SMPP v3.4 Implementation Guide for WAP

(for WAP June 2000 Conformance Release)

Version 14-February-2001

COPYRIGHT

Disclaimer

WHILST THE GREATEST CARE HAS BEEN TAKEN TO ENSURE THE ACCURACY OF THE INFORMATION AND DATA CONTAINED HEREIN, SMPP FORUM DOES NOT WARRANT THE ACCURACY OR SUITABILITY OF SAME FOR ANY SPECIFIC USE. SMPP FORUM EXPRESSLY DISCLAIMS ALL AND ANY LIABILITY TO ANY PERSON, WHETHER A PURCHASER OR OTHERWISE, IN RESPECT OF ANY CONSEQUENCES OF ANYTHING DONE OR OMITTED TO BE DONE BY ANY SUCH PERSON IN PARTIAL OR TOTAL RELIANCE UPON THE WHOLE OR ANY PART OF THE CONTENTS OF THIS PUBLICATION OR ANY DERIVATIVE THEREOF.

THE INFORMATION CONTAINED HEREIN IS BELIEVED TO BE ACCURATE AND RELIABLE. HOWEVER, SMPP FORUM ACCEPTS NO RESPONSIBILITY FOR IT'S USE BY ANY MEANS OR IN ANY WAY WHATSOEVER. SMPP FORUM SHALL NOT BE LIABLE FOR ANY EXPENSE, COSTS OR DAMAGE THAT MAY RESULT FROM THE INFORMATION CONTAINED HOWSOEVER ARISING IN THIS DOCUMENT OR ANY DERIVATIVE THEREOF.

THE INFORMATION CONTAINED IN THE WITHIN DOCUMENT AND ANY DERIVATIVE THEREOF IS SUBJECT TO CHANGE WITHOUT NOTICE.

THE SMPP DEVELOPERS FORUM LIMITED IS COMPANY NUMBER 309113, REGISTERED OFFICE GARDNER HOUSE, WILTON PLACE, DUBLIN 2.

Table of Contents

1 Introduction

1.1 Overview

The Short Message Peer To Peer (SMPP) protocol is an open, industry standard protocol designed to provide a flexible data communications interface for transfer of short message data between a Message Centre and an SMS application system.

A Bearer Message Centre can be a Short Message Service Centre (SMSC), a GSM Unstructured Supplementary Services Data (USSD) Server, a Broadcast Message Service Centre or other type of Message Centre. Examples of SMS application systems include a WAP gateway, an Email gateway, a voicemail system or other messaging gateways.

The Wireless Application Protocol is an evolving standard for Internet based wireless data applications. WAP uses bearers such as SMS, USSD, CBS and circuit switched data to deliver data to wireless handsets.

The WAP WDP and WCMP adaption for access of a WAP Proxy Server to Wireless Data Gateway is based on a subset of SMPP v3.4, refer to [WDP/WCMP ADAPT]. This document is intended to capture salient implementation guidelines for such a WAP Proxy Server or Push Proxy Gateway interworking with a Wireless Data Gateway/Bearer Message Centre that supports SMPP v3.4 and WAP-related messaging services.

1.2 Scope

Within the WAP architecture, the tunnel protocol over the underlying access protocol, between a WAP Proxy Server and a Wireless Data Gateway, is based on a subset of SMPP v3.4. This document is intended to provide recommendations for the implementation of this SMPP v3.4 subset in the context of WAP services.

SMPP supports various bearer services such as SMS, CBS and USSD and these bearers are also used within the context of WAP services. Though some of the scenarios described in this document are also relevant in the context of the Bearer Message Centre being a CBC or a USSD Server, this document focuses in the main on the use of SMPP as an interface to a SMSC Bearer Message Centre.

In general, there is no definite requirement for the secured storage of WAP-related messages, however it should be noted that a MC architecture may support a store & forward mechanism and may provide this for potential use by WAP push applications.

1.3 Glossary

ACK	Acknowledgement
CBC	Cell Broadcast Centre
CBS	Cell Broadcast Service
CDMA	Code Division Multiple Access
HTTP	HyperText Transfer Protocol
HLR	Home Location Register
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
ESME	External Short Message Entity.
NACK	Negative Acknowledgement
MO	Mobile Originated
MS	Mobile Subscriber
PAP	Push Access Protocol
PDU	Protocol Data Unit
POTA	Push Over The Air Protocol
PPG	Push Proxy Gateway
SIA	Session Initiation Application
SME	Short Message Entity
SMPP	Short Message Peer to Peer Protocol
SMS	Short Message Service
SMSC	Short Message Service Centre
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
URI	Uniform Resource Identifier
USSD	Unstructured Supplementary Services Data
WAP	Wireless Application Protocol
WBXML	WAP Binary XML
WCMP	Wireless Control Message Protocol
WDP	Wireless Datagram Protocol
WML	Wireless Markup Language
WSP	Wireless Session Protocol
WTLS	Wireless Transport Layer Security
WTP	Wireless Transaction Protocol
WWW	World Wide Web
XML	Extensible Markup Language

1.4 Reference

Ref.	Document Title	Document Number	Version Number
[PAP]	WAP Push Access Protocol	Wap Forum http://www.wapforum.org	Version 08-Nov-1999
[POTA]	WAP Push OTA Protocol	WAP-189-PushOTA Wap Forum http://www.wapforum.org	Version 17-Feb-2000
[SMPP]	Short Message Peer to Peer Protocol Specification v3.4	SMPP Developer Forum http:/www.smpp.org	Version 12-Oct-1999 Issue 1.2
[WBXML]	WAP Binary XML Content Format	WAP-192-WBXML-20000515 Version 1.3 Wap Forum http://www.wapforum.org	Version 15-May-2000
[WCMP]	Wireless Application Protocol Wireless Control Message Protocol Specification	WAP-202-WCMP Wap Forum http://www.wapforum.org	Version 19-Feb-2000
[WDP]	Wireless Application Protocol Wireless Datagram Protocol Specification	WAP-200-WDP Wap Forum http://www.wapforum.org	Version 19-Feb-2000
[WDP/WCMP ADAPT]	WDP and WCMP Wireless Data Gateway Adaption	WAP Forum http://www.wapforum.org	Version 05-Nov-1999
[WML]	Wireless Application Protocol Wireless Markup Language Specification	WAP-191-WML WAP Forum http://www.wapforum.org	19-Feb-2000 Version 1.3
[WSP]	Wireless Application Protocol Wireless Session Protocol Specification	WAP-203-WSP WAP Forum http://www.wapforum.org	Version 4-May-2000
[WTLS]	Wireless Application Protocol Wireless Transport Layer Security Specification	WAP-199-WTLS WAP Forum http://www.wapforum.org	Version 18-Feb-2000
[WTP]	Wireless Application Protocol Wireless Transaction Protocol Specification	WAP-201-WTP WAP Forum http://www.wapforum.org	Version 19-Feb-2000

2 WAP Services

The Wireless Session Protocol currently supports services which are suitable and intended for Mobile Station based browsing applications. WSP provides both push and “WSP Get” data transfer.

The “WSP Get” mechanism which is supported in WAP 1.1 is based on the request/response mechanism in HTTP/1.1 with the MS subscriber requesting data.

The Push mechanism which includes both confirmed and unconfirmed data push is a key feature of WAP 1.2. In this case the MS subscriber is not restricted to only receiving information which he has specifically requested, but can receive unsolicited data from the Server.

The IANA assigned standard IP port for SMPP is 2775. The SMPP Forum strongly recommends that SMPP servers and clients use this port.

Note 1: In the following scenarios the normal SMPP session initiation, data transfer and session termination are illustrated for completeness in the message flows. It is however recommended good practice that the WAP Proxy Server or Push Proxy Gateway remains permanently bound to the Message Centre.

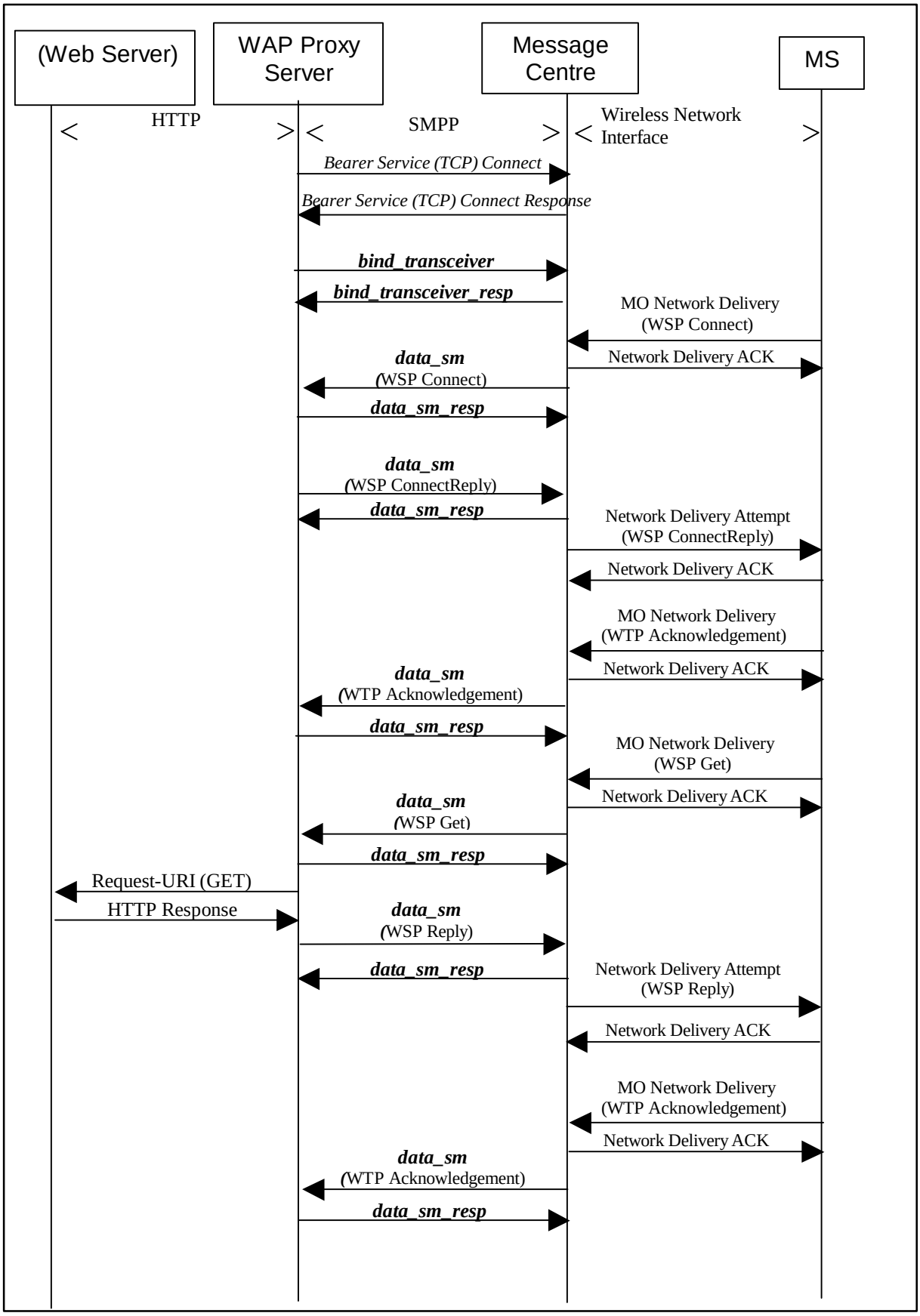
Note 2: An SMPP session between a Bearer Message Centre (SMSC) and a WAP Proxy Server/Push Proxy Gateway is initiated by the a WAP Proxy Server/Push Proxy Gateway first establishing a network connection (e.g. TCP/IP) with the Bearer Message Centre (SMSC) and then issuing an SMPP Bind request to open an SMPP session. A WAP Proxy Server/Push Proxy Gateway wishing to submit and receive messages may establish a network connection (e.g. TCP/IP) and a Transceiver SMPP session, or alternatively may establish two network connections (e.g. TCP/IP) and two SMPP sessions (Transmitter and Receiver) to the Bearer Message Centre (SMSC).

