Client-Server Protocol Transport Bindings

Approved Version 1.3 – 23 Jan 2007

Open Mobile Alliance

OMA-TS-IMPS_CSP_Transport-V1_3-20070123-A

Contents

Figures

Tables

1. Scope

The Instant Messaging and Presence Service (IMPS) includes four primary features:

- Presence
- Instant Messaging
- Groups
- Shared Content

Presence is the key enabling technology for IMPS. It includes client device availability (my phone is on/off, in a call), user status (available, unavailable, in a meeting), location, client device capabilities (voice, text, GPRS, multimedia) and searchable personal statuses such as mood (happy, angry) and hobbies (football, fishing, computing, dancing). Since presence information is personal, it is only made available according to the user's wishes - access control features put the control of the user presence information in the users' hands.

Instant Messaging (IM) is a familiar concept in both the mobile and desktop worlds. Desktop IM clients, two-way SMS and two-way paging are all forms of Instant Messaging. Wireless Village IM will enable interoperable mobile IM in concert with other innovative features to provide an enhanced user experience.

Groups or chat are a fun and familiar concept on the Internet. Both operators and end-users are able to create and manage groups. Users can invite their friends and family to chat in group discussions. Operators can build common interest groups where end-users can meet each other online.

Shared Content allows users and operators to setup their own storage area where they can post pictures, music and other multimedia content while enabling the sharing with other individuals and groups in an IM or chat session.

These features, taken in part or as a whole, provide the basis for innovative new services that build upon a common interoperable framework.

2. References

2.1 Normative References

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL: http://www.openmobilealliance.org

[JSR120] "Wireless Messaging API for Java 2 Micro Edition". Java Community Process. August 2002. URL: http://jcp.org/aboutJava/communityprocess/final/jsr120/index.html

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

[RFC2616] "Hypertext Transfer Protocol – HTTP/1.1". Fielding R.; Gettys J.; Mogul J.; Frystyk H.; Masinter L.; Leach P.; Berners-Lee T., June 1999. URL: http://www.ietf.org/rfc/rfc2616.txt

[TCPTunnel] "Tunneling TCP based protocols through Web proxy servers", Luotonen, A., URL: http://www.web-cache.com/Writings/Internet-Drafts/draft-luotonen-web-proxy-tunneling-01.txt

[TIAEIA-637] "ANSI/TIA/EIA-637-B: Short Message Service for Wideband Spread Spectrum Systems", 2002. URL: http://global.ihs.com/search_res.cfm?RID=TIA&INPUT_DOC_NUMBER=TIA%2FEIA%2D637

[TS 23.038] "Alphabets and Language-specific Information (Release 5), 3GPP TS 23.038 v5.0.0", 3rd Generation Partnership Project, March 2002. URL: ftp://ftp.3gpp.org/Specs/archive/23_series/23.038/23038-500.zip

[TS 23.040] "Technical Realization of the Short Message Service (Release 5), 3GPP TS 23.040 v5.4.0", 3rd Generation Partnership Project, June 2002. URL: ftp://ftp.3gpp.org/Specs/archive/23_series/23.040/23040-540.zip

[TS 24.011] "Point-to-Point (PP) Short Message Service (SMS) Support on Mobile Radio Interface (Release 5), 3GPP TS 24.011 v5.1.0", 3rd Generation Partnership Project, December 2002. URL: ftp://ftp.3gpp.org/Specs/archive/24_series/24.011/24011-510.zip

[WAPWDP] "Wireless Datagram Protocol, Version 14-Jun-2001", Open Mobile Alliance™, WAP-259-WDP, URL: http://www.openmobilealliance.org

[WAPWSP] "Wireless Session Protocol Specification, Version 05-July-2001", Open Mobile Alliance™, WAP-230-WSP, URL: http://www.openmobilealliance.org

2.2 Informative References

[Arch] "IMPS Architecture", Version 1.3, Open Mobile Alliance™, OMA-AD-IMPS-V1_3, URL: http://www.openmobilealliance.org

[CSP] "Client-Server Protocol Session and Transactions", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_CSP-V1_3, URL: http://www.openmobilealliance.org

[CSP DataType] "Client-Server Protocol Data Types", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_CSP_Data_Types-V1_3, URL: http://www.openmobilealliance.org

[CSP PTS] "Client-Server Protocol Plain Text Syntax", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_CSP_PTS-V1_3, URL: http://www.openmobilealliance.org

[CSP Trans] "Client-Server Protocol Transport Bindings", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_CSP_Transport-V1_3, URL: http://www.openmobilealliance.org

[CSP WBXML] "Client-Server Protocol Binary XML Definition and Examples", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_CSP_WBXML-V1_3, URL: http://www.openmobilealliance.org

[CSP XMLS] "Client-Server Protocol XML Syntax", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_CSP_XMLS, URL: http://www.openmobilealliance.org

[PA] "Presence Attributes", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_PA-V1_3, URL: http://www.openmobilealliance.org

[PA XMLS] "Presence Attribute XML Syntax", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_PA_XMLS-V1_3, URL: http://www.openmobilealliance.org

[SSP] "Server-Server Protocol Semantics", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_SSP-V1_3, URL: http://www.openmobilealliance.org

[SSP Syntax] "Server-Server Protocol XML Syntax", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_SSP_XMLS-V1_3, URL: http://www.openmobilealliance.org

[SSP Trans] "Server-Server Protocol Transport Binding", Version 1.3, Open Mobile Alliance™, OMA-TS-IMPS_SSP_Transport-V1_3, URL: http://www.openmobilealliance.org

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

None.

