SIPCORE Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 16, 2018

C. Holmberg
Ericsson
M. Arnold
Metaswitch Networks

March 15, 2018

Push Notification with the Session Initiation Protocol (SIP)
draft-ietf-sipcore-sip-push-09

Abstract

This document describes how a Push Notification Service (PNS) can be used to awake suspended Session Initiation Protocol (SIP) User Agents (UAs), for the UA to be able to receive and send SIP requests. The document defines new SIP URI parameters and new feature-capability indicators that can be used in SIP messages to indicate support of the mechanism defined in this document, to exchange PNS information between the SIP User Agent (UA) and the SIP entity that will request push notifications towards the UA, and to trigger such push notification requests.

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 September 16, 2018.

Table of Contents

1. Introduction

In order to save resources (e.g., battery life) some devices (especially mobile devices) and operating systems will suspend applications when not used. In some cases, internal timers cannot be used to awake such applications, nor will incoming network traffic awake the application. Instead, the only way to awake the application is by using a Push Notification Service (PNS). Typically each operating system uses a dedicated PNS. For example, Apple iOS devices use the Apple Push Notification service (APNs) while Android devices use the Firebase Cloud Messaging (FCM) service.

Because of the restrictions above, Session Initiation Protocol (SIP) User Agents (UAs) [RFC3261] can not be awoken, in order to send re-registration SIP REGISTER requests and to receive incoming SIP requests, without using a PNS to awake the UA in order to perform those functions.

Also, without being able to use internal timers in order to awake applications, a UA will not be able to maintain connections e.g., using the SIP Outbound Mechanism [RFC5626], as it requires the UA to send periodic keep-alive messages.

This document describes how PNSs can be used to awake suspended UAs, to be able to send re-registration REGISTER requests and to receive incoming SIP requests. The document defines new SIP URI parameters and new feature-capability indicators [RFC6809] that can be used in SIP messages to indicate support of the mechanism defined in this document, to exchange PNS information between the UA and the SIP entity (realized as a SIP proxy in this document) that will request push notifications towards the UA, and to trigger such push notification requests.

NOTE: Even if a UA is able to awake (e.g., using internal timers) in order to send periodic re-registration REGISTER requests, it might still be useful to suspend the application between the sending of re-registration requests (as it will save battery life etc) and use a PNS to awake the UA when a SIP request addressed towards the UA arrives.

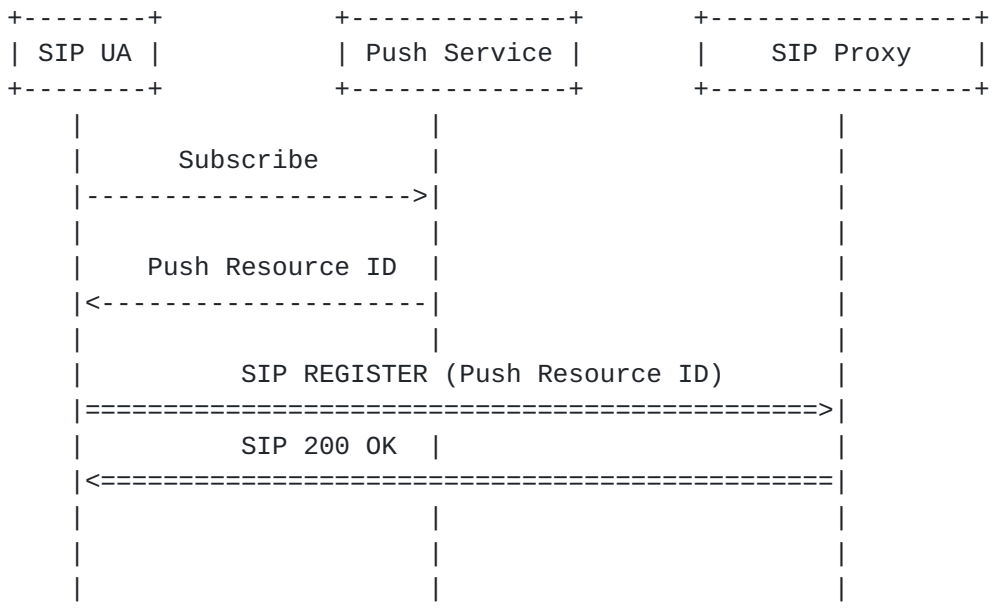
When a UA registers to a PNS, it will receive a unique Push Resource ID (PRID) associated with the push notification registration. The UA

