WAP PPG Service

Version 16-August-1999

Wireless Application Protocol
Push Proxy Gateway Service Specification

Contents

1. Scope

Wireless Application Protocol (WAP) is a result of continuous work to define an industry-wide specification for developing applications that operate over wireless communication networks. The scope for the WAP Forum is to define a set of specifications to be used by service applications. The wireless market is growing very quickly and reaching new customers and providing new services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, WAP defines a set of protocols in transport, session and application layers. For additional information on the WAP architecture, refer to "*Wireless Application Protocol Architecture Specification*" [WAP].

One part of the WAP effort is the specification of an architecture, illustrated in figure 1, which allows content to be pushed from wired networks to WAP compliant mobile devices. The scope of this document is the specification of the Push Proxy Gateway, a gateway intended to provide push connectivity between wired and wireless networks.

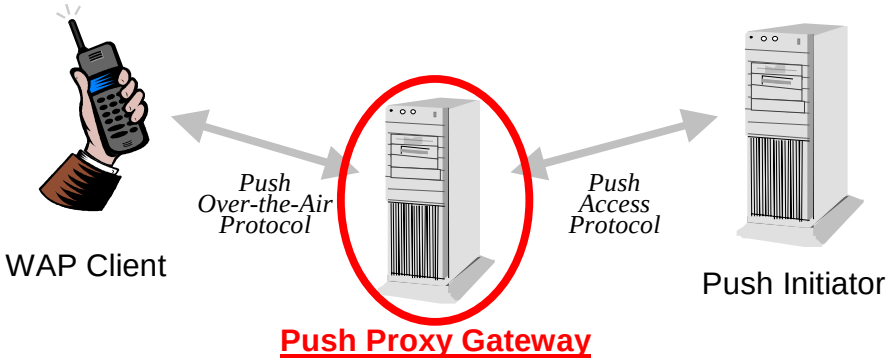

Figure 1. WAP Push Architecture

2. Document Status

This document is available online in the following formats:

- PDF format at http://www.wapforum.org/.

2.1 Copyright Notice

2.2 Errata

Known problems associated with this document are published at http://www.wapforum.org/.

2.3 Comments

Comments regarding this document can be submitted to the WAP Forum in the manner published at http://www.wapforum.org/.

3. References

3.1 Normative references

[ABNF] "Augmented BNF for Syntax Specification: ABNF", D. Crocker, Ed., P. Overell. November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt

[HTTP] "Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding, et al., June 1999. URI: http://www.ietf.org/rfc/rfc2616.txt

[ISO8601] "Data elements and interchange formats - Information interchange - Representation of dates and times", International Organization For Standardization (ISO), 1988-06-15; and

"Data elements and interchange formats - Information interchange - Representation of dates and times, Technical Corrigendum 1", International Organization For Standardization (ISO) - Technical Committee ISO/TC 154, 1991-05-01

[PushOTA] "WAP Push OTA Protocol Specification", WAP Forum, Ltd., 08-Nov-1999, URI: http://www.wapforum.org/

[PushPAP] "WAP Push Access Protocol Specification", WAP Forum, Ltd., 08-Nov-1999, URI: http://www.wapforum.org/

[PushMsg] "WAP Push Message", WAP Forum, Ltd., 16-August-1999, URI: http://www.wapforum.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. URI: http://www.ietf.org/rfc/rfc2119.txt

[RFC2387] "The MIME Multipart/related content type", E. Levinson, August 1998. URI: http://www.ietf.org/rfc/rfc2387.txt

[WAE] "Wireless Application Environment Specification", WAP Forum, 04-Nov-1999, URI: http://www.wapforum.org/

[WINA] "WAP Interim Naming Authority", WAP Forum, URL:http://www.wapforum.org/wina/

[WBXML] "Binary XML Content Format Specification", WAP Forum, 04-Nov-1999, URI: http://www.wapforum.org/

[WML] "Wireless Markup Language Specification", WAP Forum, 04-Nov-1999, URI: http://www.wapforum.org/

[XML] "Extensible Markup Language (XML)", W3C Recommendation 1998-02-10, REC-xml-19980210", T. Bray, et al, February 10, 1998. URI: http://www.w3.org/TR/REC-xml

3.2 Informative references

[PushArch] "WAP Push Architectural Overview", WAP Forum, Ltd.,08-Nov-1999, URI: http://www.wapforum.org/

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, et al., August 1998. URI: http://www.ietf.org/rfc/rfc2396.txt

[WAP] "Wireless Application Protocol Architecture Specification", WAP Forum, 30-Apr-1998. URI: http://www.wapforum.org/

4. Definitions, Abbreviations and Conventions

4.1 Definitions

The following are terms and conventions used throughout this specification.