3.3 Abbreviations

CIR	Communications Initiation Request
CSP	Client-Server Protocol
HTTP	Hypertext Transfer Protocol
NAT	Network Address Translation
OMA	Open Mobile Alliance
PTS	Plain Text Syntax
SMS	Short Message Service
WAP	Wireless Application Protocol
WSP	Wireless Session Protocol
WV	Wireless Village

4. Introduction

This document describes the binding of the session and transactions to different transports. This document describes four bindings: WSP [WAPWSP], HTTP [RFC2616], HTTPS and SMS [TS 23.040], [TIAEIA-637]. In addition, it defines the use of a Communications Initiation Request (CIR) used to initiate communication process between server and clients.

A IMPS client and server MUST support at least one transport binding, either WSP or HTTP or HTTPS or SMS. The server support for WSP can be implemented by using a WAP Gateway in front of the server and then use HTTP communication between the WAP Gateway and the IMPS server. The CIR channel is mandatory for all data channel transport bindings except the SMS binding.

5. Logical Model of Communications

Logically the IMPS transport binding is divided into two channels: a mandatory *data channel* in which all the exchange of CSP primitives is done and a conditional *CIR channel* used to activate the data channel whenever the data channel is not established, or the communication is halted in the data channel and needs to be reactivated. Both channels are depicted on Figure 1.

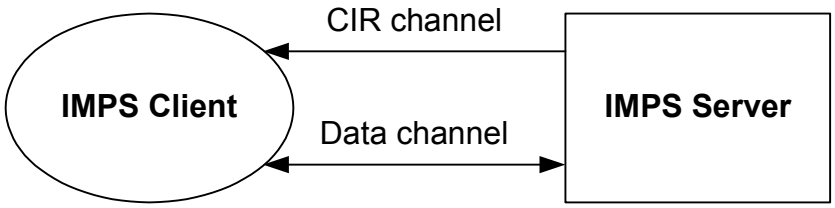


Figure 1: Logical Model of Communications

The need and use of a CIR channel depends on the protocol and bearer used in data channel. The protocol bindings in data channel are WSP, HTTP, HTTPS and SMS. In case of WSP, HTTP and HTTPS, the communication is asymmetric, i.e., it always originates from IMPS client to the server. Thus, the client can always start a transaction from the client to the server. If the IMPS server needs to start a transaction, there are two alternatives:

- The server inserts the transaction request into a response message for a pending transaction from the client to the server
- The server sends a communication initiation request message through the CIR channel to the client in order to request an immediate CSP PollingRequest message from the client to the server on the data channel. The transaction request is then inserted into the response part of the poll request.

In addition to the use described above, the CIR channel is also used to establish the data channel when the channel is not available. For instance, if a TCP/IP connection for the data channel has been disconnected, or the PDP context in 2.5G or 3G mobile networks is not allocated, the CIR channel is used to reestablish the channel connection to the server.

In the SMS technology, both the client and server can originate transactions and the data channel is always available. Thus, a separate CIR channel is not needed.

6. IMPS Session and Channel Management

The IMPS session and transaction models are independent of the IMPS transport binding and the underlying bearer protocols. The IMPS session does not require persistent underlying bearer for the data channel. The TCP/IP connection or WSP session MAY be disconnected during the session for performance reasons or it MAY be lost for some other reason. If disconnected, the client reestablishes the connection when it needs to send a request or when it receives the CIR.

Servers MAY support the usage of different bearers for the data channel within a single session due to the independence of the session and transaction models from the IMPS transport binding. Clients MUST however not depend on this behaviour since it MAY NOT be supported by the server or disabled by the operator due to security reasons. Supporting multiple bearers is beneficial for mobile clients that MAY loose connectivity over a preferred bearer while maintaining connectivity over alternative bearers.

The CIR channel is either connectionless, or connection-oriented, or based on polling. If the channel is connection-oriented, the connection needs to be persistent (for the duration of a session). However, if the server discovers that the CIR channel over standalone TCP/IP is disconnected, the server MAY notify the client that the CIR channel is disconnected by setting the CIR flag 'F' in an http POST response (see 7.2 for the details of WSP/HTTP encapsulation of CSP transactions).