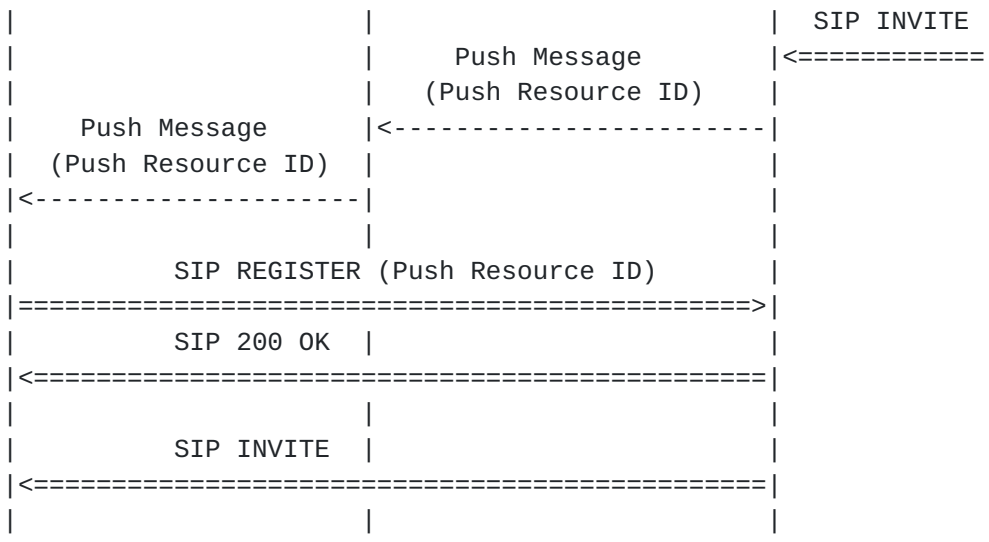
will provide the PRID to the SIP proxy that will request push notifications towards the UA in a REGISTER request.

When the proxy receives (or, if the proxy is the SIP registrar [RFC3261], initiates) a SIP request for a new dialog, or a stand-alone SIP request, addressed towards a UA, or when the proxy determines that the UA needs to send a re-registration REGISTER request, the proxy will request a push notification towards the UA, using the PNS of the UA. Once the UA receives the push notification, it will be able to send a re-registration REGISTER request and receive the incoming SIP request. The proxy will receive and forward (or, if the proxy is the registrar, process) the REGISTER request. If the push notification request was triggered by a SIP request addressed towards the UA (see above), once the REGISTER request has been accepted by the registrar, and the associated SIP 2xx response has been forwarded by the proxy towards the UA, the proxy can forward the SIP request towards the UA using normal SIP routing procedures.

Different PNSs exist today. Some are based on the 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.

