

The Wireless Communication Transfer Protocol (WCTP)

Protocol Specification

Revision: 1.0
Release Date: June 6, 2000

Status: Released
Document ID: wctp-v1r0.doc



Personal Communications Industry Association

<http://www.pcia.com>

Notice

Copyright © 1999, Personal Communications Industry Association

This specification is a work in progress for the purposes of creating an open, non-proprietary, messaging standard between wireline and wireless networks. The Personal Communications Industry Association makes no warranty of any kind with regard to this material, including, but not limited to, the documentation, function, and completeness of these specifications, and their suitability for any particular purpose. The Personal Communications Industry Association reserves the right to revise this publication and to make changes to it from time to time without obligation to notify any person or organization of such revision or changes. Personal Communications Industry Association shall not be liable for any errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this specification.

Participation in WCTP Developments

Comments, contributions, and discussions may be submitted to the Personal Communications Industry Association through its Web site <http://www.pcia.com>. PCIA members who wish to contribute to the efforts to enhance this protocol are encouraged to join the WCTP technical sub-committee. Non-PCIA members who wish to contribute suggestions for modification of the protocol or to suggest new features and capabilities are encouraged to work with a member of the technical committee to represent their ideas to the committee.

Trademarks

FLEXsuite™ is a trademark of Motorola, Inc.

ReFLEX™ is a trademark of Motorola, Inc.

WMapi™ is a trademark of Glenayre Electronics Inc.

Company names mentioned, are the trademarks of the individual company.

WCTP Committee Acknowledgement

The following committee members contributed to the development, enhancements and promotion of this specification.

Chairman

Edward "Ted" Sumner Vast Solutions edward_sumner@vast.com

Past Chairman

Wiley Wilkins Vast Solutions wiley_wilkins@vast.com

Originating Authors

Jay Moskowitz RTS Wireless jmosk@rtswireless.com

Dr. Yuri Salkinder RTS Wireless yuri.salkinder@rtswireless.com

Drafting Committee

Committee Member	Company
Dr. Alfred Adler	Motorola
Raj Cherukuri	Skytel
Dr. Michael Christiansen	Pagenet
Vic Cox	Pagemart
John Davis	Pagemart
Bruce Deer	Skytel
Don Gayton	Glenayre
Mike Johnson	Skytel
Anthony Klinkert	Pagenet
Brian Laird	Glenayre
Kyung Tae Mun	Pagemart
Jay Moskowitz	RTS Wireless
Bijan Nowroozi	Pagenet

Committee Member	Company
Garland Phillips	Motorola
Jeff Poe	Skytel
Dr. Yuri Salkinder	RTS Wireless
Karl Schlieber	Bell South Wireless Data
Bill Scott	Pagenet
Dwight Smith	Motorola
Benjamin Sun	Pagenet
Chris Wells	Glenayre
Wiley Wilkins	Pagenet

Revision History

Document Number	Date	Description
wctpv1r0.doc	June 6, 2000	Updated specification to reflect WCTP as a PCIA supported protocol. Update references to the WCTP DTD to show that it now exists at the PCIA Web site.
wctpv0r7.doc	November 1, 1999	Final revision of the first "official" version of the protocol presented to the PCIA. Accepted as a standard by the Paging Technical Committee of the Personal Communication Industry Association (PCIA). Use of the uuencode scheme for the transparent data transfer is discontinued, and the use of the XML "canonical" encoding is discouraged in favor of base64 encoding.
wctpv0r6.doc	June 18, 1999	Minor typos and inconsistencies are fixed in the use cases. All the use cases were validated through an XML parser.
wctpv0r5.doc	May 17, 1999	Release of drafting committee preliminary version 0.4 with a major portion of the 21 defined use cases included.
wctpv0r4.doc	April 9, 1999	Incorporated changes agreed to by the members of the drafting committee, developed during a series of meetings and conference calls. This is major update to the DTD and rewrite of the specification from the previous release. Changes span the entire document and are too numerous to list individually.
wctpv0r3.doc	February 22, 1999	Re-release of specification under the name of the Messaging Standards Committee instead of RTS Wireless, the original author.
wctpv0r2.doc	January 29, 1999	Initial Release by Jay Moskowitz, CTO, RTS Wireless, a.k.a. Real Time Strategies, Inc. Introductory specification released after a presentation to the Messaging Standards Committee, January 13, 1999. As a result of that meeting, certain aspects of WCTP have been eliminated from this specification so as

		to simplify its description and to avoid the addressing of implementation details which should be outside of the scope of this document.
--	--	--

Contents

- 1. INTRODUCTION 1-1**
 - 1.1 A NEW STANDARD IS NEEDED 1-1
 - 1.2 GOALS OF WCTP 1-2
 - 1.3 WCTP AND ITS RELATIONSHIP TO WAP 1-2
 - 1.4 REFERENCE MATERIALS 1-3
- 2. WCTP OVERVIEW..... 2-1**
 - 2.1 NATURE OF WCTP 2-1
 - 2.2 TRANSPORT OF WCTP 2-2
- 3. SAMPLE SCENARIOS 3-1**
 - 3.1 PULLING DATA FROM A HOST VIA A TWO-WAY PAGER 3-1
 - 3.2 OPERATING A WIRELESS DEVICE FROM THE INTERNET 3-1
 - 3.3 COMPLEX MULTI-DEVICE TRANSACTION..... 3-4
- 4. WCTP PROTOCOL PROCEDURE 4-1**
 - 4.1 WCTP COMMUNICATIONS THROUGH HTTP 4-1
 - 4.1.1 *Pushing WCTP from the Wireline to Wireless Network* 4-1
 - 4.1.2 *Pushing WCTP From the Wireless to Wireline Network*..... 4-2
 - 4.1.3 *"Pseudo-Pushing" of WCTP from the Wireless to Wireline network* 4-4
 - 4.2 TWO CATEGORIES OF WCTP USERS 4-4
 - 4.2.1 *Enterprise Hosts*..... 4-5
 - 4.2.2 *Transient Users* 4-5
 - 4.3 ADDRESSING IN WCTP..... 4-5
 - 4.3.1 *Identifiers for Wireless Devices*..... 4-5
 - 4.3.2 *Identifiers for Wireline Hosts* 4-6
 - 4.3.3 *Recipient Specification on Wireless Devices*..... 4-7
 - 4.4 WCTP ADDRESSING AND MESSAGE ROUTING 4-7
 - 4.4.1 *Message Delivery to Enterprise Hosts* 4-7
 - 4.4.2 *Message Delivery to Wireless Carriers*..... 4-9
 - 4.4.3 *Message Response Redirection to non-WCTP Destination*..... 4-9
- 5. WCTP OPERATIONS..... 5-10**
 - 5.1 WCTP OPERATIONS SUMMARY 5-10
 - 5.2 CALL FLOW BETWEEN NETWORK ELEMENTS 5-13
 - 5.3 WCTP SUBMIT MESSAGE REQUEST OPERATION..... 5-17
 - 5.3.1 *Message Payload* 5-17
 - 5.3.2 *Submit Header*..... 5-18
 - 5.4 WCTP MESSAGE REPLY OPERATION 5-19
 - 5.4.1 *Response Header*..... 5-19
 - 5.5 WCTP STATUS INFO OPERATION 5-20
 - 5.6 WCTP CONFIRMATION OPERATION 5-21
 - 5.7 WCTP POLL FOR MESSAGES OPERATION 5-22
 - 5.8 WCTP POLL RESPONSE OPERATION 5-22
 - 5.9 WCTP LOOKUP SUBSCRIBER OPERATION..... 5-23
 - 5.10 WCTP LOOKUP RESPONSE OPERATION..... 5-23
 - 5.11 WCTP SUBMIT CLIENT MESSAGE OPERATION 5-23
 - 5.12 WCTP SUBMIT CLIENT RESPONSE OPERATION 5-24
 - 5.13 WCTP CLIENT QUERY OPERATION 5-25
 - 5.14 WCTP CLIENT QUERY RESPONSE OPERATION 5-25

5.15 WCTP BIND DOMAIN ALIAS OPERATION 5-26

6. WCTP STANDARDS..... 6-1

6.1 XML SYNTAX 6-1

6.2 DATE AND TIME FORMATS 6-1

6.2.1 WCTP Date Format 6-1

6.2.2 WCTP Time Format 6-1

6.2.3 Subsecond Resolution 6-2

6.2.4 Time Zones 6-2

6.2.5 WCTP Date and Time Format..... 6-2

6.3 STATUS AND ERROR CODES 6-2

6.3.1 No HTTP Relationship 6-3

6.3.2 Status and Error Code Returned Text Strings..... 6-3

6.3.3 Categories of Status and Error Codes 6-3

6.4 WCTP OPERATION DECLARATIONS 6-4

6.5 WCTP PROTOCOL VERSION..... 6-5

7. PROTOCOL CAPABILITIES FOR FUTURE CONSIDERATION 7-1

7.1 ISSUES TO ADDRESS 7-1

A APPENDIX A - DOCUMENT TYPE DEFINITION FOR WCTP 1

B APPENDIX B - WCTP FIELD DEFINITIONS 1

C APPENDIX C - ERROR AND SUCCESS CODES..... 1

D APPENDIX D - IMPLEMENTATION NOTES, PROTOCOL BEHAVIOR..... 1

D.1 INFRASTRUCTURE ELEMENTS 1

E APPENDIX E - GENERAL EXAMPLES..... 1

E.1 REQUEST FROM WIRELESS DEVICE TO ENTERPRISE HOST 1

E.1.1 Alarm Message Sent..... 1

E.2 REQUEST FROM ENTERPRISE HOST TO A 2-WAY PAGER 2

E.2.1 Enterprise Host presents multiple choice selection to a technician..... 2

E.3 DELIVERY NOTIFICATION – WCG TO ENTERPRISE HOST 3

E.3.1 WCG informs Enterprise Host that message was successfully delivered to a 2-way pager. 3

E.4 RESPONSE FROM 2-WAY PAGER TO ENTERPRISE HOST 4

E.4.1 Technician selects one of presented choices 4

E.5 RESPONSE FROM ENTERPRISE HOST TO WIRELESS DEVICE 5

E.5.1 Enterprise Host sends a request to perform a corrective action 5

E.6 CONFIRMATION OF MESSAGE ACCEPTANCE TO ENTERPRISE HOST 6

E.6.1 Internet Gateway informs Host that message passed protocol conformance verification. 6

E.7 RESPONSE FROM WIRELESS DEVICE TO ENTERPRISE HOST 7

E.7.1 Wireless device informs Enterprise Host that corrective action started 7

E.8 RESPONSE FROM ENTERPRISE HOST TO 2-WAY PAGER 8

E.8.1 Enterprise Host informs technician that corrective action started 8

E.9 REQUEST FROM 2-WAY PAGER TO ENTERPRISE HOST 9

E.9.1 Technician asks Host to request the telemetry device to verify that corrective action was taken 9

F APPENDIX F - USE CASES 1

F.1 USE CASE #1: TRANSIENT CLIENT SENDING A MESSAGE TO A VALID CUSTOMER..... 2

F.2 USE CASE #2: TRANSIENT CLIENT SENDING A MESSAGE TO AN INVALID CUSTOMER 4

F.3 USE CASE #3: TRANSIENT CLIENT CHECKING THE STATUS OF A MESSAGE WITH A VALID TRACKING NUMBER. 6

F.4 USE CASE #4: TRANSIENT CLIENT CHECKING MESSAGE STATUS USING AN INVALID TRACKING NUMBER..... 10

F.5 USE CASE #5: A DEVICE INITIATED MESSAGE FOR A VALID DEVICE ON A DIFFERENT MESSAGING NETWORK. 11

F.6 USE CASE #6: A DEVICE INITIATED MESSAGE TO AN INVALID DEVICE ON A DIFFERENT NETWORK 11

F.7 USE CASE #7: A DEVICE INITIATED MESSAGE FOR AN INVALID HOST 12

F.8 USE CASE #8: AN ENTERPRISE HOST ORIGINATED MESSAGE TO A VALID DEVICE 14

F.9 USE CASE #9: AN ENTERPRISE HOST ORIGINATED MESSAGE TO AN INVALID DEVICE..... 16

F.10 USE CASE #10: A DEVICE INITIATED MESSAGE TO A DEFAULT APPLICATION ON AN ENTERPRISE HOST 18

F.11 USE CASE #11: A DEVICE INITIATED MESSAGE TO A SPECIFIC APPLICATION ON AN ENTERPRISE HOST 21

F.12 USE CASE #12: AN MCR MESSAGE FROM A TRANSIENT CLIENT OR ENTERPRISE HOST 24

F.12.1 A transient user originated MCR and a subsequent MCR response. 24

F.12.2 A non-polling enterprise host originated MCR and a subsequent MCR response..... 27

F.12.3 A polling enterprise host originated MCR and a subsequent MCR response 30

F.13 USE CASE #13: A BINARY MESSAGE FROM A TRANSIENT CLIENT OR ENTERPRISE HOST 34

F.13.1 A binary message originated by a transient user and a subsequent response to it..... 35

F.13.2 A binary message originated by a non-polling enterprise host and a subsequent response to it..... 38

F.13.3 A binary message originated by a polling enterprise host and a subsequent response to it..... 41

F.13.4 A binary message from a wireless application to an enterprise host and a subsequent response 44

F.14 USE CASE #14: A TRANSIENT USER OR SERVER INITIATED MESSAGE WITH AUTHORIZATION CODE 47

F.15 USE CASE #15: AN ENTERPRISE HOST ASKING FOR ALL OF ITS MESSAGES 50

F.16 USE CASE #16: A DEVICE REPLY TO AN ENTERPRISE HOST GENERATED MESSAGE 55

F.17 USE CASE #17: A DEVICE REPLY TO A TRANSIENT USER GENERATED MESSAGE 55

F.18 USE CASE #18: SENDING BACK MULTIPLE REPLIES ON A TRANSIENT USER MESSAGE STATUS QUERY 55

F.19 USE CASE #19: DISABLING REPLIES ON A MESSAGE SENT TO THE DEVICE 55

F.20 USE CASE #20: POLLING BY ENTERPRISE HOST FOR ALL MESSAGES DIRECTED AT THAT HOST..... 56

F.21 USE CASE #21: LOOKUP & SUBMIT FROM AN ENTERPRISE HOST 61

List of Figures

Figure 2-1. Simple WCTP Connection	2-3
Figure 3-1. Wireless Person to Wireline Computer	3-2
Figure 3-2. Wireline Person to Wireless Device	3-3
Figure 3-3. Telemetry Device, Two-Way Pager and Wireless Host	3-5
Figure 4-1. Pushing HTTP Requests From the Enterprise Host.....	4-2
Figure 4-2. Pushing WCTP Operations From the Wireless Network.....	4-3
Figure 5-1. Call Flow Between Network Components	5-16
Figure 5-2. WCTP Submit Request.....	5-17
Figure 5-3. WCTP Message Reply Operation	5-20
Figure 5-4. WCTP Status Info Operation	5-21
Figure 5-5. WCTP Confirmation Operation	5-22
Figure 5-6. WCTP Submit Client Message Operation.....	5-24
Figure 5-7. WCTP Client Query Response Operation.....	5-25
Figure 5-8. WCTP Bind Domain Alias Operation	5-26

1. Introduction

1.1 A New Standard is Needed

The **Wireless Communications Transfer Protocol (WCTP)** is specifically aimed at creating an easy means of passing alphanumeric and binary messages to and from wireline systems and two-way capable wireless devices. A draft proposal was submitted to the Radio Paging Community, through an ad hoc "Messaging Standards Committee", as means to foster industry input, co-operation, study, promotion and participation in its further expansion and growth as an open, non-proprietary standard. The Messaging Standards Committee is a group of paging industry manufacturers and carriers focused on rapidly creating mutually agreeable standards to address emerging applications that can not be easily realized through the utilization of existing standards. The Committee accepted the first proposal and established a WCTP drafting sub-committee to further enhance the protocol and release it in the form of this specification. The sub-committee is continuing its efforts to refine the features and capabilities of the protocol, and further feature richness will be introduced in future revisions. It is the intent of the drafting committee to provide the greatest degree of revision compatibility between releases as possible, so as to easily allow for the incorporation of new features and to provide a means for systems operating at different protocol revisions to communicate. With the completion of the first full release of the WCTP specification, the Personal Communications Industry Association agreed to adopt the protocol as a PCIA standard and support the continued activities of the working group as an official PCIA technical sub-committee.

Although introduced through the paging industry, WCTP is directly applicable for messaging to/from most other wireless technologies including PCS, GSM, and cellular.

The wireless industry today hosts a range of different protocols of varying complexities and capabilities to submit messages into a wireless network (e.g. TAP, TNPP, TDP/TME, TME-X, WMtp™, WMapi™, UCP, I4, and SMPP). In certain cases, messages are submitted within the confines of Internet Standard protocols such as the Simple Mail Transport Protocol (SMTP/Email protocol) or the HyperText Transfer Protocol (HTTP). In these cases the extent of the information that may be conveyed to the wireless network is limited due to the nature of these generalized protocols. Sometimes carriers further offer specialized software development kits providing Application Programming Interfaces (API's) which are specific to their individual networks. Oftentimes, the implementers of "wireless enabled" applications need to find the right protocol to integrate into their application for a particular network, only to find that they need to re-implement their application under another protocol for another wireless two-way network.

WCTP has been created to **provide a wireless industry specific standardized representation of messages** which are to:

- Be sent from a wireline host to a wireless device.
- Be sent from a wireless device to a wireline host.
- Facilitate inter-carrier connectivity for messages sent from a wireless device to a wireless device on another carrier's network.

This document also includes a section that specifies a number of issues that have been recognized as requiring further study. It also denotes some features and functions, which are under consideration, but were not ready for introduction in the current specification revision. In certain cases, it was determined to defer certain functionality to a later release in order to simplify the earlier releases of the specifications and implementations of the protocol.

1.2 Goals of WCTP

The major goals that are to be derived through WCTP are:

- 1) To interconnect Internet accessible systems with wireless networks.
- 2) To capitalize on Internet standards already in place or rapidly emerging which will dramatically simplify the process of implementing WCTP over other existing standards which are straightforwardly usable over the Internet.
- 3) To send and receive messages containing any type of binary data content.
- 4) To support as many wireless networks as possible.
- 5) To create an extensible architecture allowing unlimited growth to accommodate new features and capabilities.
- 6) To support a wide range of wireless devices.
- 7) To support the ability for wireless devices to send messages to other wireless devices located on other wireless carrier's networks.

1.3 WCTP and its relationship to WAP

The Wireless Application Protocol forum has received worldwide attention and at the time of this writing has grown to a membership of more than 85 corporations. A set of approximately 20 specification documents currently exists detailing the architecture and protocol specifics of the various aspects of the "WAP Protocol Suite". WAP is still an evolving specification. Many people, who are not knowledgeable about all of the current WAP specifics, believe that WAP is the all-encompassing answer to two-way communications with wireless devices. In reality, the forum is presently focused on a particular set of features that operate between a "WAP gateway" and wireless devices. It has not addressed the aspect of providing a specific standardized protocol to reach the WAP gateway itself.

The initial focus of WAP has concentrated on the support of a handheld wireless Internet browser and integrated telephony/browser functions as well as a standard set of protocols between the handset, the wireless network and the WAP Gateway to support these functions. The set of protocols established by the WAP forum are designed to support much more than browser functions. WAP discusses the desire to access the various layers of the protocol suite through a set of standardized API's, but these have not yet been established.

WCTP is complementary to WAP in that it addresses a standardized manner to move data to/from the wireless network gateway over a wired network (an area noted as not being addressed). It might even represent a means of directly presenting data directly to the internal WAP layers. As such, WCTP could potentially be studied for integration with the WAP protocol stack. Having such discussions is anticipated.

Another area where WCTP and WAP differ is in how WCTP views the wireless network. The WAP architecture calls for all wireless devices, regardless of the over-the-air protocol details of the individual wireless network, to support the WAP protocol layers. This helps to make all devices appear the same. WCTP, on the other hand, only looks toward the use of one methodology to move between the wired network and a wireless network gateway (explained shortly). It does not require that all of the wireless networks appear uniform and can be used both by WAP compatible networks as well as today's existing non-WAP networks employing the host of protocols referred to in the WCTP introduction.

1.4 Reference Materials

The following documents and Web site addresses provide information that is pertinent to this specification.

- Reference information regarding XML , Web site: <http://www.w3.org/XML>
- Motorola, FLEXsuite™ of Enabling Protocols, Document Number: 6881139B10
- Motorola, ReFLEX™ 25 Specification, Document Number: RFL25LX-033195
- Message Standard Committee Web site: <http://www.msg-stds.org>
- Personal Communications Industry Association Web site: <http://www.pcia.com>
- WAP Forum Web Site: <http://www.wapforum.org>
- ISO8601, Web Site: <http://www.iso.ch/>

2. WCTP Overview

2.1 Nature of WCTP

The Wireless Communications Transfer Protocol can be succinctly characterized through the major attributes described below. These will be clarified through the descriptions in the next sections.

- 1) There are sets of specific operations that are represented within WCTP.
- 2) Any operation may result in:
 - 2.1) No direct response.
 - 2.2) One direct response.
 - 2.3) Many resultant responses.
- 3) Operations may be initiated by:
 - 3.1) The wireline system.
 - 3.2) An infrastructure element of the wireless network creating a WCTP operation on behalf of a wireless device.
- 4) Response operations may come from:
 - 4.1) The wireline system.
 - 4.2) Infrastructure elements of the wireless network which create WCTP response messages on behalf of the wireless device or on behalf of the wireless network reporting a significant event regarding the message delivery process.
- 5) WCTP operations are represented through the Extensible Markup Language (XML).
 - 5.1) This document is not meant to be an introduction to XML. The reader is referred to the Web site <http://www.w3.org/XML> for reference material regarding this markup language. XML, like the World Wide Web's HyperText Markup Language (HTML), are subset of the Standard Generalized Markup Language (SGML). Through XML, application specific data representations may be created for the common sharing of this specific information among dissimilar systems across the Internet just as HTML allows dissimilar systems to visually paint the same display screens. XML has already surpassed a certain critical mass, being adopted across many industries because of its ease of use, human readability, and information content carrying ability. This critical mass is resulting in an explosion in the distribution of information regarding the support of XML and the creation of tools

to easily integrate XML into applications. Web browsers, such as Internet Explorer 4.0 already are running XML "under-the-hood" in order to provide services such as "Information Channels" delivering specific information of interest from the Internet to a subscribers computer. The rapid deployment of more XML support tools by the Internet development community will be bringing XML even more to the forefront and not leave it hidden under-the-hood at all. For all of these reasons, as well as many more as is evident among the materials at the referenced Web site, the incorporation of XML into WCTP, and its ease of integration, will dramatically increase the acceptance of WCTP across all application developers who wish to wirelessly enable their products.

- 5.2) WCTP concepts are represented within the element/attribute markup model of XML.
 - 5.3) The formal description of an XML based document or WCTP operation in this case, is given through a specification known as the Document Type Definition (DTD). The DTD can be used by a set of software known as an XML message parser, to validate that the syntax of a received transaction is valid. WCTP expects operations being submitted to be of a valid form or it will reject the received information. This document contains the DTD for WCTP as well as several sample message flows and their XML representations.
- 6) WCTP is not a pure document markup application but is a request/reply type protocol.

2.2 Transport of WCTP

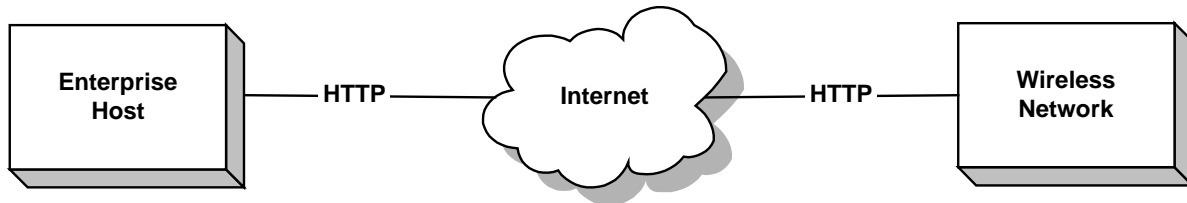
WCTP is called a "transfer" protocol in that it is transferring information content between wireline and wireless devices. The manner, in which the WCTP defined operations are transported between a pair of systems, is not defined by and is independent of WCTP. The protocol may operate over any desired transport protocol that is block oriented. Meaning that the underlying transport protocol can move a complete WCTP operational control block between the systems.

The protocol itself deliberately does not address the issues of security or authentication (possibly through digital signatures for example) of transactions in a highly secure manner. This is because these capabilities may be achieved through existing and rapidly emerging Internet standards that operate at the transport level.

Over the Internet, WCTP will *primarily* be transported through the use of the HTTP protocol normally used to move World Wide Web HTML documents between systems. HTTP communications would normally be initiated by client software on a host computer wishing to connect to the wireless network. In order to accommodate the return of solicited messages and the sending of unsolicited messages from the wireless network to a host computer, Internet "push" technology shall be used. Wherever "push" implementation will not be possible, "pseudo-push", where the transmission is initiated

by a poll from a host computer, can be used instead. This is further detailed in a later section. Because WCTP is transport independent, in some applications operating across the Internet, any of a number of different transport protocols could potentially be utilized. The current recommendation is to utilize the HTTP protocol as discussed in section 4.

The basic form of a WCTP enabled network is shown in Figure 2-1. In this configuration the wireline host uses the protocol to directly send messages into the wireless network.



000070 Jan. 11, 1999

Figure 2-1. Simple WCTP Connection

The integration of WCTP within the Wireless Network component of Figure 2-1 may incorporate a number of different infrastructure components. Configurations could involve WCTP being carried into an "Internet Gateway" that front-ends other infrastructure equipment operating under a non-WCTP protocol. More complex configurations could support an environment in which a single WCTP connection to the wireless network equipment is actually allowing messaging to dissimilar wireless networks operated by different paging and PCS carrier's. The infrastructure to support such configurations could have a front-end Internet Gateway that is forwarding traffic to other WCTP-enabled Wireless Communication Gateways for each individual network. WCTP is tasked with ensuring that sufficient information is carried to support such configurations.

Because of the extensible nature of WCTP, the carrying of non-standard or implementation specific data elements may be added to the protocol at any time to support the specialized requirements of some complex configurations. Should some of these extensions be necessary to integrate WCTP in such a configuration, it is expected that these extensions would be considered for incorporation into this specification. As specified by the XML nature of the protocol, fields tags that are not recognized by a specific WCTP implementation because they are not part of the standard protocol definition, will be ignored. This might not provide the desired operation within a particular implementation but it does allow the rest of the protocol control structures to be processed normally.

3. Sample Scenarios

To better understand how WCTP might be used in typical applications and to better understand why the protocol is carrying certain types of information, this section describes a few applications that would be implemented under the protocol. Appendix E provides several general examples of the actual coding of the WCTP transactions to support the most complex of these scenarios. Appendix F contains several specific examples of the actual WCTP coding which represents the primary cases in which it is used.

To show how WCTP may be used to not only move data between the wireline and wireless network but also within the infrastructure of the network, these examples also show the message flow between elements within the wireless network. In these examples, a front-end Internet Gateway supporting the WCTP protocol is shown. Within the wireless network a Wireless Communications Gateway (WCG) provides for an interface between the Internet Gateway and the protocols of a specific wireless network is shown.

3.1 Pulling Data From a Host Via a Two-Way Pager

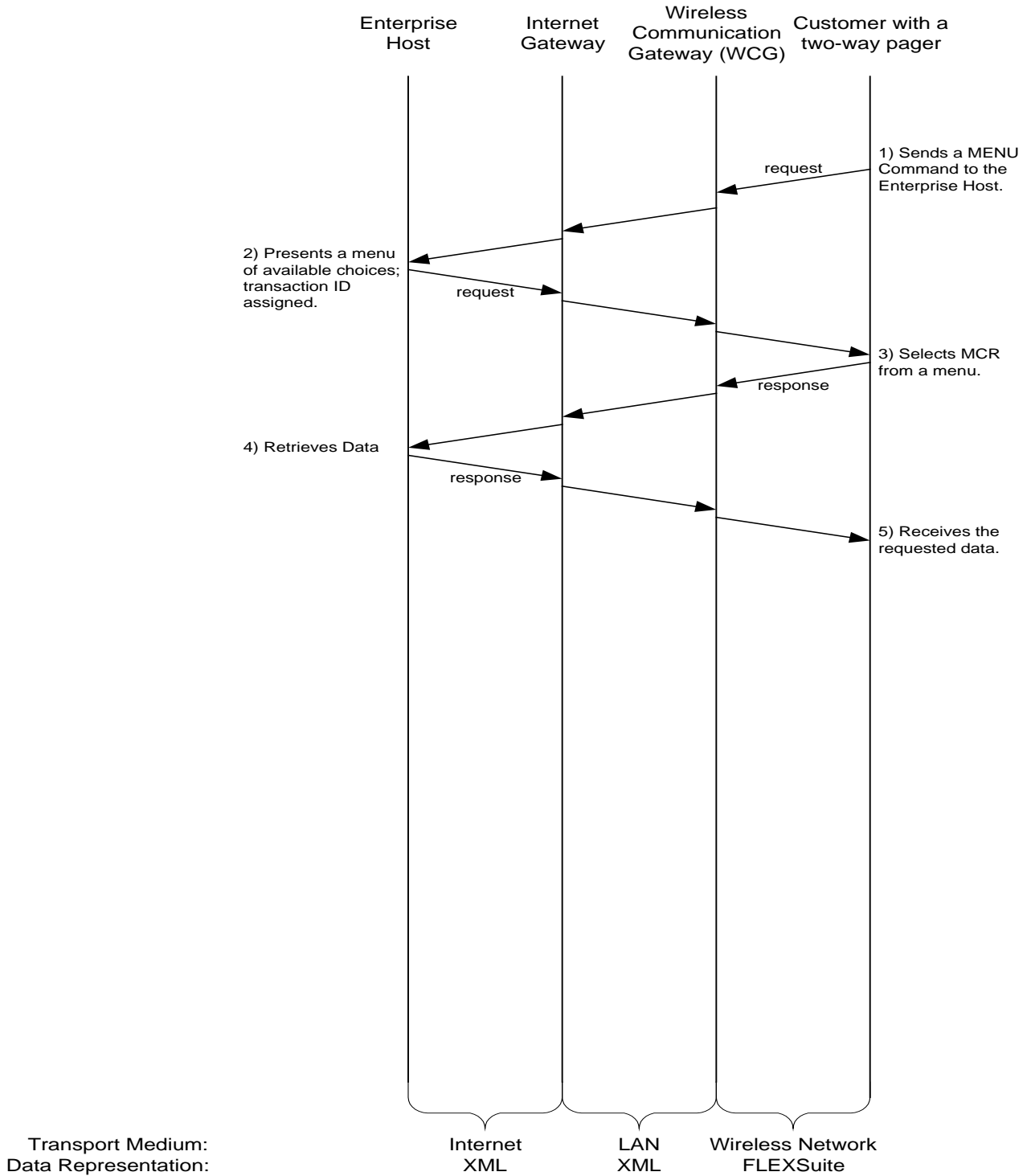
A customer with a two-way pager wants to retrieve the results of a daily report from an enterprise host. Access to the report is pre-programmed via a customer's private menu on the host. Customer initiates the exchange by sending a request to the enterprise host that contains a command to show a menu. Enterprise host responds with a menu of available actions presented as a multiple choice. Customer selects a report. Enterprise host receives her selection, retrieves the results of the report and sends them to the customer.