When the TCP/IP-based CIR channel is disconnected, and

- there was at least one connectionless CIR channel agreed during client capability negotiation, the client MAY reconnect the TCP/IP-based CIR channel at any suitable time.
- there was not any one connectionless CIR channel agreed during client capability negotiation, the client MUST reconnect the TCP/IP-based CIR channel when it has received the CIR flag 'F' in an HTTP POST response.

The relation of the channel connections and the IMPS session is illustrated in Figure 2.

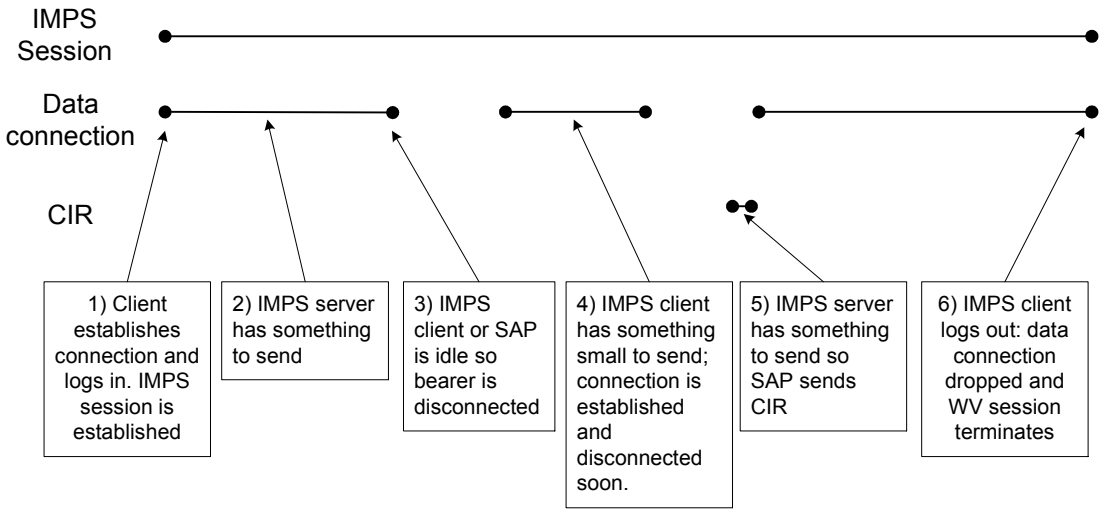


Figure 2: The relation of the IMPS session and bearer.

Clients MAY request re-establishment of a disconnected session. The reason code of the disconnection is not significant – the client MAY request re-establishment of sessions that have been terminated successfully or with some error. The feature is also suitable to switch between different transports as well as switching between clients (transferring the Session-ID from one client to another is not in the scope of the specifications). When the server detects that the ClientID is not the same as it was in the disconnected session, the server MUST force client capability negotiation in the LoginResponse primitive – the server MUST know the supported content types, and CIR channel settings.

In order to re-establish a session the client uses the Login transaction, however it needs to supply - in addition to the normal login credentials - the Session-ID from the session it wishes to restore. The server MUST NOT accept the request

if the User-ID, the password and the Session-ID do not match, and it MUST respond with error code 604. When the server is unable to provide exactly the same services that have been negotiated for the disconnected session, the server MUST reject the session re-establishment request with error code 605.

A server MAY choose not to support session re-establishment – it is an OPTIONAL feature. A server MUST reply with error code 501 when it does not support session re-establishment and it receives a session re-establishment request from the client.

When the server supports session re-establishment, it MUST maintain the properties of every disconnected session for a limited period of time. The amount of time is up to the server implementations, however it is RECOMMENDED to be at least five times the last agreed Keep-Alive-Time, and it is NOT RECOMMENDED to be more than 24 hours.

A successfully restored session carries over all credentials from the disconnected session, see *session context* in [CSP].

Note that the re-joining of group does not take place – except of course those groups that have the AutoJoin flag set to 'T' – as the server removes the user from any joined groups before the logout/disconnection takes place.

After a successful session re-establishment it is the client's responsibility to make sure that it has the up-to-date contact lists and any other data that it caches.

7. Transport Binding for WSP/HTTP/HTTPS Data Channels

7.1 Overview

The CSP transport binding alternatives for data channel are HTTP 1.1, HTTPS, WSP 1.2 or WSP 2.0. In HTTP and HTTPS binding, the bearer protocol in data channel is TCP/IP. For WSP bindings, the bearer protocol alternatives are described in WAP specifications. There is no requirement for persistent bearer connection.

The bearer connection for the data channel is always set up from the IMPS client to the IMPS SAP.

7.2 WSP/HTTP Encapsulation of CSP Transactions

The WSP and HTTP(S) are both asymmetric, client-server protocols, in which requests always originate from the client and responses from the server. In IMPS transactions, however, there is a need for symmetric transactions: the requests MAY originate from client or server.