Each PNS uses PNS-specific terminology and function names. The terminology in this document is meant to be PNS-independent. If the PNS is based on [RFC8030], the SIP proxy takes the role of the application server.






```
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 a SIP UA registers with a PNS it receives a unique Push Resource ID (PRID), which is a value associated with the registration that can be used to generate push notifications.

The format of the PRID may vary depending on the PNS.

The details regarding discovery of the PNS, and the procedures regarding 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 of the device.

4. SIP User Agent (UA) Behavior

Once the SIP UA has registered with the PNS, has received the PRID (using the protocol and procedures associated with the PNS), and when the UA wants to receive push notifications, the UA MUST send a SIP REGISTER request using normal SIP procedures. The UA MUST include the following SIP URI parameters in the SIP Contact header field URI of the REGISTER request: pn-provider, pn-prid and pn-param (if required for the specific PNS). The pn-provider URI parameter identifies the PNS, the pn-prid URI parameter contains the PRID value and the pn-param URI parameter contains additional PNS-specific information.

When the UA receives a 2xx response to the REGISTER request, if the response contains a Feature-Caps header field with a 'sip.pns' feature-capability indicator with a parameter value identifying the same PNS that was identified by the pn-provider URI parameter in the REGISTER request, the UA can assume that a SIP proxy will request push notifications towards the UA. In other cases, the UA MUST NOT assume that push notifications will be requested, and the actions taken by the UA might be dependent on implementation or deployment architecture, and are outside the scope of this document.

In addition, if the response contains a Feature-Caps header field with a 'sip.vapid' feature-capability indicator, the UA can use the Voluntary Application Server Identification VAPID) mechanism [RFC8292] to restrict push notifications to the proxy (assuming that the PNS supports VAPID).

When the UA receives a push notification, it MUST send a re-registration REGISTER request, using normal SIP procedures. If there are Network Address Translators (NATs) between the UA and the proxy, the REGISTER request will create NAT bindings that will allow incoming SIP requests to reach the UA. Once the UA has received a

2xx response to the REGISTER request, the UA might receive a SIP request for a new dialog (e.g., a SIP INVITE), or a stand-alone SIP request (e.g., a SIP MESSAGE), if such SIP request triggered the push notification request. Note that, depending on which transport protocol is used, the SIP request might reach the UA before the REGISTER response.

If the UA is able to generate re-registration REGISTER requests without recieving push notifications (e.g., using a timer), the UA MUST insert a 'sip.pnsreg' media feature tag [RFC3840] in the Contact header field of each REGISTER request. Then, if the response to the REGISTER request contains a 'sip.pnsreg' feature-capability indicator with an indicator value, the UA MUST send REGISTER requests prior to the registration expires. The indicator value indicates a minimum time (given in seconds), prior to the registration expires when the UA MUST send the REGISTER request. If the REGISTER response does not contain a a 'sip.pnsreg' feature-capability indicator, the UA only sends REGISTER requests when it receives a push notification.

Note that even if the UA is able to send re-registration REGISTER requests without receiving push notifications, the UA MUST still send a REGISTER request whenever it receives a push notification, according to the procedures in this section.

NOTE: As described above, in some cases the UA might be able to use a non-push mechanism (e.g., a timer) to awake and send re-registration REGISTER requests. This allows the UA to e.g., use timers for sending REGISTER requests, but to be suspended (in order to save battery resources etc) between sending the REGISTER requests and use push notification to awaken the UA to process incoming calls.

As long as the UA wants to receive push notifications (requested by the proxy), the UA MUST include a pn-provider, pn-prid and a pn-param (if required for the specific PNS provider) SIP URI parameter in each re-registration REGISTER request. Note that, in some cases, the PNS might update the PRID value, in which case the UA will include the new value in the pn-prid SIP URI parameter in the re-registration REGISTER request.

If the UA no longer wants to receive push notifications (requested by the proxy), the UA MUST send a re-registration REGISTER request without including the SIP URI parameters described above, or the UA MUST remove the registration.

If a UA's subscription to a PNS (and the associated PRID) is only valid for a limited time then the UA is responsible for retrieving new subscription details from the PNS and sending a REGISTER request with the updated pn parameters to the proxy prior to the expiry of

the existing subscription. If a UA's subscription to a PNS expires then the UA MUST send a REGISTER without the pn parameters (to tell the proxy that it no longer wants push notifications) or terminate the registration. The exact mechanisms for expiry and renewal of PRIDs will be PNS-specific and are outside the scope of this document.