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 by [RFC2119].

Application - A value-added data service provided to a WAP Client. The application may utilise both push and pull data transfer to deliver content

Application-Level Addressing - the ability to address push content between a particular user agent on a WAP client and push initiator on a server.

Bearer Network - a network used to carry the messages of a transport-layer protocol between physical devices. Multiple bearer networks may be used over the life of a single push session.

Client – in the context of push, a client is a device (or service) that expects to receive push content from a server. In the context of pull a client, it is a device initiates a request to a server for content or data. See also “device”.

Contact Point – address information that describes how to reach a push proxy gateway, including transport protocol addres and port of the push proxy gateway.

Content - subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent on a client. Content can both be returned in response to a user request, or being pushed directly to a client.

Content Encoding - when used as a verb, content encoding indicates the act of converting a data object from one format to another. Typically the resulting format requires less physical space than the original, is easier to process or store, and/or is encrypted. When used as a noun, content encoding specifies a particular format or encoding standard or process.

Content Format – actual representation of content.

Context – an execution space where variables, state and content are handled within a well-defined boundary.

Device – is a network entity that is capable of sending and/or receiving packets of information and has a unique device address. A device can act as either a client or a server within a given context or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server.

End-user - see “user”

Extensible Markup Language - is a World Wide Web Consortium (W3C) recommended standard for Internet mark-up languages, of which WML is one such language. XML is a restricted subset of SGML.

Multicast Message - a push message containing a single OTA client address which implicitly specifies more than OTA client address.

Push Access Protocol - a protocol used for conveying content that should be pushed to a client, and push related control information, between a Push Initiator and a Push Proxy/Gateway.

Push Framework- - the entire WAP push system. The push framework encompasses the protocols, service interfaces, and software entities that provide the means to push data to user agents in the WAP client.

Push Initiator - the entity that originates push content and submits it to the push framework for delivery to a user agent on a client.

Push OTA Protocol - a protocol used for conveying content between a Push Proxy/Gateway and a certain user agent on a client.

Push Proxy Gateway - a proxy gateway that provides push proxy services.

Push Session - A WSP session that is capable of conducting push operations.

Server - a device (or service) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client. A server may initiate a connection to a client as part of a service (push).

User - a user is a person who interacts with a user agent to view, hear, or otherwise use a rendered content. Also referred to as end-user.

User agent - a user agent (or content interpreter) is any software or device that interprets resources. This may include textual browsers, voice browsers, search engines, etc.

XML – see *Extensible Markup Language*

4.2 Abbreviations

For the purposes of this specification, the following abbreviations apply.

CPI	Capability and Preference Information
DNS	Domain Name Server
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
OTA	Over The Air
PAP	Push Access Protocol
PI	Push Initiator
PPG	Push Proxy Gateway
QOS	Quality of Service
RDF	Resource Description Framework
RFC	Request For Comments
SGML	Standard Generalized Markup Language
SI	Service Indication
SIA	Session Initiation Application
SIR	Session Initiation Request
SL	Service Loading
SSL	Secure Socket Layer
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Universal Time Co-ordinated
WAP	Wireless Application Protocol
WDP	Wireless Datagram Protocol
WSP	Wireless Session Protocol
WBXML	WAP Binary XML
WINA	WAP Interim Naming Authority
WTLS	Wireless Transport Layer Security
XML	Extensible Mark-up Language

4.3 Conventions

Within this document, `courier font` is used to identify literal names of elements, attributes, parameters, and values in referenced specifications. For example, the following table indicates that the [PushPAP] `message-state` attribute contains the `"pending"` value.

PAP Attribute	Value
message-state	“pending”

5. Introduction

This document is part of the WAP push specification suite. These specifications address the needs of a content provider seeking to “push” (i.e., send without a synchronous request) content to a client (i.e., a WAP-compliant mobile device). This is in contrast to “pull” technology, which requires a synchronous request from the client.

Push to a client is facilitated by a gateway between the wired and wireless networks. This gateway is called the Push Proxy Gateway (PPG). The purpose of this document is to specify the function of PPG.