The encapsulation of symmetric CSP transactions to asymmetric WSP/HTTP(S) methods is based on the use of WSP/HTTP(S) POST method only. The WSP/HTTP(S) POST-request, POST-response and the CSP transactions are completely separated. Each WSP/HTTP(S) POST-request MAY contain at least one CSP transaction request or CSP transaction response message. Similarly, each WSP/HTTP(S) POST-reply MAY contain at least one CSP transaction request or CSP transaction response message.

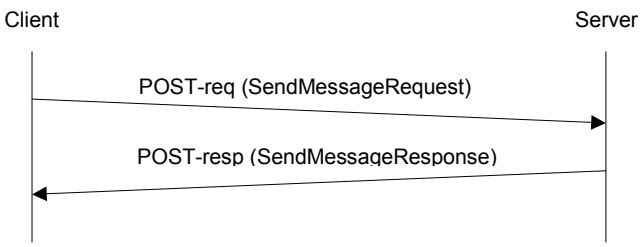
If the IMPS client supports more than one CSP request or response within single HTTP(S) POST request or response, it is indicated during the client capabilities negotiation procedure.

For each transaction at POST-reply, the IMPS server indicates whether (for server reasons) the next POST request is needed. If it is needed, but the IMPS client has nothing to send, the client SHALL WSP/HTTP(S) POST request with CSP PollingRequest primitive as content. Similarly, if server has no IMPS transaction request or reply to be sent to the IMPS client, the WSP/HTTP(S) POST response SHOULD contain no content and the 200 OK response code.

This communication continues until neither the IMPS client nor the server has CSP primitives to send. In such a case, the communication grinds to a halt. If, at this point, the IMPS client has something to send, it simply issues a WSP/HTTP(S) POST with the CSP transaction request.

If server needs to send any data (CSP request or response) to the particular client, first the server has to send a Communications Initiation Request to that client which is a signal to the IMPS client to initiate WSP/HTTP(S) POST with CSP PollRequest primitive as content.

Examples of the mapping of the message flow for the CSP SendMessageRequest and MessageNotification transactions are depicted on Figure 3.



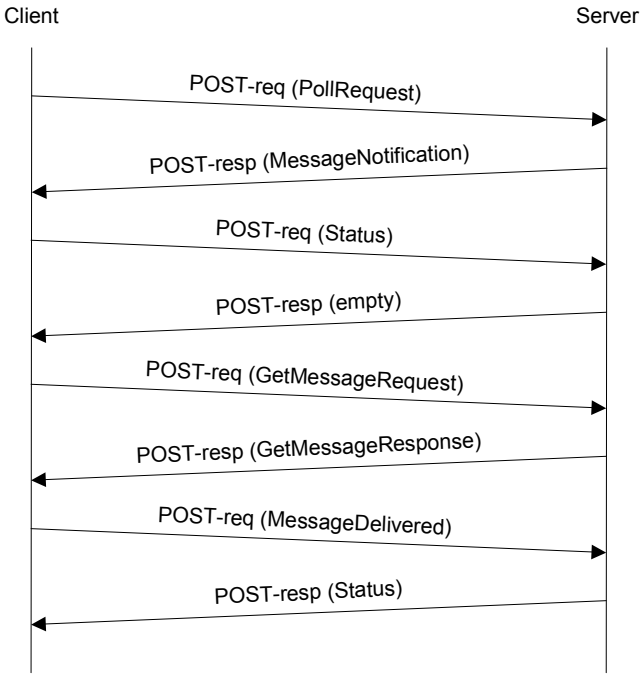


Figure 3: Examples of mapping of IMPS CSP transactions to WSP/HTTP(S)

7.3 Access Point Definition

For WAP/WSP bindings, the access point towards IMPS SAP requires the WAP access point and the URL of the IMPS SAP.

For HTTP(S) binding, the access point towards IMPS SAP requires the general ISP access point for TCP/IP and the URL of the IMPS SAP.

The URL used in WAP/WSP and HTTP(S) bindings MAY contain a non-empty path and a non-standard port number.

7.4 Redirection

The IMPS client that uses HTTP binding MUST understand standard HTTP redirection codes [RFC2616] and associated information headers. HTTP redirection mechanism allows a IMPS server to redirect clients to other servers or other parts of the same server based on load or session related information.

HTTP Redirect [RFC2616] indicates that such redirect codes as 301 (Permanent), 302 (Found) and 307 (Temporary) MUST NOT automatically redirect without being confirmed by the user. It is recognized that such interactions in the UI will give rise to a poor user experience in a real product. HTTP Redirect in IMPS HTTP Binding SHOULD be supported in the following ways:

- The client MAY automatically perform the redirect action without further user confirmation.
- Only two types of redirect are allowed in the client server communication:
  - Permanent Redirection (301): The server address is redirected forever. The client MAY save the redirected server address to flash and reuse it in the future transactions and sessions if the client supports persistent storage.
  - Temporary Session Redirection (307): The server address is redirected for this session. The client uses the redirected server address only in the context of current IMPS session. The purpose of this code would be so that the server MAY change the redirect address for different sessions, however, the client is not being redirected on every transaction. The client MAY save the redirected server address to flash and reuse it in the future transactions during this session if the client supports persistent storage.
- The caching headers, which are specified in the HTTP [RFC2616] when redirecting, SHALL be ignored by the client. The server SHOULD not enclose those caching headers when redirecting.