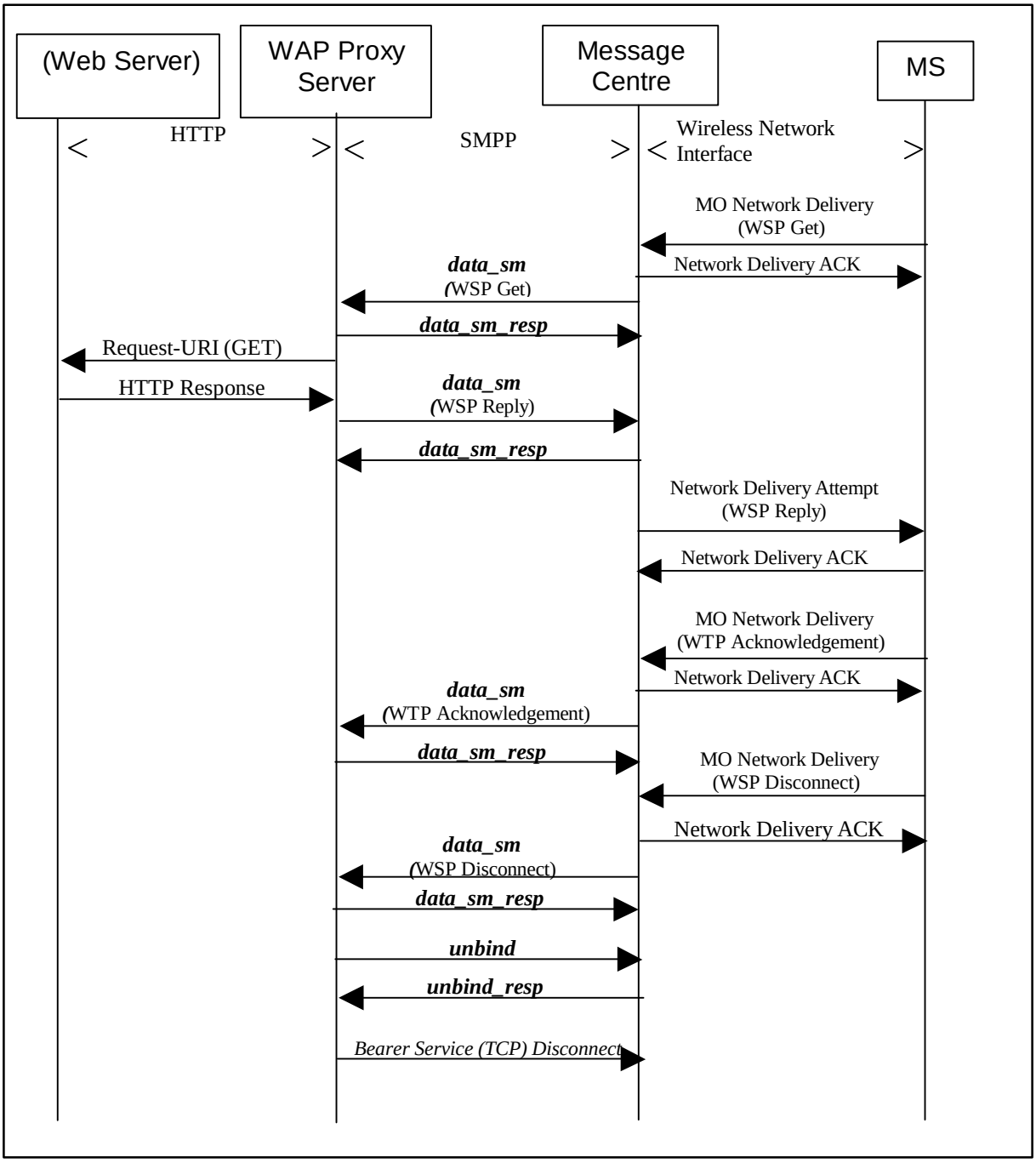
2.1 WSP Get Service – Normal Scenarios

2.1.1 Scenario 1: Successful Information Pull from an Application Node – Response encapsulated in a Single SMPP Data Transaction – Transceiver session

Use Case: An MS Subscriber Requests the Current Trading Price of a Stock.

The MS subscriber requests the service by issuing a “WSP Get” PDU (e.g. a URI request) within a WAP session using a SMSC as the Bearer Message Centre. The request is routed through the WAP Proxy Server to the Web Server.





- The WAP Proxy Server issues a Bearer Service (e.g. TCP/IP) connect to the Bearer Message Centre (SMSC) which is acknowledged and a network connection has been successfully established.

- The WAP Proxy Server binds as a transceiver using the SMPP ***bind_transceiver*** PDU. The SMSC acknowledges with a ***bind_transceiver _resp*** and SMPP session establishment is successfully complete. It is recommended good practice that the WAP Proxy Server is permanently bound in.

- The MS issues a “WSP Connect”, requesting a WSP session with a WAP Proxy Server.

- The SMSC forwards the “WSP Connect” to the WAP Proxy Server in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request and responds by generating a “WSP ConnectReply”, setting datagram mode in the SMPP ***data_sm*** request with the *source_por*t and *destination_port* optional parameters populated.

- The SMSC forwards the “WSP ConnectReply” to the MS.

- Since the “WSP Connect” is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP ConnectReply” to the WAP Proxy Server and WSP session establishment has been successfully completed.

- Once a WSP session has been successfully established, the MS can issue a “WSP Get” PDU (for example a URI request).

- The bearer Message Centre (SMSC) forwards the “WSP Get” PDU to the ESME (WAP Proxy Server) via a ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request.

- The WAP Proxy Server retrieves the information pertaining to the URI request. This normally involves the WAP Proxy Server forwarding the request to a WEB Server. To do this the WAP Proxy Server must be capable of translating requests from the WAP protocol stack (WSP, WTP, WTLS and WDP) to the WWW protocol stack (HTTP and TCP/IP). In addition the data content may have to be translated from MML or rather the compact binary format (WBXML) understood by the client MS to WWW content (HTML).

- The WEB server may provide the response data as WWW content (HTML) or WAP content (WML) and thus the WAP Proxy Server may be required to translate the WWW content to WAP content. The Proxy must then encode the responses from the WEB server into the compact binary format (WBXML) understood by the client (MS).

- The Wap Proxy Server submits the “WSP reply” which contains the data in response to the “WSP Get” to the SMSC, setting datagram mode on the ***data_sm*** request and populating the optional parameters *source_port* and *destination_port*.

- The Bearer Message Centre (SMSC) acknowledges the ***data_sm*** request and forwards the “WSP Reply” to the MS.

- Since the “WSP Get” is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP Reply” to the WAP Proxy Server.

- An MS can initiate multiple “WSP Get(s)” on a WSP session. The “WSP Get/Reply/Acknowledgement” sequence is repeated in this example message flow. When it wishes to, the client initiates a “WSP Disconnect” to indicate that it wishes to terminate the WSP session.

- The SMSC forwards the “WSP Disconnect” in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The Wap Proxy Server acknowledges the ***data_sm*** request and the WSP session is terminated.

- In the above message flow the WAP Proxy Server issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC). The Bearer Message Centre responds with an ***unbind_resp***. The SMPP ***unbind*** sequence is shown for completeness as the WAP Proxy Server should remain permanently bound to the Bearer Message Centre.

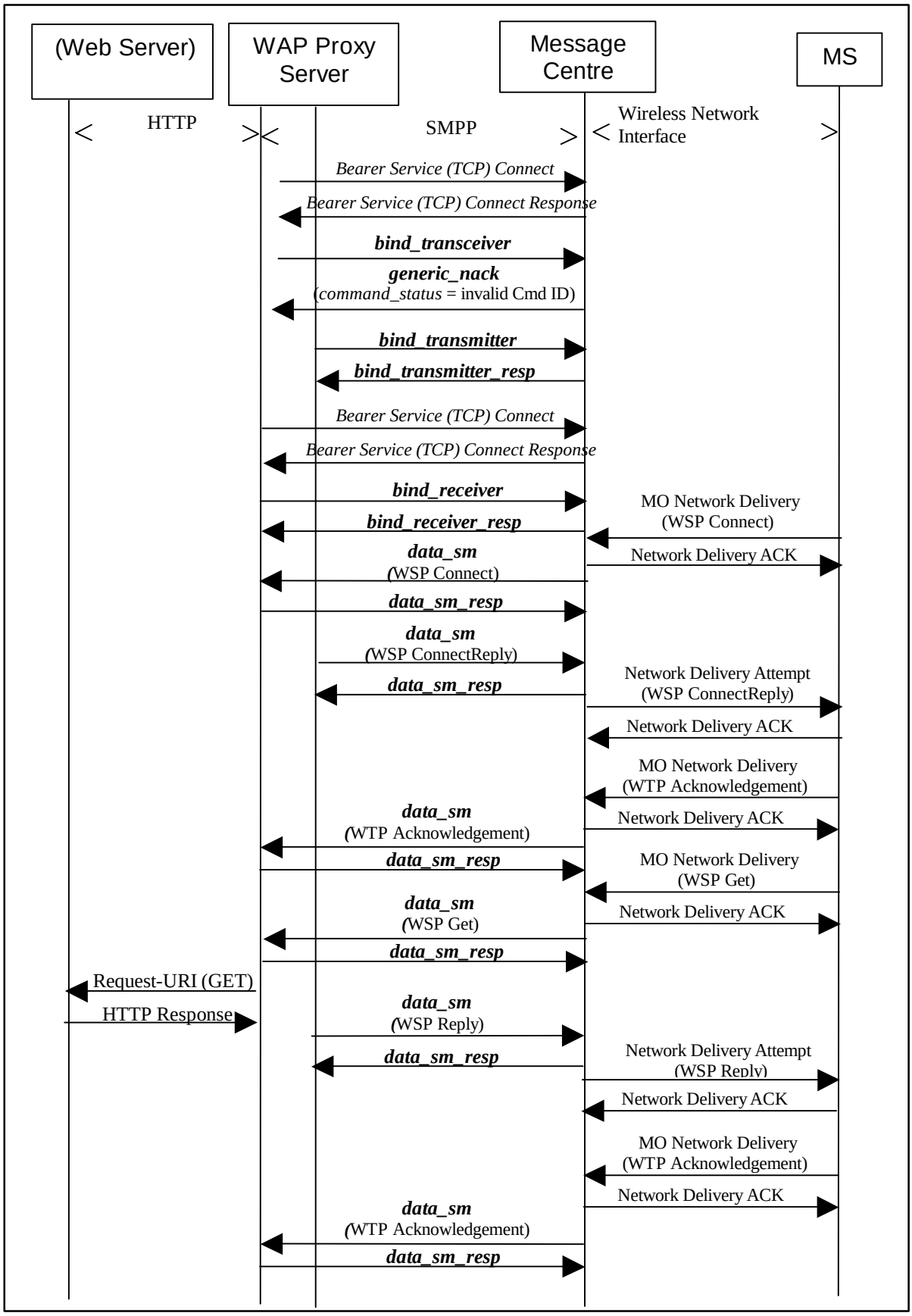
- Either Peer can initiate a network (e.g. TCP/IP) disconnect for the network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow the WAP Proxy Server issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC).