The transaction flow for this scenario is shown in Figure 3-1.

3.2 Operating a Wireless Device From the Internet

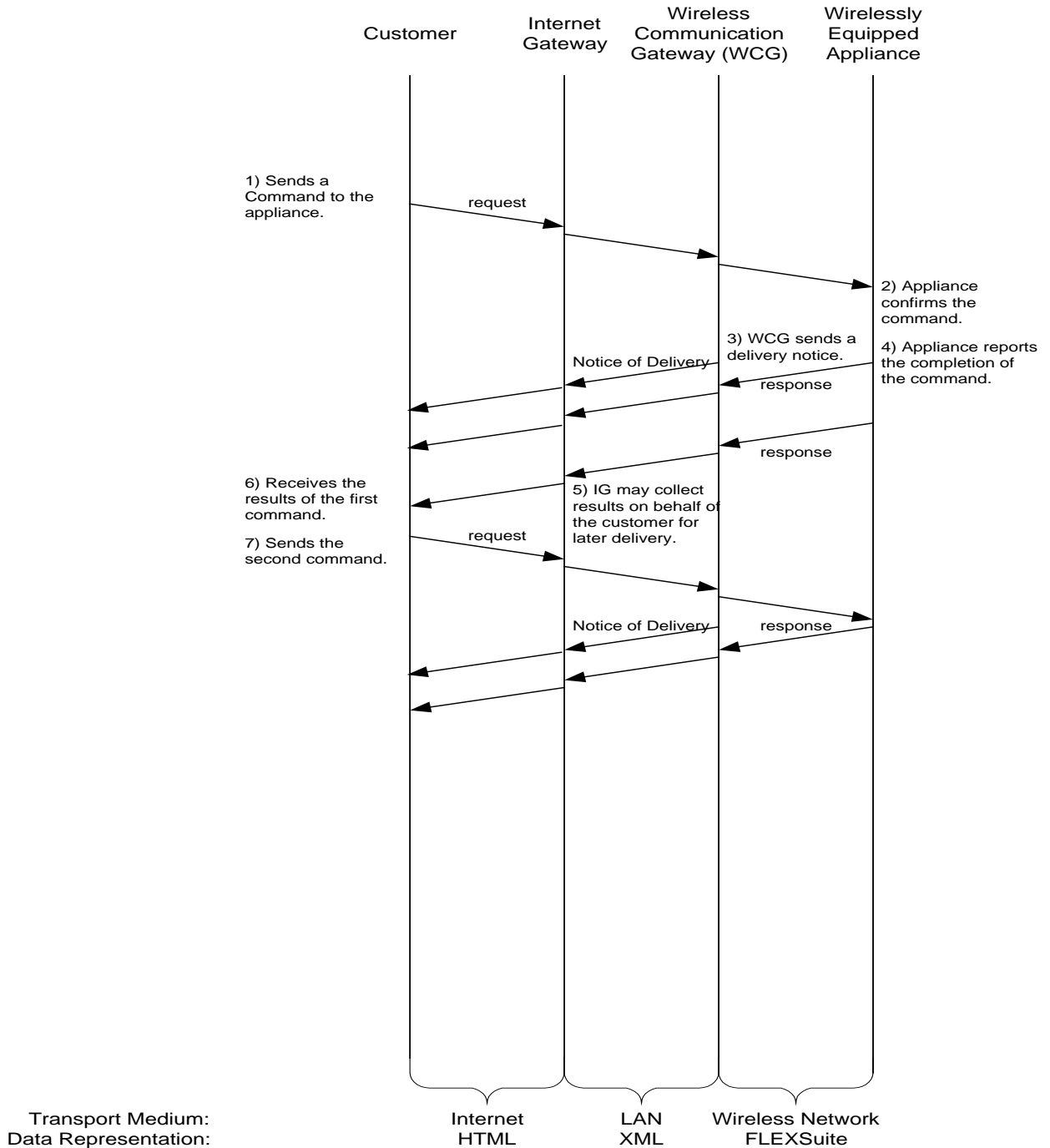
A network-connected customer wants to communicate with his wireless-equipped home appliance (for example, to program his VCR). Appliance supports a simple command set. Customer uses his browser to communicate with Internet Gateway. Internet Gateway presents a forms-based interface to the customer. Customer sends a series of commands to the appliance and receives responses with the results of command execution via Internet Gateway. Notifications of the delivery of the command to the appliance are issued.

The transaction flow for this scenario is shown in Figure 3-2.



000076 1/11/99

Figure 3-1. Wireless Person to Wireline Computer

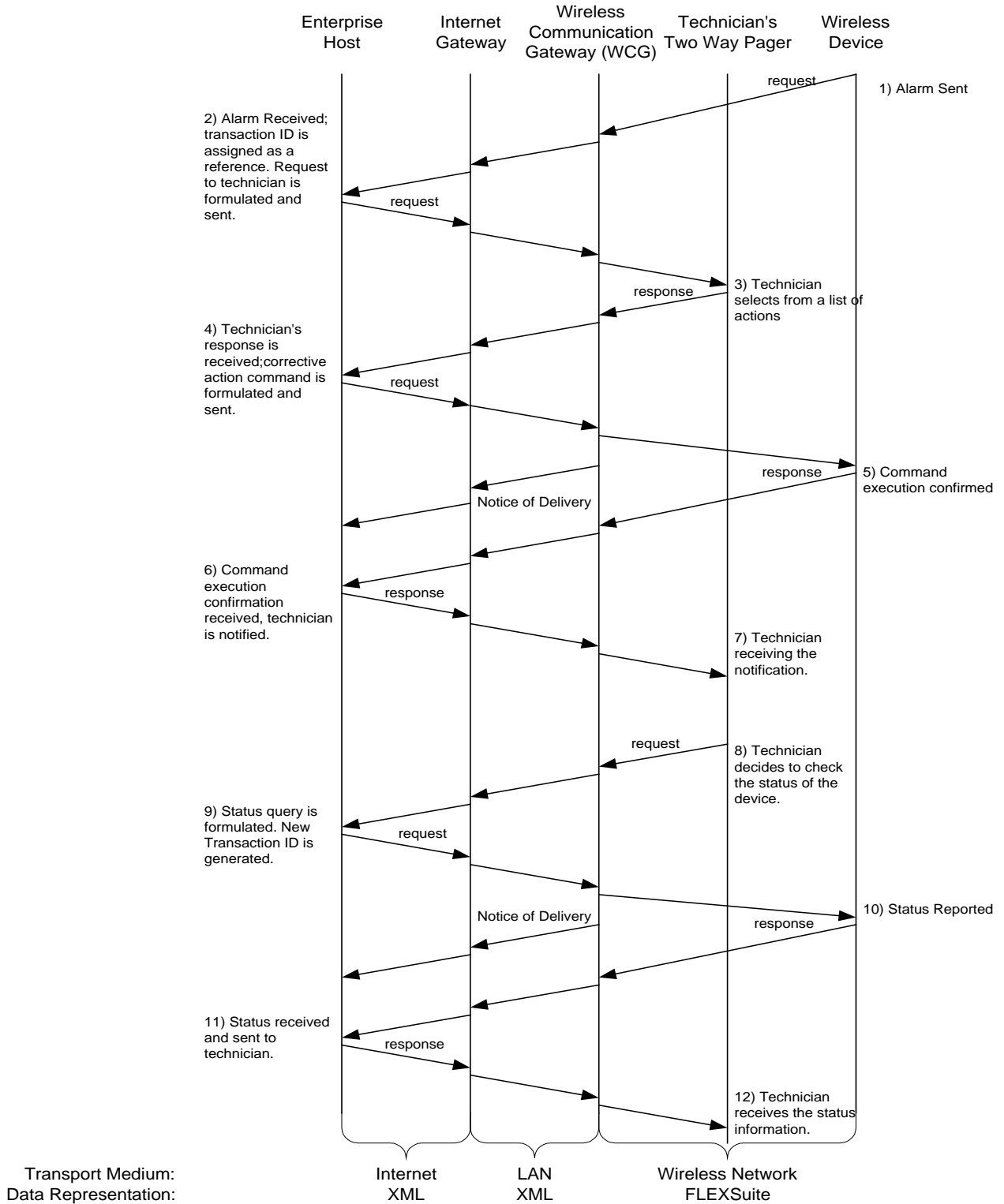


000075 1/11/99

Figure 3-2. Wireline Person to Wireless Device

3.3 Complex Multi-device Transaction

Wireless-equipped device registers abnormal condition and sends an alert to enterprise host. Enterprise host selects potential corrective actions and presents them to a technician equipped with a 2-way pager in a multiple-choice form. Technician selects corrective action in his response. Enterprise host sends appropriate command to the device. Device confirms that corrective action is executed. Enterprise host forwards notice to the technician. Later, the technician sends a request to enterprise host to inquire the state of the device. Enterprise host formulates a query and sends it to the device. Device responds. Enterprise host delivers status information to the technician. The transaction flow for this scenario is shown in Figure 3-3.



000065 1/11/99

Figure 3-3. Telemetry Device, Two-Way Pager and Wireless Host

4. WCTP Protocol Procedure

4.1 WCTP Communications through HTTP

The WCTP protocol defines a number of different control blocks which are transferred between a wireline system and an entry point, or Gateway, into the wireless network. Each control block is referred to as a WCTP Operation and is represented in an XML format as dictated by the format language syntax, or Document Type Definition (DTD) described in Appendix A. The Operation defines a function being requested or a status being reported. The Operation also carries mandatory and optional parameters associated with that function or status report.

Although any of a number of transport protocols may be used to move WCTP control blocks between a pair of systems, the HyperText Transfer Protocol (HTTP) has been initially selected as the recommended transport protocol for WCTP information. HTTP was selected as the protocol of choice because its use already addresses the issue of security and firewall protection typically found within the corporations that allow access into the Internet and are expected to be the primary users of the protocol. The firewalls maintained by the security departments of most corporations, normally allow HTTP requests to be initiated from systems which are within a corporate network. Because of this, WCTP will be able to be deployed in most corporate environments without co-ordination with the security administration of the network.

4.1.1 Pushing WCTP from the Wireline to Wireless Network

A wireline system sends (or pushes) information to the Gateway of the wireless network through the use of the HTTP protocol. This is the same protocol used by browser software to request and receive information from the World Wide Web and to forward information to Internet based servers to perform some function based upon the data forwarded (such as a search function, a database lookup, adding a name to a mailing list or ordering some goods). HTTP is a client/server protocol. An HTTP client always makes a request to an HTTP server.

HTTP supports two *major* processes. It may GET (or pull) information from a server (such as a Web page) or it may POST data to a server. Regardless of which process is requested, there is always a response sent to the request. The response always contains an HTTP protocol header field optionally followed by some returned data. HTTP opens a TCP/IP connection to send its request and the associated response is delivered on the same connection. Client software running HTTP version 1.0, opens a new TCP/IP connection to the server for each server request. HTTP version 1.1 allows the same TCP/IP connection to be used to process more than one request, and provides a more efficient means of transferring data between the client and the host for

reasons which are beyond the scope of this synopsis. Because of this extra efficiency, the WCTP protocol recommends HTTP version 1.1 be utilized when possible.

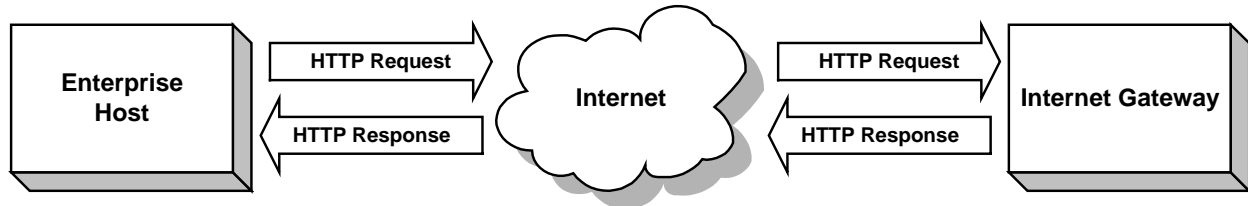


Figure 4-1. Pushing HTTP Requests From the Enterprise Host

In WCTP, the POST process is used to send a WCTP formatted request (WCTP Operation) to the WCTP server. The response by the server is a WCTP XML formatted response (another WCTP control block or Operation). The types of control blocks and their formats are discussed later in this section. Figure 4-1 shows the normal call flow from the client to the server when a wireline client sends a request to the wireless network.

4.1.2 Pushing WCTP From the Wireless to Wireline Network

The wireless network can forward data to the wireline network in the HTTP response to a previous HTTP request, as shown in Figure 4-1. But when the wireless network has unsolicited information available, it would like to push this data out to the wireline system. There are several methodologies employed within the Internet today to "Push" information from servers to client machines. Several of these Push technologies were considered by the WCTP committee. Two forms of push technology are currently recommended by this specification.

The recommended means of pushing information from the wireless network to the wireline system is also through the use of the HTTP protocol. In this case, the TCP/IP connection is initiated by the wireless network and an HTTP request carries a POST operation which contains an XML formatted WCTP Operation as its data content. The WCTP Operation carried in the HTTP response from the wireline system, indicates if the data sent has been accepted, processed and/or safe-stored by the wireline system. If so, the wireless network can drop the WCTP Operation from its queues, knowing that the information has been properly conveyed to the wireline system. Figure 4-2 shows the call flow of WCTP Operations pushed from the wireless network to the wireline system.

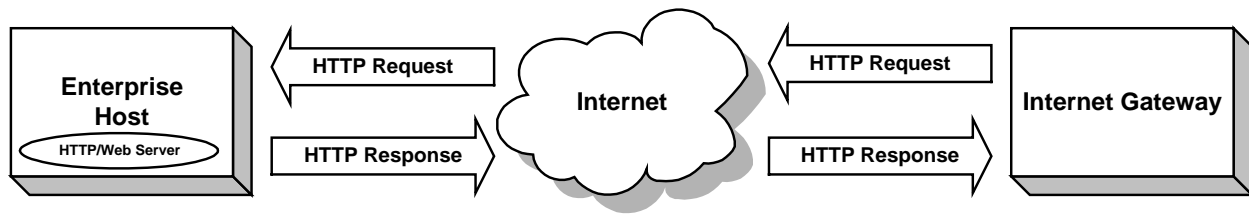


Figure 4-2. Pushing WCTP Operations From the Wireless Network

The HTTP protocol has been selected as the transport protocol of choice to push data to the wireline systems, again because of security and firewall issues. Many corporate networks are already operating Web servers where this HTTP data may be forwarded for processing. Special applications on these Web servers could process the received data without having to open special TCP ports in firewalls.

The major requirement to support this type of push technology, is that the wireline system must be running some type of HTTP based web server-like software in order to accept and process the WCTP Operations pushed from the wireless network.

In support of this form of push, addressing information which is carried by the WCTP protocol, knows the host and domain name of the wireline server which is to process the pushed data from the wireless network. The standard WCTP message routing mechanism derives the host and domain name from the domain portion of the recipient ID as it is described in section 4.4.

The HTTP protocol also allows "path information" to be included as part of the request. This path can signify the location of particular applications software on the server which is to specifically process the WCTP Operation being sent. In the case of a wireless device sending a message to wireline host for processing, this path information can be forwarded by the wireless device in order to specify that a specific application should process this message. WCTP provides a special operation, which may be implemented on the WCTP servers of wireless networks to simplify this form of routing of messages from the wireless device to an application on the wireline network. This operation also allows the wireless device to send fewer characters over-the-air while still providing for the same form of routing. A special *Bind* Operation allows the wireline system to register a unique alias name and associate a Universal Resource Indicator (URI) to be used for the push operation when that alias name is received from a wireless device. By using the alias form of addressing, the wireless device may send a few characters which could be expanded into a fairly lengthy character sequence to be posted to the wireline system. In addition, the *Bind* Operation allows the wireline system to change the destination application to process unsolicited data, without having to re-program destination routing information in the wireless device.

4.1.3 "Pseudo-Pushing" of WCTP from the Wireless to Wireline network

If the wireline network is not able to run an HTTP server to accept pushed data from the wireless network, a pseudo-form of pushing is used. When this form of push is supported, the wireless network queues the data which would have normally been pushed, and instead forwards this data in response to a *Poll For Messages* Operation which is sent from the wireline system through an HTTP request as shown in Figure 4-1. The response to the *Poll For Messages* may indicate that there is no data to return to the wireline system, or it may return one or more queued WCTP Operations in response. The *Poll For Messages* Operation can control the maximum number of possible WCTP Operations returned. If any WCTP Operations are returned, a subsequent *Poll For Messages* must carry information which indicates that the returned WCTP Operations have been acknowledged by the wireline system so that the wireless system may remove them from its queues.

For a wireline system to support this manner of push, they must be registered at the wireless network to allow a polling mode of operation. The wireline system employing polling, has a unique host identifier in the form *host.domain*, which is being used as a part of sender or recipient ID, registered with the wireless network. When this host identifier is encountered in the domain portion of the recipient ID of the WCTP Operation, the wireless network will hold the data in its queues, which it would have normally immediately pushed out using the mechanism of section 4.1.2. Several of these queued WCTP operations could be returned when the *Poll For Messages* Operation is received by the wireless network. Further discussion of message routing and delivery is presented in the section 4.4.

4.2 Two Categories of WCTP Users

There are two major categories for wireline systems, which utilize the WCTP protocol. These are the Enterprise Host and the Transient user. There are specific WCTP protocol operations for each type of user as will be discussed in this section.

The Enterprise host is a system, which operates on behalf of one or more message senders and/or message processing applications. It is capable of receiving unsolicited messages from wireless devices. These unsolicited messages may be directed to a general or specific processing application at the Enterprise host as discussed in the following sections. The wireless network will normally push status, message replies and unsolicited messages to the wireline host as soon as such information is ready to send as described in section 4.1.2. In those cases where security considerations or host limitations do not permit the pushing of information, the Enterprise host will use the pseudo-push mechanism described in section 4.1.3.

The Transient user is categorized as a message sender who may not have a permanent connection to the Internet and periodically sends messages to one or more wireless devices. The transient users may receive message status information and replies for specific messages, which they have submitted into the wireless network. They may not receive unsolicited messages from wireless devices. They can not have data pushed to them as described in section 4.1.2. Instead, they must use *Client Query*

operation to query the wireless system for information using a polling mechanism similar to that described in section 4.1.3, but only capable of returning responses to a given single original message.

4.2.1 Enterprise Hosts

In the WCTP environment, the Enterprise host is expected to act as a Gateway for one or more machines within the Enterprise. Messages may be originated from one or more users or applications at individual machines connected on a LAN or WAN to the Enterprise host. It is expected that the Enterprise host will accumulate these messages and submit them to the wireless network as if originating from a common host. Responses on behalf of these messages will be returned to the Enterprise host, which will then direct the responses to the proper recipient or application.

Unsolicited messages from wireless devices are addressed to the Enterprise host designated to process such messages, by addressing these messages in a manner described in the sections 4.3 and 4.4 of this specification.

In WCTP the Enterprise host sends information to the wireless network through the use of the HTTP protocol. It is recommended that HTTP version 1.1 (or better) is utilized, as this version of the protocol permits the transmission of multiple operations on a single TCP connection.

This release of the WCTP protocol assumes that the Enterprise host is capable of accepting messages pushed via the HTTP protocol. This may be accomplished if the Enterprise host is already running a Web Server or other application, which is capable of receiving HTTP POST's. Where this is not feasible, the "pseudo-push" technique is employed (see section 4.1.3).

Messages submitted by an Enterprise Host are assigned a Message Identifier (messageID) by the host. This Message Identifier is returned to the host in all replies and responses to the submitted message.

4.2.2 Transient Users

A WCTP transient user may submit messages to many different wireless devices. But this type of user does not receive a consolidation of all of its status responses, delivery notifications and replies returned to it automatically as in the case of the Enterprise host. Instead, the transient message sender may check for data regarding previously submitted messages. In support of such checking, the system returns a message tracking number whenever a message is submitted. This tracking number is used in subsequent *Client Query Operations*.

4.3 Addressing in WCTP

4.3.1 Identifiers for Wireless Devices

Identifiers of wireless devices have the following form:

[Wireless-Device-Identifier\[@Wireless-Carrier-Domain\]\[Path\]](#)

Wireless-Device-Identifier can represent a PIN of a wireless device, its numeric access code or a user alias established for it by its wireless carrier.

Wireless-Carrier-Domain is an Internet domain of a wireless carrier established for wireless communications.

Path is a path for an application running on a wireless device.

An "at" sign ("@"), the Wireless-Carrier-Domain and the Path are optional fields. If a message is originated from a wireless device and the Wireless-Carrier-Domain is omitted, it should be assumed that the Wireless-Device-Identifier specifies a device on the same wireless network where the message was originated. If a message is originated from a wireline host and delivered to a WCTP server of a wireless carrier and these fields are omitted, it should be assumed that the Wireless-Device-Identifier specifies a device on that carrier's network.

Examples:

```
9396655
1234567@abcwireless.com
8005551212.98765@abcwireless.com
john_doe@abcwireless.com/chess_app
john_doe/chess_app
```

4.3.2 Identifiers for Wireline Hosts

Identifiers of Enterprise Hosts have the following general form:

Entity@Domain[:Port][Path]

When used as a part of a recipient ID, **Domain** defines the address of the WCTP server where WCTP message is being delivered. Furthermore, the domain portion is used to determine if the message is to be pushed to the Enterprise host or held for polling by the Enterprise host.

Entity can represent a user name, an application alias or carry any other Enterprise host specific meaning. This information is carried in the senderID field of the Enterprise host originated message or a recipientID field of a message directed to the Enterprise host.

Both the optional **Port** and **Path** present additional routing information to the WCTP Enterprise host server. That information can be used together with Entity for internal message routing and processing.

Examples:

```
inventory@myenterprise.com
joe.user@myenterprise.com/cgi-bin/exchange-gateway
accounting@myenterprise.com/processing/cgi-bin/receive.exe?dev=WR2XYZ
```

The format of identifiers of Transient users in WCTP is not fixed. Normally, uniqueness of such an identifier cannot be guaranteed. Therefore, wireless systems cannot rely on the sender identifier for messages submitted by Transient users. For Transient users the wireless system must rely on the uniqueness of an assigned tracking number or a combination of tracking number and recipient identifier of the original message for delivery of responses. The senderID, specified on a submitted message by a transient user, must be specified for authentication when polling for status.

4.3.3 Recipient Specification on Wireless Devices

Some wireless devices have the capability to distinguish the destination routing type of unsolicited messages as being a PageTo or MailTo destinations. It is assumed that if the destination is specified via PageTo address, the WCTP protocol will be used to deliver the request when it is directed outside of the wireless carrier's own network. If the destination is specified as MailTo, the message will be delivered via SMTP protocol.

As an exception, if a wireless device does not support the alphanumeric PageTo specification or does not distinguish between PageTo and MailTo addresses, use of the WCTP protocol for message delivery can be identified by a "**msg:**" prefix in front of the destination address. For example: msg:5169396655@othercarrier.com.

4.4 WCTP Addressing and Message Routing

4.4.1 Message Delivery to Enterprise Hosts

Normally, when information is to be sent to the Enterprise host, it will be sent via the HTTP protocol through an HTTP POST operation. The Enterprise host must be running a Web Server or other application which is capable of receiving HTTP POST's in order to accept this information.

WCTP specifies two mechanisms to route data to an Enterprise host. They both involve the use of a Universal Resource Indicator (URI)¹ derived from the recipient identifier. The **Standard Routing** mechanism relies on a derivation of the name of the target Enterprise host from the domain portion of the recipient identifier by prepending a "**wctp.**" prefix (unless it is already a part of the domain specification). For example, a recipient of the message is specified as:

inventory@myenterprise.com

or as:

processing@myenterprise.com/cgi-bin/request.pl

or as:

joe.user@myenterprise.com/exchange/cgi-bin/gateway.exe

¹ In the initial release of HTTP protocol URI was referred to as a URL - Universal Resource Locator.

In all these cases the domain portion of the address is **myenterprise.com**. The name of the target Enterprise host in all these cases will be derived as **wctp.myenterprise.com**.

If the *path* portion of the address is not specified, default path **/wctp** is assumed. This allows a wctp server to be co-located on the same host with a Web server while maintaining short format of the address. To demonstrate the expansion of short form of address, in the examples above the the connection will be made to the address of wctp.myenterprise.com. The POST command of the HTTP protocol in the first example will be formed as:

```
POST /wctp HTTP/1.1
```

If the *path* portion of the address is explicitly specified, it is used as is. Thus in the second example above the POST command would be formed as:

```
POST /cgi-bin/request.pl HTTP/1.1
```

and in the third example:

```
POST /exchange/cgi-bin/gateway.exe HTTP/1.1
```

The specification of the recipient identifier inside WCTP message is preserved intact (full [Entity@Domain:Port/Path](#) designation) and delivered as a part of WCTP Operation (within the recipientID field) to the Enterprise host. The Enterprise host can use the extra information presented as entity and/or path for the delivery of the message content to a locally running application or to another host on a corporate network.

In the 3 examples above, *inventory* might be treated as a reference to a particular application, whereas *processing* and *exchange* could be the names of other servers on on MyEnterprise's intranet. The Entity name *joe.user* could be an alias of a user at the Exchange server. Such determination is beyond the scope of the WCTP and is particular to the implementation of the Enterprise host.

The second mechanism WCTP supports is known as **Alias Routing**. It is also used to simplify the addressing of messages which originate from a wireless device. If an alias is being employed, the Enterprise host may register an alias domain name with the wireless network. A message originating from a wireless device may include this alias as the domain portion of the recipient identifier. The wireless system will POST such a message to the full URI, which has been associated with the alias instead of delivering it via Standard Routing. The use of alias names reduces the number of characters, which must be sent over-the-air from a wireless device in order to route data to the message recipient. Also, an alias can be used to alleviate the requirement for an Enterprise host to comply with "wctp." naming convention. For example, the alias name "abc-status" could be assigned to represent the URI http://www.abc-messaging.com/processing/cgi-bin/status.exe. This form of a POST can use a Web server's ability to directly invoke a specific application at the server. The HTTP connection in this case will be established to <http://www.abc-messaging.com> with the following POST operation:

POST /processing/cgi-bin/status.exe HTTP/1.1

WCTP provides means for establishing domain aliases via *Bind Domain Alias* Operation. An alias may also be established by a wireless carrier through the assistance of a system administrator.

It is important to note that a domain alias for an Enterprise host is established for each wireless carrier independently. Therefore, the Enterprise host must establish its domain alias on all wireless networks from which it needs to be able to receive messages.

It is strongly recommended to use the Standard Routing whenever possible.

4.4.2 Message Delivery to Wireless Carriers

To be able to receive messages via WCTP, wireless carriers must adhere to the Standard Routing convention and establish WCTP servers at `wctp.Wireless-Carrier-Domain`. Both Enterprise hosts and Transient users will use HTTP POST to deliver messages addressed to `recipientID=device-id@Wireless-Carrier-Domain` to the URI `wctp.Wireless-Carrier-Domain`. For example, the HTTP POST command to deliver a message with `recipientID=john_doe@abcwireless.com` is

POST `http://wctp.abcwireless.com/wctp` HTTP/1.1

in accordance with the default address expansion rule discussed in the previous section.

However, it is important to note that the WCTP servers of a wireless carrier treat identifiers of wireless devices in a specific way, because of the role these servers play in front-ending deliveries to wireless devices. As it was highlighted in section 4.3.1, *path* portion of the `recipientID` for the wireless device refers to the application running on the *wireless device*, and **not** on the WCTP server of the carrier. For example, a message addressed to `1234567@abcwireless.com/chess` refers to the *chess* application running on device `1234567` of `abcwireless.com`'s wireless network. Therefore, the WCTP server of a wireless carrier has to be implemented in such a way that the *path* portion of an HTTP submission of a WCTP message is ignored and all the submissions are forwarded to the default WCTP server application.

4.4.3 Message Response Redirection to non-WCTP Destination

WCTP allows specification of where the responses to a particular message are to be sent via `sendResponsesToID` attribute of `wctp-MessageControl`. Wireless Carriers may support redirection of responses to non-WCTP destinations, like e-mail, faxes, phones, etc. To support this feature WCTP reserves 4 prefixes that can be specified in `sendResponsesToID`: **mailto:**, **faxto:**, **phoneto:** and **pageto:**.

WCTP server may reject messages requesting response redirection if the requested method of redirection is not supported.

5. WCTP Operations

5.1 WCTP Operations Summary

The WCTP protocol defines a number of "operations", each of which is represented as a "control block" in an XML form. The operations currently supported by the protocol are summarized below. The names of the various operations are of the form "wctp-OperationName".