For privacy and security reasons, the UA MUST NOT include the SIP URI parameters defined in this document in non-REGISTER request, to prevent the PNS information associated with the UA from reaching the remote peer. For example, the UA MUST NOT include the SIP URI parameters in the Contact header field of an INVITE request.

5. SIP Proxy Behavior

5.1. PNS Identifier

The PNS is identified by the pn-provider SIP URI parameter.

The protocol and format used for the push notification requests are PNS-specific, and the details for constructing and sending a push notification request are outside the scope of this specification.

5.2. Trigger Periodic Re-registration

In order to request push notifications towards a SIP UA that will trigger the UA to send re-registration SIP REGISTER requests, the SIP proxy MUST have information about when a registration will expire. The proxy either needs to be the SIP registrar, or the proxy needs to retrieve the information from the registrar using some other mechanism. Such mechanisms are outside the scope of this document.

When the proxy receives an indication that the UA needs to send a re-registration REGISTER request, the proxy requests a push notification towards the UA.

Note that the push notification needs to be requested early enough, in order for the associated re-registration REGISTER request to reach the SIP registrar before the registration expires. It is RECOMMENDED that the proxy requests the push notification at least 120 seconds before the registration expires.

If the UA has indicated, using the 'sip.pnsreg' media feature tag, that it is able to sending re-registration REGISTER requests without receiving push notifications, if the proxy does not receive a REGISTER request prior to 120 seconds before the registration expires, the proxy SHOULD request a push notification towards the UA, to trigger the UA to send a REGISTER request.

NOTE: In some cases the UA might be able to use a non-push mechanism (e.g., a timer) to awake and send re-registration REGISTER requests. Such REGISTER request will update the registration expiration timer, and the proxy does not need to request a push notification towards the UA in order to awake the UA. The proxy will still request a push notification towards the UA when the proxy receives a SIP request addressed towards the UA (Section 5.3.2). This allows the UA to e.g., use timers for sending re-registration REGISTER requests, but to be suspended (in order to save battery resources etc) between sending the REGISTER requests and use push notification to awaken the UA to process incoming calls.

5.3. SIP Request

5.3.1. REGISTER Request

The procedures in this section apply when the SIP proxy receives a SIP REGISTER request (initial REGISTER request for a registration, or a re-registration REGISTER request) that contains a pn-provider SIP URI parameter identifying a PNS. If the proxy receives a REGISTER request that does not contain a pn-provider SIP URI parameter, or that removes the registration, the proxy MUST NOT request push notifications towards the UA associated with the REGISTER request, or perform any other procedures in this section.

When the proxy receives a REGISTER request, if the REGISTER request contains a Feature-Caps header field with a 'sip.pns' feature-capability indicator, it indicates that an upstream proxy supports, and will request, push notifications towards the UA. The proxy MUST skip the rest of the procedures in this section, and process the REGISTER request using normal SIP procedures.

If the proxy considers the requested registration expiration interval [RFC3261] to be too short, the proxy MUST either send a 423 (Interval Too Brief) response to the REGISTER request or skip the rest of the procedures in this section, and process the REGISTER request using normal SIP procedures. Similarly, when the proxy receives a 2xx response to the REGISTER request (see below), if the proxy considers the registration expiration internal indicated by the registrar too short, the proxy MUST NOT insert a Feature-Caps header field with a 'sip.pns' feature-capability indicator in the response, and the proxy MUST NOT request push notifications associated with the registration. A registration expiration interval MUST be considered too short if the interval is smaller than the time prior to expiration that the proxy would request a push notification. The proxy MAY consider the interval too small based on its own policy so as to reduce load on the system.

Otherwise, if the pn-provider SIP URI parameter identifies a PNS that the proxy does not support, or if the REGISTER request does not contain all additional information required for the specific PNS, the proxy MUST either forward the request (e.g., if the proxy knows that a downstream proxy supports the PNS) 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) response, the proxy SHOULD insert a Feature-Caps header field with a 'sip.pns' feature-capability indicator in the response, identifying each PNS that the proxy supports.

If the proxy supports the PNS identified by the pn-provider SIP URI parameter, the proxy MUST insert a Feature-Caps header field with a 'sip.pns' feature-capability indicator, identifying the PNS, in the REGISTER request before forwarding the REGISTER request (unless the proxy is the registrar, in which case the proxy will terminate the REGISTER request). This will inform downstream proxies that the proxy supports, and will request, push notifications towards the UA.