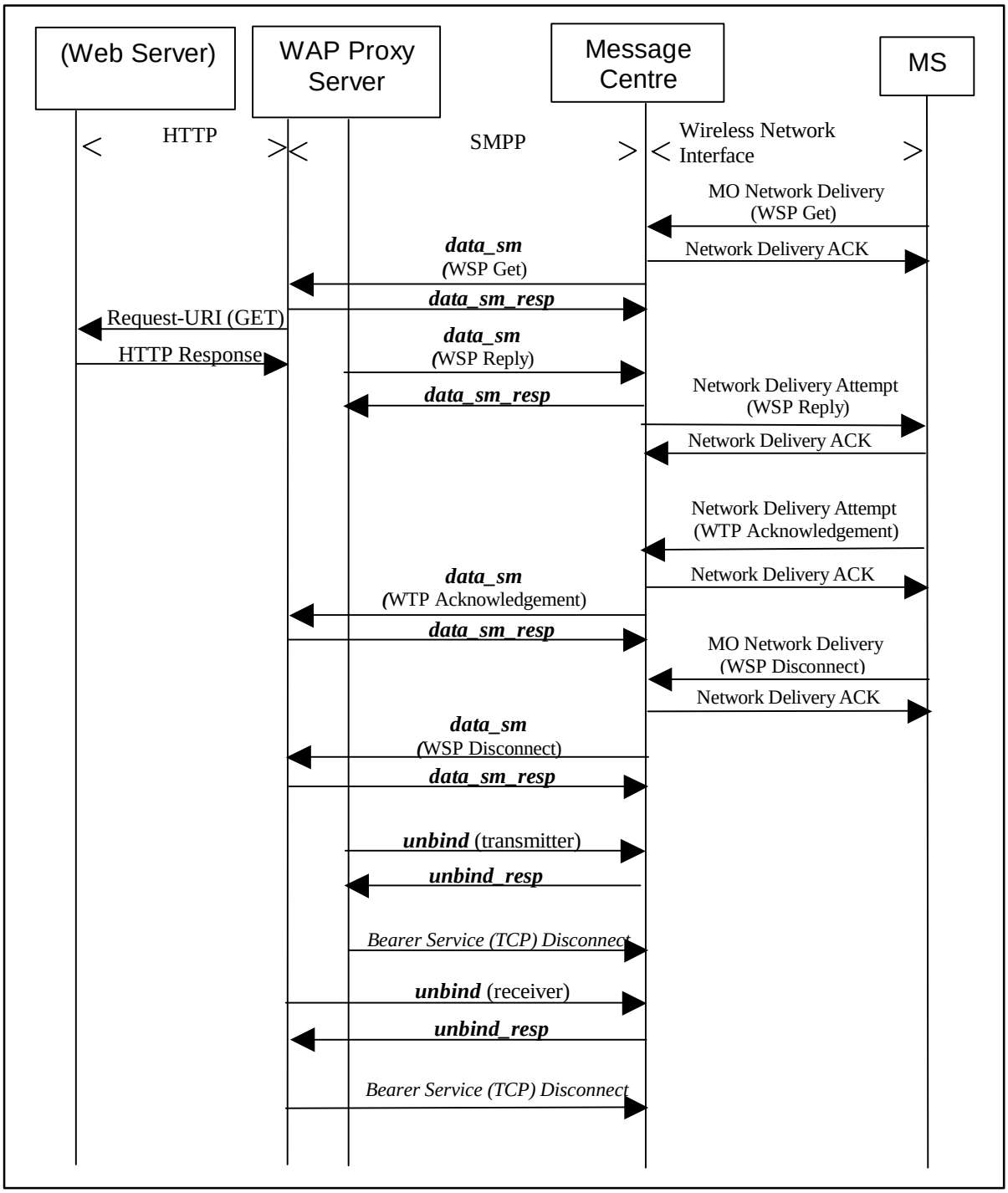
2.1.2 Scenario 2: Successful Information Pull from an Application Node – Response encapsulated in a Single SMPP Data Transaction – Transmitter and Receiver session

Use Case: An MS Subscriber Requests the Current Trading Price of a Stock.

The MS subscriber requests the service by issuing a “WSP Get” PDU (e.g. a URI request) within a WAP session using a SMSC as the Bearer Message Centre. The request is routed through the WAP Proxy Server to the Web Server.

In this scenario the Bearer Message Centre (SMSC) rejects the attempt by the WAP Proxy Server to establish a SMPP transceiver session indicating that the command is invalid. The WAP Proxy Server responds by establishing two SMPP sessions (Transmitter and Receiver) with two attendant network connections (e.g. TCP/IP), with the Bearer Message Centre.





- The WAP Proxy Server issues a Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) which is acknowledged and a network connection has been successfully established.

- The WAP Proxy Server attempts to bind as a transceiver using the SMPP ***bind_transceiver*** PDU.

- The Bearer Message Centre (SMSC) rejects the ***bind_transceiver*** request with a ***generic_nack*** setting *command_status* to "*Invalid Command ID*".

- Since the WAP Proxy Server wishes to submit and receive messages it must establish two SMPP sessions (Transmitter and Receiver) with two attendant network connections (e.g. TCP/IP).

- One network connection is already successfully established and thus the WAP Proxy Server issues a second Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) which is acknowledged, and thus a second network connection has been successfully established.

- The WAP Proxy Server binds as a transmitter using the SMPP ***bind_transmitter*** PDU and as a receiver using the SMPP ***bind_receiver*** PDU. The SMSC acknowledges with a ***bind_transmitter _resp*** and ***bind_receiver_resp*** respectively and SMPP session establishment is successfully complete. It is recommended good practice that the WAP Proxy Server is permanently bound in.

- The MS issues a "WSP Connect", requesting a WSP session with a WAP Proxy Server.

- The SMSC forwards the "WSP Connect" to the WAP Proxy Server in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request and responds by generating a "WSP ConnectReply", setting datagram mode in the SMPP ***data_sm*** request with the *source_por*t and *destination_port* optional parameters populated.

- The SMSC forwards the "WSP ConnectReply" to the MS.

- Since the "WSP Connect" is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a "WTP Acknowledgement" to acknowledge receipt of the "WSP ConnectReply" to the WAP Proxy Server and WSP session establishment has been successfully completed.

- Once a WSP session has been successfully established, the MS can issue a "WSP Get" PDU (for example a URI request).

- The bearer Message Centre (SMSC) forwards the "WSP Get" PDU to the ESME (WAP Proxy Server) via a ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request.

- The WAP Proxy Server retrieves the information pertaining to the URI request. This normally involves the WAP Proxy Server forwarding the request to a WEB Server. To do this the WAP Proxy Server must be capable of translating requests from the WAP protocol stack (WSP, WTP, WTLS and WDP) to the WWW protocol stack (HTTP and TCP/IP). In addition the data content may have to be translated from MML or rather the compact binary format (WBXML) understood by the client MS to WWW content (HTML).

- The WEB server may provide the response data as WWW content (HTML) or WAP content (WML) and thus the WAP Proxy Server may be required to translate the WWW content to WAP content. The Proxy must then encode the responses from the WEB server into the compact binary format (WBXML) understood by the client (MS).

- The Wap Proxy Server submits the “WSP reply” which contains the data in response to the “WSP Get” to the SMSC, setting datagram mode on the ***data_sm*** request and populating the optional parameters *source_port* and *destination_port*.

- The Bearer Message Centre (SMSC) acknowledges the ***data_sm*** request and forwards the “WSP Reply” to the MS.

- Since the “WSP Get” is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP Reply” to the WAP Proxy Server.

- An MS can initiate multiple “WSP Get(s)” on a WSP session. The “WSP Get/Reply/Acknowledgement” sequence is repeated in this example message flow. When it wishes to, the client initiates a “WSP Disconnect” to indicate that it wishes to terminate the WSP session.

- The SMSC forwards the “WSP Disconnect” in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The Wap Proxy Server acknowledges the ***data_sm*** request and the WSP session is terminated.

- In the above message flow the WAP Proxy Server issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC). The Bearer Message Centre responds with an ***unbind_resp***. The SMPP ***unbind*** sequence is shown for completeness as the WAP Proxy Server should remain permanently bound to the Bearer Message Centre.