In addition to the PPG, the WAP push architecture provides protocols to push content to the gateway and on to the client, additional functionality within WAP clients, new addressing schemes, and several standard message and content types. These are outside the scope of this document. For a complete overview, see [PushArch].

6. PPG Operations

This section defines the operations performed by a PPG. These operations include push submission processing, result notification, delivery cancellation, and Push Access Protocol (PAP) status query.

PPG operations are defined as handling each push submission (and subsequent operations related to its push message) independently from other push submissions. However, there may be limited interaction between push submissions. For example, a PPG implementation MAY support multiple delivery priorities. This could cause one message to affect the time at which another (eg, lower priority) message is processed, and consequently the ultimate success or failure of its delivery. Note that a PPG is not required to deliver push messages in any specific order.

6.1 Push Submission Processing

A Push Initiator (PI) triggers push message processing by sending the PPG a push message. Push submission processing includes four operations. The following three operations must be performed in order:

- push submission acceptance or rejection,
- over-the-air message delivery, if the message is accepted and can be delivered in accordance with PPG policies and PI requirements; and
- message delivery result notification, if the message is accepted and the push initiator has requested message delivery notification.

The fourth operation may, as determined by the PPG implementation, be performed at any time after push message acceptance:

- PAP push message response.

These four functions are described in this section.

6.1.1 Push Submission Acceptance or Rejection

Each PAP push submission received by the PPG is either accepted or rejected.

The PPG SHOULD accept a PAP push submission if it might ultimately be delivered to the OTA client. The PPG MUST reject any push submission containing a PAP `push-message` element that is not valid with respect to its document type definition (DTD). Additional criteria used to determine whether to accept or reject a `push-message` are implementation dependent.

An accepted, undelivered PAP push submission for which message handling (described in the next section) for over-the-air delivery have not been completed MUST have the following message status reportable:

PAP Attribute	**Value**
`Message-state`	`“pending”`

6.1.2 Over-the-Air Message Delivery

Over-the-Air message delivery consists of two functions:

- Message handling
- Over-the-air message transmission.

These functions are described in this section.

6.1.2.1 Message Handling

The PPG may transform the push message (as defined in [PushMsg]) entity contained within the push submission in preparation for over-the-air transmission. Typical reasons for transformation include compilations/optimisations for over-the-air efficiency, and translation of entities to a content type acceptable to the client. This section describes the transformations.

Entity and Header Transformation

A PPG MUST NOT transform the body of any entity which falls under the scope of a No-Transform cache control directive as defined in [HTTP]; otherwise, a PPG MAY translate entities in an implementation-dependent manner. Many PPG implementations will, for example, encode WML into a compact binary format for OTA transmission. The headers of all transformed entities MUST be revised as needed to correctly represent the transformed entity. All transformations MUST be in conformance with the requirements of [PushMsg].

A PPG MAY encode any header into a compact binary format for OTA transmission.

A PPG MUST process a [PushMsg] `X-Wap-Application-Id` (application ID) header as follows:

> If the header contains a [PushMsg] `absoluteURI` format application ID for which an `app-assigned-code` has been registered with [WINA], the PPG MUST remove any [PushMsg] `app-assigned-code` format application ID (if present) from the header and then substitute the registered `app-assigned-code` format application ID for the `absoluteURI` format application ID.
>
> If the header contains a [PushMsg] `absoluteURI` format application ID for which no `app-assigned-code` has been registered with [WINA], the PPG MUST use this value unless a [PushMsg] `app-assigned-code` format application ID is present. In this case (if the `app-assigned-code` format application ID is present), the `absoluteURI` format application ID must be removed.
>
> A header containing only a [PushMsg] `app-assigned-code` format application ID requires no substitutions or deletions.
>
> If the resulting header identifies a default application known to the client, the PPG MAY delete this header.

If no [PushMsg] `X-Wap-Application-ID` header is present in the push message, the PPG MUST, unless the client's default application ID is the WML user agent, add this header. If added, the application ID MUST be that of the WML user agent.

A PPG MAY remove any header which specifies a default value known to the client. This default may be specified in the over-the-air protocol, provisioned, or established using an implementation-dependent mechanism. For example, an X-Wap-Application-Id header might be removed if a client has only one push application, optimising over-the-air communications. X-Wap-Application-Id headers containing a registered value MUST NOT be sent over the air without being encoded in numeric format.