1) Submit Message Operations

- 1.1) In general, a message is any binary data. As a subset of this generalized format are alphanumeric messages in human readable form. Submission control blocks may also carry an indication of a particular form for the otherwise transparent binary information. For example, binary messages, which follow the Motorola FLEXsuite™ defined form are tagged as such in a transparent data type.
- 1.2) A message may be submitted by the wireline system.
- 1.3) A message submit operation is formed by the wireless network infrastructure equipment on behalf of a wireless device when an unsolicited message is received and is to be delivered to another system component in WCTP form.
- 1.4) There are two types of message submit operations. One is used for Enterprise hosts (*wctp-SubmitRequest*) and the other for transient users (*wctp-SubmitClientMessage*).
- 1.5) Implementations may impose a maximum message submission rate to the wireless network. If this rate is exceeded, an error response code is returned in lieu of accepting the message, to indicate that the allowed input rate has been exceeded. If this rate is exceeded an excessive number of times, another error response code may be returned indicating that the sending system is subject to being disabled from sending further messages.

2) Status Return Operations

- 2.1) There are several status return operations.
- 2.2) Whenever data is moved between a client and a server a first level status operation must be returned. The status indicates if the message has been successfully or unsuccessfully received so that the sender knows that the data has been safely received and can be removed from its queue of data to transfer. *wctp-Confirmation* operation is used for this purpose in interaction between Enterprise hosts and WCTP

Gateways. *wctp-SubmitClientResponse* is returned when Transient users submit a message (via *wctp-SubmitClientMessage*).

- 2.3) Message status may be sent unsolicitedly (*wctp-StatusInfo*) upon the detection of a change of state of a message. The status could be in the form of a notification of the message processing state (message queued or message delivered, for example), if such notifications were requested when the message was first sent. The status could also be in the form of a failure indication with an explanation. The failure could be indicating that the message was determined to be undeliverable for any of a number of reasons. This status could be returned quickly if, for example, it was determined that the destination ID is invalid (some implementations may not be able to check validity at the point of entry into the network and may instead perform this validation step after the message goes through the first level of acceptance). The failure status could come at a much later time if, for example, a timeout response is returned because the message could not be delivered within a time interval specified.
- 3) Return Reply Operation
 - 3.1) A reply may contain a message that has been generated to specifically respond to a message previously submitted. WCTP provides threading information to tie such a response (*wctp-MessageReply*) to the original message submitted. More than one response could potentially be returned for one message submitted.
- 4) Poll Operations
 - 4.1) Polling is used by an Enterprise host when information may not be automatically pushed to it because of security reasons or the lack of support for an HTTP server. A Poll operation (*wctp-PollForMessages*) is used by an Enterprise Host to request and potentially receive (via *wctp-PollResponse*) information (status or replies) on behalf of many messages which were previously submitted. A Poll operation may also return many unsolicited messages which are to be processed by the Enterprise host.
 - 4.2) In order to speed up the transfer of information queued at the wireless network for transfer to the host, a Poll Response operation (*wctp-PollResponse*) may return several WCTP operations (status responses, replies, delivery notifications or unsolicited messages). The system which is performing the polling operation may limit the maximum number of WCTP operation responses that may be returned, should a large number of responses be queued for delivery.
 - 4.3) The maximum number of WCTP operations returned by a single Poll Response, is implementation specific. The maximum number actually returned is the smaller of the number of responses queued, the

implementation specific maximum number returnable and the maximum number of responses specified with the Poll command.

- 4.4) Whenever there are WCTP operations returned by a Poll Response, a response sequence number is assigned to each operation returned. The sequence number of each operation received must be sent in a subsequent Poll operation to acknowledge its receipt. The Poll operation allows a list of response sequence numbers to be specified, so that the acknowledgement of receipt of multiple WCTP operations may be performed at once.
 - 4.5) If a sequence number of a particular WCTP operation returned in a response to a poll, is not acknowledged in a subsequent Poll operation, the WCTP operation will be resent in response to the following Poll operation.
 - 4.6) When there are no further WCTP operations queued at the wireless network for return to the Enterprise host, a specific indication (*wctp-NoMessages*) is returned within *wctp-PollResponse*.
 - 4.7) The wireless network may impose an implementation-specific maximum polling rate. A system may impose two maximum polling rates. One rate when a poll results in no further queued messages and a higher allowed rate when WCTP operations are returned in the poll response. System error response codes are returned in lieu of a Poll Response, to indicate when a polling rate has been exceeded. Other error response codes indicate when polling rates are excessively exceeded. If excessive polling continues, the wireless network may permanently disable the polling system from access to the wireless network.
- 5) Query Operations
- 5.1) A Client Query operation (*wctp-ClientQuery*) is a form of a poll operation used by Transient users to check for responses to a specific previously submitted message. The response to a Client Query operation (*wctp-ClientQueryResponse*) could indicate that no information is available or it may return a collection of replies and message statuses.
 - 5.2) A Subscriber Lookup operation (*wctp-LookupSubscriber*) is used by Enterprise hosts to query the WCTP server about the capabilities of a wireless device. The response to a Subscriber Lookup operation (*wctp-LookupResponse*) specifies a set of attributes describing capabilities of the device. Such information can be used to appropriately format the following submitted messages.
- 6) Bind Operation
- 6.1) The Bind operation (*wctp-BindDomainAlias*) permits an alias name to be associated with an address which will be utilized to route unsolicited

messages, status and replies to an Enterprise host. A few different types of bind operations exist to maintain the alias association information.

5.2 Call Flow Between Network Elements

The operations specified in the previous section are used to convey information between the wireline and wireless network and possibly between elements within the wireless network itself. Figure 5-1 shows several call flows between the Enterprise Host and other network elements, which were described in section 3 of this document. The diagram is assuming that messages from the wireless network that need to be returned to the Enterprise host, are being pushed directly to the host using the recommended HTTP POST methodology.

The call flows are read from the top of the diagram down, showing the types of operations that may be moving between elements in normal and error cases. The diagram also shows cases in which the user of a wireless device is responding to a specific message as well as a case in which an unsolicited message is generated from the wireless device. It should be pointed out that some wireless two-way technologies support the direct association of a message response against a specific message while some other wireless devices may send such responses as new, seemingly unsolicited messages which are not tagged to the original message. Where the device itself is not capable of tagging subsequent responses, the method of associating transmissions and responses are outside of the scope of the WCTP protocol.

Figure 5-1 helps to point out several items for the WCTP application developer on the wireline side as well as to the infrastructure equipment suppliers on the wireless network side. The definitions of the WCTP control blocks (operations) themselves do not make it apparent how a call flow may proceed. In fact, the sequence of operations which may occur, could be dependent upon the nature of the transport protocol being utilized as explained in the following paragraphs.

Referring to the figure, it is shown that when an operation is submitted (set #1) , it is possible to get an XML formatted error response posted back. The Submit operation does carry optional parameters (referred to as notifyWhen type parameters) that indicates the requirement to post back one or more delivery status conditions to the message submitter. When the requested status condition occurs, it will be queued for return to the submitter. The "Acknowledged" status shown in the second Submit in the call flow (set #2), is not one that it optionally requested. This status return is provided for implementations in which an error or status response may not be immediately available to return to the submitter, but the transport protocol being used requires a return response before closing the connection. This is the case of the submit/response HTTP protocol which is expected to be used as the primary WCTP transport protocol. The second Submit operation shown in the figure (as set #2), is assumed to be operating over an HTTP type connection. Although the Submit request received was accepted by the Internet Gateway element of the wireless network, the diagram is attempting to show the generalized case in which the operation may be determined to be un-executable or in error at some later point in its processing (such as when it is

processed by the WCG element within the wireless network). That determination may not be able to be made in a "timely" fashion and it is not desirable to leave HTTP transport connections open for longer than they need to be. The "Acknowledged" response (success code 200 of Appendix C) is returned in order to acknowledge the receipt of a valid WCTP operation which it may or may not be able to be performed.

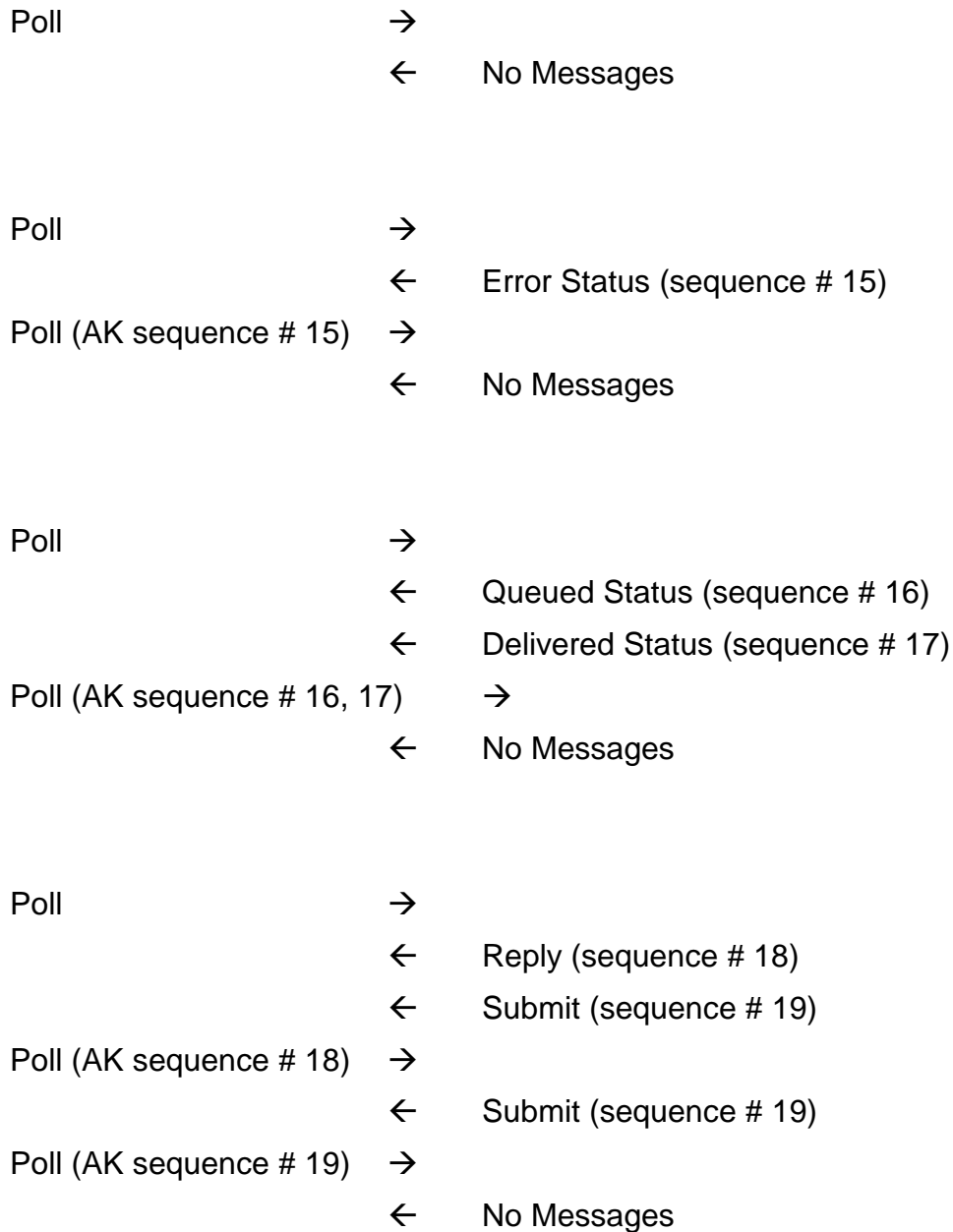
From the point of view of the creator of a WCTP application, it is only necessary to understand that an error response to a Submitted request may be returned to the application some time in the future. That a Received response status is only an intermediate response condition and does not necessarily mean that the operation was acceptable for execution.

The third Submit operation in the figure (as set #3), shows where a valid Submit operation was presented. When the message being submitted is validated, properly queued and safe-stored for delivery within a network specific element of the wireless Network, a "Message Queued" delivery status may be returned to the message sender (set #4 of the figure) to indicate that the requested operation is pending. The diagram shows the Queued Status being pushed to the Enterprise host through the HTTP protocol. As required by HTTP, a response must be sent to the pushed Queued status. That response is sent as a WCTP Acknowledgement operation (wctp-Confirmation) which confirms the guaranteed receipt of the pushed data.

The example shows the pushing of a Delivered Status notification (set #5) upon the successful delivery of the message to the wireless device, because the Submit operation had requested a "NotifyWhenDelivered" status report. When the user of the wireless device replies to the received message, a WCTP "MessageReply" type operation (set #6) is pushed to the Enterprise Host.

The example also shows the case where a wireless device initiates an unsolicited message (set #7). This type of message is pushed to the Enterprise Host as a WCTP Submit operation initiated by the wireless network. The Enterprise host may send a response to the unsolicited message (set #8) by initiating a "MessageReply" operation against the MessageID assigned by the wireless network.

The diagram shows the pushing of an Error, Queued and Delivered Status, a Reply and an Unsolicited message, to the Enterprise host. If the Enterprise host is not capable of supporting the HTTP protocol to receive this pushed information, it would have been registered with the wireless network as supporting Polling to receive the WCTP operations representing these messages. Periodically, the Enterprise host would issue a Poll to check for messages the wireless network has queued for delivery because it is unable to "push". The sequence of operations by the Enterprise could have been as follows:



When polling is being used to receive queued messages from the wireless network, the poll messages are also used to acknowledge the positive receipt of the messages returned by the wireless network. If this acknowledgement is not returned with the subsequent poll, non-confirmed messages will be returned again.

Subsequent Poll operation may be submitted by an Enterprise Host immediately after the receipt of the messages from the wireless network, unless the wireless network sent No Messages indication or explicitly specified the next poll interval.

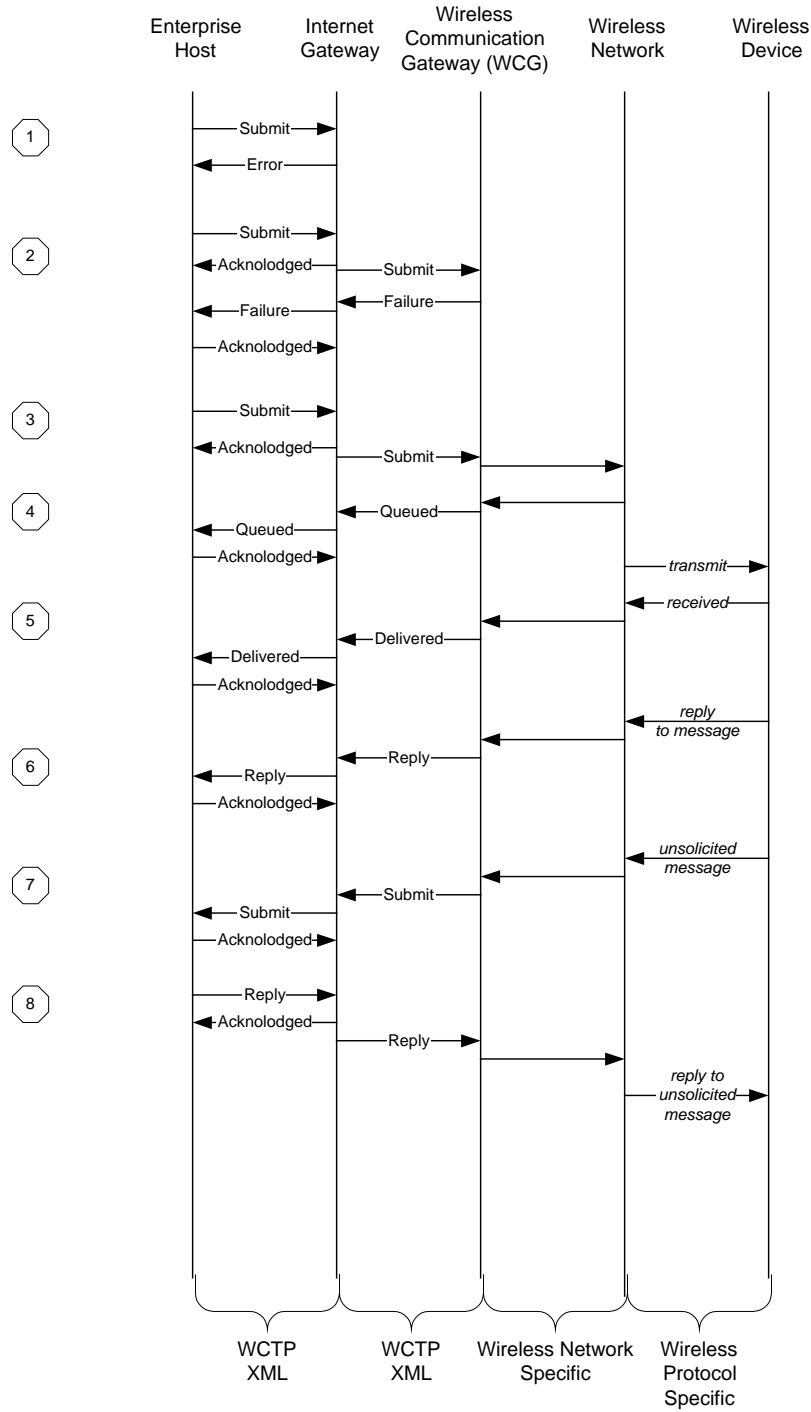


Figure 5-1. Call Flow Between Network Components

5.3 WCTP Submit Message Request Operation

The Submit Message Request operation of WCTP (wctp-SubmitRequest) is used to initiate a new message

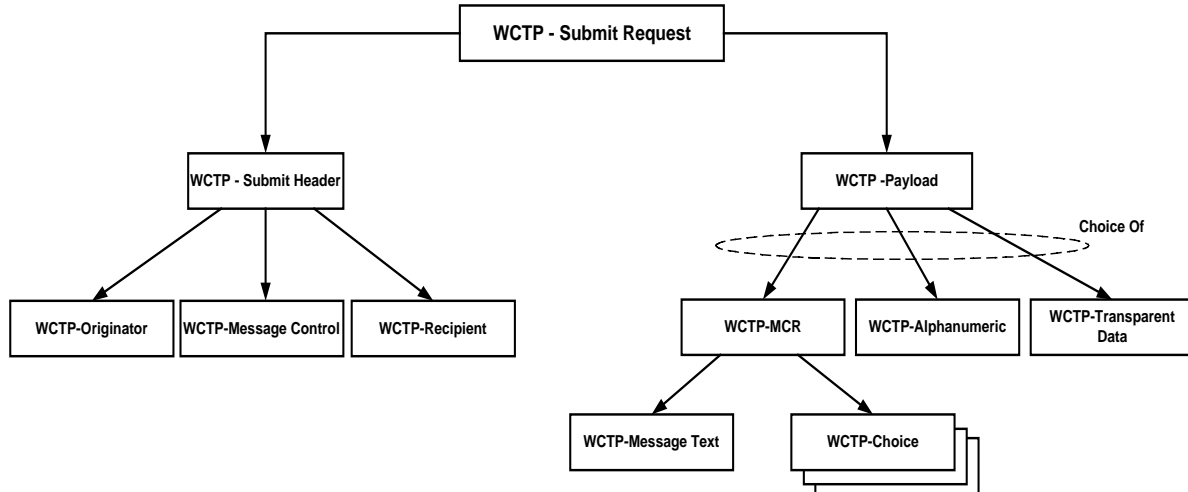


Figure 5-2. WCTP Submit Request

from the Wireline Enterprise Host or Wireless system. Figure 5-2 shows the major XML formatted elements of the Submit request. Each of these elements may specify a number of different attributes or parameters, which are associated with the element. Appendix A contains a copy of the Document Type Definition which formally specifies the XML syntax of the Submit Message Request operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

The Submit request has two major elements, the Submit Header and the Payload.

5.3.1 Message Payload

The Payload contains the message content, which is to be delivered. This content is contained within a structure that is dependent upon the data type being carried. General binary type messages, referred to as Transparent Data, may be further tagged as to a specialized form of the binary data, such as FLEXsuite. XML provides a standard canonical form for data. Numeric, alphanumeric and some 8-bit data may be specified within this form. The 8-bit binary data can be represented as an ASCII string which encodes a hexadecimal value (e.g. Ñ represents the binary octet 11010001). However, not all octets can be encoded in XML. In particular, most of the “special characters” with hexadecimal values 0x00 to 0x1F cannot be encoded in XML and considered illegal characters. Refer to the referenced XML documentation for further details.

For transfer of arbitrary binary data WCTP supports *base64* form of encoding. Specification of the encoding method is made via the *encoding* attribute, with canonical XML representation being a default.

Multiple Choice Messages (MCR) may be specified through the Submit Message Request. An MCR message is a menu of selections. The current support of MCR in WCTP allows the wireless user or application on the wireless device to specify a reply which selects one item from the list of choices received. A subsequent Message Reply, tagged back to the original message, will contain the actual text selected. Since it is the actual text returned as opposed to the index of the MCR selection, the list of choices should contain unique strings of text. If the same text appears within multiple selections, the information provided in the MCR response, does not provide sufficient information to know specifically which index of the MCR message was selected when the response was generated. If this information is required, the list of choices must contain unique strings of text. The support and implementation of MCR by a wireless device or application is device or application specific.

The MCR response message does permit responses to be returned which are not members of the list of choices. This permits an application to provide a list of "suggested" responses while still permitting the wireless device to return its own specific response. Some wireless devices, such as those that integrate reFLEX™ technology, provide an over-the-air optimized methodology for supporting this capability. Other wireless devices could implement the MCR type capability through special application code operating in the wireless device or a computer directly connected to the two-way wireless transceiver. If specialized applications code is used to support the feature, the Wireless network processing the WCTP requests must be aware of manner in which the capability is implemented in order to properly activate the MCR function of WCTP. The current release of WCTP does not support the return of multiple selections at one time.

5.3.2 Submit Header

The message header provides information as to the message originator, message recipient, and many message control parameters. One of the key parameters is the Message ID. This is a unique message identifier assigned by the message originator. This identifier will be contained in any subsequent responses directly returned on behalf of the submitted message.

Other Message Control parameters allow the message sender to control when a message is to be delivered and what types of responses should be reported. It is also possible to direct those responses to be sent to a different Enterprise host or application than the sender of the original message.

A transaction ID may be submitted with the message. If such an ID is included, it will be returned in subsequent responses. A transaction ID is used by the message originator to tag a group of individually submitted requests. These requests might even be sent to a number of different devices that may even span several different wireless networks. The transaction ID may be used to indicate that all of these requests were part of a particular instance of an application on behalf of a sequence of operations that the

application may have been performing. In our complex multi-device application of section 3.3, each of the individual operations performed would have had the same transaction identifier as they were all involved in resolving a particular alarm condition that was first reported through a message from a remote telemetry device. If that same device had initiated a second alert due to another alarm condition occurring, presumably a different transaction identifier would have been associated with the sequence of operations, which would be performed to resolve that independent alarm condition.

5.4 WCTP Message Reply Operation

The Message Reply operation of WCTP (`wctp-MessageReply`) is used to report message replies returned on behalf of other messages. This includes message replies returned on behalf of yet other message replies. Figure 5-3 shows the major XML elements of the Message Reply operation. Each of these elements may specify a number of different attributes or parameters, which are associated with the element. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Message Response operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

The Message Reply contains two main elements, the Message Response Header and the Payload. Specification of the message payload follows the same rules as listed in the Section 5.3.1.

5.4.1 Response Header

The Response Header provides information with regard to the message originator, the message recipient, and control information to further specify how the response should be processed. The Originator portion of the Response Header indicates who is initiating the response. In the case of a response from a wireless device, this would specify the unique identification code of the device as specified by the wireless network on which this device operates.

The Recipient information of the Response Header would normally contain the address of the sender of the original message. If, when the message was submitted, it was indicated that responses should be sent to a different destination other than the message originator, this destination information will appear in the Recipient Header instead.

The Message Control element of the Message Response operation contains the unique Message Identifier, which ties the response back to a particular submitted message associated with the message originator. The structure of the Message Control element is identical to its namesake from the Submit Header of the Submit Message Request operation (see Section 5.3.2).

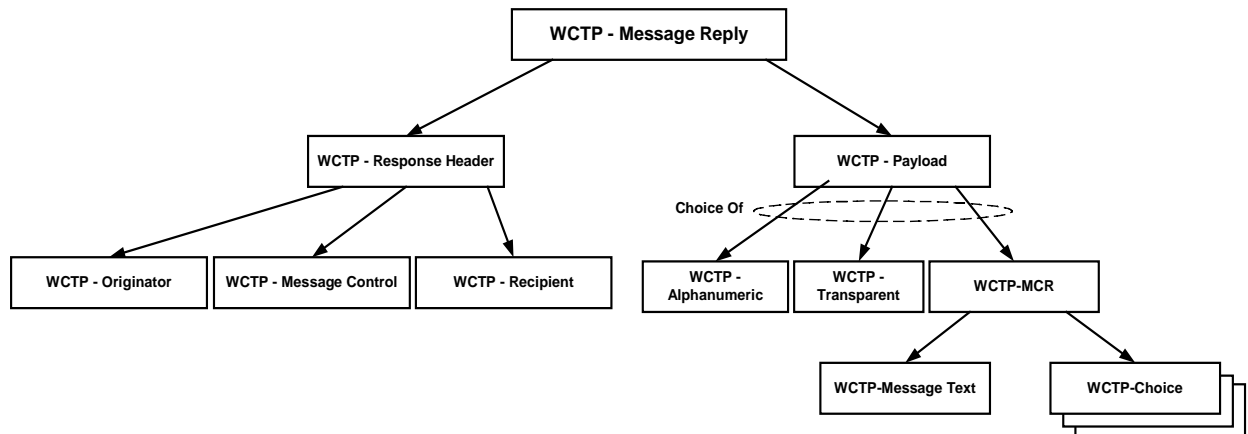


Figure 5-3. WCTP Message Reply Operation

5.5 WCTP Status Info Operation

The Status Info operation of WCTP (wctp-StatusInfo) is used to report both message delivery failures as well as delivery notifications returned on behalf of other messages. Figure 5-4 shows the major XML elements of the Status Info operation. Each of these elements may specify a number of different attributes or parameters, which are associated with the element. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Status Info operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

The Status Info contains two main elements, the Response Header and either a Failure or a Notification.

The Failure response may indicate a permanent or temporary failure of message delivery. The specifics of the failure condition are reported via a standardized error code and optional error text.

The Notification response may indicate a message delivery status condition such as "message accepted by the wireless device" or "message read or processed by the wireless device". The types of notification status responses desired may be specified when the message is created.

The Response Header provides information with regard to the message originator, the message recipient, and control information to further specify how the response should be processed. Specification of the Response Header follows the rules as listed in the Section 5.4.1.

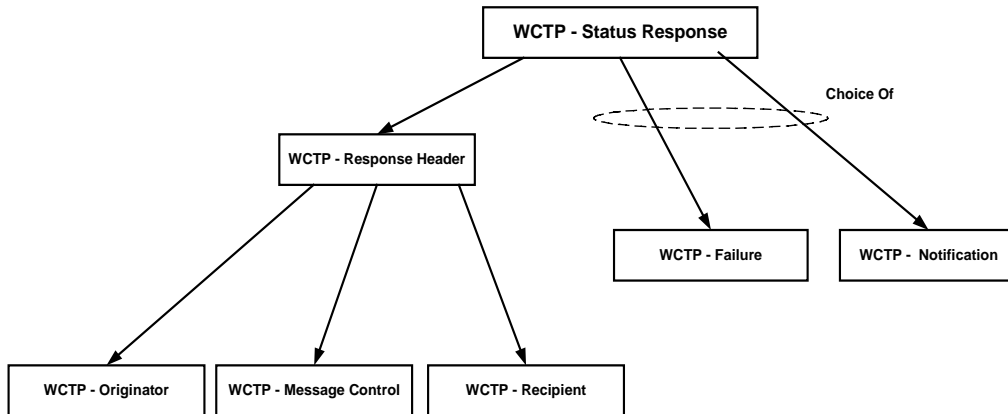


Figure 5-4. WCTP Status Info Operation

5.6 WCTP Confirmation Operation

The Confirmation operation of WCTP (wctp-Confirmation) is used to report both message delivery failures as well as delivery notifications returned on behalf of other messages. Figure 5-5 shows the major XML elements of the Confirmation operation. Each of these elements may specify a number of different attributes or parameters, which are associated with the element. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Confirmation operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

The Confirmation is intended to be used in a return portion of HTTP POST operation or as an embedded element inside the Poll For Messages operation during the polling session. In both cases the Confirmation conveys positive or negative acknowledgement of a received message.

Structurally, the Confirmation operation contains either the Success or the Error element.

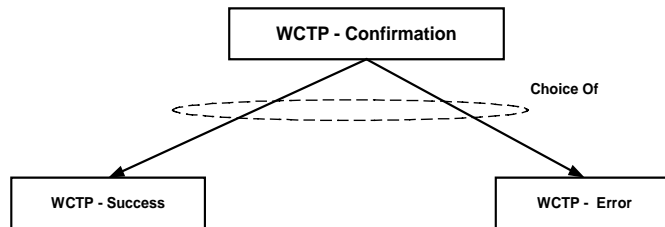


Figure 5-5. WCTP Confirmation Operation

5.7 WCTP Poll For Messages Operation

The Poll For Messages operation (`wctp-PollForMessages`) is used by wireline systems that are not able to accept pushed data from the wireless network and have to use Polling mechanism to collect message response and unsolicited messages directed to them. This “pseudo-push” mechanism was discussed in a section 4.1.3. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Poll For Messages operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

The Poll For Messages operation is initiated by a wireline system to solicit message responses and unsolicited messages accumulated for it by a WCTP server of a wireless carrier. The response to a Poll For Messages operation, the Poll Response operation, may provide a batch of messages, each identified by an assigned sequence number. The following Poll For Message operation must acknowledge the receipt of messages received in the prior poll by providing a Message Received element with the sequence number for each received message in the batch.