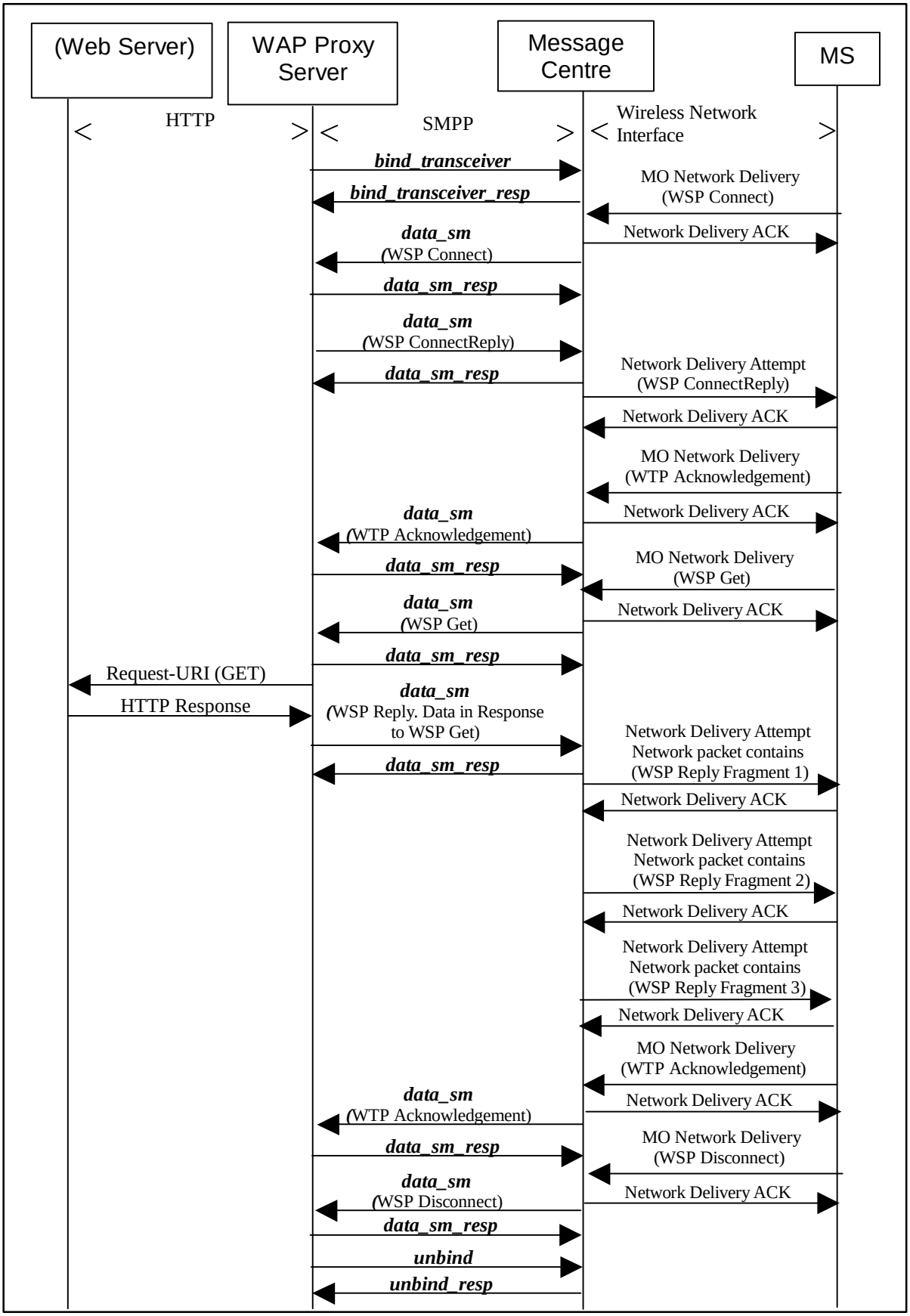
- Either Peer can initiate a network (e.g. TCP/IP) disconnect for each network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow the WAP Proxy Server issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC) for each network connection.

2.1.3 Scenario 3: Successful Information Pull from an Application Node – Long Response encapsulated in a Single SMPP Data Transaction - Segmentation on Message Centre

Use Case: An MS Subscriber Requests a list of downtown theatres

The MS subscriber requests the service by issuing a “WSP Get” PDU (e.g. a URI request) within a WAP session using a SMSC as the Bearer Message Centre. The request is routed through the WAP Proxy Server to the Web Server

The Message Centre fragments the response to the ”WSP Get” and forwards the response as Multiple Transactions across the Wireless Network Interface.



- The WAP Proxy Server issues a Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) which is acknowledged and a network connection has been successfully established.

- The WAP Proxy Server binds as a transceiver using the SMPP ***bind_transceiver*** PDU. The SMSC acknowledges with a ***bind_transceiver _resp*** and SMPP session establishment is successfully complete. It is recommended good practice that the WAP Proxy Server is permanently bound in.

- The MS issues a "WSP Connect" requesting a WSP session with a WAP Proxy Server.

- The SMSC forwards the "WSP Connect" to the WAP Proxy Server in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request and responds by generating a "WSP ConnectReply", setting datagram mode in the SMPP ***data_sm*** request with the *source_por*t and *destination_port* optional parameters populated.

- The SMSC forwards the "WSP ConnectReply" to the MS.

- Since the "WSP Connect" is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a "WTP Acknowledgement" to acknowledge receipt of the "WSP ConnectReply" to the WAP Proxy Server and WSP session establishment has been successfully completed.

- Once a WSP session has been successfully established, the MS can issue a "WSP Get" PDU (for example a URI request).

- The Bearer Message Centre (SMSC) forwards the "WSP Get" PDU to the ESME (WAP Proxy Server) via a ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request.

- The WAP Proxy Server retrieves the information pertaining to the URI request. This normally involves the WAP Proxy Server forwarding the request to a WEB Server. To do this the WAP Proxy Server must be capable of translating requests from the WAP protocol stack (WSP, WTP, WTLS and WDP) to the WWW protocol stack (HTTP and TCP/IP). In addition the data content may have to be translated from MML or rather the compact binary format (WBXML) understood by the client MS to WWW content (HTML).

- The WEB server may provide the response data as WWW content (HTML) or WAP content (WML) and thus the WAP Proxy Server may be required to translate the WWW content to WAP content. The Proxy must then encode the responses from the WEB server into the compact binary format (WBXML) understood by the client (MS).

- The Wap Proxy Server submits the "WSP reply" (which contains the data in response to the "WSP Get") to the SMSC, setting datagram mode and populating the *source_port* and *destination_port* optional parameters.

- The Bearer Message Centre (SMSC) acknowledges the ***data_sm*** request with a ***data_sm_resp*** and forwards the "WSP Reply" to the MS. In this instance the message length is greater than can be accommodated in a single delivery packet on the Mobile Network and thus the SMSC performs segmentation and forwards the data as three seperate packets.

- Since the “WSP Get” is a class 2 WTP transaction (see [WTP] and [WSP]), following the successful receipt (of all fragments in this scenario) of the “WSP Reply” the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP Reply” to the WAP Proxy Server.

- An MS can initiate multiple “WSP Get(s)” on a WSP session. When it wishes to, the client initiates a “WSP Disconnect” to indicate that it wishes to terminate the WSP session.

- The SMSC forwards the “WSP Disconnect” in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The Wap Proxy Server acknowledges the ***data_sm*** request and the WSP session is terminated.
- In the above message flow the WAP Proxy Server issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC). The Bearer Message Centre responds with an ***unbind_resp***. The SMPP ***unbind*** sequence is shown for completeness as the WAP Proxy Server should remain permanently bound to the Bearer Message Centre.

- Either Peer can initiate a network (e.g. TCP/IP) disconnect for the network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow the WAP Proxy Server issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC).

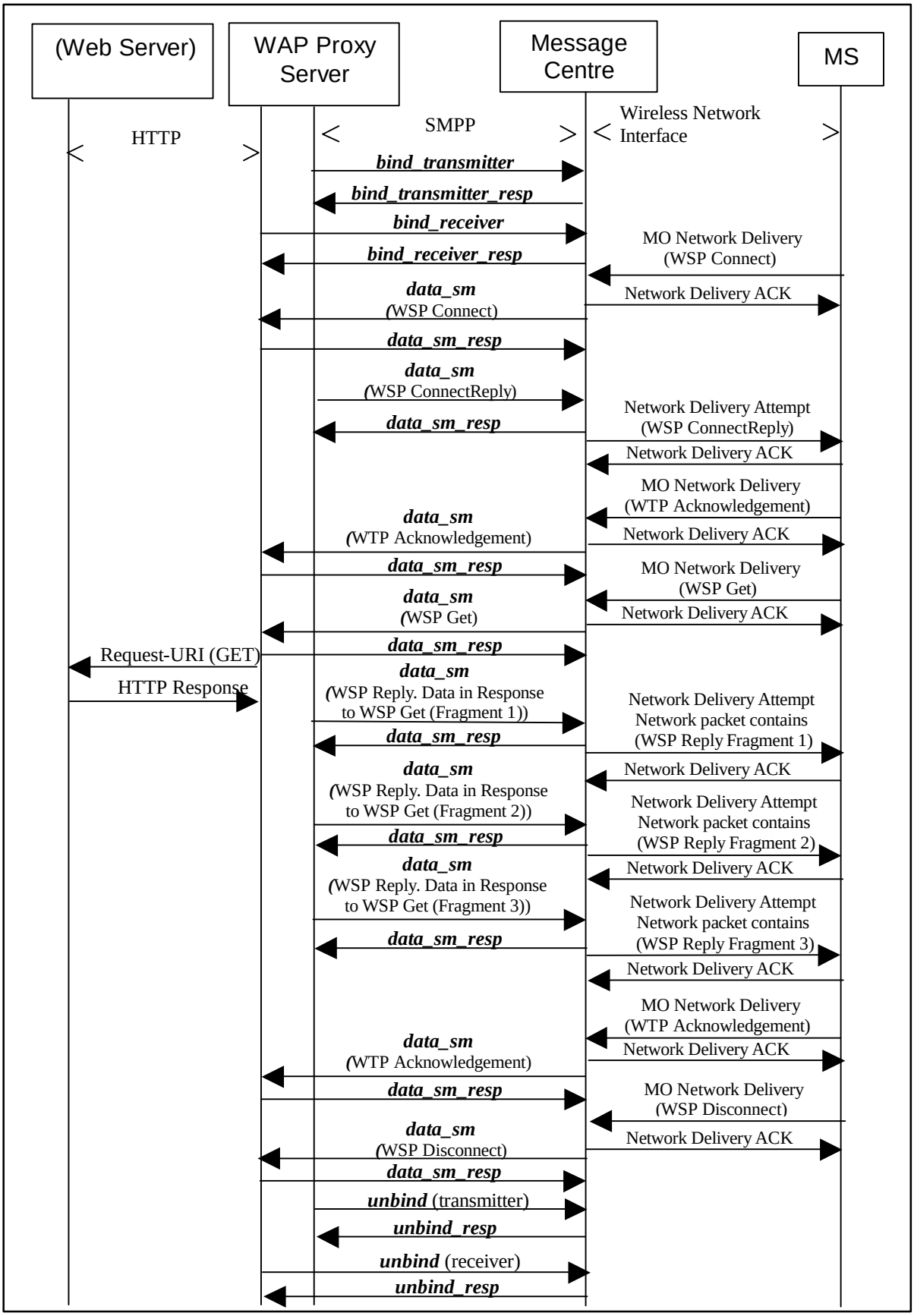
Note: It is recommended that Scenarios where the WAP Proxy Server/Push Proxy Gateway and the Bearer Message Centre (SMSC) would both concurrently perform fragmentation should be avoided.