Message State

For each push submission for which errors are encountered in the steps above, or for which it is apparent that successful message delivery is not possible, message delivery MUST NOT be attempted. Note that this may cause a PAP `resultnotification-message` to be sent. Messages which fail the entity and header transformation process MUST have the following status reportable:

PAP Attribute	Value
`message-state`	`"undeliverable"`
`code`	`"transformation-failure"`

If message handling is successfully completed, an undelivered message MUST have the following status reportable:

PAP Attribute	Value
`message-state`	"pending"

6.1.2.2 Over-the-Air Transmission

The purpose of this function is to deliver messages to the OTA client. Key elements of this function are selection of confirmed or unconfirmed push, and message delivery. A PPG implementation may include tests for message expiration and cancellation, message retransmission and delivery timeout, bearer and session management.

Bearer Network Selection

If the QOS section of the PAP `push-message` element requires a specific bearer and/or network to be used, the PPG MUST use the specified bearer and/or network, or fail to deliver the message with the following messages status reportable:

PAP Attribute	Value
`message-state`	"undeliverable"
`desc`	An appropriate, implementation-dependent value
`event-time`	Time or estimated time of failure

Session Selection/Creation

The PPG may use an existing session, or take implementation-dependent action (eg, use the OTA session initiator mechanisms) to create a suitable session. If this fails (e.g., if there is a timeout) the following messages status MUST be reportable:

PAP Attribute	Value
`message-state`	"undeliverable"
`desc`	An appropriate, implementation-dependent value
`event-time`	Time or estimated time of failure

Delivery Time Constraints

If the PPG supports delivery time constraints, the PPG MUST NOT deliver the push message prior to the PAP `deliver-after-timestamp` time and MUST, if unable to deliver by the PAP `deliver-before-timestamp` time, fail with the following message status reportable:

PAP Attribute	Value
message-state	"expired"
desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of failure

Delivery

Absent any errors, the PPG MUST deliver either a confirmed (Po-ConfirmedPush) or unconfirmed (Po-Push or Po-Unit-Push) [PushOTA] push primitive. The type of primitive (confirmed or unconfirmed) depends on the PAP delivery-method attribute and implementation-dependent PPG policies. If this attribute is "unconfirmed" a PPG MUST send a Po-Push.req or Po-Unit-Push.req, or reject the push message if it is incapable. If the PAP delivery-method attribute is "confirmed", a PPG MUST send a Po-ConfirmedPush.req primitive or reject the push message if it is incapable. Otherwise, the PPG may use an implementation-dependent policy to select the type of primitive. The remaining process depends on the type of push as follows:

Unconfirmed Push

If the PPG sends a Po-Push.req or Po-Unit-Push.req primitive, the following message status MUST be reportable:

PAP Attribute	Value
message-state	"delivered"
delivery-method	"unconfirmed"
event-time	Time or estimated time of delivery

Confirmed Push

If the PPG sends a Po-ConfirmedPush.req primitive, the outcome depends as follows on whether or not the push message is acknowledged:

Success: If a Po-ConfirmedPush.res primitive indicating successful delivery to the OTA client, possibly after a PPG's implementation-dependent retries, is received, the following message status MUST be reportable:

PAP Attribute	Value
message-state	"delivered"
delivery-method	"confirmed"
event-time	Time or estimated time of delivery

Failure due to abort: If the PPG receives an OTA Po-PushAbort.ind primitive indicating an aborted push attempt the following message status MUST be reportable:

PAP Attribute	Value
`message-state`	`"aborted"`
`code`	PAP-specified representation of the abort parameter specified in [PushOTA]
`desc`	An appropriate, implementation-dependent value
`event-time`	Time or estimated time of aborted delivery attempt

Failure due to confirmation timeout: If the PPG does not receive an OTA `Po-ConfirmedPush.cnf` primitive indicating a successful push within an implementation-dependent period, the following message status MUST be reportable:

PAP Attribute	**`Value`**
`Message-state`	`"timeout"`
`desc`	An appropriate, implementation-dependent value
`event-time`	Time or estimated time of last delivery attempt

6.2 Result Notification

The PPG MUST, if requested by the push initiator during push message submission, send at least one PAP `resultnotification-message` to the push initiator or its designee.

6.2.1 Time of Result Notification

A result notification, if requested, should be sent as soon as practical after the completion (successful or unsuccessful) of the Over-the-Air message delivery process.