The sequence of Poll For Messages/Poll Response operations is repeated until the Poll Response returns an indication that there are no more messages to deliver.

To limit the size of a batch of messages being delivered, the Poll For Messages operations may optionally specify the maximum number of messages returned in one Poll Response.

5.8 WCTP Poll Response Operation

The Poll Response operation (`wctp-PollResponse`) is sent by a WCTP server of a wireless carrier to a wireline system in a response to the Poll For Messages operation. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Poll Response operation. Appendix B contains a description of

each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

Each Poll Response operation may contain a batch of messages destined to the wireline system, which have been queued on its behalf by a WCTP server of a wireless carrier. Each message in a batch is assigned a sequence number. Receipt of the messages delivered via Poll Response is acknowledged based upon this sequence number, with the following Poll For Messages operation (see section 5.7).

5.9 WCTP Lookup Subscriber Operation

The Lookup Subscriber operation of WCTP (`wctp-LookupSubscriber`) is used by an Enterprise host to ask a carrier's WCTP server about the capabilities of a subscriber's messaging device. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Lookup Subscriber operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

Typically an Enterprise host would do a Lookup Subscriber operation to get information about a specific subscriber's messaging device which the Enterprise Host could use when formulating a Submit Message operation for that subscriber.

5.10 WCTP Lookup Response Operation

The Lookup Response operation of WCTP (`wctp-LookupResponse`) is sent by a WCTP server of a wireless carrier to an Enterprise host in a response to the Lookup Subscriber operation. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Lookup Subscriber operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

Typically an Enterprise host would use the information about a specific subscriber's messaging device to formulate a viable set of attributes for a Submit Message operation for that subscriber.

5.11 WCTP Submit Client Message Operation

The Submit Client Message operation of WCTP (`wctp-SubmitClientMessage`) is used to initiate a new message

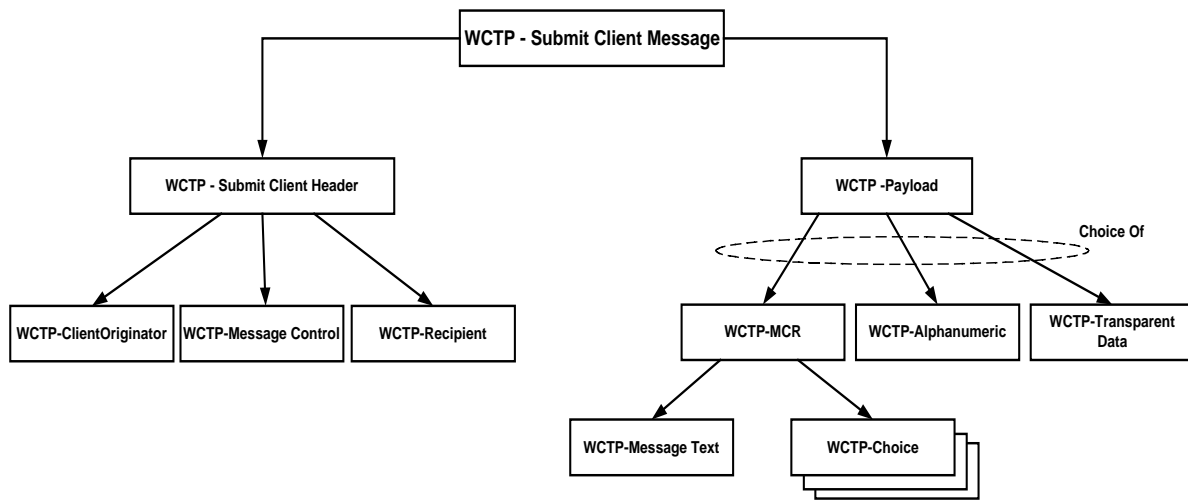


Figure 5-6. WCTP Submit Client Message Operation

from the transient user to the WCTP server of a wireless system. Figure 5-6 shows the major XML elements of the Submit Client Message operation. Each of these elements may specify a number of different attributes or parameters, which are associated with the element. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Submit Client Message operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

The Submit Client Message has two major elements, the Submit Client Header and the Payload.

The Submit Client Header element is analogous to the Submit Header, described in section 5.3.2, however it is tailored for use by transient users. In particular, there is no requirement for the sender ID to be unique. Also, there is no message ID supplied with this operation.

The structure of the Payload element is described in the section 5.3.1.

5.12 WCTP Submit Client Response Operation

The Submit Client Response operation (`wctp-SubmitClientResponse`) is sent by a WCTP server of a wireless carrier to a transient user in a response to the Submit Client Message operation. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Submit Client Response operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

The Submit Client Response operation provides positive or negative acknowledgement of acceptance of the message submitted by the transient user by a wireless carrier. In

the case of positive acknowledgement the WCTP server provides to the message sender a tracking number assigned to the message. Such tracking number in a combination with the sender ID and the recipient ID of the message may be subsequently used to check for the replies and status notifications through a Client Query Operation.

5.13 WCTP Client Query Operation

The Client Query operation (wctp-ClientQuery) is used by a transient user to query a WCTP server of a wireless carrier for replies and status notifications to the previously submitted message. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Query Client Replies operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

To perform the Query Client Replies operation the sender ID, the recipient ID and the tracking number of the original message must be supplied.

5.14 WCTP Client Query Response Operation

The Client Query Response operation (wctp-ClientQueryResponse) is sent by a WCTP server of a wireless carrier to a transient user in the response to the Query Client Replies operation. Figure 5-7 shows the major XML elements of the Submit Client Message operation. Appendix A contains a copy of the Document Type Definition, which formally specifies the XML syntax of the Query Client Response operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

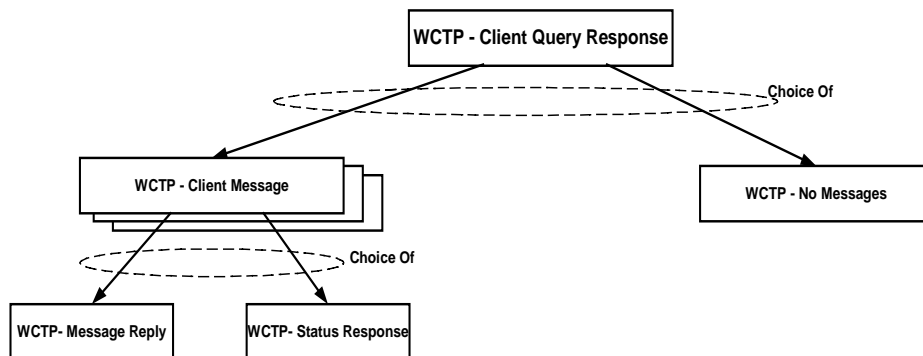


Figure 5-7. WCTP Client Query Response Operation

The Client Query Response operation provides to the requestor a set of Client Messages. Each Client Message contains either a Message Reply or a Status Info to the message identified by a combination of the sender ID, the recipient ID and the tracking number as specified in the Query Client Replies request.

Alternatively, the No Messages element is used to indicate to the requestor that there are no received replies or responses to the original message.

Note: each time the Client Query operation is submitted, the Client Query Response operation will return a set of replies and status notifications with the respect to the original message. WCTP servers will hold this information for an implementation-specific period of time.

5.15 WCTP Bind Domain Alias Operation

The Bind Domain Alias operation of WCTP (wctp-BindDomainAlias) is used to establish

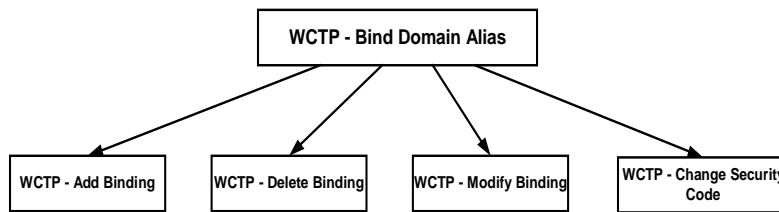


Figure 5-8. WCTP Bind Domain Alias Operation

and maintain a domain alias for a wireline Enterprise Host on a particular carrier's wireless network. Figure 5-8 shows the major XML formatted elements of the Bind Domain Alias request. Each of these elements may specify a number of different attributes or parameters, which are associated with the element. Appendix A contains a copy of the Document Type Definition which formally specifies the XML syntax of the Bind Domain Alias operation. Appendix B contains a description of each of the attributes associated with the major elements of the operation. The description explains how these attributes are used.

The Bind Domain Alias has four sub-operations.

Add Binding is used to establish a new domain alias for the Enterprise Host on the WCTP gateway of the carrier where the message is being sent. The domain alias

specifies a URI where the messages destined for the domain should be delivered. The security code is necessary for verification of authority to perform the binding. This sub-operation establishes an alternative message routing mechanism for the cases where the standard mechanism of delivery mechanism of URI derivation from the domain name is impossible or undesirable (see the discussion of addressing and routing in section 4.4).

Delete Binding is used to eliminate existing domain alias and therefore terminate an alternative message routing for the domain.

Modify Binding is used to change the delivery URI for the existing domain alias.

Change Security Code is used to change the security code for the existing domain alias.

6. WCTP Standards

This section describes aspects of the WCTP protocol that are common to all types of operations, whereas the prior sections discussed the specific operations themselves.

6.1 XML Syntax

WCTP uses XML as the format to define its operations. WCTP operations must be formatted in accordance with the XML 1.0 specification [W3C-WD-xml]. Furthermore, these operations must be "well-formed" and valid according to the WCTP Data Type Definition. The meaning of well-formed and valid is described in the XML specifications referenced.

6.2 Date and Time Formats

This section describes the date and time format used by WCTP in all contexts where a parseable date is required. The format shown here is a selected profile or options form ISO8601:1988 (with Technical Corrigendum 1 applied), hereinafter referred to as ISO8601.

6.2.1 WCTP Date Format

The format for a date string in WCTP Date Format is:

CCYY-MM-DD

where CCYY is the four-digit year (century and year, as described in ISO8601), MM is a two-digit month number, DD is the two-digit ordinal number of the day with in the calendar month, and the separator character is a "hyphen" ("-"). This format is the *Extended Format* described in ISO8601 section 5.2.1.1, with the separator as described in ISO8601 section 4.4 and WCTP implementations must use this format for all date strings specified as being in WCTP Date Format.

Note that specifying a Date without a time rarely makes sense; see 6.3.5 for how to specify both.

6.2.2 WCTP Time Format

The format for a time string in WCTP Time Format is:

HH:MM:SS

where HH is the two-digit hour in 24 hour notation ranging from 00 to 24 (this is not a typo), MM is the two-digit minute ranging from 00 to 59, SS is the two-digit seconds ranging from 00 to 59, and the separator character is a "colon" (:). This format is the *Extended Format* described in ISO8601 section 5.3.1.1, with the separator as described

in ISO8601 section 4.4. WCTP implementations must use this format for all time strings specified as being in WCTP Time Format.

Note that midnight has two representations:

00:00:00

24:00:00

This is deliberate, and in accordance with ISO8601 section 5.3.2.

6.2.3 Subsecond Resolution

WCTP Time Format for representing subsecond granularity follows ISO8601 section 5.3.1.3, and thus uses a "comma" (,) separator and an arbitrary number of digits representing the fraction down to whatever level of precision is appropriate. Thus, the format for time with subsecond resolution is:

HH:MM:SS,S

where the "comma" (,) is a literal character (ISO8601 separator) and "s" after the comma is "to the right of the decimal mark" and indicates the subsecond value. The number of digits in the subsecond value, and the precision of the subsecond value, and the ability of a given implementation to honor that precision, are quality of implementation issues and are not specified by WCTP.

6.2.4 Time Zones

All times specified within WCTP must be specified using GMT (UTC). Implementations are expected to translate these times into the appropriate local time presentation format before utilizing the time in the manner in which it is being requested.

6.2.5 WCTP Date and Time Format

When a Date and Time need to be specified in a single string, the WCTP Date and Time format is:

CCYY-MM-DDTHH:MM:SS,S

where "T" (upper case letter T) is a literal character [ISO8601 designator). This format is the Extended Format of calendar date and time of day as described in ISO8601 section 5.4.1 clause (a).

WCTP operations must not specify invalid combinations of fields, such as February 31. WCTP implementations should reject invalid combinations of fields, rather than trying to interpret them.

6.3 Status and Error Codes

WCTP standardizes all of the status and error codes returned in a WCTP Response operation. It utilizes the familiar Internet protocol paradigm of three-digit status values in responses to protocol operations. This paradigm was chosen because it is well

understood and is suited to both machine-to-machine communication and human interpretation.

6.3.1 No HTTP Relationship

There is no relationship between the status codes in WCTP and the status codes at the HTTP transport level. As already described above, HTTP is merely an instance of a transport mechanism for WCTP operations, and any WCTP implementations utilizing HTTP must appropriately honor HTTP status or error conditions at the transport level. The semantics of completing the HTTP transport operation do not affect the semantics of the WCTP operations, in any way.

Throughout the rest of this discussion, an HTTP status of "200 OK" (or "100 CONTINUE" if applicable) is implicit for the transport of the WCTP operations.

6.3.2 Status and Error Code Returned Text Strings

WCTP Response operations which return success or error codes, are restricted to specifying one of the standard codes from Appendix C. The error and success responses may optionally return an error text or success text attribute. The text string returned in this attribute is implementation specific but must be consistent with the general description of the associated status or error code value (specified in Appendix C) which is required in status response operations. WCTP implementations must process the three-digit numeric value returned. WCTP implementations should not use the text string returned for any purposes other than as informational data with no defined semantics. A typical use for the text string is to report it to an error or call processing log file for later analysis by humans. WCTP implementations must not utilize the free-form body of the WCTP status or error response for any other purpose than as informational data with no defined semantics. Again, a typical use for the text string is to report it to an error or call processing log file for later analysis by humans. For error responses, the free-form data returned may assist in tracking down XML data encoding problems in WCTP implementations.

Error and success responses may optionally return an arbitrary text string within the error or success element. In the case of error responses, this text string could contain portions of the WCTP XML syntax in which errors were found resulting in the inability to process the requested operation. For example, the following XML fragment could be returned when a date value is determined to be incorrect:

```
<wctp-Failure errorCode="404" errorText="Invalid date/time format">  
    deliveryBefore="January 29, 1999" is not a valid WCTP date/time format  
</wctp-Failure>
```

6.3.3 Categories of Status and Error Codes

The defined status and error codes are specified in Appendix C of this specification. Each code contains a three-digit numeric value and a general description of the status or error code.

Implementations of WCTP may treat unrecognized codes not defined in Appendix C of the WCTP specification, or any 9xx series code, in an implementation specific manner.

The general categories in which a status or error definition may fall is:

- 2xx: "Success" Codes
- 3xx: "Protocol Violation" Error Codes
- 4xx: "Content Error" Codes
- 5xx: "Message Processing" Error Codes (Permanent Errors)
- 6xx: "Message Processing" Error Codes (Temporary Errors)
- 9xx: "Experimental" Codes
 - Implementations may experiment with new codes and new features by employing codes values in the 9xx series, without fear of interfering with future versions of the WCTP specification. The 900 series codes will never be specified in an officially released WCTP specification. Presumably, once an experimental implementation is deemed to be operational, the developer of this new capability may submit a request to the appropriate WCTP Standards Committee to request that an official status or error code designation be allocated.

The set of error codes referred to as permanent errors are those error conditions that will not change if the WCTP operation is retried. The set of error codes referred to as temporary errors represent certain error conditions of a temporary nature. A re-submission of the WCTP operation at a later time may result in a successful operation.

6.4 WCTP Operation Declarations

A WCTP operation must begin with a suitable XML document type definition declaration such as:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
```

Any alternate form of the above declaration is acceptable. The requirement is simply that a DTD must be specified somehow.

This declares the operation to be valid XML according to the supplied DTD. The specific URL must be a functional URL that will return the DTD defining the version of the WCTP protocol used to create this operation.

6.5 WCTP Protocol Version

The version number of the WCTP protocol being used, appears as a parameter in each WCTP operation performed. The filename of the standard WCTP Document Type Definition (DTD) being used to define the operations, appears in the <!DOCTYPE> specification which is included with each XML based document as shown in the examples of WCTP transactions in the Appendix E. The version number of the DTD is not necessarily the same as the version number of this specification.

Each WCTP operation contains the `wctpVersion` attribute which specifies the version of the protocol being used. The value of the attribute is of the form <major release number>.<minor release number>, for example: `wctpVersion="1.0"`.

The version number is also reflected by the filename of the DTD. The filename is of the form **wctpv**<major release number>-<minor release number>.**dtd**.. For example, `wctpv1-0.dtd` is the name associated with the first release of WCTP.

The current version number of this protocol can be found by looking at the filename which appears in Appendix A of this specification, which contains a copy of the most recent DTD.

7. Protocol Capabilities for Future Consideration

This specification describes an operational set of features and functions through which WCTP may be utilized between wireline hosts and wireless devices as well as between wireless devices. It was the intent of the WCTP Committee to create a protocol which can meet the goals stated earlier in this specification and to insure that a specification permitting initial implementations be released in a timely fashion. This specification addresses a sufficient set of the WCTP protocol specifics so as to permit the protocol to be utilized. There are many issues which have been deemed as requiring further study for inclusion or addressing by the protocol, but not so critical as to hold off a specification release until the issue is resolved. Many other issues are expected to be taken under consideration and review by the committee before the next release of the specification.

This section lists many issues which have already been recognized as requiring further investigation. The study of these issues and many others and the integration of new features, functions and capabilities into the WCTP protocol and specification as a result of these studies, will further solidify the role of this protocol in servicing the needs of the wireless industry.

7.1 Issues to Address

Some of the issues, which have been recognized as requiring further study, and are under consideration for inclusion in a future release of the protocol specification are described in this section. In some cases, certain features, if added to the protocol, might not be supportable in certain wireless networks. So a means to indicate that protocol capabilities are not supported in particular instances will also be taken under consideration as issues are reviewed.

These are some of the issues which are under consideration for support within the protocol and/or specification:

- 1) *"Cancel Undelivered Message" Operation*
 - 1.1) Provide an operation to cancel a message which has been entered into the wireless network but not yet delivered to the wireless device.
- 2) *Request Message Status Operation*
 - 2.1) The primary intention of the *Request Message Status* operation is to provide message delivery status information. The need for such an operation is questionable as the "submit" operation already provides a means to insure that delivery status information is guaranteed to be returned to the message submitter through the message Acknowledgment process described in this specification.

- 2.2) Given this procedure, it is questionable if there is a need to interrogate the network for status as opposed to assuming the status if no further status responses messages have been received.
- 2.3) It is further questioned if it should be the responsibility of the WCTP protocol, or more specifically the network infrastructure equipment, to retain reported message deliver status information for re-posting of the same information under a status inquiry request.
- 3) *Request Network Characteristics Operation.*
 - 3.1) Provide a means by which applications can determine if the network to which it is communicating, has certain restrictions which should be adhered to. For example, a maximum polling rate, a maximum size message, a maximum rate at which messages can be sent to one or more devices.
 - 3.2) Determine if and how a means should be provided to allow or disallow message truncation for long messages.
- 4) Multiple recipients with a common message.
- 5) Group transmission with individual responses?
- 6) Support "Allow Response Before" a specified time within the protocol and carrier based servers that implement it.
 - 6.1) Would automatically drop late responses to a submitted message rather than having the message sender have to check if certain responses should be ignored.
- 7) What, if any, protocol specific features should exist in support of Over-the-Air programming?
- 8) What retry and timeout parameters should be recommended by the specification?
- 9) Provide additional information with regard to security issues.
- 10) Determine if other forms of Push technology should be standardized into the protocol.
 - 10.1) Review the support of other transport protocols to carry WCTP Operations.
- 11) Investigate use of WCTP protocol with multi-byte character encoding in XML.
 - 11.1) Discuss the implications of the support for multi-byte character sets in this specification.

A Appendix A - Document Type Definition for WCTP

```

<!-- ++++++----- -->
<!-- file-url='http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd' -->
<!-- ++++++----- -->

<!-- Wireless Communication Transfer Protocol (WCTP) -->

<!-- ===== -->
<!-- Declaration of wctp-Operation -->
<!-- ===== -->
<!ELEMENT wctp-Operation      ( wctp-SubmitRequest
                               | wctp-MessageReply
                               | wctp-StatusInfo
                               | wctp-Confirmation
                               | wctp-PollForMessages
                               | wctp-PollResponse
                               | wctp-SubmitClientMessage
                               | wctp-SubmitClientResponse
                               | wctp-ClientQuery
                               | wctp-ClientQueryResponse
                               | wctp-BindDomainAlias
                               | wctp-LookupSubscriber
                               | wctp-LookupResponse
                               ) >

<!ATTLIST wctp-Operation
          wctpVersion          CDATA   #REQUIRED
          >

<!-- ===== -->
<!-- Declaration of wctp-SubmitRequest -->
<!-- ===== -->

<!ELEMENT wctp-SubmitRequest  ( wctp-SubmitHeader
                               , wctp-Payload
                               ) >

<!ELEMENT wctp-SubmitHeader   ( wctp-Originator
                               , wctp-MessageControl
                               , wctp-Recipient
                               ) >

<!ATTLIST wctp-SubmitHeader
          submitTimestamp      CDATA   #IMPLIED
          >

<!ELEMENT wctp-Originator      EMPTY
          >

<!ATTLIST wctp-Originator
          senderID             CDATA   #REQUIRED
          securityCode         CDATA   #IMPLIED
          miscInfo             CDATA   #IMPLIED
          >

<!ELEMENT wctp-MessageControl  EMPTY
          >

<!ATTLIST wctp-MessageControl
          messageID            CDATA   #REQUIRED

```



```

transactionID          CDATA #IMPLIED
sendResponsesToID     CDATA #IMPLIED
allowResponse         ( true | false ) "true"
notifyWhenQueued      ( true | false ) "false"
notifyWhenDelivered   ( true | false ) "false"
notifyWhenRead        ( true | false ) "false"
deliveryPriority       ( HIGH | NORMAL | LOW ) "NORMAL"
deliveryAfter         CDATA #IMPLIED
deliveryBefore        CDATA #IMPLIED
preformatted          ( true | false ) "false"
allowTruncation       ( true | false ) "true"
>

<!ELEMENT wctp-Recipient EMPTY
>

<!ATTLIST wctp-Recipient
recipientID          CDATA #REQUIRED
authorizationCode    CDATA #IMPLIED
>

<!ELEMENT wctp-Payload ( wctp-MCR
| wctp-Alphanumeric
| wctp-TransparentData
) >

<!ELEMENT wctp-MCR ( wctp-MessageText
, wctp-Choice+
) >

<!ELEMENT wctp-MessageText (#PCDATA)
>

<!ELEMENT wctp-Choice (#PCDATA)
>

<!-- ===== -->
<!-- Declaration of wctp-MessageReply -->
<!-- ===== -->

<!ELEMENT wctp-MessageReply ( wctp-ResponseHeader
, wctp-Payload
) >

<!ELEMENT wctp-ResponseHeader ( wctp-Originator
, wctp-MessageControl
, wctp-Recipient
) >

<!ATTLIST wctp-ResponseHeader
responseToMessageID CDATA #REQUIRED
responseTimestamp   CDATA #IMPLIED
respondingToTimestamp CDATA #IMPLIED
onBehalfOfRecipientID CDATA #IMPLIED
>

<!-- ===== -->
<!-- Declaration of wctp-StatusInfo -->
<!-- ===== -->

<!ELEMENT wctp-StatusInfo ( wctp-ResponseHeader
, ( wctp-Failure
| wctp-Notification )

```

```

) >

<!ELEMENT wctp-Failure          (#PCDATA)
>

<!ATTLIST wctp-Failure
  errorCode          CDATA #REQUIRED
  errorText         CDATA #IMPLIED
>

<!ELEMENT wctp-Notification     EMPTY
>

<!ATTLIST wctp-Notification
  type              ( QUEUED | DELIVERED | READ ) #REQUIRED
>

<!-- ===== -->
<!-- Declaration of wctp-Confirmation -->
<!-- ===== -->

<!ELEMENT wctp-Confirmation     ( wctp-Failure
| wctp-Success
) >

<!ELEMENT wctp-Success          (#PCDATA)
>

<!ATTLIST wctp-Success
  successCode       CDATA #REQUIRED
  successText       CDATA #IMPLIED
>

<!-- ===== -->
<!-- Declaration of wctp-PollForMessages -->
<!-- ===== -->

<!ELEMENT wctp-PollForMessages ( wctp-MessageReceived*
) >

<!ATTLIST wctp-PollForMessages
  pollerID          CDATA #REQUIRED
  securityCode      CDATA #IMPLIED
  maxMessagesInBatch CDATA #IMPLIED
>

<!ELEMENT wctp-MessageReceived ( wctp-Failure
| wctp-Success
) >

<!ATTLIST wctp-MessageReceived
  sequenceNo        CDATA #REQUIRED
>

<!-- ===== -->
<!-- Declaration of wctp-PollResponse -->
<!-- ===== -->

<!ELEMENT wctp-PollResponse     ( wctp-Message+
| wctp-NoMessages
) >

<!ATTLIST wctp-PollResponse
  minNextPollInterval CDATA #IMPLIED
>

<!ELEMENT wctp-Message          ( wctp-SubmitRequest

```

```

        | wctp-MessageReply
        | wctp-StatusInfo
        | wctp-LookupResponse
    ) >
<!ATTLIST wctp-Message
    sequenceNo          CDATA #REQUIRED
>

<!ELEMENT wctp-NoMessages    EMPTY
>

<!-- ===== -->
<!-- Declaration of wctp-StatusRequest -->
<!-- ===== -->

<!ELEMENT wctp-StatusRequest    EMPTY
>

<!-- to be defined -->

<!-- ===== -->
<!-- Declaration of wctp-CancelRequest -->
<!-- ===== -->

<!ELEMENT wctp-CancelRequest    EMPTY
>

<!-- to be defined -->

<!-- ===== -->
<!-- Declaration of wctp-SubmitClientMessage -->
<!-- ===== -->

<!ELEMENT wctp-SubmitClientMessage ( wctp-SubmitClientHeader
    , wctp-Payload
    ) >

<!ELEMENT wctp-SubmitClientHeader ( wctp-ClientOriginator
    , wctp-ClientMessageControl?
    , wctp-Recipient
    ) >

<!ATTLIST wctp-SubmitClientHeader
    submitTimestamp      CDATA #IMPLIED
>

<!ELEMENT wctp-ClientOriginator    EMPTY
>

<!ATTLIST wctp-ClientOriginator
    senderID              CDATA #REQUIRED
    miscInfo              CDATA #IMPLIED
>

<!ELEMENT wctp-ClientMessageControl    EMPTY
>

<!ATTLIST wctp-ClientMessageControl
    sendResponsesToID    CDATA #IMPLIED
    allowResponse        ( true | false ) "true"
    notifyWhenQueued     ( true | false ) "false"
    notifyWhenDelivered  ( true | false ) "false"
    notifyWhenRead       ( true | false ) "false"
    deliveryPriority      ( HIGH | NORMAL | LOW) "NORMAL"
    deliveryAfter         CDATA #IMPLIED
    deliveryBefore       CDATA #IMPLIED

```

```

    preformatted          ( true | false ) "false"
    allowTruncation      ( true | false ) "true"
  >

<!-- ===== -->
<!-- Declaration of wctp-SubmitClientResponse -->
<!-- ===== -->

<!ELEMENT wctp-SubmitClientResponse ( wctp-Failure
                                       | wctp-ClientSuccess
                                       ) >

<!ELEMENT wctp-ClientSuccess          (#PCDATA)
>

<!ATTLIST wctp-ClientSuccess
  successCode          CDATA #REQUIRED
  successText          CDATA #IMPLIED
  trackingNumber        CDATA #REQUIRED
>

<!-- ===== -->
<!-- Declaration of wctp-ClientQuery -->
<!-- ===== -->

<!ELEMENT wctp-ClientQuery           EMPTY
>

<!ATTLIST wctp-ClientQuery
  senderID             CDATA #REQUIRED
  recipientID          CDATA #REQUIRED
  trackingNumber        CDATA #REQUIRED
>

<!-- ===== -->
<!-- Declaration of wctp-ClientQueryResponse -->
<!-- ===== -->

<!ELEMENT wctp-ClientQueryResponse ( wctp-ClientMessage+
                                       | wctp-NoMessages
                                       | wctp-Failure
                                       ) >

<!ATTLIST wctp-ClientQueryResponse
  minNextPollInterval CDATA #IMPLIED
>

<!ELEMENT wctp-ClientMessage          ( wctp-ClientMessageReply
                                       | wctp-ClientStatusInfo
                                       ) >

<!ELEMENT wctp-ClientMessageReply     ( wctp-ClientResponseHeader
                                       , wctp-Payload
                                       ) >

<!ELEMENT wctp-ClientStatusInfo       ( wctp-ClientResponseHeader
                                       , ( wctp-Failure
                                           | wctp-Notification
                                           )
                                       ) >

<!ELEMENT wctp-ClientResponseHeader   ( wctp-Originator
                                       , wctp-Recipient
                                       ) >

```

```

<!ATTLIST wctp-ClientResponseHeader
    responseTimestamp      CDATA #IMPLIED
    respondingToTimestamp  CDATA #IMPLIED
>

<!-- ===== -->
<!-- Declaration of wctp-BindDomainAlias -->
<!-- ===== -->

<!ELEMENT wctp-BindDomainAlias      ( wctp-AddBinding
    | wctp-DeleteBinding
    | wctp-ModifyBinding
    | wctp-ChangeSecurityCode
    ) >

<!ELEMENT wctp-AddBinding           EMPTY
>

<!ATTLIST wctp-AddBinding
    domainAlias      CDATA #REQUIRED
    deliverToURI     CDATA #REQUIRED
    securityCode     CDATA #REQUIRED
>

<!ELEMENT wctp-DeleteBinding       EMPTY
>

<!ATTLIST wctp-DeleteBinding
    domainAlias      CDATA #REQUIRED
    securityCode     CDATA #REQUIRED
>