2.1.4 Scenario 4: Successful Information Pull from an Application Node – Long Response encapsulated in Multiple SMPP Data Transactions – WDP Segmentation on WAP Proxy Server.

Use Case: An MS Subscriber Requests a list of downtown theatres

The MS subscriber requests the service by issuing “WSP Get” PDU (e.g. a URI request) within a WAP session using a SMSC as the Bearer Message Centre. The request is routed through the WAP Proxy Server to the Web Server.

The WAP Proxy Server fragments the response to the Pull Request and submits it as Multiple Transactions to the Message Centre.



- In this message flow the WAP Proxy Server wishes to establishes two SMPP sessions (Transmitter and Receiver) and thus must establish two network (e.g. TCP/IP) connections. The WAP Proxy Server issues a Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) for each of the network connections it wishes to establish. The Bearer Message Centre acknowledges each (TCP) connect request and two network connections are successfully established.

- The WAP Proxy Server binds as a transmitter using the SMPP ***bind_transmitter*** PDU and as a receiver using the SMPP ***bind_receiver*** PDU. The SMSC acknowledges with a ***bind_transmitter _resp*** and ***bind_receiver_resp*** respectively and SMPP session establishment is successfully complete. It is recommended good practice that the WAP Proxy Server is permanently bound in.

- The MS issues a “WSP Connect” requesting a WSP session with a WAP Proxy Server.

- The SMSC forwards the “WSP Connect” to the WAP Proxy Server in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request and responds by generating a “WSP ConnectReply”, setting datagram mode in the SMPP ***data_sm*** request with the *source_por*t and *destination_port* optional parameters populated.

- The SMSC forwards the “WSP ConnectReply” to the MS.

- Since the “WSP Connect” is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP ConnectReply” to the WAP Proxy Server and WSP session establishment has been successfully completed.

- Once a WSP session has been successfully established, the MS can issue a “WSP Get” PDU (for example a URI request).

- The Bearer Message Centre (SMSC) forwards the “WSP Get” PDU to the ESME (WAP Proxy Server) via a ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request.

- The WAP Proxy Server retrieves the information pertaining to the URI request. This normally involves the WAP Proxy Server forwarding the request to a WEB Server. To do this the WAP Proxy Server must be capable of translating requests from the WAP protocol stack (WSP, WTP, WTLS and WDP) to the WWW protocol stack (HTTP and TCP/IP). In addition the data content may have to be translated from MML or rather the compact binary format (WBXML) understood by the client MS to WWW content (HTML).

- The WEB server may provide the response data as WWW content (HTML) or WAP content (WML) and thus the WAP Proxy Server may be required to translate the WWW content to WAP content. The Proxy must then encode the responses from the WEB server into the compact binary format (WBXML) understood by the client (MS).

- The Wap Proxy Server fragments the response data from the WEB server as the data is larger than what the network can support. Having performed the appropriate fragmentation and network encoding the WAP Proxy server submits the “WSP reply” (which contains the data in response to the “WSP Get”) as several fragments to the SMSC, setting datagram mode and populating the *source_port* and *destination_port* optional parameters. The fragments may be submitted synchronously or asynchronously.

- For each fragment the Bearer Message Centre (SMSC) generates a ***data_sm_resp*** to the Wap Proxy Server and forwards the data to the MS.

- Since the “WSP Get” is a class 2 WTP transaction (see [WTP] and [WSP]), following the successful receipt (of all fragments in this scenario) of the “WSP Reply” the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP Reply” to the WAP Proxy Server.

- An MS can initiate multiple “WSP Get(s)” on a WSP session. When it wishes to, the client initiates a “WSP Disconnect” to indicate that it wishes to terminate the WSP session.

- The SMSC forwards the “WSP Disconnect” in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The Wap Proxy Server acknowledges the ***data_sm*** request and the WSP session is terminated.

- In the above message flow the WAP Proxy Server issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC) for both the transmitter and receiver SMPP sessions. The Bearer Message Centre responds with an ***unbind_resp*** for each ***unbind*** request. The SMPP ***unbind*** sequence is shown for completeness as the WAP Proxy Server should remain permanently bound to the Bearer Message Centre.

- Either Peer can initiate a network (e.g. TCP/IP) disconnect for each network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow (although not illustrated in the diagram) the WAP Proxy Server issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC) for each network connection.

Note 1: In the case where the MS is accessed via a TDMA network, fragmentation being performed on the WAP Proxy Server is not recommended.

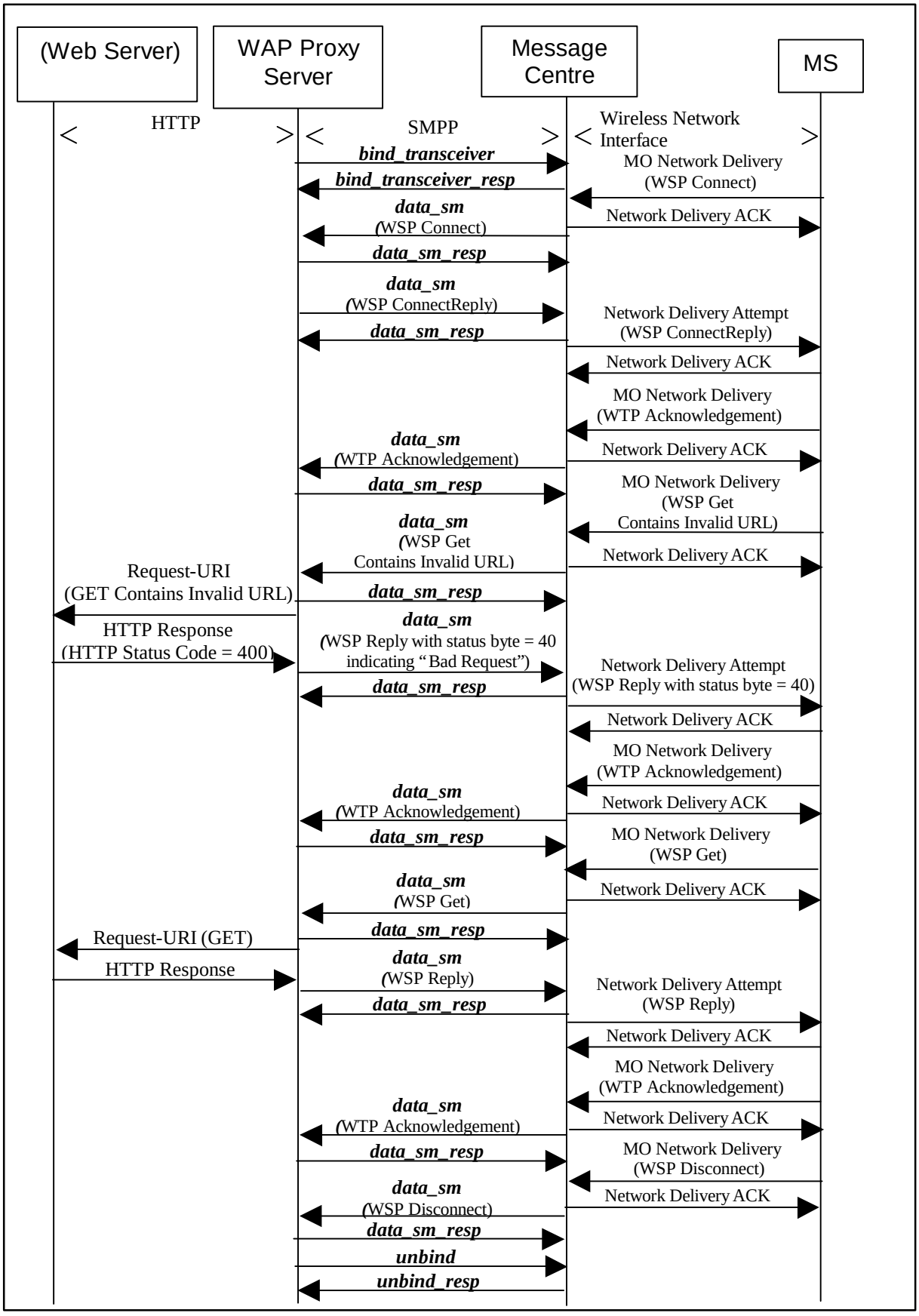
Note 2: It is recommended that Scenarios where the WAP Proxy Server/Push Proxy Gateway and the Bearer Message Centre (SMSC) would both concurrently perform fragmentation should be avoided.

2.2 WSP Get Service – Error Scenarios

2.2.1 Scenario 5: Unsuccessful Information Pull from an Application Node – MS supplies invalid URL content in the "WSP Get" PDU.

Use Case: An MS Subscriber requests a service but supplies invalid URL content in the "WSP Get" PDU.

The MS subscriber requests the service by issuing a "WSP Get" PDU (e.g. a URI request) within a WAP session using a SMSC as the Bearer Message Centre. The request is routed through the WAP Proxy Server to the Web Server



- The WAP Proxy Server issues a Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) which is acknowledged and a network connection has been successfully established.

- The WAP Proxy Server binds as a transceiver using the SMPP ***bind_transceiver*** PDU. The SMSC acknowledges with a ***bind_transceiver _resp*** and SMPP session establishment is successfully complete. It is recommended good practice that the WAP Proxy Server is permanently bound in.

- The MS issues a “WSP Connect” requesting a WSP session with a WAP Proxy Server.

- The SMSC forwards the “WSP Connect” to the WAP Proxy Server in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request and responds by generating a “WSP ConnectReply”, setting datagram mode in the SMPP ***data_sm*** request with the *source_por*t and *destination_port* optional parameters populated.

- The SMSC forwards the “WSP ConnectReply” to the MS.

- Since the “WSP Connect” is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP ConnectReply” to the WAP Proxy Server and WSP session establishment has been successfully completed.

- Once a WSP session has been successfully established, the MS can issue a “WSP Get” PDU (for example a URI request). In this instance the MS issues a “WSP Get” with an invalid URI request.

- The Bearer Message Centre (SMSC) forwards the “WSP Get” PDU to the ESME (WAP Proxy Server) via a ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request.

- The WAP Proxy Server retrieves the information pertaining to the URI request. This normally involves the WAP Proxy Server forwarding the request to a WEB Server. To do this the WAP Proxy Server must be capable of translating requests from the WAP protocol stack (WSP, WTP, WTLS and WDP) to the WWW protocol stack (HTTP and TCP/IP). In addition the data content may have to be translated from MML or rather the compact binary format (WBXML) understood by the client MS to WWW content (HTML).