6.2.2 Result Notification Contents

The PAP `resultnotification-message` indicates the reportable message status, which includes the message state and other information as specified earlier in this document. The status should reflect the message just before, within the limits of practicality, sending the result notification.

6.3 PAP Status Query

This OPTIONAL function provides message status on receipt of a PAP `statusquery-message`.

The status query reply indicates the reportable message status, which includes the message state, and other information as specified earlier in this document. The status should reflect the message just before, within the limits of practicality, sending the result notification.

6.4 Delivery Cancellation

This OPTIONAL function allows delivery cancellation of a pending push message.

If the PPG supports cancellation of a push message, and the message is in a state from which delivery cancellation may be assured, the PPG MUST cancel delivery of the message as requested by a PAP `cancel-message`, and the following message status MUST be reportable:

PAP Attribute	**Value**
`message-state`	`"cancelled"`
`desc`	An appropriate, implementation-dependent value
`event-time`	Time or estimated time of cancellation

If the PPG cannot assure cancellation of the message delivery, it MUST reject the delivery cancellation.

Successful cancellation of a push message will trigger a delivery result notification, if requested during the push message submission.

7. Client Addressing

Push Initiators are able to identify clients to the PPG using a special textual address format. The PPG MUST transform these addresses into a form that can be used to deliver over the wireless network. Conversely, the PPG MUST transform network-specific addresses into the textual address format for communication to a Push Initiator. If a Push Initiator has used a particular address value to identify a client in a request sent to the PPG, this address value MUST be used when referring to this client in the corresponding response and any subsequent result notification.

A client address is composed of a client specifier and a PPG specifier. Inclusion of the PPG specifier provides a mechanism to ensure that the address is unambiguous, permitting requests to be routed through proxies. The PPG specifier does not necessarily identify a physical PPG, and is not required to be the hostname of the PPG receiving the address from a PI.

There are multiple types of client specifiers. A PPG MUST support at least one of these client specifier types:

a) User-defined identifiers

b) Device addresses

User-defined identifiers are arbitrary values that are mapped to wireless network addresses in an unspecified manner. The PPG has complete control over which bearer-level address will be used in delivering the push message to the client. The user-defined identifier MAY be expanded to several bearer-level addresses for one or more clients. In this case the PPG MUST interact with the Push Initiator in the same way as when the user-defined identifier maps to a single bearer-level address. The interpretation of user-defined identifiers is based on a mutual understanding between the Push Initiator and the PPG. This permits them to be assigned values that are useful for the application using push services. For instance, they could be e-mail addresses.

Device addresses use static values from well-known network address spaces. One example is telephone numbers in the public land mobile network (PLMN). The PPG MAY use any of the client's bearer-level addresses in delivering the push message to the client. How the PPG determines this is not specified, but may be based, for instance, on the characteristics of the bearers used by the client.

The bearer-level address may invoke a point-to-multipoint delivery in the wireless network, for example, using cell broadcast. In this case there still MUST be a single result notification, if one has been requested.

7.1 Client Address Format

The external representation of addresses processed by the PPG is defined using ABNF [ABNF]. The format is compatible with Internet e-mail addresses [RFC822]. The PPG MUST be able to parse this address format, and it MUST be able to determine whether it supports the specified address type or not.

```
wappush-address = ["/"] wappush-client-address ["/"] "@" ppg-specifier

wappush-client-address = "WAPPUSH" "=" client-specifier
ppg-specifier = dom-fragment *( "." dom-fragment )
dom-fragment = ( ALPHA / DIGIT ) *( ALPHA / DIGIT / "-" )

client-specifier = ( user-defined-identifier / device-address )
user-defined-identifier = ( escaped-value ext-qualifiers "/TYPE=USER" )
device-address = ( global-phone-number ext-qualifiers "/TYPE=PLMN" )
      / ( ipv4 ext-qualifiers "/TYPE=IPv4" )
      / ( ipv6 ext-qualifiers "/TYPE=IPv6" )
      / ( escaped-value ext-qualifiers "/TYPE=" address-type )
address-type = 1*address-char
; A network bearer address type identifier registered with [WINA]
address-char = ( ALPHA / DIGIT / "_" )

ext-qualifiers = *( "/" keyword "=" value )
; for future extensions, e.g. special well-known user-defined identifier types
keyword = 1*( DIGIT / ALPHA / "-" )
```

```
value = 1*( %x20-2E / %x30-3C / %x3E-7E )
escaped-value = 1*( safe-char )
; the actual value escaped to use only safe characters by replacing
; any unsafe-octet with its hex-escape
safe-char = ALPHA / DIGIT / "+" / "-" / "." / "%" / "_"
unsafe-octet = %x00-2A / %x2C / %x2F / %x3A-40 / %x5B-60 / %x7B-FF
hex-escape = "%" 2HEXDIG      ; value of octet as hexadecimal value

global-phone-number = "+" 1*( DIGIT , written-sep )
written-sep = ( "-" / "." )
ipv4 = 1*3DIGIT 3( "." 1*3DIGIT )   ; IPv4 address value
ipv6 = 4HEXDIG 7( ":" 4HEXDIG )     ; IPv6 address per RFC 2373
```