7.5 HTTP Headers

All headers MUST conform to [RFC2616].

7.6 Error Handling

Support for the standard HTTP error responses is mandatory for both client and server implementations.

When a client or server implementation receives an error response in a POST-reply, it SHALL:

- assume that all of the transaction requests in the corresponding POST-request are invalid; and
- make no assumptions about the HTTP headers, content type and content.

When a client or server implementation generates an HTTP error response, it MAY set the HTTP headers, content type and content to any appropriate value.

A client or server implementation that detects an error in the received XML data in a POST-request SHALL reply with an HTTP error response with code 400, BAD_REQUEST.

8. Transport Binding for CIR Channel

8.1 Transport Alternatives and Message Format

The CIR channel is a push-type channel that can be implemented as connectionless or connection-oriented channel. The purpose of the CIR channel is to carry communication initiation requests from the server to the client only. It does not carry any CSP primitives.

For the CIR channel, the following bindings are defined:

- WAP 1.2 or WAP 2.0 push using WSP unit push message and SMS as a bearer
- WAP 1.2 or WAP 2.0 push using WSP unit push message and UDP/IP as a bearer
- Standalone SMS binding
- Standalone UDP/IP binding
- Standalone TCP/IP binding
- Standalone HTTP binding

The IMPS client MAY support one or more CIR channel bindings that are indicated in the client capability negotiation defined in CSP.

If CIR channel is connection-oriented, the connection of CIR channel and data channel are independent of each other.

In general, the bindings of CIR channel and data channel are independent. However, if the binding for data channel is WSP, the CIR channel binding MUST use WAP 1.2/2.0 CIR bindings.

The communication initiation request message is textual message in the following format:

- WVCI <CSP-version> <Session-cookie>
- Where:
  - CSP-version MUST be the version number of the IMPS specification that the client is currently using. Major version and minor version numbers MUST be separated by the dot (“.”).
  - *Session-cookie* MUST be the client-defined session cookie generated at every client login.

The default character-encoding scheme MUST be UTF-8, unless it is specified otherwise in the specific binding section(s). With UTF-8 the minimum supported set of characters MUST be ISO-8859-1 (Latin 1).

The HTTP binding CIR Channel does not use a textual CIR message. Instead, the HTTP reply code is used to trigger the communication on the data channel.

8.1.1 WAP Push Binding

To be able initiate a WAP Push request, the IMPS server MUST be provisioned with an address of WAP PPG and support WAP Push Access Protocol. The IMPS server uses the push submission operation to send CIR to the terminal. Each push message SHOULD contain one CIR. Content type of the content entity of a PAP request is “application/vnd.wv.csp.cir”.

The use of WAP Push does not require that the IMPS client has active PDP context. The Push Proxy Gateway MAY use a SMS bearer to send the initiation request or, if a PDP context is already active and the IP address is known, it MAY push the message over TCP or UDP.

The IMPS client in a mobile handset MAY provide its mobile number in the CSP protocol login transaction (as a part of Client ID). If the mobile number is not present, the IMPS SAP MUST be able to obtain the mobile number if it is required.

For offline notifications (see chapter 9 Transport Binding for Offline Notifications), the MSISDN MUST be available between sessions. Therefore the MSISDN, received via the login transaction MUST be either cached between sessions or the IMPS SAP MUST be able to obtain the mobile number if it is required.

8.1.2 Standalone UDP/IP Binding

In the case of a standalone UDP/IP binding, the IMPS server sends the client the CIR messages enclosed in UDP datagrams. Each UDP datagram contains exactly one CIR message. To use this binding, IMPS client MUST be able to receive UDP datagrams directly from the IMPS server.

The IMPS client MAY accept the CIR request either to default UDP port defined in this document or to provide the UDP port in the capability negotiation phase of client login.

Due to the small size of the CIR message, it is guaranteed that the UDP will not be fragmented or rejected because of size.

After a successful complete login procedure that includes client capability and service negotiation, the IMPS client MUST send an UDP/IP packet to the server comprising the message “HELO” with the Session-ID as a parameter, as defined in section 8.1.3. The packet carrying the "HELO" message allows the server to discover the client's IP address and to overcome possible NATs. The IMPS server replies to the client’s “HELO” message with an “OK” message. The HELO message MUST be sent to the IP and port set by the server in the ClientCapabilityResponse message.