- The WEB server may provide the response data as WWW content (HTML) or WAP content (WML) and thus the WAP Proxy Server may be required to translate the WWW content to WAP content. The Proxy must then encode the responses from the WEB server into the compact binary format (WBXML) understood by the client (MS). In this case the response from the Web server will indicate "HTTP error code 400 - Bad Request".

- The Wap Proxy Server generates a WSP reply (setting the status field in the WSP reply to 40 indicating "Bad Request"). The Wap Proxy Server submits the “WSP Reply” in response to the “WSP Get” to the SMSC, setting datagram mode on the ***data_sm*** request and populating the optional parameters *source_port* and *destination_port*.

- The Bearer Message Centre (SMSC) generates a ***data_sm_resp*** to the Wap Proxy Server and forwards the data to the MS.

- Since the “WSP Get” is a class 2 WTP transaction (see [WTP] and [WSP]), following the successful receipt of the “WSP Reply” the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP Reply” to the WAP Proxy Server.

- An MS can initiate multiple “WSP Get(s)” on a WSP session. In this instance the MS re-attempts the “WSP Get” supplying valid URL content and receiving and acknowledging the “WSP Reply” with the requested content.

- When it wishes to, the client initiates a “WSP Disconnect” to indicate that it wishes to terminate the WSP session.

- The SMSC forwards the “WSP Disconnect” in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The Wap Proxy Server acknowledges the ***data_sm*** request and the WSP session is terminated.

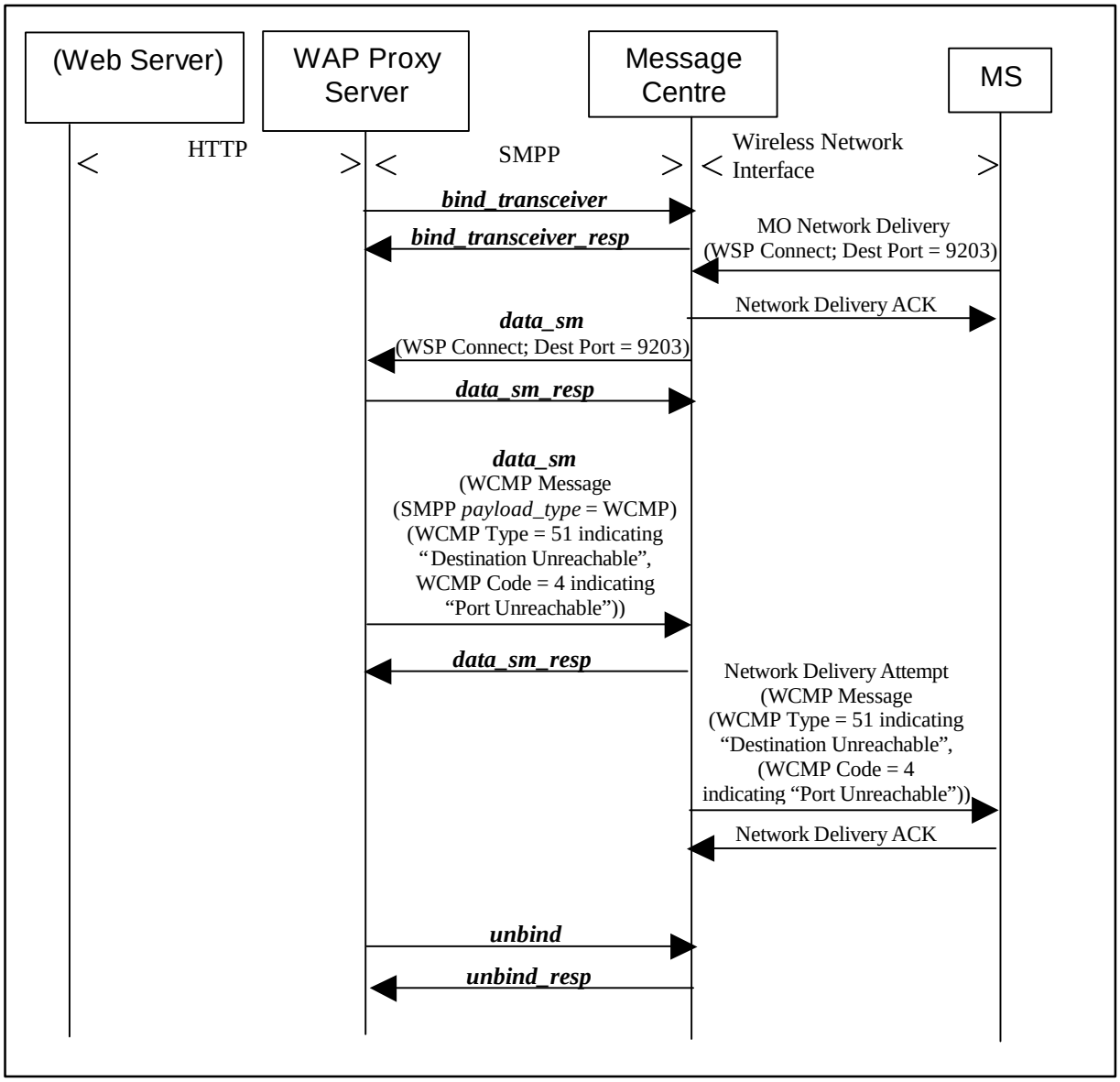
- In the above message flow the WAP Proxy Server issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC). The Bearer Message Centre responds with an ***unbind_resp***. The SMPP ***unbind*** sequence is shown for completeness as the WAP Proxy Server should remain permanently bound to the Bearer Message Centre.

- Either Peer can initiate a network (e.g. TCP/IP) disconnect for the network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow the WAP Proxy Server issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC).

2.2.2 Scenario 6: An MS Subscriber requests secure connection-oriented session service – The WAP Proxy Server indicates the service is not supported using WCMP to indicate "Port Unreachable".

Use Case: An MS Subscriber requests a secure connection-oriented session service in the WSP connect, the WAP Proxy Server indicates the service is not supported and uses WCMP to indicate "Port Unreachable".

The MS subscriber requests a service from the WAP Proxy Server such as WAP secure connection-oriented session service, by issuing a WSP Connect (with destination port 9203) to initiate a WSP WAP session using an SMSC as the Bearer Message Centre. The WAP Proxy Server is unable to respond to the request, because the requested service is not available (thus the WDP port assigned to this particular service is not open on the WAP Proxy Server) and thus the WDP entity on the WAP Proxy Server returns a WCMP message with "*Port unreachable*".



- The WAP Proxy Server issues a Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) which is acknowledged and a network connection has been successfully established.

- The WAP Proxy Server binds as a transceiver using the SMPP ***bind_transceiver*** PDU. The SMSC acknowledges with a ***bind_transceiver _resp*** and SMPP session establishment is successfully complete. It is recommended good practice that the WAP Proxy Server is permanently bound in.

- The MS issues a "WSP Connect" requesting a WAP WSP secure connection-oriented session service (ie destination port set to 9203) with a WAP Proxy Server, using a SMSC as the Bearer Message Centre.

- The SMSC forwards the “WSP Connect” to the WAP Proxy Server in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The WAP Proxy Server acknowledges the SMPP ***data_sm*** request.

- Since the requested service is not available on the WAP Proxy Server and if the WAP Proxy server supports WCMP, the WDP entity on the WAP Proxy Server should originate a WCMP message (with a WCMP type = 51 indicating “Destination unreachable, and a WCMP Code = 4 indicating a “port unreachable error”), setting payload_type = WCMP in the ***data_sm*** and submitting this to the SMSC. As usual datagram mode is set on the ***data_sm*** request to the Bearer Message Centre (SMSC). For the destination unreachable WCMP message, the port information is encapsulated in the body of the WCMP message and thus port information carried in a WCMP message will be encoded in the *message_payload* field of ***data_sm*** (and not in the optional parameters *source_port* and *destination_port* as with normal WDP data).

- The Bearer Message Centre (SMSC) generates a ***data_sm_resp*** to the Wap Proxy Server and forwards the data to the MS.

- In the above message flow the WAP Proxy Server issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC). The Bearer Message Centre responds with an ***unbind_resp***. The SMPP ***unbind*** sequence is shown for completeness as the WAP Proxy Server should remain permanently bound to the Bearer Message Centre.

- Either Peer can initiate a network (e.g. TCP/IP) disconnect for the network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow the WAP Proxy Server issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC).

2.3 Push Service – Normal Scenarios

In the following scenarios datagram mode is recommended as the *esm_class* setting in ***data_sm*** when submitting data to the Bearer Message Centre (SMSC).

Alternatively it may be appropriate for Push Service to set “Store and Forward” mode as the *esm_class* setting, in conjunction with an appropriate setting for the *qos_time_to_live* parameter, in ***data_sm*** when submitting data to the Bearer Message Centre (SMSC).

