SIPCORE Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 8, 2018

C. Holmberg
Ericsson

January 4, 2018

Push Notification with the Session Initiation Protocol (SIP)
draft-ietf-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 July 8, 2018.

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

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).

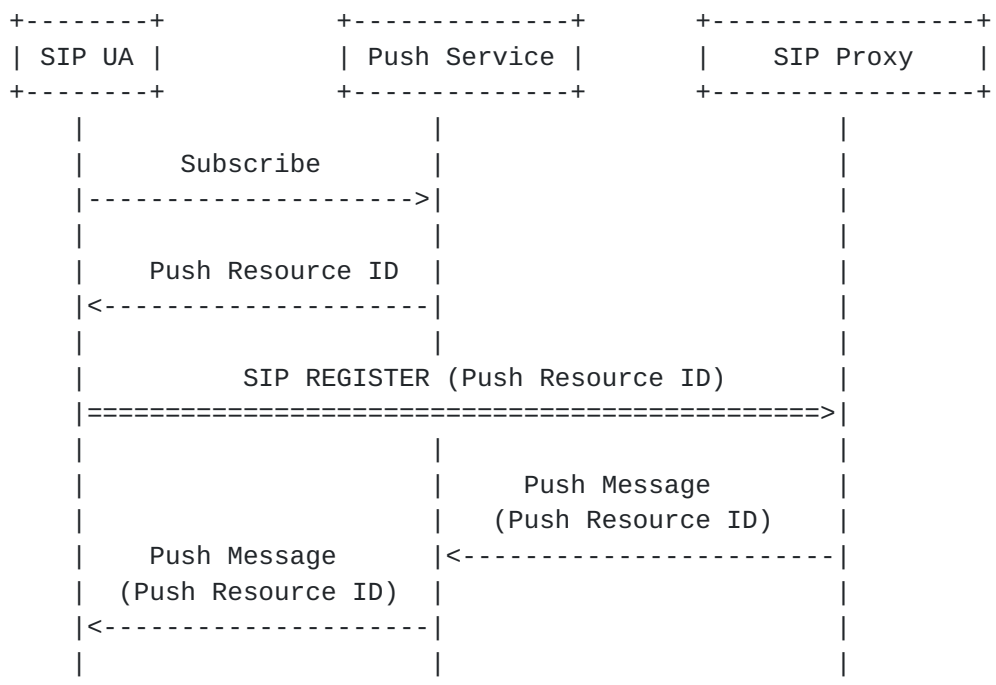
Due to the restriction above, applications can not be woken up by non-push notification traffic. This means that a suspended SIP UA will not be able to receive an incoming SIP request (e.g., a SIP INVITE request), or to send periodic re-registration requests.

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 Notification Service (PNS), 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 (or, in case the SIP proxy is also registrar, initiates) a SIP request for a new dialog, 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. 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. The proxy can use the receipt of the REGISTER request as a trigger to forward SIP request towards the UA, using normal SIP routing procedures.