If the proxy inserted a Feature-Caps header field with a 'sip.pns' feature-capability indicator in the REGISTER request (see above), when the proxy receives (or, in case the proxy is the SIP registrar, creates) a 2xx response to the REGISTER request, the proxy MUST insert a Feature-Caps header field with a 'sip.pns' feature-capability indicator in the response, identifying the PNS. This will inform the UA that the proxy supports, and will request, push notifications towards the UA. The proxy MUST only indicate support of the same PNS that was identified in the pn-provider SIP URI parameter in the REGISTER request.

In addition, if the proxy supports, and will use, the VAPID mechanism, the proxy MUST insert a Feature-Caps header field with a 'sip.vapid' feature-capability indicator in the response. The header field parameter contains the public key identifying the proxy [RFC8292].

In addition, if the proxy received a 'sip.pnsreg' media feature tag in the REGISTER request, the proxy SHOULD include a 'sip.pnsreg' feature-capability indicator with an indicator value bigger than 120 in the response, unless the proxy always want to request push notifications to trigger the UA to send a REGISTER request.

5.3.2. Initial Request for Dialog or Stand-Alone Request

The procedures in this section apply when the SIP proxy has indicated that it supports, and will request, push notifications towards the SIP UA.

When the proxy receives (or, in case the proxy is the registrar, creates) a SIP request for a new dialog (e.g., a SIP INVITE request) or a stand-alone SIP request (e.g., a SIP MESSAGE request) addressed towards 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 requests a push notification towards the UA, using the PRID included in the pn-prid SIP URI parameter and the PNS identified by the pn-provider SIP URI parameter.

The push notification will trigger the UA to send a re-registration REGISTER request. The proxy will process the REGISTER request and the associated response as described in Section 5.3.1. In case of a 2xx response to the REGISTER request, once the proxy has forwarded the REGISTER response towards the UA, if the contact in the REGISTER response matches the Request-URI of the SIP request to be forwarded, the proxy can also forward the SIP request towards the UA, using normal SIP procedures. If the contact of the most recent REGISTER 2xx response and Request-URI do not match, the proxy MUST reject the SIP request with a 404 (Not Found) response.

NOTE: If the proxy is not the registrar, it needs to store (or be able to retrieve) the contact of the most recent REGISTER 2xx response, to be able to compare it with the Request-URI of the request to be forwarded towards the UA.

In case of non-2xx response to the REGISTER request, the proxy MUST reject the SIP request with a 404 (Not Found) response.

If the push notification request fails (see PNS-specific documentation for details), the proxy MUST reject the SIP request with a 555 (Push Notification Service Not Supported) response.

NOTE: As described above, the reason the proxy needs to wait for the REGISTER response before forwarding the SIP request is to make sure that the REGISTER request has been accepted by the SIP registrar, and that the registered contact matches the Request-URI of the SIP request to be forwarded.

If the proxy does not receive the REGISTER request from the UA within a given time after the proxy has requested the push notification, the proxy MUST reject the request with a 404 (Not Found) response. The time value is set based on local policy.

The proxy MUST NOT include the SIP request as payload in the requested push message.

If the proxy has knowledge that the UA is awake, and that the UA is able to receive the SIP request without first sending a REGISTER request, the proxy MAY choose to not request a push notification towards the UA (and wait for the associated REGISTER request and 2xx response) before it tries to forward the SIP request towards the UA. The mechanisms for getting such knowledge might be dependent on implementation or deployment architecture, and are outside the scope of this document.

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 sending the re-registration SIP REQUEST the UA will establish such a NAT binding.

If the UA and the SIP proxy are operating in an environment where one or more NATs might be located between the UA and the proxy, when the proxy receives a REGISTER request from the UA it can store the IP address and port information from where it received the request. Then, when the proxy forwards a SIP request towards the UA, it uses that IP address and port, instead of the IP address and port associated with the Request-URI of the request. If the REGISTER request created one or more NAT bindings, this will ensure that the SIP request reaches the UA.

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 identified in a 'pn-provider' SIP URI parameter, or that the server failed to request a push notification from the push notification service.

The use of the SIP 555 response code is defined for SIP REGISTER responses, responses to SIP requests initiating dialogs and responses to stand-alone SIP requests.

7.2. sip.pns Feature-Capability Indicator

The sip.pns feature-capability indicator, when included in a Feature-Caps header field of a SIP REGISTER request or a SIP 2xx response to a REGISTER request, indicates that the entity associated with the indicator supports, and will use, the SIP push mechanism and the push

notification service identified by the indicator value. When included in a 555 (Push Notification Service Not Supported) response to a REGISTER request, the the indicator indicates that the entity associated with the indicator supports the SIP push mechanism, and the push notification service(s) identified by the indicator value. The values defined for the pn-provider SIP URI parameter are used as indicator values.