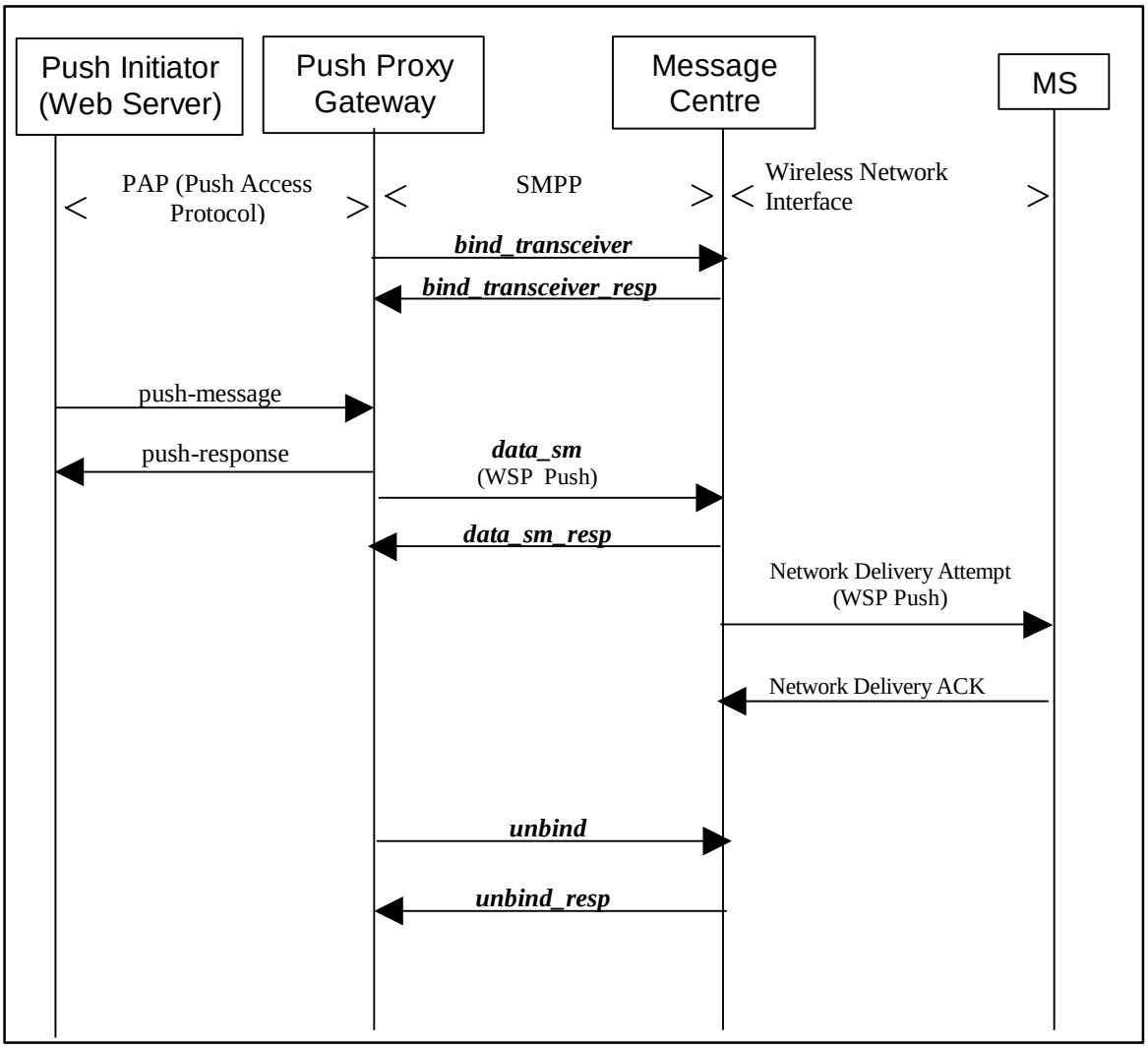
2.3.1 Scenario 7: Connectionless Unconfirmed Push Service

Use Case: A Push Initiator pushes one-way out-of-session information to an MS Subscriber with minimum transaction overhead

This service offers minimum transaction overhead (no session initiation, termination or confirmation transactions) and is suitable for pushing data which does not require confirmation or guarantee of delivery.

Connectionless in this context refers to the fact that there is no WSP session between the MS client and the Push Proxy Gateway. There is however a SMPP connection between the Push Proxy Gateway and the Bearer Message Centre. Thus the normal SMPP session initiation and session termination are illustrated in the message flow. It is recommended good practice that the Push Proxy Gateway remains permanently bound to the Message Centre and session initiation and termination are illustrated in this message flow for completeness.

In the following message flow the Bearer Message Centre is an SMSC, it is worth noting that a Cell Broadcast Centre is also a suitable Bearer Message Centre for connectionless unconfirmed push where there is more than one target MS.



- The Push Proxy Gateway issues a Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) which is acknowledged and a network connection has been successfully established.

- The Push Proxy Gateway binds as a transceiver using the SMPP ***bind_transceiver*** PDU. The SMSC acknowledges with a ***bind_transceiver _resp*** and SMPP session establishment is successfully complete. It is recommended good practice that the Push Proxy Gateway is permanently bound in.

- The Push Initiator issues a Push Access Protocol (PAP) push-message.

- The Push Proxy Gateway generates a push-response informing the Push Initiator that the request has been accepted by the PPG. This does not indicate a guarantee of delivery of the information to the mobile device.

- The Push Proxy Gateway converts transactions from the Web Server WWW protocol stack (HTTP and TCP/IP) to the WAP protocol stack (Push OTA, WSP, WTP, WTLS and WDP). The WEB server may provide the data as WWW content (HTML) or WAP content (WML) and

thus the PPG may be required to translate the WWW content to WAP content and encode it as WBXML.

- The PPG forwards the data it received from the Push Initiator to the SMSC in a “WSP Push” PDU, setting datagram mode in the SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated. This can typically be WML data encoded as WBXML.

- The SMSC generates a ***data_sm_resp*** to the Push Proxy Gateway indicating that the SMSC has accepted the “WSP Push”.

- The SMSC forwards the data to the MS.

- In the above message flow the Push Proxy Gateway issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC). The Bearer Message Centre responds with an ***unbind_resp***. The SMPP ***unbind*** sequence is shown for completeness as the Push Proxy Gateway should remain permanently bound to the Bearer Message Centre.

- Either Peer can initiate a network (e.g. TCP/IP) disconnect for the network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow the Push Proxy Gateway issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC).

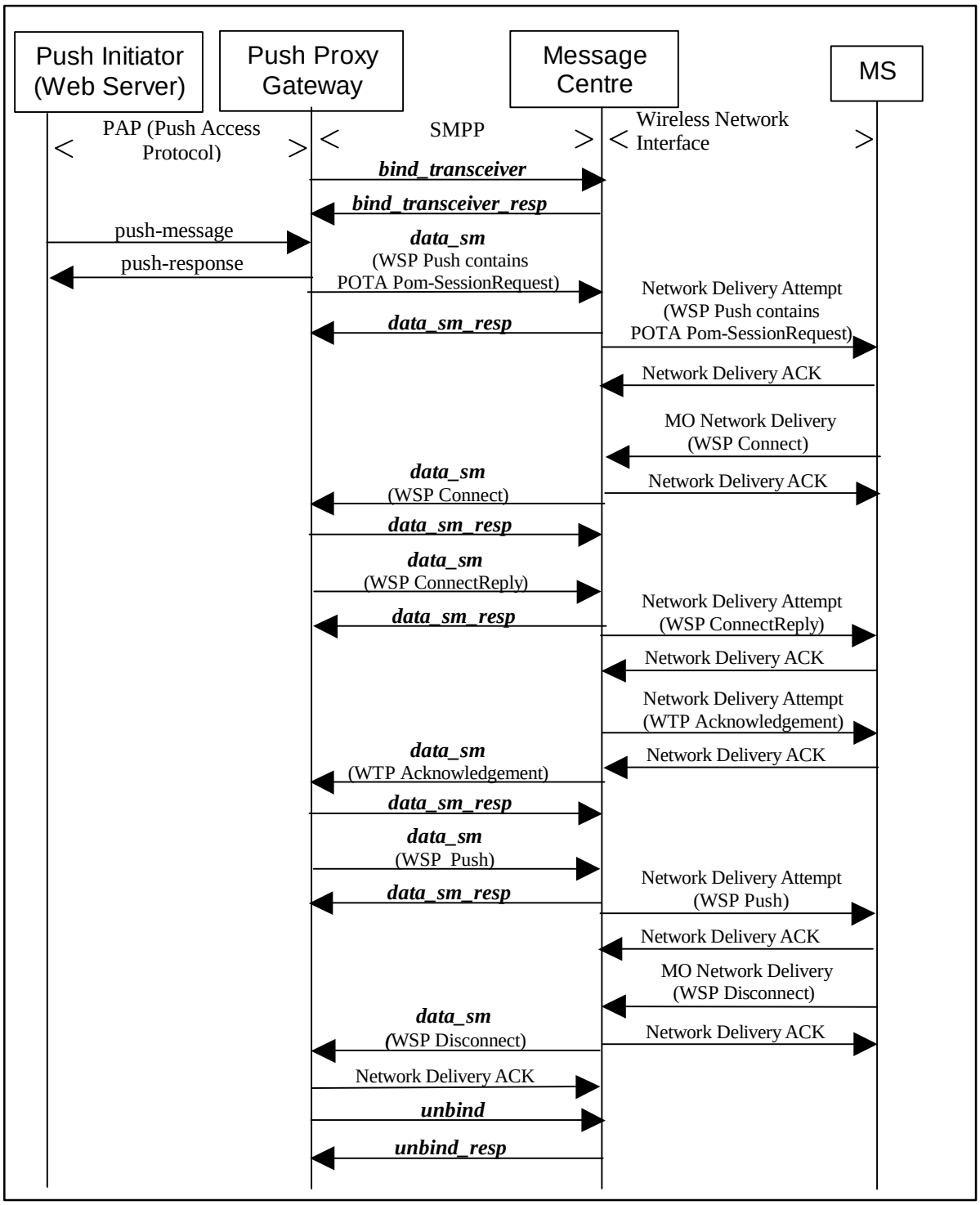
2.3.2 Scenario 8: Connection-Oriented Unconfirmed Push Service

Use Case: A Push Initiator pushes one-way information to an MS Subscriber via a WSP session between the Push Proxy Gateway and the MS.

A connection oriented session service offers the Push initiator the ability to push information, which does not require confirmation or guarantee of delivery, to the mobile device at any time during a session.

The Push Proxy Gateway may request a push session between it and the mobile device if one does not already exist. However, the Push Proxy Gateway cannot create a session directly. Instead, the Push Proxy Gateway sends a Pom-SessionRequest POTA message to the session initiation application, SIA, residing on the mobile device. The SIA then initiates a new session to the Push Proxy Gateway over which information delivery can then occur.

Once Session Initiation has been completed the Unconfirmed Push Service can involve a Push Initiator pushing multiple push data transactions over a WSP session.



- The Push Proxy Gateway issues a Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) which is acknowledged and a network connection has been successfully established.

- The Push Proxy Gateway binds as a transceiver using the SMPP ***bind_transceiver*** PDU. The SMSC acknowledges with a ***bind_transceiver _resp*** and SMPP session establishment is successfully complete. It is recommended good practice that the Push Proxy Gateway is permanently bound in.

- The Push Initiator issues a Push Access Protocol (PAP) push-message. The Push Proxy Gateway generates a push-response informing the Push Initiator that the request has been accepted by the PPG. This does not indicate a guarantee of delivery of the information to the mobile device.

- A WSP Session does not previously exist so a session must be created before the information can be pushed.

- The Push Proxy Gateway sends a “Pom-SessionRequest POTA message” mapped to a “WSP Push” to the session initiation application, SIA (residing on the mobile device), setting datagram mode in the SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated. It should be noted that the WSP Push (containing the Pom-SessionRequest) is a connectionless unconfirmed push and if the request is unsuccessful it is up to the Push Proxy Gateway to reattempt the transaction.

- The SMSC forwards the “WSP Push” (containing the Pom-SessionRequest) to the MS

- The MS (or rather the SIA) responds to the Pom-SessionRequest by issuing a “WSP Connect”, requesting a WSP session with the Push Proxy Gateway.

- The SMSC forwards the “WSP Connect” in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The PPG acknowledges the SMPP ***data_sm*** request and responds by generating a “WSP ConnectReply”, setting datagram mode in the SMPP ***data_sm*** request with the *source_por*t and *destination_port* optional parameters populated.