In some cases the public address of the client may be changed (i.e., by a NAT application) and by that not allow the server to communicate with the client. To prevent this from happening, the IMPS client SHOULD periodically send “PING” messages over UDP/IP to maintain its address. The delay between such PING messages is network dependant and thus outside the scope of this document. The server must respond to these messages with the “OK” message. If the client doesn’t receive an “OK” message, it MUST send the “HELO” message again.

Since the IP address assigned to the client may also change due to network considerations (e.g., by GGSN or PDSN), the PING message contains also the session ID, similar to the HELO message. In this case, the server will be able to identify the PING, even if the IP has changed and update its routing table accordingly.

An example of data traffic on the UDP/IP-based CIR channel (“C→S” indicates client originated messages, “S→C” indicates server originated messages) is:

C→S: HELO abcd123

S→C: OK

S→C: WVCI 1.3 cookie123

C→S: PING abcd123

S→C: OK

The character-encoding scheme MUST be UTF-8. The minimum supported set of characters MUST be ISO-8859-1 (Latin 1).

Due to the behavior of some NAT servers, it is RECOMMENDED that whenever the server sends a message to the client, it uses the same IP and port used by the client as destination address. This SHOULD be the same address as appeared in the ClientCapabilityResponse.

8.1.3 Standalone TCP/IP Binding

TCP/IP binding uses a persistent connection from the IMPS client to server to provide a low-latency always-on CIR channel.

The IMPS client is responsible for setting up the TCP/IP connection and maintaining its persistency.

The IMPS client opens the CIR TCP/IP connection to the server right after a successful login procedure including client capability and service negotiation. The IP address and port for the CIR channel are provided by the server in the capability negotiation. The IP address and port are valid throughout the session.

As soon as a connection opens, the client MUST send the authentication message “HELO” with Session ID as a parameter. This allows the IMPS server to associate the new TCP/IP connection with one of the existing sessions. If the IMPS server does not receive a “HELO” message in 10 seconds after a new connection has been opened from the client or the received Session ID is unknown, the server MUST terminate the connection. The IMPS server replies to the client’s “HELO” message with an “OK” message. The client is not allowed to open more than one connection to the IMPS server.

In some cases a TCP/IP connection MAY be closed by the intermediate network entities, or a connection MAY be broken due to network problems. To prevent this from happening or to be able to recover, the IMPS client SHOULD periodically send “PING” messages over an opened connection to determine if it is still available. The server MUST respond to these messages with the “OK” message. If client doesn’t receive an “OK” message or detects that the connection is broken, it MUST open a new TCP/IP connection and send the “HELO” message again.

When a server has any data (CSP request or response) that needs to be sent to the client, it sends a CIR message over the TCP/IP connection associated with this client.

All client and server originated messages MUST be terminated with a <CR><LF> (carriage return, line feed) sequence.

The character-encoding scheme MUST be UTF-8. The minimum supported set of characters MUST beis ISO-8859-1 (Latin 1).

The connection establishment for a standalone TCP/IP binding for CIR channel MAY not work directly when the IMPS client is behind a firewall or proxy. The technology alternatives to facilitate the connection initiation and management are:

HTTP Tunnelling [TCPTunnel]

SOCK4

SOCK5

An example of data traffic on the TCP/IP-based CIR channel (“C→S” indicates client originated messages, “S→C” indicates server originated messages) is:

<client opened TCP/IP connection to the server>

C→S: HELO abcd123

S→C: OK

S→C: WVCI 1.3 cookie123

C→S: PING

S→C: OK

<client closes TCP/IP connection>

8.1.4 Standalone SMS Binding for CIR

The standalone SMS binding for the CIR channel uses either GSM short message or CDMA IS-637 short message technology to facilitate the CIR channel. The IMPS client and the short message service center MUST support both mobile-originated (MO) and mobile-terminated (MT) short messages.

The standalone SMS binding for the CIR channel supports both GSM SMS [TS 23.040] and CDMA SMS IS-637 [TIAEIA-637]. Each SMS message SHALL contain exactly one CIR message. The encoding of SMS messages for CIR SHALL be the GSM 7-bit default alphabet defined in [TS 23.038].

After a successful complete login procedure that includes client capability and service negotiation, the IMPS client MUST send an SMS to the server comprising the message “HELO” with the Session-ID as a parameter, as defined in section 8.1.3. The SMS carrying the "HELO" message allows the server to discover the client's mobile number. The IMPS client SHALL NOT periodically send a “PING” message over the CIR channel. The encoding of the “HELO” message SHALL be the GSM 7-bit default alphabet [TS 23.038].

For offline notifications (see chapter 9 Transport Binding for Offline Notifications), the MSISDN MUST be available between sessions. Therefore the MSISDN received via the “HELO” message must be cached between sessions.