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.




```
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-provider=acme;
  pn-param=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 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 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 (using the protocol and procedures associated with the PNS), 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-provider, a pn-prid and a pn-param (if required for the specific PNS provider) SIP URI parameter to the SIP Contact header field URI of the request. The pn-provider URI parameter contains the PNS provider, the pn-prid URI parameter contains the PRID value and the pn-param URI parameter contains additional PNS-specific information.

When the SIP UA receives a 200 (OK) response to the SIP REGISTER request, if the response does not contain a Feature-Caps header field with a '+sip.pns' header field parameter, or if the response contains a Feature-Caps header field with a '+sip.pns' header field parameter with a parameter value that the UA does not support, the UA cannot assume that push notifications will be triggered by a SIP proxy. The actions taken by the UA might be dependent on implementation or deployment architecture, and are outside the scope of this document.

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 REGISTER request will create NAT bindings allowing incoming SIP requests to reach the UA. If the SIP proxy triggered the push notification because it wants to forward a SIP request towards the UA, the receipt of the REGISTER request can be used by the proxy as a trigger to forward the request.

As long as the SIP UA wants the SIP proxy to continue sending push requests, the UA MUST include a pn-provider, pn-prid and a pn-param (if required for the specific PNS provider) SIP URI parameter in every re-registration SIP REGISTER request sent towards the proxy.

Note that, in some cases, the PNS might update the PRID value, in which case the pn-prid URI parameter within the re-registration 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 UA MUST send a SIP REGISTER request without the URI parameters described above, or remove the registration.

If the SIP UA expects to receive payload in the push notification, the UA MAY add a pn-enckey and a pn-encsec SIP Contact header field SIP URI parameter, in order to allow encryption of the data using the mechanism in [RFC8291]. The pn-enckey URI parameter contains the public key, and the pn-encsec URI parameter contains the authentication secret [RFC8291]. The format of such payload is outside the scope of this document.

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 PNS provider.

5. SIP Proxy Behavior

5.1. PNS Provider Information

The PNS provider is retrieved from the pn-provider SIP URI parameter.

The protocol and format used for the push request depends on the PNS 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 proxy needs to have information about when those re-registrations are to be performed. The proxy either needs to contain the SIP registrar functionality, or the 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 proxy triggers a push request towards the push notification server associated with the PRID.

Note that the re-registration needs to be triggered early enough, in order for the re-registration request to reach the registrar before the registration expires.

5.3. SIP Request

When the SIP proxy receives a SIP REGISTER request that contains a pn-provider SIP URI parameter value that the proxy does not support, or if the REGISTER request does not contain all information required for the specific PNS provider, the proxy MUST either forward the request (e.g., if the proxy is aware of another proxy that supports the PNS provider) or send a SIP 555 (Push Notification Service Not Supported) response to the REGISTER request. If the proxy sends a SIP 555 (Push Notification Service Not Supported), the proxy SHOULD insert a Feature-Caps header field with a '+sip.pns' header field parameter in the response, indicating the PNS supported by the proxy.

If the SIP proxy supports the pn-provider SIP URI parameter value, when the proxy receives (or, in case the proxy is the SIP registrar, creates) a 200 (OK) response to the REGISTER request, the proxy MUST insert a Feature-Caps header field with a '+sip.pns' header field parameter in the response, in order to inform the SIP UA that the proxy supports the PNS indicated by the pn-provider SIP URI parameter value.

When the SIP proxy receives (or, in case the proxy is the SIP registrar, creates) 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-provider, a pn-prid and a pn-param (if required for the specific PNS provider) SIP URI parameter, the proxy triggers a push request towards the PNS associated with the PRID. After that the proxy forwards the SIP request towards the 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 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 SIP proxy is able to assume that the SIP UA is awake, and that the UA is able to receive the SIP request, the proxy MAY choose to not trigger a push notification request before trying to forward the SIP request towards the UA. The mechanisms for making such assumption might be dependent on implementation or deployment architecture, and are outside the scope of this document.

If the SIP proxy is not able to contact the push notification provider, or to forward the SIP request to the SIP UA, the proxy MUST reject the SIP request.

6. Network Address Translator (NAT) Considerations

Whenever the SIP UA receives a push notification, if the 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 UA will establish such NAT binding.

7. Grammar

7.1. 555 (Push Notification Service Not Supported) Response Code

The 555 response code is added to the "Server-Error" Status-Code definition. 555 (Push Notification Service Not Supported) is used to indicate that the server did not support the push notification service specified in a 'pn-provider' SIP URI parameter.

The use of the SIP 555 response code is defined for SIP REGISTER responses. Usage with other SIP methods is undefined.

7.2. sip.pns Feature-Capability Indicator

The sip.pns feature-capability indicator is used in a SIP 555 (Push Notficiation Service Not Supported) response to indicate which push notification services the sender of the response supports.