```
pns-fc          = "+sip.pns" EQUAL LDQUOT pns-list RDQUOT
pns-list        = pns *(COMMA pns)
pns             = tag-value

; tag-value as defined in RFC 3840
```

7.3. sip.vapid Feature-Capability Indicator

The sip.vapid feature-capability indicator, when included in a SIP 2xx response to a SIP REGISTER request, indicates that the entity associated with the indicator supports, and will use, the Voluntary Application Server Identification (VAPID) [RFC8292] mechanism when requesting push notifications towards the SIP UA associated with the SIP registration. The indicator value is a public key identifying the entity, that can be used by a SIP UA to restrict subscriptions to that entity.

```
vapid-fc          = "+sip.vapid" EQUAL LDQUOT vapid RDQUOT
vapid             = tag-value

; tag-value as defined in RFC 3840
```

7.4. sip.pnsreg Feature-Capability Indicator

The sip.pnsreg feature-capability indicator, when included in a SIP 2xx response to a SIP REGISTER request, indicates that the entity associated with the indicator expects to receive re-registration REGISTER requests from the SIP UA associated with the registration before the registration expires, without the entity having to request push notifications towards the SIP UA in order to trigger the REGISTER requests. The indicator value is the minimum value (given in seconds) before the registration expireation when the entity expects to receive the REGISTER request.

```
pns-fc          = "+sip.pnsreg" EQUAL LDQUOT reg RDQUOT
reg             = 1*DIGIT

; DIGIT as defined in RFC 3261
```

7.5. sip.pnsreg Media Feature Tag

The sip.pnsreg media feature tag, when included in the SIP Contact header field of a SIP REGISTER request, indciates that the SIP UA associated with the tag is able to send re-registration REGISTER requests associated with the registration without being awaken by push notifications. The media feature tag has no values.