The IMPS client SHALL accept the CIR request through the standalone SMS CIR port defined in section 10.5. However, the CDMA IS-637 SMS does not include a port number or any other field for differentiation between recipient applications. For this purpose, the WAP WDP for IS-637 SMS, which is defined in section 6.5 of WDP specification [WAPWDP], MUST be used.

In order for the SMS-based CIR channel to work properly through a J2ME platform, the guidelines defined in J2ME Wireless Messaging API [JSR120] MUST be followed.

8.1.5 Standalone HTTP Binding

The HTTP binding is used by clients that cannot establish any other CIR channel. The client periodically polls on the CIR channel for a CIR trigger. When a CIR trigger is received, the client performs a CSP PollingRequest transaction to enable the server-initiated transaction in the same way as for the other CIR channels.

Polling is very resource consuming when it comes to bandwidth and server load. To minimize this overhead, the periodic polling is only done for CIR with a minimal HTTP GET on a non-persistent HTTP connection.

The URL to be used for the CIR poll is provided by the server in the capability negotiation. The format of the address is such as the server can identify the session but for security reasons the actual Session ID SHOULD never be revealed in the HTTP binding. Example of poll URL:

```
MyServiceProvider.com/poll?pc=1234567
```

The 'pc=123456' is a poll cookie that the server generates and internally uses to map to the real session.

The URL is valid throughout the session and the client closes the HTTP connection after each poll. The HTTP binding always uses HTTP even if the data channel uses HTTPS.

The IMPS client is responsible for setting up the HTTP connection and to do the polling. The minimum time between two polls is given by the server during client capability negotiation. The IMPS client starts to poll after successfully logs in and has completed appropriate client capability and service negotiation. .

It is RECOMMENDED that clients implement an adaptive polling policy. Normally, when polling requests return a CIR trigger to do a CSP PollingRequest Transaction, the client needs to initiate the next polling request after the minimum interval. However, in the case of empty responses, it SHOULD gradually increase the polling intervals (up to 10 seconds or more). This significantly decreases the server load and reduces unnecessary network traffic.

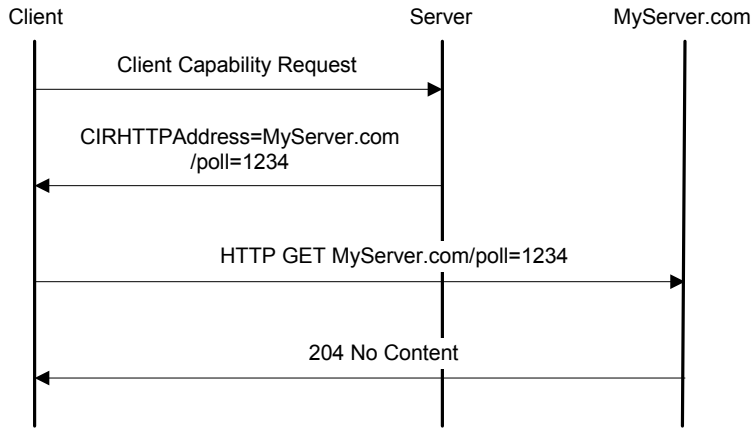


Figure 4: HTTP binding for CIR channel – the server does not have any queued requests or responses.

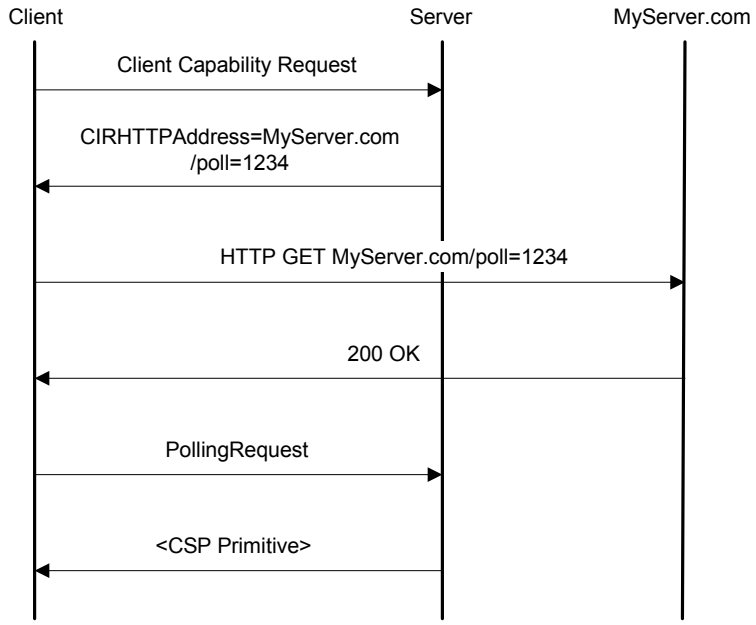


Figure 5: HTTP binding for CIR channel – the server has some queued requests or responses.

The Client uses the HTTP GET method to poll on the HTTP binding.