```
sip.pns         = "<" pns-list ">"
pns-list        = pns *(COMMA pns)
pns             = pvalue

; pvalue as defined in RFC 3261
```

The value of the pns is identical to the corresponding pn-provider SIP URI parameter for the push notification service associated with the value.

7.3. SIP URI Parameters

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-provider / pn-param / pn-prid / pn-enccode /
                   pn-enckey
pn-provider     = "pn-provider" EQUAL pvalue
pn-param        = "pn-param" EQUAL pvalue
pn-prid         = "pn-prid" EQUAL pvalue
pn-enccode      = "pn-enccode" EQUAL pvalue
pn-enckey       = "pn-enckey" EQUAL pvalue

; 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 PNS associated with the value is provided. That specification MUST contain the following information:

- The value of the pn-provider SIP URI parameter.
- How the pn-prid SIP URI parameter value is retrieved and set by the SIP UA.
- How the pn-param SIP URI parameter (if required for the specific PNS provider) value is retrieved and set by the SIP UA.
- Whether there are any restrictions regarding usage of payload encryption [RFC8291] with the associated PNS.

9. pn-provider, pn-param and pn-prid URI Parameters for Apple Push Notification service

When the Apple Push Notification service (APNs) is used, the PNS-related SIP URI parameters are set as described below.

The value of the pn-provider URI parameter is "apns".

Example: pn-provider = apns

The value of the pn-param URI parameter is 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-param = DEF123GHIJ.com.yourcompany.yourexampleapp

The value of the pn-prid URI parameter is 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-provider, pn-param and pn-prid URI Parameters for Google Firebase Cloud Messaging (FCM) push notification service

When Firebase Cloud Messaging (FCM) is used, the PNS related URI parameters are set as described below.

The value of the pn-provider URI parameter is "fcm".

The value of the pn-param URI parameter is the Sender ID.

The value of the pn-prid URI parameter is the Registration token, which is 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

12.1. SIP URI Parameters

This section defines new SIP URI Parameters that extend the "SIP/SIPS URI Parameters" sub-registry [RFC3969] under the sip-parameters registry: http://www.iana.org/assignments/sip-parameters.

12.1.1. pn-provider

Parameter Name: pn-provider

Predefined Values: No

Reference: RFC XXXX

12.1.2. pn-param

Parameter Name: pn-param

Predefined Values: No

Reference: RFC XXXX

12.1.3. pn-prid

Parameter Name: pn-prid

Predefined Values: No

Reference: RFC XXXX

12.1.4. pn-enckey

Parameter Name: pn-enckey

Predefined Values: No

Reference: RFC XXXX

12.1.5. pn-enccode

Parameter Name: pn-enccode

Predefined Values: No

Reference: RFC XXXX

12.2. SIP Response Code

This section defines a new SIP response code that extends the "Response Codes" sub-registry [RFC3261] under the sip-parameters registry: http://www.iana.org/assignments/sip-parameters.

Response Code Number: 555

Default Reason Phrase: Push Notification Service Not Supported

12.3. SIP Global Feature-Capability Indicator

This section defines a new feature-capability indicator that extends the "SIP Feature-Capability Indicator Registration Tree" sub-registry [RFC6809] under the sip-parameters registry: http://www.iana.org/assignments/sip-parameters.

Name: sip.pns

Description: This feature-capability indicator, when included in a Feature-Caps header field of a REGISTER response, indicates that the server supports the SIP push mechanism. The value is a list of the push notification services supported by the server.

Reference: [RFCXXXX]

12.4. 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-provider 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 (PNS)

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. Acknowledgements

Thanks to Mickey Arnold, Paul Kyzivat, Dale Worley, Ranjit Avasarala, Martin Thomson, Mikael Klein, Susanna Sjoholm and Kari-Pekka Perttula for reading the text, and providing useful feedback.

14. References

14.1. 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>.

[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>.

[RFC6809] Holmberg, C., Sedlacek, I., and H. Kaplan, "Mechanism to Indicate Support of Features and Capabilities in the Session Initiation Protocol (SIP)", RFC 6809, DOI 10.17487/RFC6809, November 2012, <https://www.rfc-editor.org/info/rfc6809>.

[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>.

14.2. Informative References

[RFC8291] Thomson, M., "Message Encryption for Web Push", RFC 8291, DOI 10.17487/RFC8291, November 2017, <https://www.rfc-editor.org/info/rfc8291>.

Author's Address

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com