<!ELEMENT wctp-ModifyBinding       EMPTY
>

<!ATTLIST wctp-ModifyBinding
    domainAlias      CDATA #REQUIRED
    deliverToURI     CDATA #REQUIRED
    securityCode     CDATA #REQUIRED
>

<!ELEMENT wctp-ChangeSecurityCode  EMPTY
>

<!ATTLIST wctp-ChangeSecurityCode
    domainAlias      CDATA #REQUIRED
    oldSecurityCode  CDATA #REQUIRED
    newSecurityCode  CDATA #REQUIRED
>

<!-- ===== -->
<!-- Common elements to WCTP commands -->
<!-- ===== -->

<!ELEMENT wctp-Alphanumeric        (#PCDATA)
>

<!ELEMENT wctp-TransparentData     (#PCDATA)
>

<!ATTLIST wctp-TransparentData
    type              ( OPAQUE | FLEXSuite ) "OPAQUE"
    encoding          ( standard | base64 ) "standard"
>

```

```

<!-- ===== -->
<!-- Declaration of wctp-LookupSubscriber -->
<!-- ===== -->

<!ELEMENT wctp-LookupSubscriber      ( wctp-Originator
                                     , wctp-LookupMessageControl
                                     , wctp-Recipient
                                     ) >

<!ELEMENT wctp-LookupMessageControl  EMPTY
>

<!ATTLIST wctp-LookupMessageControl
  messageID          CDATA  #REQUIRED
  transactionID      CDATA  #IMPLIED
  sendResponsesToID CDATA  #IMPLIED
>

<!-- ===== -->
<!-- Declaration of wctp-LookupResponse -->
<!-- ===== -->

<!ELEMENT wctp-LookupResponse      ( wctp-Originator
                                     , wctp-Recipient
                                     , ( wctp-LookupData
                                       | wctp-Failure
                                       )
                                     ) >

<!ATTLIST wctp-LookupResponse
  responseToMessageID CDATA  #REQUIRED
  transactionID       CDATA  #IMPLIED
>

<!ELEMENT wctp-LookupData          EMPTY
>

<!ATTLIST wctp-LookupData
  maxMessageLength  CDATA  #REQUIRED
  mcrSupported      ( true | false ) #IMPLIED
  canRespond        ( true | false ) #IMPLIED
>

```

B Appendix B - WCTP Field Definitions

Field Tag	O/M	Field Value	Comments
Attributes of wctp-Operation			
wctpVersion	M	Char String	Version of the WCTP protocol used to communicate the WCTP operation.
Attributes of wctp-SubmitHeader			
submitTimestamp	M	Time Stamp	Date/Time when request was created.
Attributes of wctp-Originator			
senderID	M	Char String	Unique identifier of the message submitter. The length of the senderID string is limited to 254 octets.
securityCode	O	Char String	Password validating sender as being allowed to send under the specified senderID.
miscInfo	O	Char String	Information about the message sender such as a domain name. This field is intended for logging purposes only. The information provided is not utilized by the protocol in any manner.
Attributes of wctp-MessageControl			
messageID	M	Char String	Message reference number of this message submission. The message ID must be unique for a particular sender ID for the duration of the lifetime of this message transmission and the receipt of one or more responses. The length of the messageID string is limited to 254 octets.
transactionID	O	Char String	Identifier to tie together multiple message submissions into a common set primarily for logging purposes and application-level session management. There is no requirement for this value to be unique. The information content of this field is not utilized by the protocol. The length of the transactionID string is limited to 254 octets.
sendResponsesToID	O	Char String	Unique identifier where responses should be returned. Responses are returned to the originator if this field is not specified. Responses can be directed to non-WCTP destination, such as e-mail, fax, phone. Such destination is identified with one of the reserved prefixes: mailto: , faxto: , phoneto: , pageto: .
allowResponse	O	Char String	If set to false, indicates that no recipient response should be returned.
notifyWhenQueued	O	Char String	True if a notification response should be sent when the message is queued to the recipient.

Field Tag	O/M	Field Value	Comments
notifyWhenDelivered	O	Char String	True if a notification response should be sent when the message is delivered to the recipient.
notifyWhenRead	O	Char String	True if a notification response should be sent when the message is read by the recipient. This capability may be dependent on the features of the recipient.
deliveryPriority	O	Char String	Delivery urgency of the message.
deliveryAfter	O	Time Stamp	Message should be held until the time specified. Must be smaller than the DeliveryBefore value if one is specified.
deliveryBefore	O	Time Stamp	Message should be delivered by the time specified or automatically canceled. Must be larger than the DeliveryAfter value if one is specified. If specified, the DeliveryAfter function is performed before DeliveryBefore function.
allowTruncation	O	Char String	True if Message payload can be truncated. False if payload truncation is impossible (for example, messages with binary payloads may not be truncatable).
preformatted	O	Char String	True if Message payload is preformatted and should not be modified in any way.
Attributes of wctp-Recipient			
recipientID	M	Char String	Unique identifier of a message recipient. The length of the recipientID string is limited to 254 octets.
authorizationCode	O	Char String	Password for recipients that are programmed not to accept messages without the proper authorization.
Sub-Elements of wctp-Payload			
wctp-MCR	O	Element	Multiple choice response definition.
Sub-Elements of wctp-MCR			
wctp-MessageText	O	Char String	Body of the alphanumeric message to send.
wctp-Choice	O	Element	Definition of one possible response choice.
Attributes of wctp-ResponseHeader			
responseToMessageID	M	Char String	Message Identifier of the message to which this response refers.
responseTimestamp	O	Time Stamp	Date/Time when response was created (and not the time it is delivered via WCTP).
respondingToTimestamp	O	Time Stamp	Date/Time of message to which this response refers.
onBehalfOfRecipientID	O	Char String	Recipient ID to which this response refers.
Attributes of wctp-Notification			
type	M	Char String	Indicates the type of notification being reported.
Content of wctp-Failure and wctp-Error			
<body of element>	O	Char String	The body of this element may contain a portion of the data to which this error refers.
Attributes of wctp-Failure and wctp-Error			

Field Tag	O/M	Field Value	Comments
errorCode	M	Char String	WCTP standard numeric error value representing the type of error being reported.
errorText	O	Char String	Description of the error condition.
Attributes of wctp-Success			
successCode	M	Char String	WCTP standard numeric success code of the response being reported.
successText	O	Char String	Additional information regarding the success of the operation.
Attributes of wctp-PollForMessages			
pollerID	M	Char String	Unique identifier of the system polling for messages.
securityCode	O	Char String	Password validating poller as being allowed to send poll requests under the specified pollerID.
maxMessagesInBatch	O	Char String	Maximum number of messages to be returned in response to this poll request.
Attributes of wctp-PollResponse			
minNextPollInterval	O	Char String	Minimum interval before the next Poll request can be issued by the poller where this poll response is directed.
Attributes of wctp-MessageReceived			
sequenceNo	M	Char String	Sequence number of the message received in response to the previous Poll request. Used as an acknowledgement of the receipt of the message.
Attributes of wctp-Message			
sequenceNo	M	Char String	Sequence number of the message being sent as a part of a response to the Poll request.
Attributes of wctp-SubmitClientHeader			
submitTimestamp	M	Time Stamp	Date/Time when request was created.
Attributes of wctp-ClientOriginator			
senderID	M	Char String	Identifier of the message submitter. Uniqueness of this identifier for client submissions cannot be guaranteed.
miscInfo	O	Char String	Information about the message sender such as a user name. This field is intended for logging purposes only. The information provided is not utilized by the protocol in any manner.
Attributes of wctp-ClientMessageControl			
sendResponsesToID	O	Char String	Unique identifier where responses should be returned. Responses are returned to the originator if this field is not specified. Responses can be directed to non-WCTP destination, such as e-mail, fax, phone. Such destination is identified with one of the reserved prefixes: mailto: , faxto: , phoneto: , pageto: .
allowResponse	O	Char String	If set to false, indicates that no recipient response should be returned.
notifyWhenQueued	O	Char String	True if a notification response should be sent when the message is queued to the

Field Tag	O/M	Field Value	Comments
			recipient.
notifyWhenDelivered	O	Char String	True if a notification response should be sent when the message is delivered to the recipient.
notifyWhenRead	O	Char String	True if a notification response should be sent when the message is read by the recipient. This capability may be dependent on the features of the recipient.
deliveryPriority	O	Char String	Delivery urgency of the message.
deliveryAfter	O	Time Stamp	Message should be held until the time specified. Must be smaller than the DeliveryBefore value if one is specified.
deliveryBefore	O	Time Stamp	Message should be delivered by the time specified or automatically canceled. Must be larger than the DeliveryAfter value if one is specified. If specified, the DeliveryAfter function is performed before DeliveryBefore function.
allowTruncation	O	Char String	True if Message payload can be truncated. False if payload truncation is impossible (for example, messages with binary payloads may not be truncatable).
preformatted	O	Char String	True if Message payload is preformatted and should not be modified in any way.
Attributes of wctp-ClientSuccess			
successCode	M	Char String	WCTP standard numeric success code of the response being reported.
successText	O	Char String	Additional information regarding the success of the operation.
trackingNumber	M	Char String	Tracking number assigned to the submitted message, which is being acknowledged.
Attributes of wctp-QueryClientReplies			
senderID	M	Char String	Sender ID of the message replies to which are being queried.
recipientID	M	Char String	Recipient ID of the message replies to which are being queried.
trackingNumber	M	Char String	Tracking number assigned to the message replies to which are being queried.
Attributes of wctp-BindDomainAlias			
domainAlias	M	Char String	Domain specification that is to be treated as an alias for delivery to the specified URI.
deliverToURI	M	Char String	URI where messages destined for the specified domain alias are to be posted.
Attributes of wctp-TransparentData			
type	O	Char String	Type of transparent data being carried in the wctp-TransparentData element.
encoding	O	Char String	Encoding of transparent data being carried in the wctp-TransparentData element.

C Appendix C - Error and Success Codes

Code	Definition	Comments
200 Series Success Codes		
200	Acknowledged	The operation has been received and a first level of validation has been performed. Further validation may result in additional responses. Other responses to this message, such as delivery notifications, may occur as per the operations requested.
300 Series Protocol Violation Error Codes		
300	Operation not supported	The requested WCTP operation is not supported by this system.
301	Input can not be parsed	Input message is garbled. Invalid XML format. Could be caused by binary data or non-XML data being sent.
302	XML validation error	Input is not a well formed XML document and cannot be processed properly.
303	Version error	The version of the input data is not supported.
304	Cannot accept operation	The requested operation can not be performed until messages are pulled using the PollForMessages operation.
400 Series Content Error Codes		
400	Function not supported	The requested function, is not supported for this operation.
401	Invalid SenderID	The senderID is invalid.
402	Invalid security code	The security code for this senderID is invalid.
403	Invalid recipientID	The recipientID is invalid.
404	Invalid authorization code	The authorization code for this recipientID is invalid.
405	Invalid Date or Time	A date and/or time value is not in the

Code	Definition	Comments
	Format	WCTP Standard date and/or time format.
406	Date/Time Ranges specify an unsupported combination	A combination of a request to send a message after a specific time but before another time is incorrectly specified. The 'before' date/time value must be beyond the 'after' date/time value.
407	Invalid date/time period	The specified date and/or time is beyond the time range supported by the system.
408	Date/time specified has already expired	The date/time specified has already passed and the operation requested can not be performed as specified.
409	Domain Alias is not unique	The Domain Alias is already in use. A different alias name must be assigned.
410	Domain Alias access denied	The Domain Alias presented is not owned by the system attempting to reference or delete it.
500 Series Message Processing Error Codes (Permanent)		
500	Message has timed out	The message could not be delivered within a network specific maximum delivery time period.
501	Message has expired	The message could not be delivered within the delivery time period specified when it was submitted.
502	No further responses possible	The specified message ID will no longer be able to receive responses because of network or device limitations.
503	Sender permanently disabled due to excessive polling or traffic	Manual intervention by the operator of the WCTP server is required. The system performing polling has ignored warnings to reduce the polling rate or warnings that traffic levels are over the allowed quota. Service has been barred.
504	Unknown Message	Message reference (a combination of senderID, recipientID and a

Code	Definition	Comments
	Reference	trackingNumber) presented in ClientQuery is not recognized.
600 Series Message Processing Error Codes (Temporary)		
600	Poll rate exceeded	Warning. Polling for messages and status is occurring too frequently. The error text, which accompanies this status, provides a value (in seconds) for the maximum allowed polling rate.
601	Excessive polling	Warning. Polling is occurring at a rate beyond the allowed limitations. If this continues, the polling system will be permanently disabled (see error 503).
602	Traffic rate exceeded	Warning. The rate at which messages are entering the system exceeds the allowed quota. Slow down the message entry rate.
603	Traffic rate excessive	Warning. The rate at which traffic is entering the system exceeds the allowed quota. If this continues the sender will be permanently disabled (see error 504).
900 Series Experimental Error Codes		
900-999	Implementation specific	Implementations may experiment with new codes and new features employing these codes, without fear of interfering with future versions of the WCTP specification, by utilizing code values within this range of numbers. The 900 series codes will never be specified in an officially released WCTP specification. Presumably, once an experimental implementation is deemed to be operational, the developer of this new capability may submit a request to the appropriate WCTP Standards Committee to request that an official status or error code designation be allocated.

D Appendix D - Implementation Notes, Protocol Behavior

This section provides information that is considered specific to certain implementations of the protocol. The protocol does not dictate the manner in which the protocol is actually integrated into a wireless network. It does not define how or if certain protocol features are actually implemented within a particular implementation. Some of these issues will be denoted in this section.

D.1 Infrastructure Elements

Within a wireless network a WCTP Gateway may be created to accept messages that adhere to the WCTP protocol and provides responses according to the protocol specification. This Gateway may communicate with other infrastructure elements, specific to a particular implementation of a wireless network, to create WCTP response messages on behalf of the wireless device or on behalf of other elements of the wireless network reporting a significant event regarding the message delivery process. The infrastructure elements act as a Proxy/Gateway for the wireless device. These elements understand the wireless network specifics and the native protocols used to communicate within those networks. These elements map those specifics into the WCTP standardized representations in order to provide uniform appearances regardless of the wireless network over which data is being carried. These elements "do what is necessary" to provide this mapping. This may include the maintenance of transactional information on these network elements, through the "lifetime" of certain messages, in order to make up for the inability of certain specific networks to carry all of the information provided within WCTP.

E Appendix E - General Examples

This section contains examples of the specific WCTP operations executed, to perform the operations described in the Complex Multi-device Transaction, described in section 3.3 of this specification.

E.1 Request From Wireless Device to Enterprise Host

E.1.1 Alarm Message Sent

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader
      submitTimestamp="1999-03-31T18:15:00"
    >
      <wctp-Originator
        senderID="1234567@abcwireless.com"
        securityCode="OpenSesame"
      />
      <wctp-MessageControl
        messageID="1C936BA3"
      />
      <wctp-Recipient
        recipientID="controlcenter@myenterprise.com"
        authorizationCode="HelloWorld"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-TransparentData
        type="FLEXsuite"
        encoding="base64"
      >
        ...           - base64-encoded binary FLEXsuite message
      </wctp-TransparentData>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>
```

E.2 Request from Enterprise Host to a 2-way pager

E.2.1 Enterprise Host presents multiple choice selection to a technician

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader
      submitTimestamp="1999-03-31T18:18:00">
      <wctp-Originator
        senderID="controlcenter@myenterprise.com"
        miscInfo="Enterprise Corp. Operations Control Center"
      />
      <wctp-MessageControl
        messageID="46264399"
        transactionID="19990331.PN.1234567.001"
        notifyWhenQueued="true"
        notifyWhenDelivered="true"
        notifyWhenRead="true"
        deliveryPriority="HIGH"
      />
      <wctp-Recipient
        recipientID="john.tech@skypage.com"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-MCR>
        <wctp-MessageText>
          pH threshold exceeded. Location: Louisville, KY.
          TankNumber: 1234. DeviceId: Abcwireless.1234567
          Select a Corrective Action
        </wctp-MessageText>
        <wctp-Choice>
          Raise temperature 1 degree.
        </wctp-Choice>
        <wctp-Choice>
          Lower temperature 1 degree.
        </wctp-Choice>
        <wctp-Choice>
          Raise pressure 1 atm.
        </wctp-Choice>
        <wctp-Choice>
          Lower pressure 1 atm.
        </wctp-Choice>
      </wctp-MCR>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>

```


E.3 Delivery notification – WCG to Enterprise Host

E.3.1 WCG informs Enterprise Host that message was successfully delivered to a 2-way pager.

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-StatusInfo>
    <wctp-ResponseHeader
      responseToMessageID="46264399"
      responseTimestamp="1999-03-31T18:19:00"
      respondingToTimestamp="1999-03-31T18:18:00"
      onBehalfOfRecipientID="john.tech@skypage.com"
    >
      <wctp-Originator
        senderID="wcg-3.abcwireless.com"
        miscInfo="WCG for Skypage Network"
      />
      <wctp-MessageControl
        messageID="32456"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="controlcenter@myenterprise.com"
      />
    </wctp-ResponseHeader>
    <wctp-Notification
      type="DELIVERED"
    />
  </wctp-StatusInfo>
</wctp-Operation>
```

E.4 Response from 2-way pager to Enterprise Host

E.4.1 Technician selects one of presented choices

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-MessageReply>
    <wctp-ResponseHeader
      responseToMessageID="46264399"
      responseTimestamp="1999-03-31T18:25:00"
      respondingToTimestamp="1999-03-31T18:18:00"
    >
      <wctp-Originator
        senderID="john.tech@skypage.com"
        miscInfo="John The Technician"
      />
      <wctp-MessageControl
        messageID="33"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="controlcenter@myenterprise.com"
      />
    </wctp-ResponseHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        Raise temperature 1 degree.
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-MessageReply>
</wctp-Operation>
```

E.5 Response from Enterprise Host to Wireless Device

E.5.1 Enterprise Host sends a request to perform a corrective action

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader
      submitTimestamp="1999-03-31T8:30:00"
    >
      <wctp-Originator
        senderID="controlcenter@myenterprise.com"
        securityCode="HelloWorld"
        miscInfo="Enterprise Corp. Operations Control Center"
      />
      <wctp-MessageControl
        messageID="46264423"
        transactionID="19990331.PN.1234567.001"
        notifyWhenQueued="true"
        notifyWhenDelivered="true"
        notifyWhenRead="true"
        deliveryPriority="HIGH"
      />
      <wctp-Recipient
        recipientID="1234567@abcwireless.com"
        authorizationCode="OpenSesame"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-TransparentData
        type="FLEXsuite"
        encoding="base64"
      >
        ...           - base64-encoded binary FLEXsuite message
      </wctp-TransparentData>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>
```

E.6 Confirmation of message acceptance to Enterprise Host

E.6.1 Internet Gateway informs Host that message passed protocol conformance verification.

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message acknowledged"
    >
      Your message is being processed by the ABC wireless network
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>
```

E.7 Response from Wireless Device to Enterprise Host

E.7.1 Wireless device informs Enterprise Host that corrective action started

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-MessageReply>
    <wctp-ResponseHeader
      responseToMessageID="46264423"
      responseTimestamp="1999-03-31T18:35:00"
      respondingToTimestamp="1999-03-31T18:30:00"
    >
      <wctp-Originator
        senderID="1234567@abcwireless.com"
      />
      <wctp-MessageControl
        messageID="1C936BA3"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="controlcenter@myenterprise.com"
      />
    </wctp-ResponseHeader>
    <wctp-Payload>
      <wctp-TransparentData
        type="FLEXsuite"
        encoding="base64"
      >
        ...           - base64-encoded binary FLEXsuite message
      </wctp-TransparentData>
    </wctp-Payload>
  </wctp-MessageReply>
</wctp-Operation>
```

E.8 Response from Enterprise Host to 2-way pager

E.8.1 Enterprise Host informs technician that corrective action started

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-MessageReply>
    <wctp-ResponseHeader
      responseToMessageID="33"
      responseTimestamp="1999-03-31T18:37:00"
      respondingToTimestamp="1999-03-31T18:25:00"
    >
      <wctp-Originator
        senderID="controlcenter@myenterprise.com"
      />
      <wctp-MessageControl
        messageID="1C936BA3"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="john.tech@skypage.com"
      />
    </wctp-ResponseHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        Corrective action (Raise Temperatue 1 deg) has been
        initiated. Location: Louisville, KY.
        TankNumber: 1234. DeviceId: Abcwireless.1234567
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-MessageReply>
</wctp-Operation>
```

E.9 Request from 2-way pager to Enterprise Host

E.9.1 Technician asks Host to request the telemetry device to verify that corrective action was taken

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader>
      submitTimestamp="1999-03-31T19:45:00"
    >
      <wctp-Originator>
        senderID="john.tech@skypage.com"
        miscInfo="John The Technician"
      />
      <wctp-MessageControl>
        messageID="44"
      />
      <wctp-Recipient>
        recipientID="controlcenter@myenterprise.com"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        QUERY STATE Device=Abcwireless.1234567
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>
```

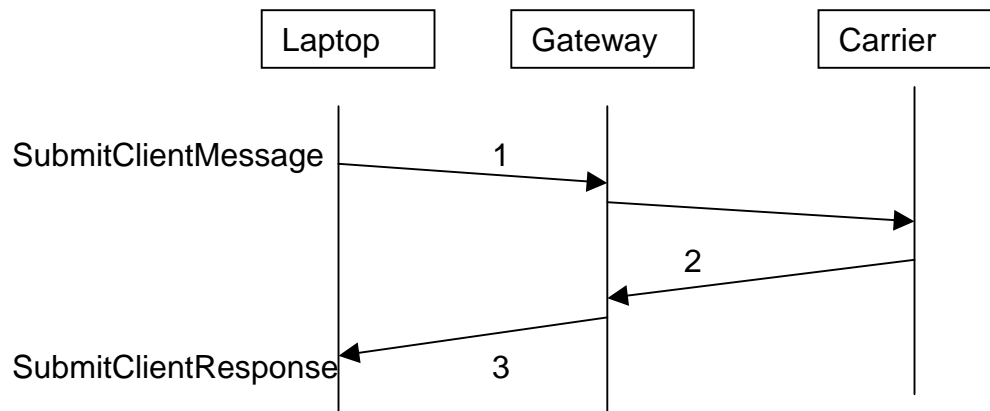
F Appendix F - Use Cases

This section will show examples of how the following types of transactions are represented through WCTP:

1. Transient user sending a message to a valid customer
2. Transient user sending a message to an invalid customer
3. Transient user checking the status of a message with valid tracking number
4. Transient user checking the status of a message with invalid tracking number
5. A device initiated message for a valid device on a different messaging network
6. A device initiated message for an invalid device on a different messaging network
7. A device initiated message for an invalid host
8. An enterprise host initiated message to a valid device on a messaging network.
9. An enterprise host initiated message to an invalid device on a messaging network
10. A device initiated message to a default application on an enterprise host
11. A device initiated message to a specific application on an enterprise host
12. A MCR message from a Transient user or enterprise host
13. A binary message from a Transient user or enterprise host
14. A Transient user or server initiated message to a device with usage code enabled
15. An enterprise host asking for all of it's messages
16. A device reply to an enterprise host generated message
17. A device reply to a Transient user generated message
18. Sending back Multiple replies on a Transient user message status query
19. Disabling replies on a message sent to the device
20. Polling by enterprise host for all messages directed at that host
21. Lookup & Submit from an Enterprise host

F.1 Use Case #1: Transient client sending a message to a valid customer

Assume recipient on same network/carrier as WCTP Gateway.



1. Submit message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientMessage>
    <wctp-SubmitClientHeader
      submitTimestamp="1999-03-31T19:45:00"
    >
      <wctp-ClientOriginator
        senderID="mylaptop@myisp.com"
      />
      <wctp-Recipient
        recipientID="userId@MyCarrier.com"
      />
    </wctp-SubmitClientHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        Test page from my laptop to my pager
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-SubmitClientMessage>
</wctp-Operation>
  
```

2. MyCarrier accepts request, queues message for delivery and returns tracking number.

3. Request accepted:

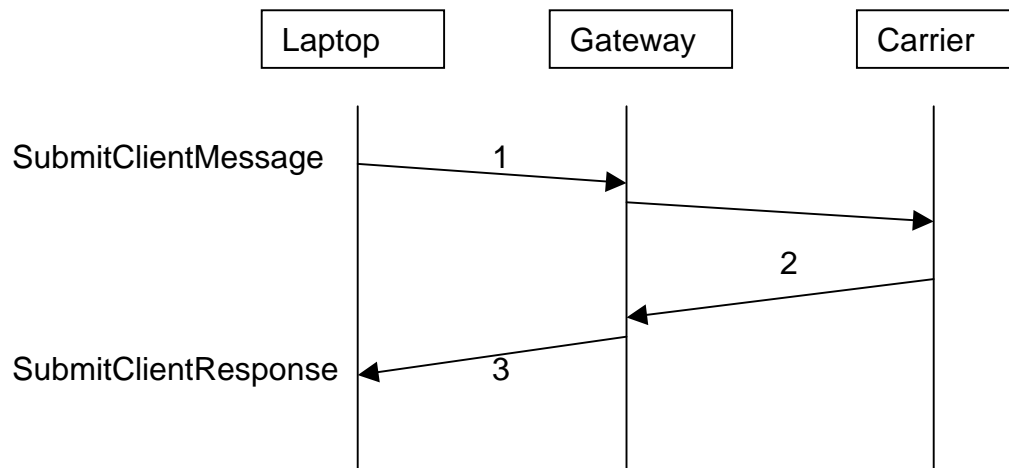
```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientResponse>
  
```

```
<wctp-ClientSuccess
  successCode="200"
  successText="Message accepted"
  trackingNumber="5579"
>
  MyCarrier is happy to send your message as page!
</wctp-ClientSuccess>
</wctp-SubmitClientResponse>
</wctp-Operation>
```

F.2 Use Case #2: Transient client sending a message to an invalid customer

Assume recipient on same network/carrier as WCTP Gateway.



1. Submit message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientMessage>
    <wctp-SubmitClientHeader
      submitTimestamp="1999-03-31T19:45:00"
    >
      <wctp-ClientOriginator
        senderID="mylaptop@myisp.com"
      />
      <wctp-Recipient
        recipientID="myUserId@MyCarrier.com"
      />
    </wctp-SubmitClientHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        Test page from my laptop to my pager
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-SubmitClientMessage>
</wctp-Operation>
  
```

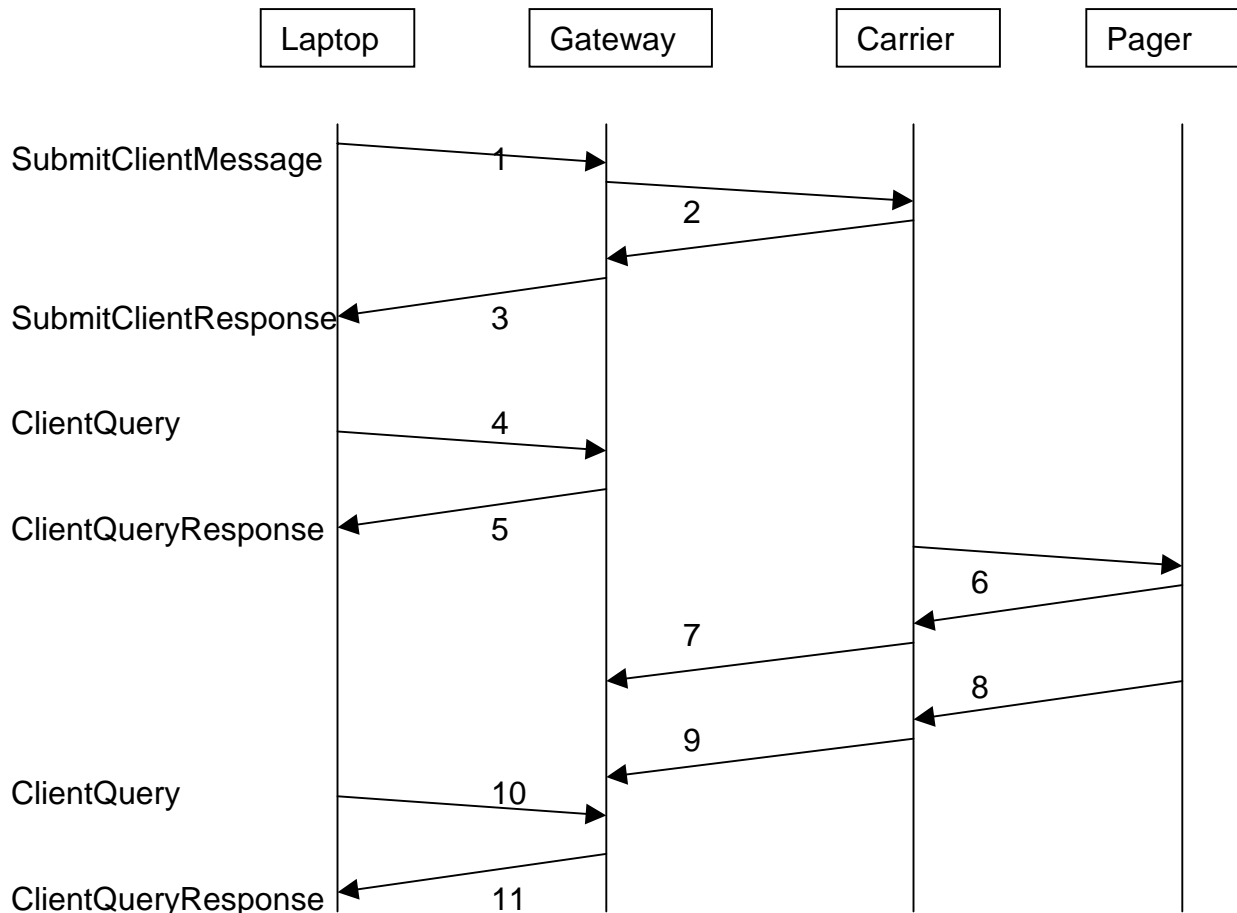
2. MyCarrier rejects the request because myUserId is not a valid subscriber.