```
pns-mt          = "+sip.pnsreg"
```

7.6. 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-provider     = "pn-provider" EQUAL pvalue
pn-param        = "pn-param" EQUAL pvalue
pn-prid         = "pn-prid" 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 pn-prid and pn-param are specific to the pn-provider value.

Parameter value chapters that are not part of pvalue needs to be escaped, as defined in RFC 3261.

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:

- o The value of the pn-provider SIP URI parameter.
- o How the pn-prid SIP URI parameter value is retrieved and set by the SIP UA.
- o How the pn-param SIP URI parameter (if required for the specific PNS provider) value is retrieved and set by the SIP UA.

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 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 Project 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 Project ID and Registration token:

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

11. pn-provider, pn-param and pn-prid URI Parameters for RFC 8030 (Generic Event Delivery Using HTTP Push)

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 "webpush".

The value of the pn-param URI parameter is the push message subscription resource URI.

The value of the pn-prid URI parameter is the push URI.

See RFC 8030 for more details:

https://tools.ietf.org/html/rfc8030

Note that encryption for web push [RFC8291] is not used, therefore parameters for message encryption are not defined in this specification. Web push permits the sending of a push message without a payload without encryption.

12. Security Considerations

Different mechanisms exist for authenticating and authorizing devices and users registering with a PNS. The mechanisms for authorizing and authenticating the users are PNS-specific, and are outside the scope of this document. In addition to the information that needs to be exchanged between a device and the PNS in order to establish a push notification subscription, the mechanism defined in this document does not require any additional information to be exchanged between the device and the PNS.

Typically, the PNS also requires the SIP proxy requesting push notifications to be authenticated and authorized by the PNS. In some cases the PNS also require the SIP application (or the SIP application developer) to be identified in order for the application to request push notifications.

If the push notification related information carried in SIP could be used by a malicious middleman to trigger push notifications towards a device, operators MUST ensure that the SIP signalling is properly secured from malicious middlemen, e.g., using encryption.

[RFC8292] defines a mechanism which allows a proxy to create a identity itself to a PNS, by signing a JWT sent to the PNS using a key pair. The public key serves as an identifier of the proxy, and can be used by devices to restrict push notifications to the proxy associated with the key.

The mechanism in this document does not require a proxy to include any payload (in addition to possible payload used for the PNS itself) when requesting push notifications.

13. IANA considerations

13.1. 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.

13.1.1. pn-provider

Parameter Name: pn-provider

Predefined Values: No

Reference: RFC XXXX

13.1.2. pn-param

Parameter Name: pn-param

Predefined Values: No

Reference: RFC XXXX

13.1.3. pn-prid

Parameter Name: pn-prid

Predefined Values: No

Reference: RFC XXXX

13.2. 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

13.3. SIP Global Feature-Capability Indicator

13.3.1. sip.pns

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 SIP REGISTER request or a SIP 2xx response to a REGISTER request, indicates that the entity associated with the indicator supports, and will use, the SIP push mechanism and the push notification service identified by the indicator value. When included in a 555 (Push Notification Service Not Supported) response to a REGISTER request, the the indicator indicates that the entity associated with the indicator supports the SIP push mechanism, and the push notification service(s) identified by the indicator value.

Reference: [RFCXXXX]

Contact: IESG (iesg@ietf.org)

13.3.2. sip.vapid

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.vapid

Description: This feature-capability indicator, when included in a SIP 2xx response to a SIP REGISTER request, indicates that the entity associated with the indicator supports, and will use, the Voluntary Application Server Identification (VAPID) mechanism when requesting push notifications towards the SIP UA associated with the SIP registration. The indicator value is a public key identifying the entity, that can be used by a SIP UA to restrict subscriptions to that entity.

Reference: [RFCXXXX]

Contact: IESG (iesg@ietf.org)

13.3.3. sip.pnsreg

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.pnsreg

Description: This feature-capability indicator, when included in a SIP 2xx response to a SIP REGISTER request, indicates that the entity associated with the indicator expects to receive re-registration REGISTER requests from the SIP UA associated with the registration before the registration expires, without the entity having to request push notifications towards the SIP UA in order to trigger the REGISTER requests. The indicator value is the minimum value (given in seconds) before the registration expireation when the entity expects to receive the REGISTER request.

Reference: [RFCXXXX]

Contact: IESG (iesg@ietf.org)

13.4. SIP Media Feature Tag

13.4.1. sip.pnsreg

This section defines a new media feature tag that extends the "SIP Media Feature Tag Registration Tree" sub-registry [RFC3840] under the Media Feature Tag registry: https://www.iana.org/assignments/media-feature-tags/media-feature-tags.xhtml.

Media feature tag name: sip.pnsreg

Summary of the media feature indicated by this feature tag: This media feature tag, when included in the SIP Contact header field of a SIP REGISTER request, indciates that the SIP UA associated with the tag is able to send re-registration REGISTER requests associated with the registration without being awaken by push notifications.

Values appropriate for use with this feature tag: none

Related standards or documents: [RFCXXXX]

Security considerations: This media feature tag does not introduce new security considerations, as it simply indicates support for a basic SIP feature. If an attacker manages to remove the media feature tag, push notifications towards the client will be requested.

Contact: IESG (iesg@ietf.org)

13.5. 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-provider values.

When a SIP URI pn-provider value is registered in the sub-registry, it needs to meet the "Expert Review" policies defined in [RFC8126].

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]
webpush	Generic Event Delivery Using HTTP Push	[RFC XXXX]

14. Acknowledgements

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

15. References

15.1. 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>.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, DOI 10.17487/RFC3840, August 2004, <https://www.rfc-editor.org/info/rfc3840>.

[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>.

[RFC8292] Thomson, M. and P. Beverloo, "Voluntary Application Server Identification (VAPID) for Web Push", RFC 8292, DOI 10.17487/RFC8292, November 2017, <https://www.rfc-editor.org/info/rfc8292>.

15.2. Informative References

[RFC5626] Jennings, C., Ed., Mahy, R., Ed., and F. Audet, Ed., "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, DOI 10.17487/RFC5626, October 2009, <https://www.rfc-editor.org/info/rfc5626>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

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

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Michael Arnold
Metaswitch Networks
100 Church Street
Enfield EN2 6BQ
United Kingdom

Email: Michael.Arnold@metaswitch.com