Each value of a `user-defined-identifier` is a sequence of arbitrary octets. They can be safely embedded in this address syntax only by escaping potentially offending values. The conversion to `escaped-value` is done by replacing each instance of `unsafe-octet` by a `hex-escape` which encodes the numeric value of the octet.

7.2 Client Address Examples

Addresses using user-defined identifiers:

```
WAPPUSH=john.doe%40wapforum.org/TYPE=USER@ppg.carrier.com
; user-defined identifier for john.doe@wapforum.org

wappush=47397547589/type=user@carrier.com
; user-defined identifier for 47397547589

WAPPUSH=47397547589/TYPE=USER@Carrier.com
; equivalent to previous one

WAPPUSH=+155519990730/TYPE=USER@ppg.carrier.com
; user-defined identifier that looks like a phone number
```

Addresses using device addresses:

```
WAPPUSH=+155519990730/TYPE=PLMN@ppg.carrier.com
; device address for a phone number of some wireless network

WAPPUSH=FEDC:BA98:7654:3210:FEDC:BA98:7654:3210/TYPE=IPv6@carrier.com
; device address for an IP v6 address

WAPPUSH=195.153.199.30/TYPE=IPv4@ppg.carrier.com
; device address for an IP v4 address
```

8. Static Conformance Requirements

This static conformance clause defines a minimum set of features that should be implemented to support Service Loading. A feature can be optional (O), mandatory (M) or conditional (C). If optional features have labels (O.<n>), support of at least one in the group of options labelled by the same numeral is required. Whether a feature applies or not is in some cases dependent on the status of another item, e.g. the PPG_OPS_013 item is only applicable, and mandatory, if CNFPUSH is supported.

8.1 Push Proxy Gateway Features

8.1.1 Predicates

These items are only used as predicates and do not state any requirements on the implementation.

Item	Functionality	Reference	Status
CNFPUSH	Confirmed push is supported	-	O

8.1.2 Operations

Item	Functionality	Reference	Status
PPG_OPS_001	Push Submission Rejection	6.1.1	M
PPG_OPS_002	Incomplete message handling reportable	6.1.1	M
PPG_OPS_003	Entity transformation under the scope of a No-Transform cache control directive	6.1.2.1	M
PPG_OPS_004	Revising headers of transformed entities	6.1.2.1	M
PPG_OPS_005	X-Wap-Application-Id header processing	6.1.2.1	M
PPG_OPS_006	Registered X-Wap-Application-Id value sent over-the-air in numeric encoded format	6.1.2.1	M
PPG_OPS_007	Reportable message states	6.1.2.1	M
PPG_OPS_008	Bearer Network Selection (QOS)	6.1.2.2	M
PPG_OPS_009	Reporting of failed Session Selection/Creation	6.1.2.2	M
PPG_OPS_010	Delivery Time Constraints	6.1.2.2	M
PPG_OPS_011	Delivery	6.1.2.2	M
PPG_OPS_012	Reportable status associated with unconfirmed push	6.1.2.2	M
PPG_OPS_013	Reportable statuses associated with confirmed push	6.1.2.2	CNFPUSH:M
PPG_OPS_014	Sending of `resultnotification-message`	6.2	M
PPG_OPS_015	PAP Status Query	6.3	O
PPG_OPS_016	Delivery Cancellation	6.4	O
PPG_OPS_017	Message cancellation request	6.4	PPG_OPS_016:M

8.1.3 Client Addressing

Item	Functionality	Reference	Status
PPG_ADD_001	Client Addressing	7	M
PPG_ADD_002	Support for user-defined identifiers	7	O.1
PPG_ADD_003	Support for device addresses	7	O.1
PPG_ADD_004	Support for client address format	7.1	M