3. Request rejected:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientResponse>
    <wctp-Failure
      errorCode="403"
      errorText="Invalid SenderID"
    >
      The subscriber "myUserId" does not exist at MyCarrier
    </wctp-Failure>
  </wctp-SubmitClientResponse>
</wctp-Operation>
```

F.3 Use Case #3: Transient client checking the status of a message with a valid tracking number

Assume recipient on same network/carrier as WCTP Gateway.



1. Submit message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientMessage>
    <wctp-SubmitClientHeader
      submitTimestamp="1999-03-31T19:45:00"
    >
      <wctp-ClientOriginator
        senderID="mylaptop@myisp.com"
      />
      <wctp-ClientMessageControl
        notifyWhenQueued="true"
        notifyWhenDelivered="true"
        notifyWhenRead="true"
      />
    </wctp-SubmitClientHeader>
  </wctp-SubmitClientMessage>
</wctp-Operation>
    
```

```

        />
        <wctp-Recipient
            recipientID="userId@MyCarrier.com"
        />
    </wctp-SubmitClientHeader>
    <wctp-Payload>
        <wctp-Alphanumeric>
            Test page from my laptop to my pager
        </wctp-Alphanumeric>
    </wctp-Payload>
</wctp-SubmitClientMessage>
</wctp-Operation>

```

2. MyCarrier accepts request, queues message for delivery then returns tracking number and the time the message was queued for delivery.

3. Request accepted:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-SubmitClientResponse>
        <wctp-ClientSuccess
            successCode="200"
            successText="Message accepted"
            trackingNumber="5579"
        >
            MyCarrier is happy to send your message as page!
        </wctp-ClientSuccess>
    </wctp-SubmitClientResponse>
</wctp-Operation>

```

4. Check for status:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-ClientQuery
        senderID="mylaptop@myisp.com"
        recipientID="userId@MyCarrier.com"
        trackingNumber="5579"
    />
</wctp-Operation>

```

5. Receive queued status (which is the only notification asked for, which has occurred):

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-ClientQueryResponse>
        <wctp-ClientMessage>
            <wctp-ClientStatusInfo>
                <wctp-ClientResponseHeader
                    responseTimestamp="1999-03-31T19:45:02"
                    respondingToTimestamp="1999-03-31T19:45:00"
                >
                    <wctp-Originator

```

```

        senderId="userId@MyCarrier.com"
    />
    <wctp-Recipient
        recipientId="mylaptop@myisp.com"
    />
</wctp-ClientResponseHeader>
<wctp-Notification
    type="QUEUED"
/>
</wctp-ClientStatusInfo>
</wctp-ClientMessage>
</wctp-ClientQueryResponse>
</wctp-Operation>

```

6. MyCarrier delivers page to pager (confirmed by pager).
7. MyCarrier returns a delivery notification associated with tracking number 5579, this sender & recipient.
8. Subscriber reads message on pager and pager notifies MyCarrier.
9. MyCarrier returns a read notification associated with tracking number 5579, this sender & recipient.
10. Check for status again:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-ClientQuery
        senderID="mylaptop@myisp.com"
        recipientID="userId@MyCarrier.com"
        trackingNumber="5579"
    />
</wctp-Operation>

```

11. Receive queued, delivered & read statuses:

```

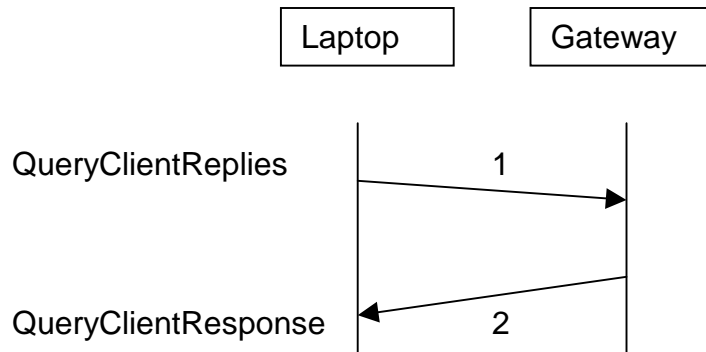
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-ClientQueryResponse>
        <wctp-ClientMessage>
            <wctp-ClientStatusInfo>
                <wctp-ClientResponseHeader
                    responseTimestamp="1999-03-31T19:45:02"
                    respondingToTimestamp="1999-03-31T19:45:00"
                >
                    <wctp-Originator
                        senderId="userId@MyCarrier.com"
                    />
                    <wctp-Recipient
                        recipientId="mylaptop@myisp.com"
                    />
                </wctp-ClientResponseHeader>
                <wctp-Notification
                    type="QUEUED"
                />
            </wctp-ClientStatusInfo>

```

```
</wctp-ClientMessage>
<wctp-ClientMessage>
  <wctp-ClientStatusInfo>
    <wctp-ClientResponseHeader
      responseTimestamp="1999-03-31T19:45:32"
      respondingToTimestamp="1999-03-31T19:45:00"
    >
      <wctp-Originator
        senderId="userId@MyCarrier.com"
      />
      <wctp-Recipient
        recipientId="mylaptop@myisp.com"
      />
    </wctp-ClientResponseHeader>
    <wctp-Notification
      type="DELIVERED"
    />
  </wctp-ClientStatusInfo>
</wctp-ClientMessage>
<wctp-ClientMessage>
  <wctp-ClientStatusInfo>
    <wctp-ClientResponseHeader
      responseTimestamp="1999-03-31T19:46:09"
      respondingToTimestamp="1999-03-31T19:45:00"
    >
      <wctp-Originator
        senderId="userId@MyCarrier.com"
      />
      <wctp-Recipient
        recipientId="mylaptop@myisp.com"
      />
    </wctp-ClientResponseHeader>
    <wctp-Notification
      type="READ"
    />
  </wctp-ClientStatusInfo>
</wctp-ClientMessage>
</wctp-ClientQueryResponse>
</wctp-Operation>
```


F.4 Use Case #4: Transient client checking message status using an invalid tracking number

Assume recipient on same network/carrier as WCTP Gateway.



1. Check for status:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-ClientQuery
    senderID="mylaptop@myisp.com"
    recipientID="userId@MyCarrier.com"
    trackingNumber="5580"
  />
</wctp-Operation>
  
```

2. Receive response:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-ClientQueryResponse>
    <wctp-Failure>
      errorCode="504"
      errorText="Unknown message reference"
    >
      The trackingNumber "5580", senderID "mylaptop@myisp.com" and
      recipientID "userId@MyCarrier.com" do not resolve to a message
      which can be queried
    </wctp-Failure>
  </wctp-ClientQueryResponse>
</wctp-Operation>
  
```

F.5 Use Case #5: A device initiated message for a valid device on a different messaging network

F.6 Use Case #6: A device initiated message to an invalid device on a different network

F.7 Use Case #7: A device initiated message for an invalid host

Sample operation follows.

1. Transaction is started as an unsolicited message from device is received at WCG for carrier "Skypage":

Sender: 1234567
 Recipient: Pageto:jo_chen@eterprise.com
 Payload: "meeting at 2 pm Wed"

Note: eterprise.com is misspelled.

2. WCG determines this message should be delivered via WCTP based on "Pageto:" portion of Recipient and a wctp-SubmitRequest is generated. It then sends wctp-SubmitRequest to Internet Gateway;
3. Internet Gateway consults its AliasRoutingTable and fails to find entry for "eterprise.com", therefore determines the wctp-SubmitRequest should be delivered via Standard Routing;
4. Internet Gateway consults its PollRegistrationTable and fails to find an entry for "eterprise.com", therefore determines the message should be sent out immediately;
5. Internet Gateway constructs XML message and attempts to deliver it using HTTP Full-Request:

POST http://wctp.eterprise.com/wctp HTTP/1.1

with Entity-Body as follows:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader
      submitTimestamp="1999-04-13T18:15:00"
    >
      <wctp-Originator
        senderID="1234567@skypage.com"
      />
      <wctp-MessageControl
        messageID="1C936BA3"
      />
      <wctp-Recipient
        recipientID="jo_chen@eterprise.com"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        "meeting at 2 pm Wed."
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>
```

6. Internet Gateway receives the following HTTP response:

HTTP/1.1 404 Not Found

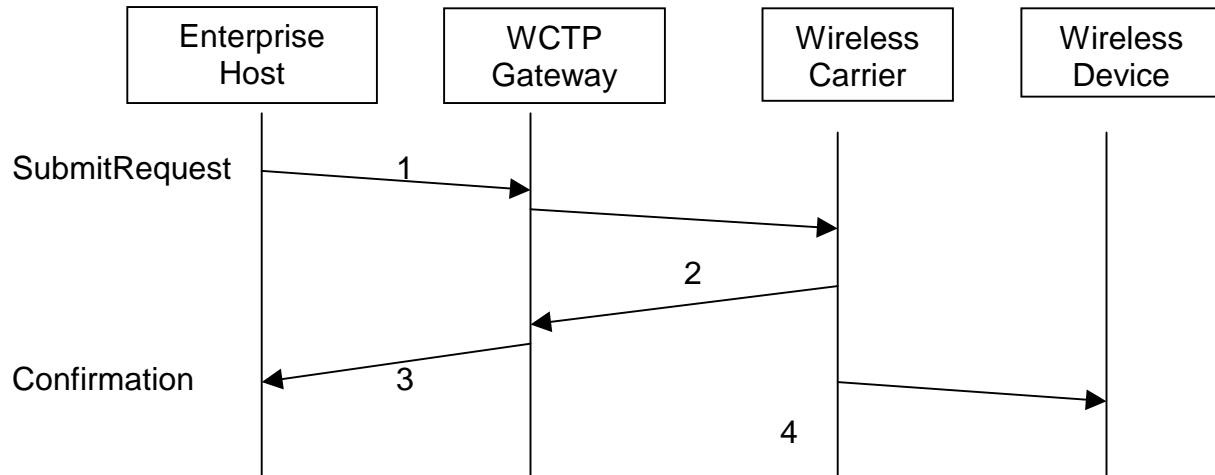
7. Internet Gateway may notifies the wireless device of the error condition by sending message to WCG:

Recipient: 1234567

Payload: "Host not found: eterprise.com"

8. WCG forwards the above message to wireless device. Transaction is completed.

F.8 Use Case #8: An enterprise host originated message to a valid device



1. Enterprise host submits message:

```

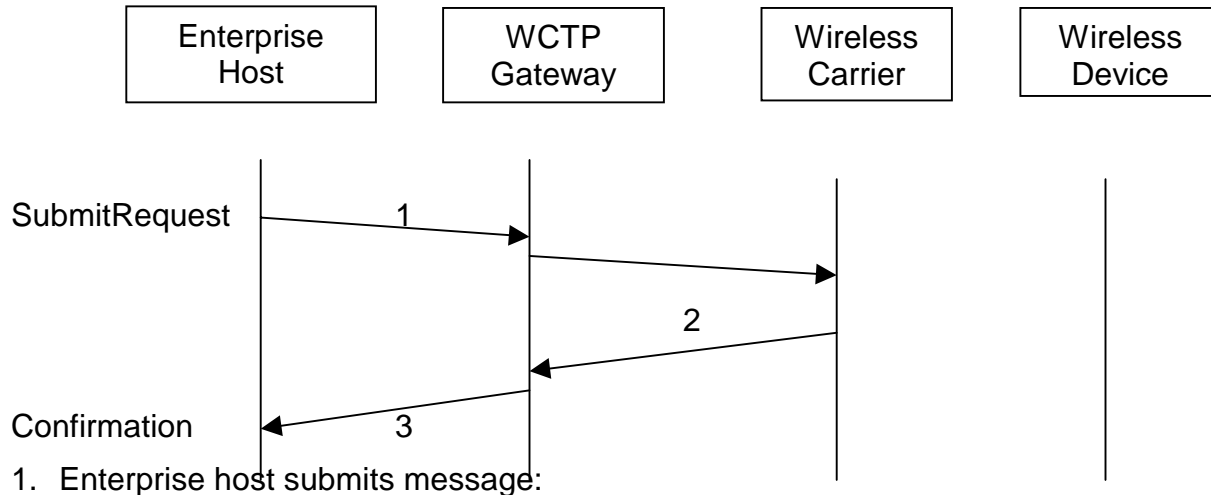
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
<wctp-SubmitRequest>
  <wctp-SubmitHeader
    submitTimestamp="1999-04-19T18:18:00">
    <wctp-Originator
      senderID="controlcenter@myenterprise.com"
    />
    <wctp-MessageControl
      messageID="46264399"
    />
    <wctp-Recipient
      recipientID="1234567"
    />
  </wctp-SubmitHeader>
  <wctp-Payload>
    <wctp-Alphanumeric>
      This message is to a valid recipientID on this messaging network.
    </wctp-Alphanumeric>
  </wctp-Payload>
</wctp-SubmitRequest>
</wctp-Operation>
  
```

2. Carrier accepts request. The carrier determines that the pin is valid. The carrier queues message for delivery.
3. Gateway replies to indicate submission request accepted:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
<wctp-Confirmation>
  <wctp-Success
    successCode="200"
  >
    Message to 1234567 accepted.
  </wctp-Success>
</wctp-Confirmation>
</wctp-Operation>
```

4. Carrier delivers message to device.

F.9 Use Case #9: An enterprise host originated message to an invalid device



```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
<wctp-SubmitRequest>
  <wctp-SubmitHeader>
    submitTimestamp="1999-04-19T18:18:00">
    <wctp-Originator>
      senderID="controlcenter@myenterprise.com"
    />
    <wctp-MessageControl>
      messageID="46264399"
    />
    <wctp-Recipient>
      recipientID="1234567"
    />
  </wctp-SubmitHeader>
  <wctp-Payload>
    <wctp-Alphanumeric>
      This message is to an invalid recipientID on this messaging network
    </wctp-Alphanumeric>
  </wctp-Payload>
</wctp-SubmitRequest>
</wctp-Operation>
  
```

- Carrier receives request; but as part of validation the carrier determines that the recipientID does not exist on this network; an error message is sent back to the gateway.
- Gateway replies to indicate submission error:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
<wctp-Confirmation>
  <wctp-Failure>
  
```

```
        errorCode="403"  
        errorText="Invalid recipientID"  
    >  
        RecipientID 1234567 is invalid.  
    </wctp-Failure>  
</wctp-Confirmation>  
</wctp-Operation>
```


F.10 Use Case #10: A device initiated message to a default application on an enterprise host

Sample sequence of operation:

1. Transaction is started as an unsolicited message from device is received at WCG for carrier "skypage":

Sender: 1234567

Recipient: Pageto:inventory@oriental_imports.com

Payload: "SELECT quantity FROM inventory WHERE sku=3328"
2. WCG determines this message should be delivered via WCTP based on "PageTo:" portion of Recipient and a wctp-SubmitRequest is generated. It then sends wctp-SubmitRequest to Internet Gateway.
3. Internet Gateway consults its AliasRoutingTable and fails to find entry for "oriental_imports.com", therefore determines the wctp-SubmitRequest should be delivered via Standard Routing.
4. Internet Gateway consults its PollRegistrationTable and fails to find an entry for "oriental_imports.com", therefore determines the message should be sent out immediately.
5. Internet Gateway constructs XML message and delivers it using HTTP Full-Request:

POST http://wctp.oriental_imports.com/wctp HTTP/1.1

with Entity-Body as follows:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader>
      submitTimestamp="1999-04-13T18:15:00"
    >
      <wctp-Originator>
        senderID="1234567@skypage.com"
      />
      <wctp-MessageControl>
        messageID="1C936BA3"
      />
      <wctp-Recipient>
        recipientID="inventory@oriental_imports.com"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        "SELECT quantity FROM inventory WHERE sku=3328"
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>
```

6. EH WCTP server receives XML message and sends back wctp-Confirmation using HTTP Full-Response:

HTTP/1.1 200 OK

with Entity-Body as follows:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      "messageID 1C936BA3 received"
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>
```

7. EH WCTP server extracts wctp-SubmitRequest from message. Based on recipientID and the server's DispatchLookupTable, the server determines the message should be handled by /usr/local/bin/inventory_app. It then dispatches the message to inventory_app, which has been started earlier.

8. Inventory_app processes the message and sends reply to EH WCTP server:

Sender: inventory
 Recipient: 1234567@skypage.com
 Payload: "quantity=17"

9. Based on recipientID, EH WCTP server sends HTTP Full-Request to

POST http://wctp.skypage.com/wctp HTTP/1.1

with the following formatted XML wctp-MessageReply as Entity-Body:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-MessageReply>
    <wctp-ResponseHeader
      reponseToMessageID="1C936BA3"
      responseTimestamp="1999-04-13T18:15:30"
    >
      <wctp-Originator
        senderID="inventory@oriental_imports.com"
      />
      <wctp-MessageControl
        messageID="331"
      />
      <wctp-Recipient
        recipientID="1234567@skypage.com"
      />
    </wctp-ResponseHeader>
    <wctp-Payload>
```

```
<wctp-Alphanumeric>  
  "quantity=17"  
</wctp-Alphanumeric>  
</wctp-Payload>  
</wctp-MessageReply>  
</wctp-Operation>
```

10. Internet Gateway receives XML message, extracts wctp-MessageReply, and sends to carrier skypage's WCG according to recipeintID:

Sender: inventory@oriental_imports.com

Recipient: 1234567

Payload: "quantity=17"

11. Skypage's WCG sends the above message to device. Transaction is completed.

F.11 Use Case #11: A device initiated message to a specific application on an enterprise host

Sample sequence of operation:

1. A wctp-BindDomainAlias message is received at Internet Gateway as follows:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-BindDomainAlias>
    <wctp-AddBinding
      domainAlias="inventory_oi"
      deliverToURI="http://oriental_imports.com/usr/local/bin/inventory_app"
      securityCode="oi_trusts_skypage"
    />
  </wctp-BindDomainAlias>
</wctp-Operation>
```

2. Internet Gateway receives the wctp-BindDomainAlias and stores tuple ("inventory_oi", "http://oriental_imports.com/usr/local/bin/inventory_app", "oi_trusts_skypage") in its AliasRoutingTable. From now on, all messages destined to "inventory_oi" will be delivered to the URI using securityCode as indicated in the tuple.
3. Transaction started as an unsolicited message from device is received at WCG for carrier "skypage":

Sender: 1234567

Recipient: Pageto:inventory_oi

Payload: "SELECT quantity FROM inventory WHERE sku=3328"
4. WCG determines this message should be delivered via WCTP based on "PageTo:" portion of Recipient and a wctp-SubmitRequest is to be generated. It then sends wctp-SubmitRequest to Internet Gateway;
5. Internet Gateway consults its AliasRoutingTable and finds an entry for "inventory_oi", therefore determines to deliver the wctp-SubmitRequest to http://oriental_imports.com/usr/local/bin/inventory_app.
6. Internet Gateway consults its PollRegistrationTable and fails to find an entry for oriental_imports.com and therefore determines the message should be delivered immediately using HTTP Full-Request:

POST http://wctp.oriental_imports.com/usr/local/bin/inventory_app HTTP/1.1

with Entity-Body as follows:

```
<?xml version="1.0"?>
```

```

<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader
      submitTimestamp="1999-04-13T18:15:00"
    >
      <wctp-Originator
        senderID="1234567@skypage.com"
        securityCode="oi_trusts_skypage"
      />
      <wctp-MessageControl
        messageID="1C936BA3"
      />
      <wctp-Recipient
        recipientID="inventory_oi"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        "SELECT quantity FROM inventory WHERE sku=3328"
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>

```

7. Inventory_app receives XML message and sends back wctp-Confirmation as HTTP Full-Response:

HTTP/1.1 200 OK

with Entity-Body as follows:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message acknowledged"
    >
      "message ID=1C936BA3 received."
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>

```

8. Inventory_app extracts and process wctp-SubmitRequest from message.
 9. Inventory_app formats wctp-MessageReply as follows:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-MessageReply>
    <wctp-ResponseHeader
      reponseToMessageID="1C936BA3"
      responseTimestamp="1999-04-13T18:15:30"
    >

```

```
>
  <wctp-Originator
    senderID="inventory_oi"
  />
  <wctp-MessageControl
    messageID="331"
  />
  <wctp-Recipient
    recipientID="1234567@skypage.com"
  />
</wctp-ResponseHeader>
<wctp-Payload>
  <wctp-Alphanumeric>
    "quantity=17"
  </wctp-Alphanumeric>
</wctp-Payload>
</wctp-MessageReply>
</wctp-Operation>
```

Based on recipientID, inventory_app sends the formatted XML message as Entity-Body of HTTP Full-Request:

POST <http://wctp.skypage.com/wctp> HTTP/1.1

10. Internet Gateway receives XML message, extracts wctp-MessageReply, and sends to carrier skypage's WCG according to recipientID:

Sender: inventory_oi

Recipient: 1234567

Payload: "quantity=17"

11. Skypage's WCG sends the above message to device. Transaction is completed.

Note: Similar transactions in the future will only execute steps 3 through 11 until the routing alias is removed via wctp-DeleteBinding.

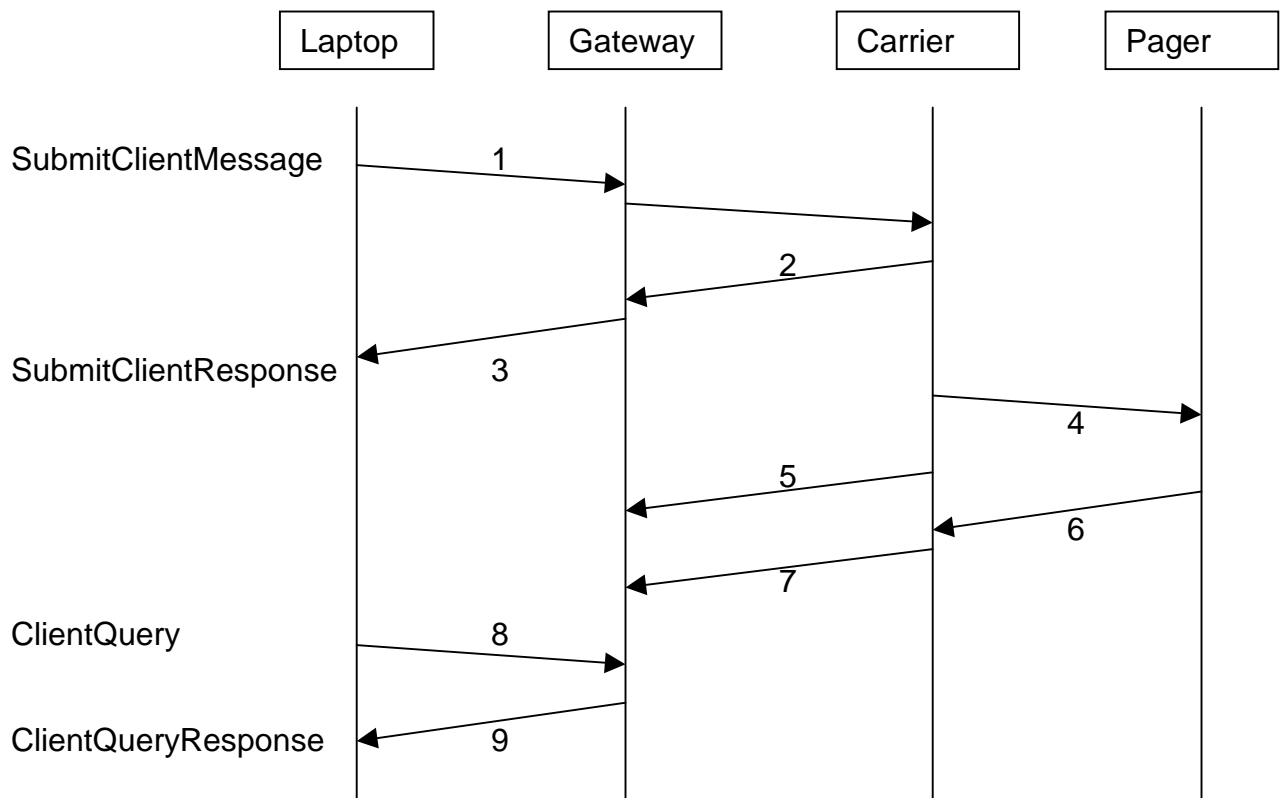
F.12 Use Case #12: An MCR message from a transient client or enterprise host

There are actually 3 different use cases here:

- A transient user originated MCR and a subsequent MCR response
- A non-polling enterprise host originated MCR and a subsequent MCR response
- A polling enterprise host originated MCR and a subsequent MCR response

F.12.1 A transient user originated MCR and a subsequent MCR response.

Assume recipient on same network/carrier as WCTP Gateway.



1. Submit message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientMessage>
    <wctp-SubmitClientHeader
      submitTimestamp="1999-03-31T19:45:00"
    >
  </wctp-SubmitClientHeader>
</wctp-SubmitClientMessage>
</wctp-Operation>
    
```

```

>
  <wctp-ClientOriginator
    senderID="mylaptop@myisp.com"
  />
  <wctp-ClientMessageControl
    allowResponse="true"
    notifyWhenDelivered="true"
  />
  <wctp-Recipient
    recipientID="userId@MyCarrier.com"
  />
</wctp-SubmitClientHeader>
<wctp-Payload>
  <wctp-MCR>
    <wctp-MessageText>
      Test MCR page from my laptop to my pager
    </wctp-MessageText>
    <wctp-Choice>
      text of MCR choice 1
    </wctp-Choice>
    <wctp-Choice>
      text of MCR choice 2
    </wctp-Choice>
    <wctp-Choice>
      text of MCR choice 3
    </wctp-Choice>
  </wctp-MCR>
</wctp-Payload>
</wctp-SubmitClientMessage>
</wctp-Operation>

```

2. MyCarrier accepts request, queues message for delivery and returns tracking number.

3. Request accepted:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientResponse>
    <wctp-ClientSuccess
      successCode="200"
      successText="Message accepted"
      trackingNumber="5579"
    >
      Thank you for using MyCarrier
    </wctp-ClientSuccess>
  </wctp-SubmitClientResponse>
</wctp-Operation>

```

4. MyCarrier delivers page to pager

5. MyCarrier returns "delivered" notification associated with tracking number 5579, this sender & recipient to the gateway.

6. Subscriber sends MCR response #2 from pager

7. MyCarrier returns an MCR response associated with tracking number 5579, this sender & recipient to the gateway.

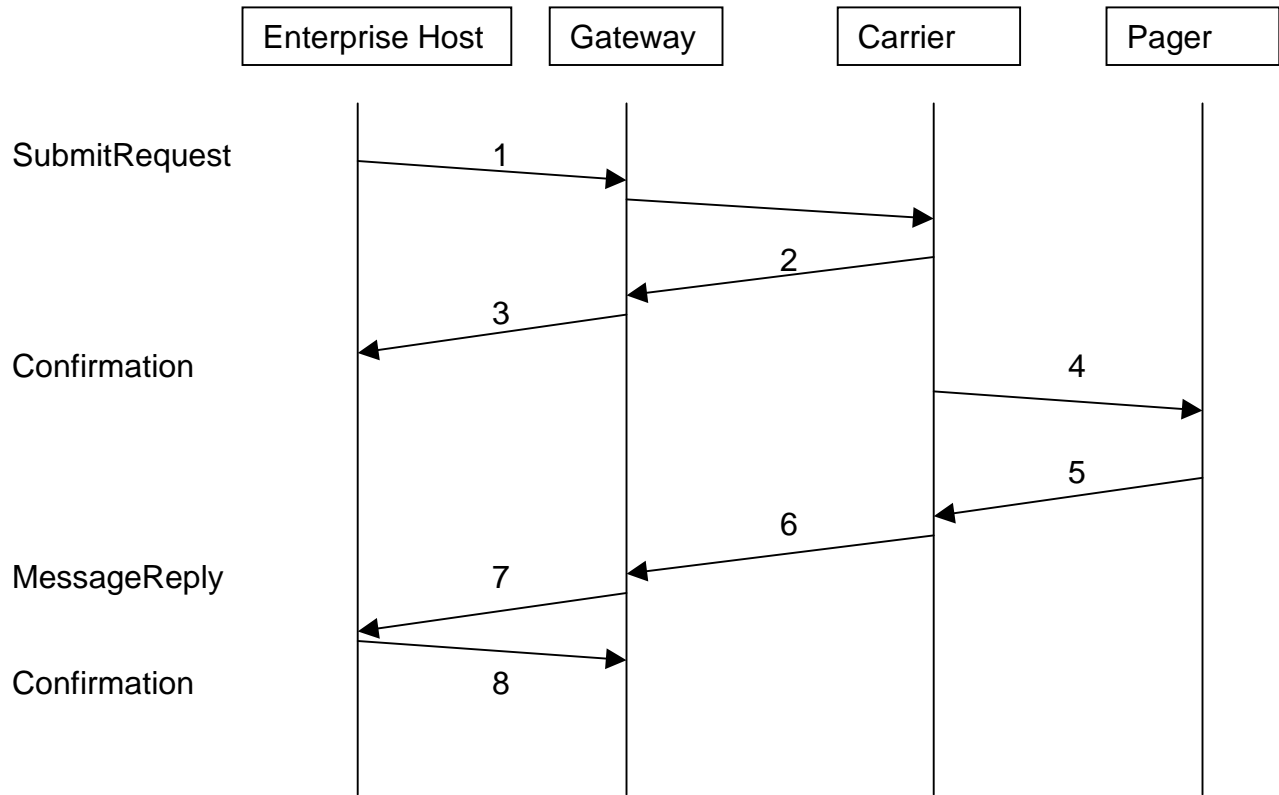
8. Check for response(s):

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-ClientQuery
    senderID="mylaptop@myisp.com"
    recipientID="userId@MyCarrier.com"
    trackingNumber="5579"
  />
</wctp-Operation>
```