- The SMSC forwards the “WSP ConnectReply” to the MS.

- Since the “WSP Connect” is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP ConnectReply” to the WAP Proxy Server and WSP session establishment has been successfully completed.

- The Push Proxy Gateway converts transactions from the Web Server WWW protocol stack (HTTP and TCP/IP) to the WAP protocol stack (Push OTA, WSP, WTP, WTLS and WDP). The WEB server may provide the data as WWW content (HTML) or WAP content (WML) and thus the PPG may be required to translate the WWW content to WAP content and encode it as WBXML.

- The PPG forwards the data it received from the Push Initiator to the SMSC in a “WSP Push” PDU, setting datagram mode in the SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated. This will typically be WML data encoded as WBXML.

- The SMSC generates a ***data_sm_resp*** to the Push Proxy Gateway indicating that the SMSC has accepted the “WSP Push”.

- The SMSC forwards the data to the MS.

- In the above message flow the Push Proxy Gateway issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC). The Bearer Message Centre responds with an ***unbind_resp***. The SMPP ***unbind*** sequence is shown for completeness as the Push Proxy Gateway should remain permanently bound to the Bearer Message Centre.

- Either Peer can initiate a network (e.g. TCP/IP) disconnect for the network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow the Push Proxy Gateway issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC).

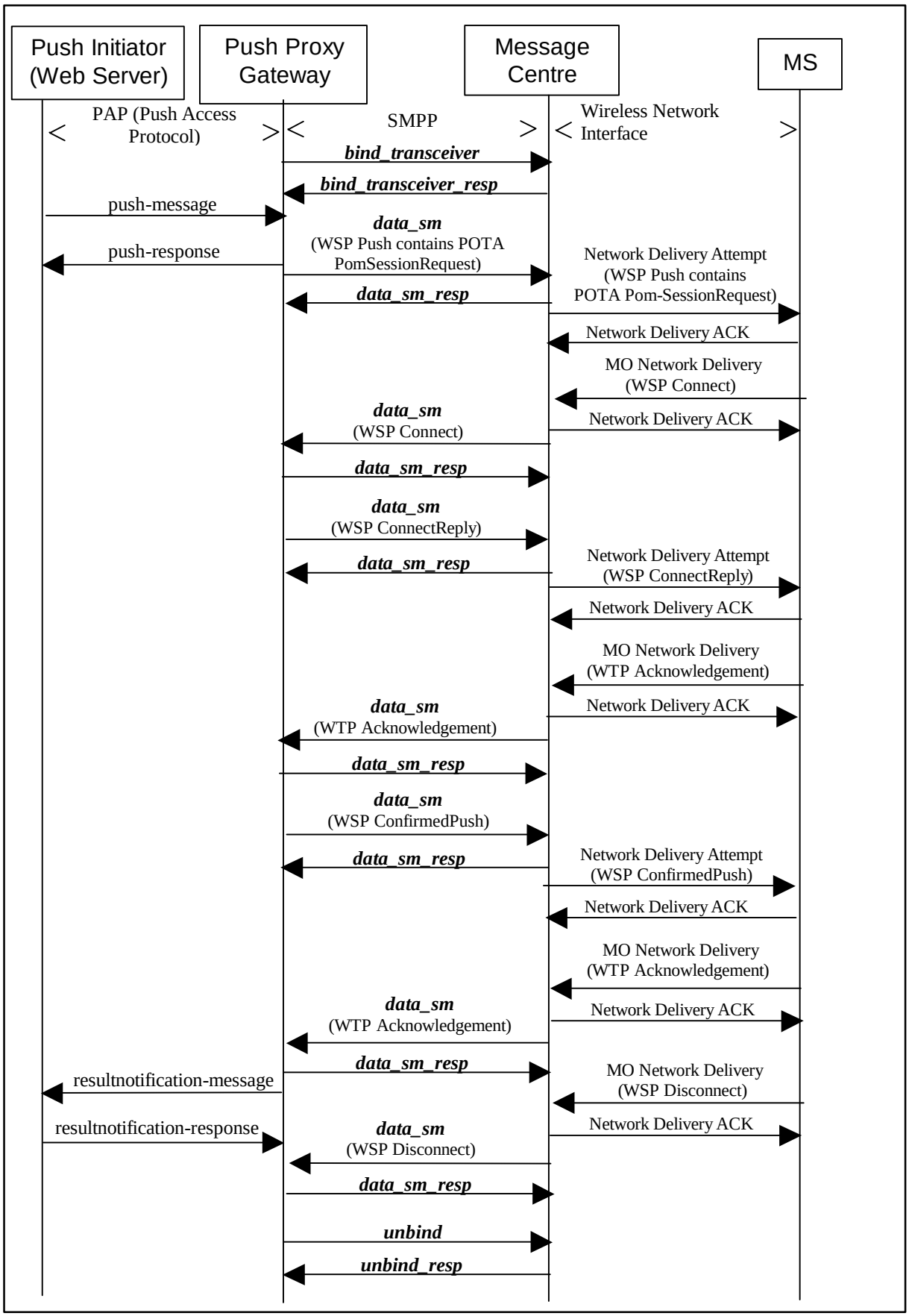
2.3.3 Scenario 9: Connection-Oriented Confirmed Push Service

Use Case: A Push Initiator pushes one-way information to an MS Subscriber via a WSP session between the Push Proxy Gateway and the MS.

A connection oriented session with confirmed push service offers the Push initiator the ability to push information and to receive confirmation that the pushed information was delivered to the mobile device.

The Push Proxy Gateway may request a push session between it and the mobile device if one does not already exist. However, the Push Proxy Gateway cannot create a session directly. Instead, the Push Proxy Gateway sends a Pom-SessionRequest POTA message to the session initiation application, SIA, residing on the mobile device. The SIA then initiates a new session to the Push Proxy Gateway over which information delivery can then occur.

Once Session Initiation has been completed the Confirmed Push Service can involve a Push Initiator pushing multiple push data transactions over a WSP session.



- The Push Proxy Gateway issues a Bearer Service (TCP) connect to the Bearer Message Centre (SMSC) which is acknowledged and a network connection has been successfully established.

- The Push Proxy Gateway binds as a transceiver using the SMPP ***bind_transceiver*** PDU. The SMSC acknowledges with a ***bind_transceiver _resp*** and SMPP session establishment is successfully complete. It is recommended good practice that the Push Proxy Gateway is permanently bound in.

- The Push Initiator issues a Push Access Protocol (PAP) push-message. The Push Proxy Gateway generates a push-response informing the Push Initiator that the request has been accepted by the PPG. This does not indicate a guarantee of delivery of the information to the mobile device.

- A WSP Session does not previously exist so a session must be created before the information can be pushed.

- The Push Proxy Gateway sends a "Pom-SessionRequest POTA message" mapped to a "WSP Push" to the session initiation application, SIA (residing on the mobile device), setting Datagram mode in the SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated. It should be noted that the WSP Push (containing the Pom-SessionRequest) is a connectionless unconfirmed push and if the request is unsuccessful it is up to the Push Proxy Gateway to reattempt the transaction.

- The SMSC forwards the "WSP Push" (containing the Pom-SessionRequest) to the MS

- The MS (or rather the SIA) responds to the Pom-SessionRequest by issuing a "WSP Connect", requesting a WSP session with the Push Proxy Gateway.

- The SMSC forwards the "WSP Connect" in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The PPG acknowledges the SMPP ***data_sm*** request and responds by generating a "WSP ConnectReply", setting datagram mode in the SMPP ***data_sm*** request with the *source_por*t and *destination_port* optional parameters populated.

- The SMSC forwards the "WSP ConnectReply" to the MS.

- Since the "WSP Connect" is a class 2 WTP transaction (see [WTP] and [WSP]), the MS issues a "WTP Acknowledgement" to acknowledge receipt of the "WSP ConnectReply" to the WAP Proxy Server and WSP session establishment has been successfully completed.

- The Push Proxy Gateway converts transactions from the Web Server WWW protocol stack (HTTP and TCP/IP) to the WAP protocol stack (Push OTA, WSP, WTP, WTLS and WDP). The WEB server may provide the data as WWW content (HTML) or WAP content (WML) and thus the PPG may be required to translate the WWW content to WAP content and encode it as WBXML.

- The PPG forwards the data it received from the Push Initiator to the SMSC in a "WSP ConfirmedPush" PDU, setting Datagram mode in the SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated. This will typically be WML data encoded as WBXML.

- The SMSC generates a ***data_sm_resp*** to the Push Proxy Gateway indicating that the SMSC has accepted the “WSP Push”.

- The SMSC forwards the data to the MS.

- Following the successful receipt of the “WSP ConfirmedPush” the MS issues a “WTP Acknowledgement” to acknowledge receipt of the “WSP ConfirmedPush” to the WAP Proxy Server.

- The SMSC forwards the “WTP Acknowledgement” setting Datagram mode in the SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The PPG acknowledges the SMPP ***data_sm*** request and generates a resultnotification-message indicating to the Push Initiator that the pushed information was delivered to the mobile device.

- The Push Initiator acknowledges the resultnotification-message with a resultnotification-response.

- The client may initiate a “WSP Disconnect” to indicate that it wishes to terminate the WSP session.

- The SMSC forwards the “WSP Disconnect” in a SMPP ***data_sm*** request with the *source_port* and *destination_port* optional parameters populated.

- The PPG acknowledges the ***data_sm*** request and the WSP session is terminated.

- In the above message flow the PPG issues an SMPP ***unbind*** to the Bearer Message Centre (SMSC). The Bearer Message Centre responds with an ***unbind_resp***. The SMPP ***unbind*** sequence is shown for completeness as the PPG should remain permanently bound to the Bearer Message Centre.

- Either Peer can initiate a network (e.g. TCP/IP) disconnect for the network connection when the unbind phase has been successfully completed (In some cases a TCP disconnect can be issued before the unbind phase has completed). In this message flow the Push Proxy Gateway issues a Bearer Service (TCP) disconnect to the Bearer Message Centre (SMSC).

Change History

Date	Editor	Summary of changes
050500	Hugh Carr (Logica)	Initial Draft Vers
180800	Hugh Carr (Logica)	Added Push Service Scenarios and detail added as regards WSP and SMPP.
070900	Hugh Carr (Logica)	Updated with review input from Ericsson.
071200	Hugh Carr (Logica)	Updated with review input from WAP SMSEG – including WCMP Scenario from Nokia amongst other review input.
230101	Hugh Carr (Logica)	Added WAP June 2000 Conformance Release to title and updated references section where appropriate to reference WAP June 2000 Conformance Release Documents.
140201	Hugh Carr (Logica)	Changed title in heading to “SMPP Implementation Guide for WAP”. Updated Company Name including Logo.