is located behind a Network Address Translator (NAT), the UA might need to take actions in order to establish a binding in the NAT, in order for an incoming SIP request to reach the UA. By performing the re-registration the SIP UA will establish such NAT binding.

7. Grammar

The section defines new SIP URI parameters, by extending the grammar for "uri-parameter" as defined in [[RFC3261](#)]. The ABNF is as follows:

uri-parameter =/ pn-prid / pn-type / pn-encode / pn-enckey

pn-prid = "pn-prid" EQUAL pvalue
pn-type = "pn-type" EQUAL pns-provider COLON pns-param
pn-enccode = "pn-enccode" EQUAL pvalue
pn-enckey = "pn-enckey" EQUAL pvalue

pns-provider = pvalue ; Colon (":") characters MUST be escaped
pns-param = pvalue ; Colon (":") characters MUST be escaped

; pvalue as defined in [RFC 3261](#)
; EQUAL as defined in [RFC 3261](#)
; COLON as defined in [RFC 3261](#)

The format and semantics of pns-param is specific to a given pns-provider value.

8. PNS Registration Requirements

When a new value is registered to the PNS Sub-registry, a reference to a specification which describes the push notification service associated with the value is provided. That specification MUST contain the following information:

- o How the values for the pn-prid parameter is retrieved and set by the SIP UA.
- o The format of the pns-param part of the pns-type parameter, and how the value of the pns-param part is retrieved and set by the SIP UA.
- o Whether there are any restrictions regarding usage of payload encryption [[I-D.ietf-webpush-encryption](#)] with the associated push notification service.

9. pn-prid and pn-type URI parameters for Apple Push Notification service

When the Apple Push Notification service (APNs) is used, the value of the pn-type URI parameter pns-provider parameter part is "apns". The pns-param part contains the APNs App ID, which is encoded by two values, separated by a period (.): Team ID and Bundle ID. The Team ID is provided by Apple and is unique to a development team. The Bundle ID is unique to a development team, and is a string that will can match a single application or a group of applications.

Example: pn-type = apns:DEF123GHIJ.com.yourcompany.yourexampleapp

When the Apple Push Notification service (APNs) is used, pn-type URI parameter pns-prid parameter part contains the device token, which is

a unique identifier assigned by Apple to a specific app on a specific device.

Example: pn-prid = 00fc13adff78512

For more information on the APNs App ID:

<https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/AppID.html>

For more information on the APNs device token:

https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html#//apple_ref/doc/uid/TP40008194-CH8-SW13

10. pn-prid and pn-type URI parameters for Google Firebase Cloud Messaging (FCM) push notification service

When Firebase Cloud Messaging (FCM) is used, the value of the pn-type URI parameter pns-provider parameter part is "fcm". The pns-param part contains the Sender ID.

When Firebase Cloud Messaging (FCM) is used, pn-type URI parameter pns-prid parameter part contains the Registration token, which generated by the FCM SDK for each client app instance.

For more information on the Sender ID and Registration token:

<https://firebase.google.com/docs/cloud-messaging/concept-options>

11. Security considerations

In addition to the information exchanged between a device and its PNS in order to establish a push notification subscription, the mechanism in this document does not require entities to provide any additional information to the PNS.

Push notification mechanisms provide different methods to ensure that malicious user cannot trigger push notifications to a device. Users of the mechanism in this document MUST take measures to prevent push notifications from being sent to a device from a malicious user.

In case entities do want to include payload in the push notifications, this document defines the means for using end-to-end payload encryption between the entity sending the push request and

the entity receiving the associated push notification.

[12.](#) IANA considerations

This specification defines new SIP URI parameters that extend the registry created by [[RFC3969](#)]:

[12.1.](#) pn-prid

Parameter Name: pn-prid

Predefined Values: No

Reference: RFC XXXX

[12.2.](#) pn-type

Parameter Name: pn-type

Predefined Values: No

Reference: RFC XXXX

[12.3.](#) pn-enckey

Parameter Name: pn-enckey

Predefined Values: No

Reference: RFC XXXX

[12.4.](#) pn-encode

Parameter Name: pn-enccode

Predefined Values: No

Reference: RFC XXXX

Holmberg

Expires May 27, 2018

[Page 10]

Internet-Draft

SIP PUSH

November 2017

[12.5.](#) PNS Sub-registry Establishment

This section creates a new sub-registry, "PNS", under the sip-parameters registry: <http://www.iana.org/assignments/sip-parameters>.

The purpose of the sub-registry is to register SIP URI pn-type values.

This sub-registry is defined as a table that contains the following three columns:

Value: The token under registration

Description: The name of the push notification service

Document: A reference to the document defining the registration

This specification registers the following values:

Value	Description	Document
-----	-----	-----
apns	Apple Push Notification service	[RFC XXXX]
fcm	Firebase Cloud Messaging	[RFC XXXX]

13. References

13.1. Normative references

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.

Holmberg

Expires May 27, 2018

[Page 11]

Internet-Draft

SIP PUSH

November 2017

- [RFC3969] Camarillo, G., "The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)", [BCP 99](#), [RFC 3969](#), DOI 10.17487/RFC3969, December 2004, <<https://www.rfc-editor.org/info/rfc3969>>.
- [RFC8030] Thomson, M., Damaggio, E., and B. Raymor, Ed., "Generic Event Delivery Using HTTP Push", [RFC 8030](#), DOI 10.17487/RFC8030, December 2016, <<https://www.rfc-editor.org/info/rfc8030>>.

13.2. Informative references

- [I-D.ietf-webpush-encryption]
Thomson, M., "Message Encryption for Web Push", [draft-ietf-webpush-encryption-09](#) (work in progress), September 2017.

Author's Address

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Holmberg

Expires May 27, 2018

[Page 12]