9. Receive queued responses as well as any returned statuses:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-ClientQueryResponse>
    <wctp-ClientMessage>
      <wctp-ClientStatusInfo>
        <wctp-ClientResponseHeader
          responseTimestamp="1999-03-31T19:46:10"
          respondingToTimestamp="1999-03-31T19:45:00"
        >
          <wctp-Originator
            senderId="userId@MyCarrier.com"
          />
          <wctp-Recipient
            recipientId="mylaptop@myisp.com"
          />
        </wctp-ClientResponseHeader>
        <wctp-Notification
          type="DELIVERED"
        />
      </wctp-ClientStatusInfo>
    </wctp-ClientMessage>
    <wctp-ClientMessage>
      <wctp-ClientMessageReply>
        <wctp-ClientResponseHeader
          responseTimestamp="1999-03-31T21:16:09"
          respondingToTimestamp="1999-03-31T19:45:00"
        >
          <wctp-Originator
            senderId="userId@MyCarrier.com"
          />
          <wctp-Recipient
            recipientId="mylaptop@myisp.com"
          />
        </wctp-ClientResponseHeader>
        <wctp-Payload>
          <wctp-Alphanumeric>
            text of MCR choice 2
          </wctp-Alphanumeric>
        </wctp-Payload>
      </wctp-ClientMessageReply>
    </wctp-ClientMessage>
  </wctp-ClientQueryResponse>
</wctp-Operation>
```

F.12.2 A non-polling enterprise host originated MCR and a subsequent MCR response



1. Enterprise host submits MCR message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader
      submitTimestamp="1999-03-31T18:18:00"
    >
      <wctp-Originator
        senderID="controlcenter@myenterprise.com"
      />
      <wctp-MessageControl
        messageID="46264399"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="john.tech@skypage.com"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-MCR>
        <wctp-MessageText>
          pH threshold exceeded. Location: Louisville, KY.
          TankNumber: 1234. DeviceId: Abcwireless.1234567
          Select a Corrective Action
        </wctp-MessageText>
      </wctp-MCR>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>
  
```

```

        <wctp-Choice>
            Raise temperature 1 degree.
        </wctp-Choice>
        <wctp-Choice>
            Lower temperature 1 degree.
        </wctp-Choice>
        <wctp-Choice>
            Raise pressure 1 atm.
        </wctp-Choice>
        <wctp-Choice>
            Lower pressure 1 atm.
        </wctp-Choice>
    </wctp-MCR>
</wctp-Payload>
</wctp-SubmitRequest>
</wctp-Operation>

```

2. SkyPage accepts request, queues message for delivery.
3. Gateway replies to indicate submission request accepted:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-Confirmation>
        <wctp-Success
            successCode="200"
            successText="Message accepted"
        >
            Thank you for using SkyPage
        </wctp-Success>
    </wctp-Confirmation>
</wctp-Operation>

```

4. SkyPage delivers page to pager.
5. Subscriber sends MCR response #1 from pager
6. SkyPage sends MCR response associated with messageId 46264399.
7. Gateway delivers MCR response:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-MessageReply>
        <wctp-ResponseHeader
            responseToMessageID="46264399"
            responseTimestamp="1999-03-31T18:25:00"
            respondingToTimestamp="1999-03-31T18:18:00"
        >
            <wctp-Originator
                senderID="john.tech@skypage.com"
                miscInfo="John The Technician"
            />
            <wctp-MessageControl
                messageID="33"
                transactionID="19990331.PN.1234567.001"
            />
        <wctp-Recipient

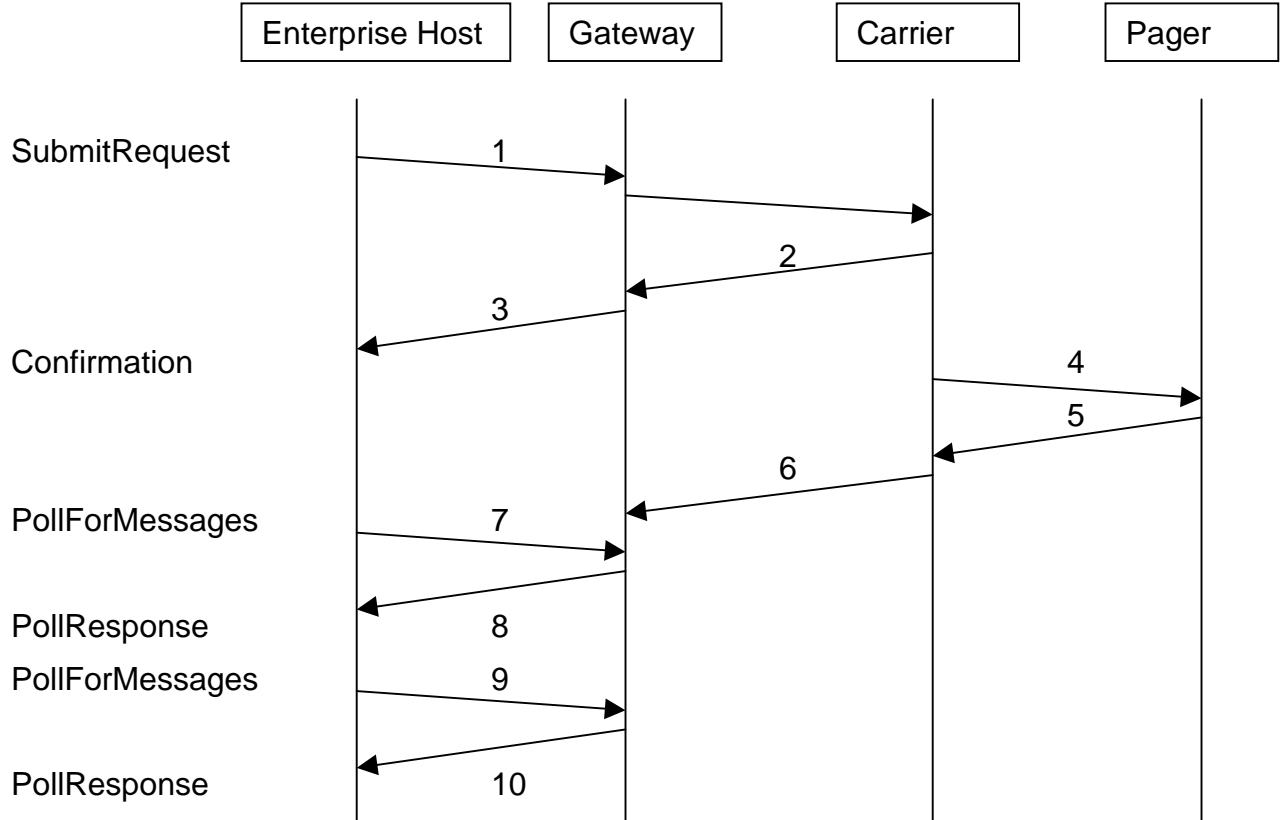
```

```
        recipientID="controlcenter@myenterprise.com"
    />
</wctp-ResponseHeader>
<wctp-Payload>
    <wctp-Alphanumeric>
        Raise temperature 1 degree.
    </wctp-Alphanumeric>
</wctp-Payload>
</wctp-MessageReply>
</wctp-Operation>
```

8. Enterprise host confirms receipt of MCR response:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-Confirmation>
        <wctp-Success
            successCode="200"
            successText="Message accepted"
        >
            Thankyou for using MyEnterprise
        </wctp-Success>
    </wctp-Confirmation>
</wctp-Operation>
```

F.12.3 A polling enterprise host originated MCR and a subsequent MCR response



1. Enterprise host submits MCR message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader>
      <wctp-SubmitHeader
        submitTimestamp="1999-03-31T18:18:00">
      <wctp-Originator
        senderID="controlcenter@myenterprise.com"
      />
    </wctp-SubmitHeader>
    <wctp-MessageControl
      messageID="46264399"
      transactionID="19990331.PN.1234567.001"
    />
    <wctp-Recipient
      recipientID="john.tech@skypage.com"
    />
  </wctp-SubmitHeader>
  <wctp-Payload>
    <wctp-MCR>
      <wctp-MessageText>
        pH threshold exceeded. Location: Louisville, KY.
        TankNumber: 1234. DeviceId: Abcwireless.1234567
        Select a Corrective Action
      </wctp-MessageText>
    </wctp-MCR>
  </wctp-Payload>
</wctp-SubmitRequest>
</wctp-Operation>
  
```

```

        </wctp-MessageText>
        <wctp-Choice>
            Raise temperature 1 degree.
        </wctp-Choice>
        <wctp-Choice>
            Lower temperature 1 degree.
        </wctp-Choice>
        <wctp-Choice>
            Raise pressure 1 atm.
        </wctp-Choice>
        <wctp-Choice>
            Lower pressure 1 atm.
        </wctp-Choice>
    </wctp-MCR>
</wctp-Payload>
</wctp-SubmitRequest>
</wctp-Operation>

```

2. SkyPage accepts request, queues message for delivery.

3. Gateway replies to indicate submission request accepted:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-Confirmation>
        <wctp-Success
            successCode="200"
            successText="Message accepted"
        >
            Thankyou for using SkyPage
        </wctp-Success>
    </wctp-Confirmation>
</wctp-Operation>

```

4. SkyPage delivers page to pager.

5. Subscriber sends MCR response #1 from pager

6. SkyPage queues MCR response associated with messageId 46264399 for MyEnterprise.com

7. Enterprise host polls to for any queued responses and notifications:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-PollForMessages
        pollerID="myenterprise.com"
        securityCode="qwerty"
        maxMessagesInBatch="10"
    />
</wctp-Operation>

```

8. Gateway replies to poll with queued MCR response:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation

```

```

SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse>
    <wctp-Message
      sequenceNo="57"
    >
      <wctp-MessageReply>
        <wctp-ResponseHeader
          responseToMessageID="46264399"
          responseTimestamp="1999-03-31T18:25:00"
          respondingToTimestamp="1999-03-31T18:18:00"
        >
          <wctp-Originator
            senderID="john.tech@skypage.com"
            miscInfo="John The Technician"
          />
          <wctp-MessageControl
            messageID="33"
            transactionID="19990331.PN.1234567.001"
          />
          <wctp-Recipient
            recipientID="controlcenter@myenterprise.com"
          />
        </wctp-ResponseHeader>
        <wctp-Payload>
          <wctp-Alphanumeric>
            Raise temperature 1 degree.
          </wctp-Alphanumeric>
        </wctp-Payload>
      </wctp-MessageReply>
    </wctp-Message>
  </wctp-PollResponse>
</wctp-Operation>

```

9. Enterprise host confirms receipt of MCR response in next poll:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="myenterprise.com"
    securityCode="qwerty"
    maxMessagesInBatch="10"
  >
    <wctp-MessageReceived
      sequenceNo="57"
    >
      <wctp-Success
        successCode="200"
        successText="Message accepted"
      >
        Thank you for using MyEnterprise
      </wctp-Success>
    </wctp-MessageReceived>
  </wctp-PollForMessages>
</wctp-Operation>

```

10. Gateway replies to poll:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse>
    <wctp-NoMessages/>
  </wctp-PollResponse>
</wctp-Operation>
```

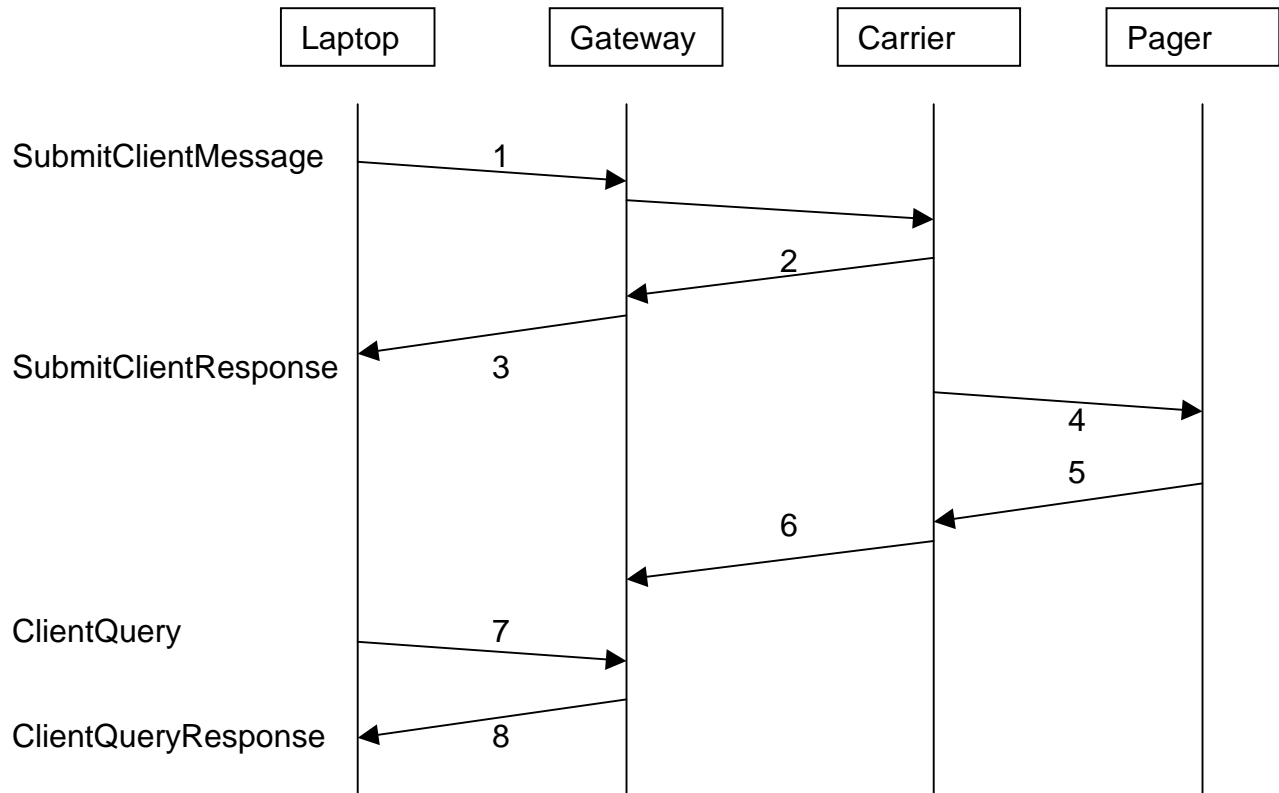

F.13 Use Case #13: A binary message from a transient client or enterprise host

There are 4 distinct cases:

- A binary message originated by a transient user and a subsequent response to it
- A binary message originated by a non-polling enterprise host and a subsequent response to it
- A binary message originated by a polling enterprise host and a subsequent response to it
- A binary message originated by an application running on a wireless device sent to an enterprise host and a subsequent response to it

F.13.1 A binary message originated by a transient user and a subsequent response to it

Assume recipient on same network/carrier as WCTP Gateway.



1. Submit message (uses canonical XML encoding):

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientMessage>
    <wctp-SubmitClientHeader
      submitTimestamp="1999-03-31T19:45:00"
    >
      <wctp-ClientOriginator
        senderID="mylaptop@myisp.com"
        ...
      />
      <wctp-ClientMessageControl
        allowResponse="true"
        ...
      />
      <wctp-Recipient
        recipientID="userId@MyCarrier.com"
        ...
      />
    </wctp-SubmitClientHeader>
    <wctp-Payload>
  
```

```

    <wctp-TransparentData
      type="OPAQUE"
    >
      &#x041F;&#x0440;&#x0438;&#x0432;&#x0435;&#x0442;&#x0021;
    </wctp-TransparentData>
  </wctp-Payload>
</wctp-SubmitClientMessage>
</wctp-Operation>

```

2. MyCarrier accepts request, queues message for delivery and returns tracking number.

3. Request accepted:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientResponse>
    <wctp-ClientSuccess
      successCode="200"
      successText="Message accepted"
      trackingNumber="5579"
    >
      Thank you for using MyCarrier
    </wctp-ClientSuccess>
  </wctp-SubmitClientResponse>
</wctp-Operation>

```

4. MyCarrier delivers page to pager.

5. Subscriber sends binary response from pager

6. MyCarrier queues the response associated with tracking number 5579, this sender & recipient.

7. Check for response(s):

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-ClientQuery
    senderID="mylaptop@myisp.com"
    recipientID="userId@MyCarrier.com"
    trackingNumber="5579"
  />
</wctp-Operation>

```

8. Receive queued response:

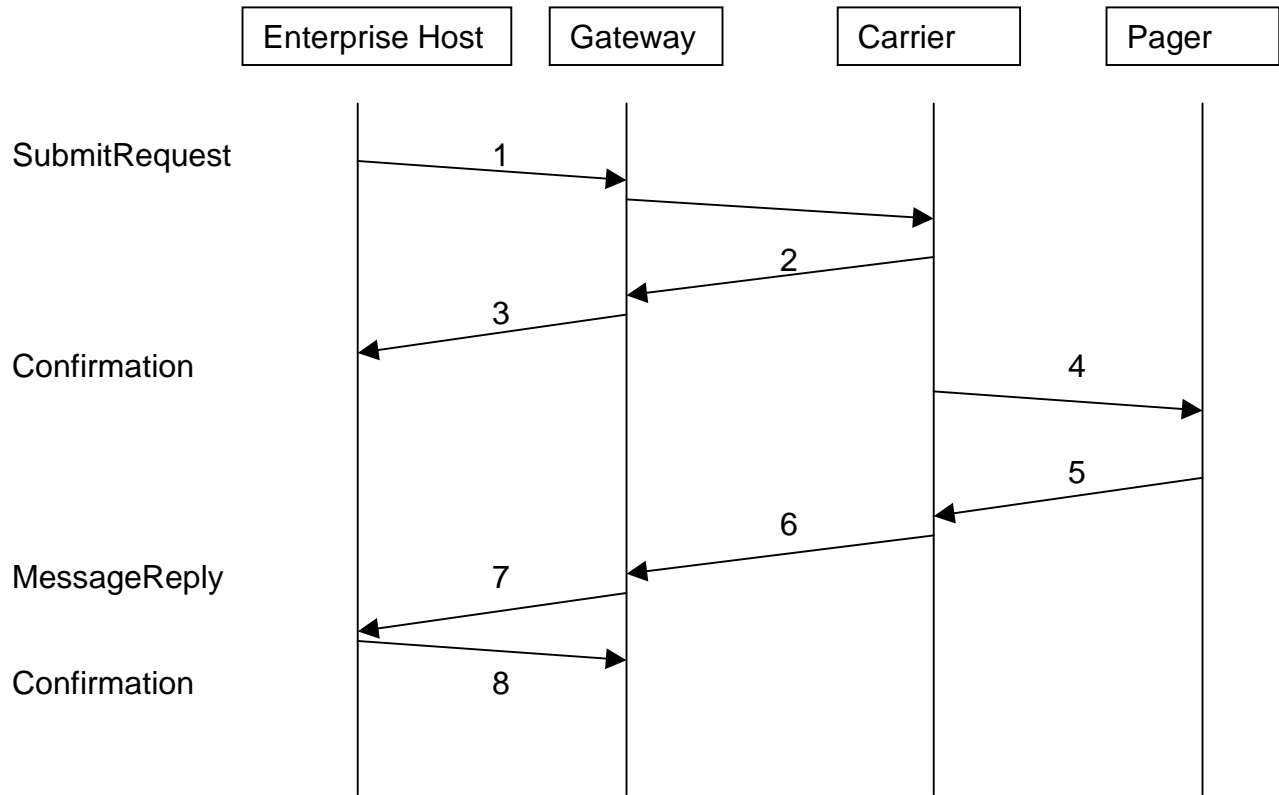
```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-ClientQueryResponse>
    <wctp-ClientMessage>
      <wctp-MessageReply>
        <wctp-ResponseHeader>

```

```
<wctp-Originator
  senderId="userId@MyCarrier.com"
/>
<wctp-MessageControl
  messageId="92"
/>
<wctp-Recipient
  recipientId="mylaptop@myisp.com"
/>
</wctp-ResponseHeader>
<wctp-Payload>
  <wctp-TransparentData>
&#x041D;&#x0443;&#x0020;&#x043F;&#x0440;&#x0438;&#x0432;&#x0435;&#x0442;
  </wctp-TransparentData>
  </wctp-Payload>
</wctp-MessageReply>
</wctp-ClientMessage>
</wctp-ClientQueryResponse>
</wctp-Operation>
```

F.13.2 A binary message originated by a non-polling enterprise host and a subsequent response to it



1. Enterprise host submits binary message (encoded as base64):

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader>
      submitTimestamp="1999-03-31T18:18:00"
    >
    <wctp-Originator>
      senderID="controlcenter@myenterprise.com"
    />
    <wctp-MessageControl>
      messageID="46264399"
      transactionID="19990331.PN.1234567.001"
    />
    <wctp-Recipient>
      recipientID="smartDevice@skypage.com"
    />
  </wctp-SubmitHeader>
  <wctp-Payload>
    <wctp-TransparentData>
      type="OPAQUE"
      encoding="base64"
    >
  </wctp-Payload>
</wctp-Operation>
  
```

```

        PD9YTUwgdMvYc2lrbj0iMS4wIiBlb
    </wctp-TransparentData>
</wctp-Payload>
</wctp-SubmitRequest>
</wctp-Operation>

```

2. SkyPage accepts request, queues message for delivery.
3. Gateway replies to indicate submission request accepted:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      Thank you for using SkyPage
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>

```

4. SkyPage delivers page to pager.
5. Subscriber sends binary response from pager
6. SkyPage sends the response associated with messageId 46264399.
7. Gateway delivers the response (also encoded in base64):

```

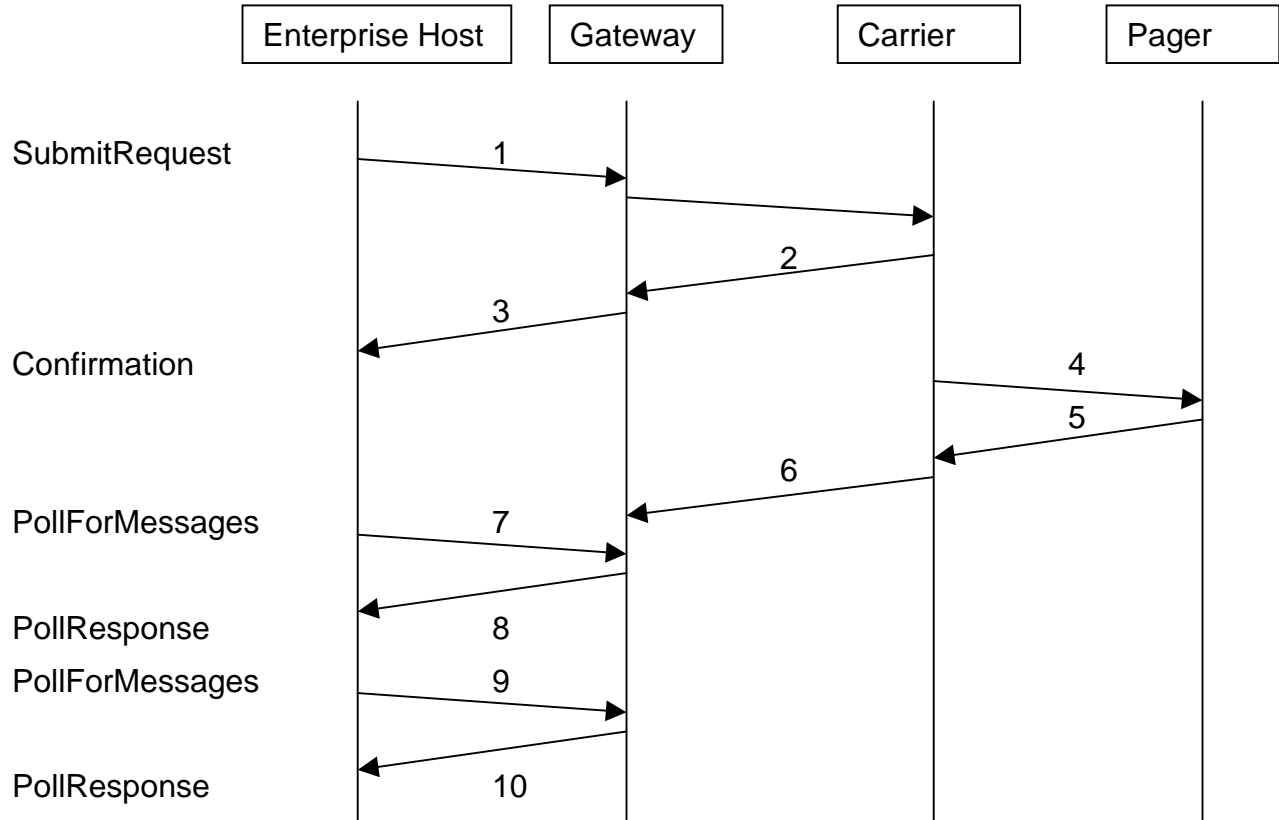
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-MessageReply>
    <wctp-ResponseHeader
      responseToMessageID="46264399"
      responseTimestamp="1999-03-31T18:25:00"
      respondingToTimestamp="1999-03-31T18:18:00"
    >
      <wctp-Originator
        senderID="smartDevice@skypage.com"
      />
      <wctp-MessageControl
        messageID="33"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="controlcenter@myenterprise.com"
      />
    </wctp-ResponseHeader>
    <wctp-Payload>
      <wctp-TransparentData
        encoding="base64"
      >
        HJG76knmjkJHKJH78GjkjhUY87Qtyu==
      </wctp-TransparentData>
    </wctp-Payload>
  </wctp-MessageReply>
</wctp-Operation>

```

8. Enterprise host confirms receipt of MCR response:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      Thank you for using MyEnterprise
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>
```

F.13.3 A binary message originated by a polling enterprise host and a subsequent response to it



1. Enterprise host submits binary message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader>
      <wctp-SubmitHeader
        submitTimestamp="1999-03-31 18:18:00">
      <wctp-Originator
        senderID="controlcenter@myenterprise.com"
      ...
    />
    <wctp-MessageControl
      messageID="46264399"
      transactionID="19990331.PN.1234567.001"
    ...
  />
  <wctp-Recipient
    recipientID="john.tech@skypage.com"
  />
</wctp-SubmitHeader>
<wctp-Payload>
  <wctp-TransparentData
    type="OPAQUE"
    encoding="base64"
  
```



```

    >
      HJG76knmjkJHKJH78GjkhUY87Qtyu
    </wctp-TransparentData>
  </wctp-Payload>
</wctp-SubmitRequest>
</wctp-Operation>

```

2. SkyPage accepts request, queues message for delivery.
3. Gateway replies to indicate submission request accepted:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      Thank you for using SkyPage
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>

```

4. SkyPage delivers page to pager.
5. Subscriber sends binary response from pager
6. SkyPage queues the response associated with messageId 46264399 for MyEnterprise.com
7. Enterprise host polls to for any queued responses and notifications:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="MyEnterprise"
    securityCode="qwerty"
    maxMessagesInBatch="10"
  />
</wctp-Operation>

```

8. Gateway replies to poll with queued response:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse>
    <wctp-Message
      sequenceNo="1"
    >
      <wctp-MessageReply>
        <wctp-ResponseHeader
          responseToMessageID="46264399"
          responseTimestamp="1999-03-31T18:25:00"
          respondingToTimestamp="1999-03-31T18:18:00"
        >

```

```

    >
      <wctp-Originator
        senderID="john.tech@skypage.com"
        miscInfo="John The Technician"
      />
      <wctp-MessageControl
        messageID="33"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="controlcenter@myenterprise.com"
      />
    </wctp-ResponseHeader>
    <wctp-Payload>
      <wctp-TransparentData
        encoding="base64"
      >
        HJG76knmjkJHKJH78GjkhUY87Qtyu
      </wctp-TransparentData>
    </wctp-Payload>
  </wctp-MessageReply>
</wctp-Message>
</wctp-PollResponse>
</wctp-Operation>

```

9. Enterprise host confirms receipt of the response in next poll:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="MyEnterprise"
    securityCode="qwerty"
    maxMessagesInBatch="10"
  >
    <wctp-MessageReceived
      sequenceNo="1"
    >
      <wctp-Success
        successCode="200"
        successText="Message accepted"
      >
        Thank you for using MyEnterprise
      </wctp-Success>
    </wctp-MessageReceived>
  </wctp-PollForMessages>
</wctp-Operation>

```

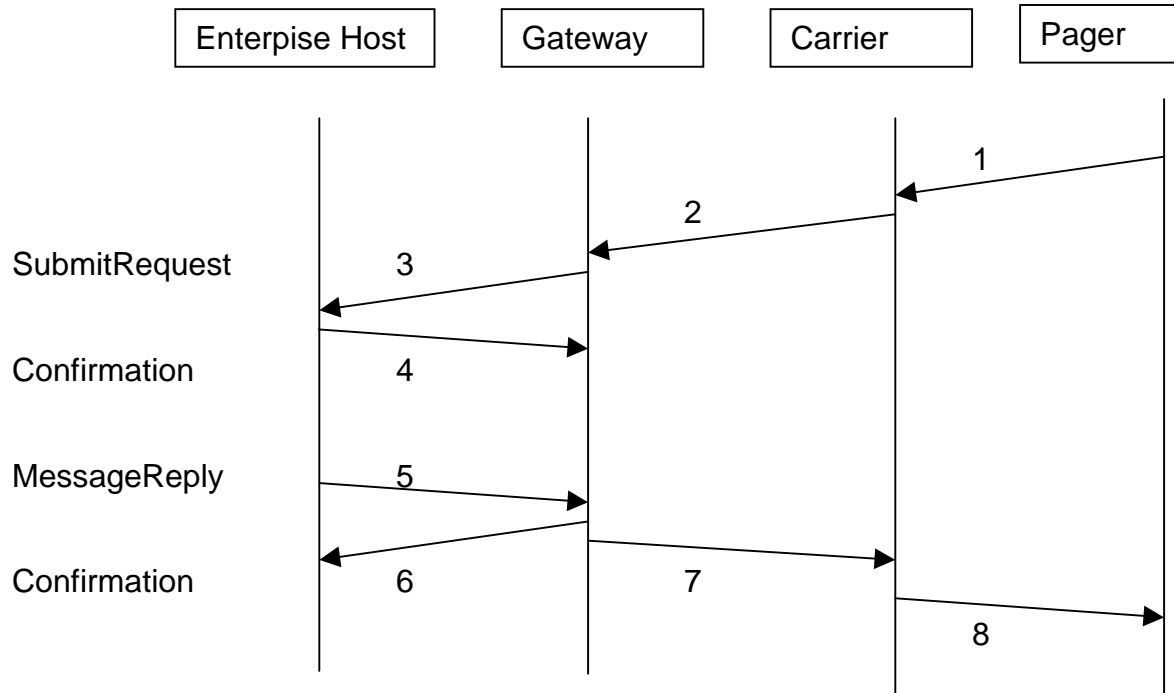
10. Gateway replies to poll:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse>
    <wctp-MessageNoMessages/>
  </wctp-PollResponse>
</wctp-Operation>

```

F.13.4 A binary message from a wireless application to an enterprise host and a subsequent response



1. Wireless device submits binary FLEXsuite message
2. SkyPage queues the message with messageID 46 for MyEnterprise.com
3. Gateway delivers the message (encoded in base64):

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader
      submitTimestamp="1999-03-31T18:18:00">
      <wctp-Originator
        senderID="telemetryDevice@skypage.com"
      />
    <wctp-MessageControl
      messageID="46"
      transactionID="19990331.PN.1234567.001"
    />
    <wctp-Recipient
      recipientID="controlcenter@myEnterprise.com"
    />
  </wctp-SubmitHeader>
  <wctp-Payload>
    <wctp-TransparentData
      type="FLEXsuite"
      encoding="base64"
    >
      PD9YTUwgdmVyc2lvcj0iMS4wIiBlb
    </wctp-TransparentData>
  </wctp-Payload>

```

```
</wctp-SubmitRequest>
</wctp-Operation>
```

4. Enterprise host confirms receipt of the response:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation>
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      Thank you for using MyEnterprise
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>
```

5. Enterprise host replies with a binary FLEXsuite message (also encoded in base64):

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-MessageReply>
    <wctp-ResponseHeader
      responseToMessageID="46264399"
      responseTimestamp="1999-03-31T18:25:00"
      respondingToTimestamp="1999-03-31T18:18:00"
    >
      <wctp-Originator
        senderID="controlcenter@myenterprise.com"
      />
      <wctp-MessageControl
        messageID="33"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="telemetryDevice@skypage.com"
      />
    </wctp-ResponseHeader>
    <wctp-Payload>
      <wctp-TransparentData
        type="FLEXsuite"
        encoding="base64"
      >
        HFHG8768jUYTUyTuyuu6jh47YH
      </wctp-TransparentData>
    </wctp-Payload>
  </wctp-MessageReply>
</wctp-Operation>
```

6. Gateway confirms the receipt of the reply:

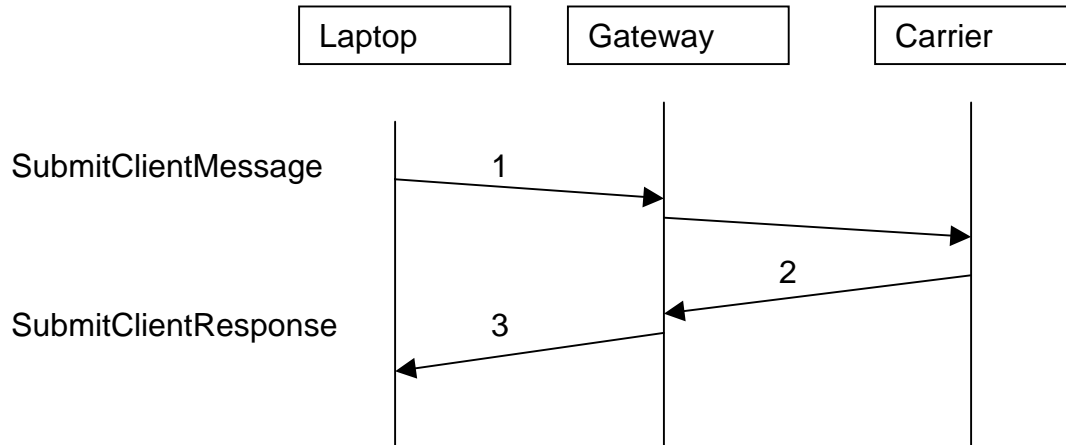
```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation>
```

```
<wctp-Confirmation>
  <wctp-Success
    successCode="200"
    successText="Message accepted"
  >
    Thank you for using SkyPage
  </wctp-Success>
</wctp-Confirmation>
</wctp-Operation>
```

7. SkyPage queues the reply for delivery
8. Message is delivered to the wireless device.

F.14 Use Case #14: A Transient user or server initiated message with authorization code

The following is the case of using the correct authorization code to send the message. Assume recipient on same network/carrier as WCTP Gateway.



1. Submit message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientMessage>
    <wctp-SubmitClientHeader
      submitTimestamp="1999-03-31T19:45:00"
    >
      <wctp-ClientOriginator
        senderID="mylaptop@myisp.com"
      />
      <wctp-Recipient
        recipientID="1234567"
        authorizationCode="myAuthorizationCode"
      />
    </wctp-SubmitClientHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        Test page from my laptop to my pager
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-SubmitClientMessage>
</wctp-Operation>
  
```

2. MyCarrier accepts request, validates the authorization code, queues message for delivery, and returns tracking number.

3. Request accepted:

```

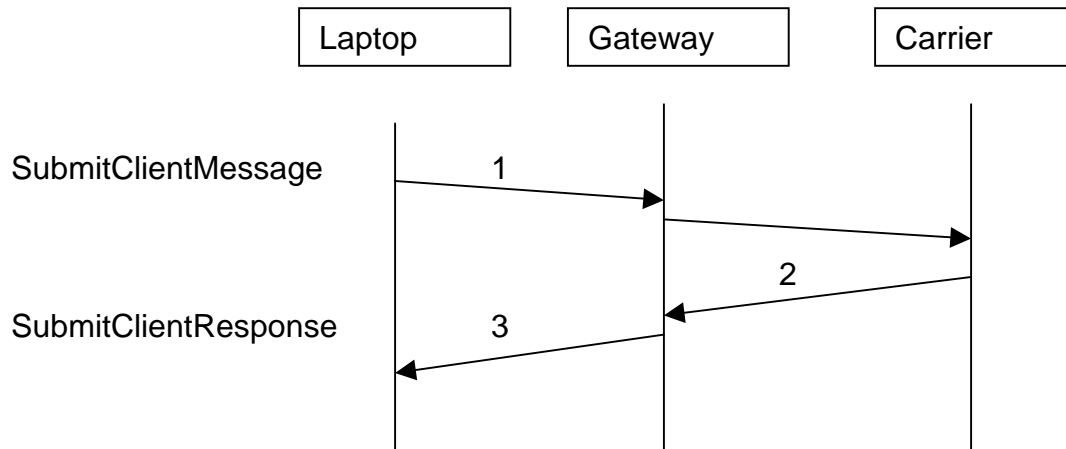
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion=" 1.0" >
  
```

```

<wctp-SubmitClientResponse>
  <wctp-ClientSuccess
    successCode="200"
    successText="Message accepted"
    trackingNumber="5579"
  >
    MyCarrier is happy to send your message as page!
  </wctp-ClientSuccess>
</wctp-SubmitClientResponse>
</wctp-Operation>

```

The following is the case of using the incorrect authorization code to send the message. Assume recipient on same network/carrier as WCTP Gateway.



1. Submit message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientMessage>
    <wctp-SubmitClientHeader
      submitTimestamp="1999-03-31T19:45:00"
    >
      <wctp-ClientOriginator
        senderID="mylaptop@myisp.com"
      />
      <wctp-Recipient
        recipientID="userId@MyCarrier.com"
        authorizationCode="myIncorrectAuthorizationCode"
      />
    </wctp-SubmitClientHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        Test page from my laptop to my pager
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-SubmitClientMessage>
</wctp-Operation>

```

2. MyCarrier accepts request, checks the authorization code, determines that the authorization code is incorrect, and returns the associated error code.
3. Return error code.

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitClientResponse>
    <wctp-Failure
      errorCode="404"
      errorText="Invalid authorization code"
    >
      Supplied authorization code for userId is invalid.
    </wctp-Failure>
  </wctp-SubmitClientResponse>
</wctp-Operation>
```


F.15 Use Case #15: An enterprise host asking for all of its messages

Sample sequence of operation follows. Note that values of some parameters are shown for illustration purpose only and in no way imply suggested values for implementation (e.g. maxReturnPerPoll).

1. Enterprise host johnson_international.com is not able to run an HTTP server. Wireless carrier therefore sets up an entry for johnson_international.com in PollRegistrationTable.
2. Transaction is started as a message from device is received at WCG for carrier "Skypage" with recipient as "Pageto:joe@johnson_international.com".
3. WCG determines this message should be delivered via WCTP based on "PageTo:" portion of Recipient and a wctp-Operation (one of wctp-SubmitRequest, wctp-StatusInfo, wctp-MessageReply) is generated for this message. It then sends the message to Internet Gateway.
4. Internet Gateway consults its PollRegistrationTable and finds an entry for "johnson_international.com". It therefore holds the message at the Gateway.
5. Steps 2 through 4 is repeated 3 times, resulting in the construction and holding of the 3 messages
6. EH "johnson_international.com" polls skypage for its messages by sending an HTTP Full-Request as follows:

POST <http://wctp.skypage.com/wctp> HTTP/1.1

with Entity-Body:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="johnson_international.com"
    maxMessagesInBatch="2"
  />
</wctp-Operation>
```

maxMessagesInBatch attributes indicates that no more than 2 messages at a time can be returned to EH of johnson_international.com.

7. Internet Gateway retrieves the held messages based on pollerID and sends an HTTP Full-Response for wctp-PollResponse as follows:

HTTP/1.1 200 OK

with Entity-Body:

```
<?xml version="1.0"?>
```

```

<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse>
    <wctp-Message
      sequenceNo ="1"
    >
      <wctp-MessageReply>
        <wctp-ResponseHeader
          responseToMessageID="26"
          responseTimestamp="1999-04-13T18:15:30"
        >
          <wctp-Originator
            senderID="1234567@skypage.com"
          />
          <wctp-MessageControl
            messageID="11936B93"
          />
          <wctp-Recipient
            recipientID="joe@johnson_international.com"
          />
        </wctp-ResponseHeader>
        <wctp-Payload>
          <wctp-Alphanumeric>
            "wed. 11:30 lunch sounds fine."
          </wctp-Alphanumeric>
        </wctp-Payload>
      </wctp-MessageReply>
    </wctp-Message>
    <wctp-Message
      sequenceNo="2"
    >
      <wctp-SubmitRequest>
        <wctp-SubmitHeader
          submitTimestamp="1999-04-13T18:16:00"
        >
          <wctp-Originator
            senderID="2234567@skypage.com"
          />
          <wctp-MessageControl
            messageID="11936B94"
          />
          <wctp-Recipient
            recipientID="sales@johnson_international.com"
          />
        </wctp-SubmitHeader>
        <wctp-Payload>
          <wctp-Alphanumeric>
            "ship 1999 fall brochure to KC field office"
          </wctp-Alphanumeric>
        </wctp-Payload>
      </wctp-SubmitRequest>
    </wctp-Message>
  </wctp-PollResponse>
</wctp-Operation>

```

8. An HTTP Full-Request from EH "johnson_international.com" is sent to Internet Gateway, requesting more messages and acking received ones, as follows:

POST <http://wctp.skypage.com/wctp> HTTP/1.1

with Entity-Body:

```
<?xml version="1.0"?>
```

```

<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="johnson_international.com"
  >
    <wctp-MessageReceived
      sequenceNo="1"
    >
      <wctp-Success
        successCode="200"
        successText="Message accepted"
      >
        Thank you for using Johnson International
      </wctp-Success>
    </wctp-MessageReceived>
    <wctp-MessageReceived
      sequenceNo="2"
    >
      <wctp-Success
        successCode="200"
        successText="Message accepted"
      >
        Thank you for using Johnson International
      </wctp-Success>
    </wctp-MessageReceived>
  </wctp-PollForMessages>
</wctp-Operation>

```

9. An HTTP Full-Response for wctp-PollResponse is generated at Internet Gateway to send remaining message, as follows:

HTTP/1.1 200 OK

with Entity-Body:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse>
    <wctp-Message
      sequenceNo="3"
    >
      <wctp-StatusInfo>
        <wctp-ResponseHeader
          reponseToMessageID="27"
          responseTimestamp="1999-04-13T18:16:07"
        >
          <wctp-Originator
            senderID="3234567@skypage.com"
          />
          <wctp-MessageControl
            messageID="11936B95"
          />
          <wctp-Recipient
            recipientID="dispatch@johnson_international.com"
          />
        </wctp-ResponseHeader>
      </wctp-StatusInfo>
    </wctp-Message>
  </wctp-PollResponse>
</wctp-Operation>

```

```

        <wctp-Failure
            errorCode="103"
            errorText="device out of range"
        >
            Device 9876543 is out of range.
        </wctp-Failure>
    </wctp-StatusInfo>
</wctp-Message>
</wctp-PollResponse>
</wctp-Operation>

```

10. An HTTP Full-Request from EH "johnson_international.com" is sent to Internet Gateway, requesting more messages and acking received one, as follows:

POST <http://wctp.skypage.com/wctp> HTTP/1.1

with Entity-Body:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-PollForMessages
        pollerID="johnson_international.com"
    >
        <wctp-MessageReceived
            sequenceNo="3"
        >
            <wctp-Success
                successCode="200"
                successText="Message accepted"
            >
                Thank you for using Johnson International
            </wctp-Success>
        </wctp-MessageReceived>
    </wctp-PollForMessages>
</wctp-Operation>

```

11. Since there are no more messages for johnson_international.com, an HTTP Full-Response for wctp-PollResponse is generated at Internet Gateway to send remaining message, as follows:

HTTP/1.1 200 OK

with Entity-Body:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
    <wctp-PollResponse
        minNextPollInterval="5"
    >
        <wctp-NoMessages/>
    </wctp-PollResponse>
</wctp-Operation>

```

Since there is no more messages held, Internet Gateway chooses to force johnson_international.com to poll at least 5 minutes later. Transaction is completed at this point.

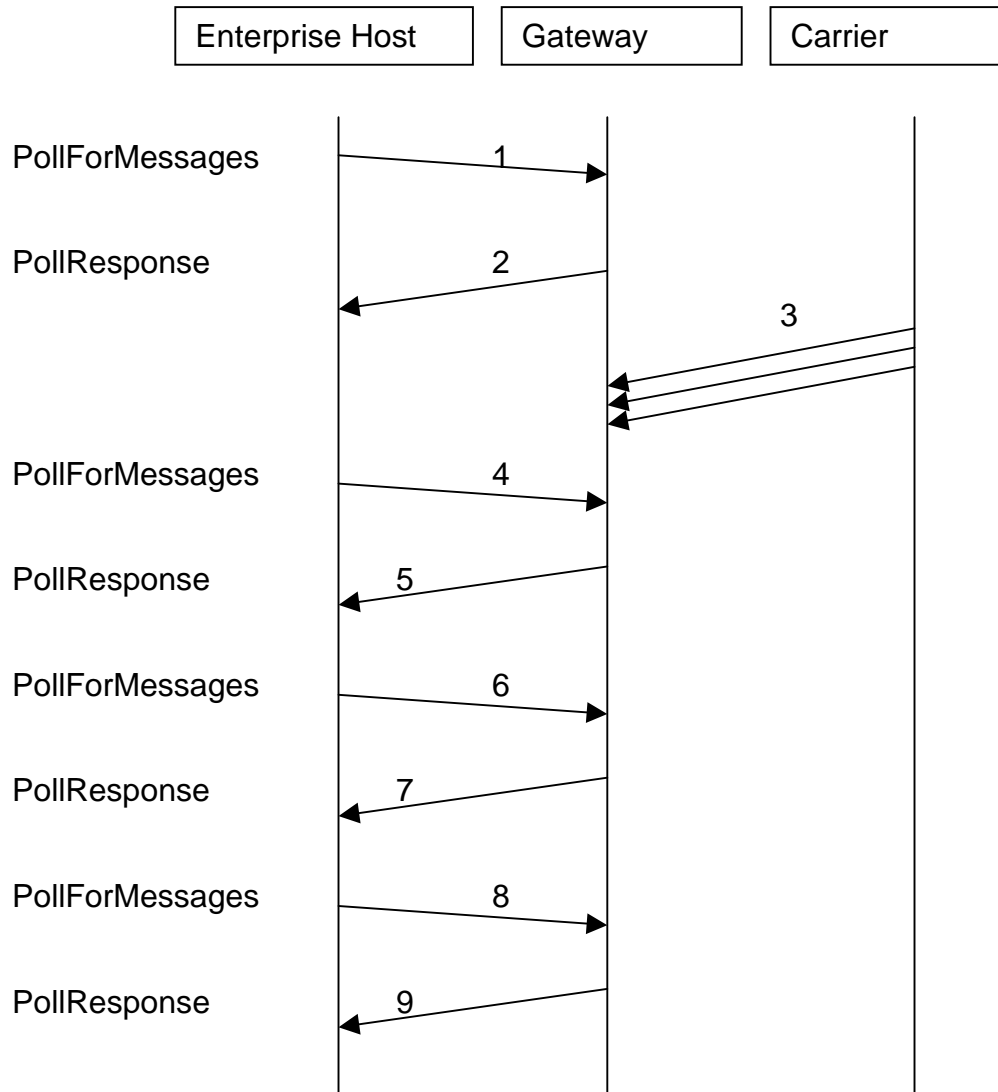
F.16 Use Case #16: A device reply to an enterprise host generated message

F.17 Use Case #17: A device reply to a Transient user generated message

F.18 Use Case #18: Sending back Multiple replies on a Transient user message status query

F.19 Use Case #19: Disabling replies on a message sent to the device

F.20 Use Case #20: Polling by enterprise host for all messages directed at that host



1. Enterprise host polls to for any queued messages:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="MyEnterprise"
    securityCode="qwerty"
    maxMessagesInBatch="2"
  />
</wctp-Operation>
    
```

2. Gateway replies to poll indicating there are no messages queued:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse
    minNextPollInterval="60"
  >
    <wctp-NoMessages>
    </wctp-NoMessages>
  </wctp-PollResponse>
</wctp-Operation>
```

3. Carrier forwards a notification, an MCR, and a pager originated message bound for MyEnterprise to the Gateway to be queued.

4. Enterprise host polls again:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="MyEnterprise"
    securityCode="qwerty"
    maxMessagesInBatch="2"
  />
</wctp-Operation>
```

5. Gateway replies to poll:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse>
    <wctp-Message
      sequenceNo="57"
    >
      <wctp-StatusInfo>
        <wctp-ResponseHeader
          responseToMessageID="46036782"
          responseTimestamp="1999-03-31T18:24:00"
          respondingToTimestamp="1999-03-12T12:37:00"
        >
          <wctp-Originator
            senderID="wcg-3.abcwireless.com"
            miscInfo="WCG for SkyPage"
          />
          <wctp-MessageControl
            messageID="27"
            transactionID="19990312.PN.7561243.001"
          />
          <wctp-Recipient
            recipientID="helpdesk@myenterprise.com"
          />
        </wctp-ResponseHeader>
        <wctp-Notification
          type="DELIVERED"
        />
      </wctp-StatusInfo>
```



```

</wctp-Message>
<wctp-Message
  sequenceNo="58"
  >
  <wctp-MessageReply>
    <wctp-ResponseHeader
      responseToMessageID="46264399"
      responseTimestamp="1999-03-31T18:25:00"
      respondingToTimestamp="1999-03-31T18:18:00"
    >
      <wctp-Originator
        senderID="john.tech@skypage.com"
        miscInfo="John The Technician"
      />
      <wctp-MessageControl
        messageID="33"
        transactionID="19990331.PN.1234567.001"
      />
      <wctp-Recipient
        recipientID="controlcenter@myenterprise.com"
      />
    </wctp-ResponseHeader>
    <wctp-Payload>
      <wctp-Alphanumeric>
        Raise temperature 1 degree.
      </wctp-Alphanumeric>
    </wctp-Payload>
  </wctp-MessageReply>
</wctp-Message>
</wctp-PollResponse>
</wctp-Operation>

```

6. Enterprise host polls again acknowledging the received messages:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="MyEnterprise"
    securityCode="qwerty"
    maxMessagesInBatch="10"
  >
    <wctp-MessageReceived
      sequenceNo="57"
    >
      <wctp-Success
        successCode="200"
        successText="Message accepted"
      >
        Thank you for using MyEnterprise
      </wctp-Success>
    </wctp-MessageReceived>
    <wctp-MessageReceived
      sequenceNo="58"
    >
      <wctp-Failure
        errorCode="502"
        errorText="No further responses possible"
      >
        The message associated with id 33 exceeded the retention
        threshold and has been deleted.
      </wctp-Failure>
    </wctp-MessageReceived>
  </wctp-PollForMessages>
</wctp-Operation>

```

```

</wctp-PollForMessages>
</wctp-Operation>

```

7. Gateway replies to poll with last queued message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollResponse
    minNextPollInterval="1"
  >
    <wctp-Message
      sequenceNo="59"
    >
      <wctp-SubmitRequest>
        <wctp-SubmitHeader
          submitTimestamp="1999-03-31T18:30:00"
        >
          <wctp-Originator
            senderID="bill.repair@skypage.com"
            miscInfo="Bill The Repairman"
          />
          <wctp-MessageControl
            messageID="47"
            transactionID="19990331.PN.9876547.001"
          />
          <wctp-Recipient
            recipientID="helpdesk@myenterprise.com"
          />
        </wctp-SubmitHeader>
        <wctp-Payload>
          <wctp-Alphanumeric>
            Please send specs for part # A/24d
          </wctp-Alphanumeric>
        </wctp-Payload>
      </wctp-SubmitRequest>
    </wctp-Message>
  </wctp-PollResponse>
</wctp-Operation>

```

8. Enterprise host polls again acknowledging the received message:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-PollForMessages
    pollerID="MyEnterprise"
    securityCode="qwerty"
    maxMessagesInBatch="2"
  >
    <wctp-MessageReceived
      sequenceNo="59"
    >
      <wctp-Success
        successCode="200"
        successText="Message accepted"
      >
        Thank you for using MyEnterprise
      </wctp-Success>
    </wctp-MessageReceived>
  </wctp-PollForMessages>
</wctp-Operation>

```

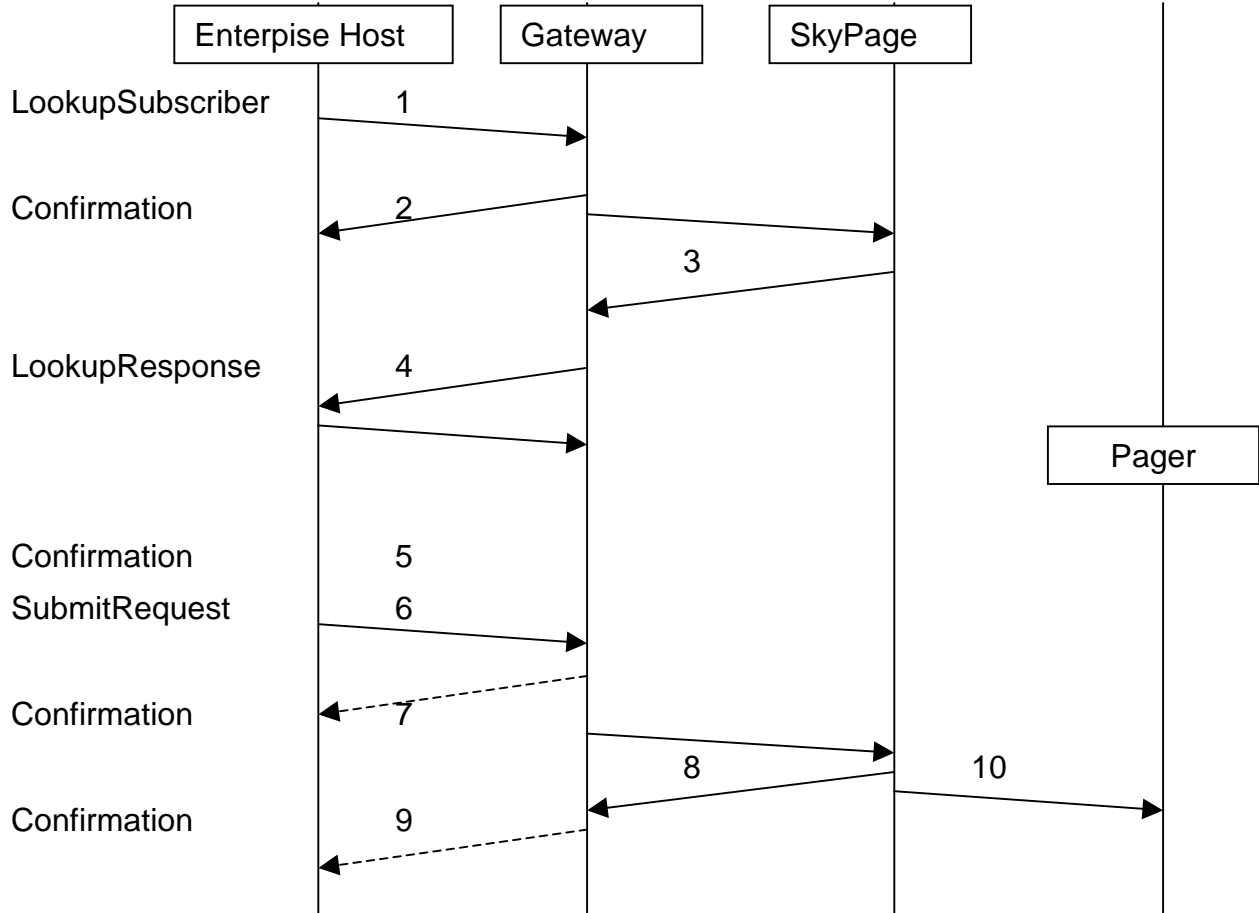
```
</wctp-MessageReceived>  
</wctp-PollForMessages>  
</wctp-Operation>
```

9. Gateway replies to poll indicating there are no messages queued:

```
<?xml version="1.0"?>  
<!DOCTYPE wctp-Operation  
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">  
<wctp-Operation wctpVersion="1.0">  
    <wctp-PollResponse  
        minNextPollInterval="60"  
    >  
        <wctp-NoMessages/>  
    </wctp-PollResponse>  
</wctp-Operation>
```

F.21 Use Case #21: Lookup & Submit from an Enterprise host

Consider the case of a non-polling enterprise host:



1. Enterprise host requests a subscriber lookup:

```

<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-LookupSubscriber>
    <wctp-Originator
      senderID="autoHelp@myenterprise.com"
    />
    <wctp-LookupMessageControl
      messageID="46264399"
      transactionID="19990331.PN.1234567.001"
    />
    <wctp-Recipient
      recipientID="newUser@skypage.com"
    />
  </wctp-LookupSubscriber>
</wctp-Operation>

```

2. Gateway returns confirmation of receipt of the lookup request:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      Thank you for using SkyPage
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>
```

3. Gateway determines the capabilities of the subscribers messaging device. This may be done by querying various SkyPage equipment.

4. Gateway returns the requested capabilities for subscriber:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-LookupResponse
    responseToMessageID="46264399"
    transactionID="19990331.PN.1234567.001"
  >
    <wctp-Originator
      senderID="newUser@skypage.com"
    />
    <wctp-Recipient
      recipientID="autoHelp@myenterprise.com"
    />
    <wctp-LookupData
      maxMessageLength="300"
      mcrSupported="true"
      canRespond="true"
    />
  </wctp-LookupResponse>
</wctp-Operation>
```

5. Enterprise host confirms the receipt:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      Thank you for using MyEnterprise
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>
```

6. Now the Enterprise host knows it can compose a message with MCR(s) up to 300 characters. Suppose the Enterprise host wants to send context sensitive help for example. If canRespond above was not true, then the responding and address book related choices below would have been left out (if unable to compose a response then the device has no need for an address book). If mcrSupported had been false, then no MCRs would have been included and some brief help text might have been substituted along with a phone number or web page where more help could be found. The Enterprise host then submits the customized message for the subscriber:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
  SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-SubmitRequest>
    <wctp-SubmitHeader
      submitTimestamp="1999-03-31T18:18:00"
    >
      <wctp-Originator
        senderID="autoHelp@myenterprise.com"
      />
      <wctp-MessageControl
        messageID="46264399"
        transactionID="19990331.PN.1234567.001"
        allowResponse="true"
        preformatted="true"
      />
      <wctp-Recipient
        recipientID="newUser@skypage.com"
      />
    </wctp-SubmitHeader>
    <wctp-Payload>
      <wctp-MCR>
        <wctp-MessageText>
          Su: Auto Help Info

          H H I |
          HHH I |
          H H I .

        </wctp-MessageText>
        <wctp-Choice>
          Help replying
        </wctp-Choice>
        <wctp-Choice>
          Help composing
        </wctp-Choice>
        <wctp-Choice>
          Help address book
        </wctp-Choice>
        <wctp-Choice>
          More help topics
        </wctp-Choice>
      </wctp-MCR>
    </wctp-Payload>
  </wctp-SubmitRequest>
</wctp-Operation>
```

7. Gateway could optionally return confirmation of receipt of the submit request if it is unable get confirmation from SkyPage in a timely manner. Such a confirmation would look like:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      Thank you for using SkyPage
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>
```

8. SkyPage accepts request, queues message for delivery.
9. If the submit was refused by Skypage then the Gateway would return a Failure. Otherwise, if the Gateway had not already done so (step 6 above), it would return a Success as shown below:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
    SYSTEM "http://www.pcia.com/wireres/protocol/dtd/wctpv1-0.dtd">
<wctp-Operation wctpVersion="1.0">
  <wctp-Confirmation>
    <wctp-Success
      successCode="200"
      successText="Message accepted"
    >
      Thank you for using SkyPage
    </wctp-Success>
  </wctp-Confirmation>
</wctp-Operation>
```

10. SkyPage delivers the message to the subscriber's messaging device respecting the preformatted attribute by not modifying or adding to the content of the page in any way (timestamps included). At some future time the subscriber may respond via MCR.