If CSP messages are queued on the server, the server replies with a '200 OK'. The reply substitutes for the textual CIR message that is used in other CIR channel bindings. The client then needs to perform a CSP PollRequest Transaction to retrieve the server initiated transaction.

If there are no queued CSP messages on the server, the server replies with '204 No Content'.

9. Transport Binding for Offline Notifications

9.1 Transport Alternatives and Message Format

Offline notifications are used to inform a user of events when being offline (not in IMPS session). The CIR channel MUST be used to transport these notifications (see chapter 8 Transport Binding for CIR Channel).

Not all bindings from CIR can be used for offline notifications due to the lack of addressing information.

Following bindings defined in the CIR channel MAY be used for offline notifications:

- WAP 1.2 or WAP 2.0 push using WSP unit push message and SMS as a bearer
- WAP 1.2 or WAP 2.0 push using WSP unit push message and UDP/IP as a bearer
- Standalone SMS binding

The CIR channel binding that the server MUST use for offline notifications MUST be negotiated during client capability negotiation using the SupportedOfflineBearer setting. The CIR channel that is used for offline notifications is independent from the negotiation of CIR channel binding for Communication Initiation Request. This means that the CIR channel binding for offline notification MAY be different than the CIR channel binding for Communication Initiation Request, however when the CIR channel bindings are the same the client is still REQUIRED to include SupportedOfflineBearer in the negotiation.

The offline notification is based on the textual message in the following format:

```
WVON <CSP-version> <Offline-Notification-Type> <Client-ID> [Offline-Notification-Content]
```

Where:

```
CSP-version is the version number of the IMPS specification that the client used before the
session was disconnected. Major version and minor version numbers MUST be separated by the dot
(".").
```

- Client-ID MUST be the unique identifier of the client to whom the session belonged to before the client logged out/was disconnected – since multiple clients MAY reside in the same device and the MSISDN agreed as offline message bearer might have been reused by another client in the same device.
- Offline-Notification-Type indicates the type of notification. See Table 1: Offline notification types.
- Offline-Notification-Content is a free text message to be shown to the end-user. The text MUST NOT contain sensitive information (such as names, User-ID, password, E.164 number, etc), and it MUST NOT be longer than 200 characters.

The total length of an offline message notification might exceed the length of a single short message in some cases, thus the server MUST send such short message using concatenated short messages (SM-TP level concatenation).

Examples:

```
WVON 1.3 OMF http://1.2.3.4:80/IMPSAPP Our valued customer, Your IMPS offline storage is full -
255 new messages. All new messages arriving to you from now on will be rejected.

WVON 1.3 OMR http://1.2.3.4:80/IMPSAPP Our valued customer, Your IMPS client is offline, while You
do not have an IMPS offline storage. We recommend keeping your client online to avoid undesired
message loss.

WVON 1.3 OMS http://1.2.3.4:80/IMPSAPP Our valued customer, you have 25 offline messages waiting
for pickup. You are using 10% of your offline storage capacity currently.
```

Offline-Notification-Type	Description
OMF	The offline message storage of the user is completely full, and the server will reject the new instant messages from now on until the user logs in and retrieves some messages.
OMR	The server does not provide offline storage for the user, thus a received instant message – and the server will reject the subsequent instant messages.
OMS	The server has successfully received and stored an instant message while the client was offline.

Table 1: Offline notification types

The default character-encoding scheme MUST be UTF-8, unless it is specified otherwise in the specific binding section(s). With UTF-8 the minimum supported set of characters MUST be ISO-8859-1 (Latin 1).

For detailed description about the possible bindings see chapter 8 Transport Binding for CIR Channel.

10.SMS Transport for Mobile Clients

Clients and servers MAY use SMS transport as a communication medium in order to support IMPS functionality. The SMS transport bindings use the GSM short message technology to facilitate the IMPS transactions. In the transport binding, the IMPS client and the short message service center MUST support both mobile-originating (MO) short messages as well as mobile-terminating (MT) short messages. Due to the symmetric nature of SMS transport, the CIR channel is not needed.

The SMS transport uses the Plain Text Syntax for message encoding. Please refer to [CSP PTS] for Plain Text Syntax. Due to the extra features and limitations of SMS transport, some transactions are not available while SMS transport is utilized. The availability of transactions are described in [CSP PTS].

10.1 Overview

In the SMS transport binding, the IMPS client communicates with the IMPS SAP through a SMSC. The CSP transactions and session document [CSP] as well as the relevant Plain Text Syntax document [CSP PTS] describe the SMS application level communication.

The SMSC MUST be able to route the messages from the IMPS client to the IMPS SAP. For this purpose, the short message is sent to a recipient that identifies the IMPS SAP as a special, IN-type number. The SMSC MUST have the capability to route messages using this special number to the IMPS SAP. When the IMPS SAP sends a message to the IMPS client, the SMSC is able to deliver the message directly when the recipient is identified with a mobile number. The architecture is depicted in Figure 6.

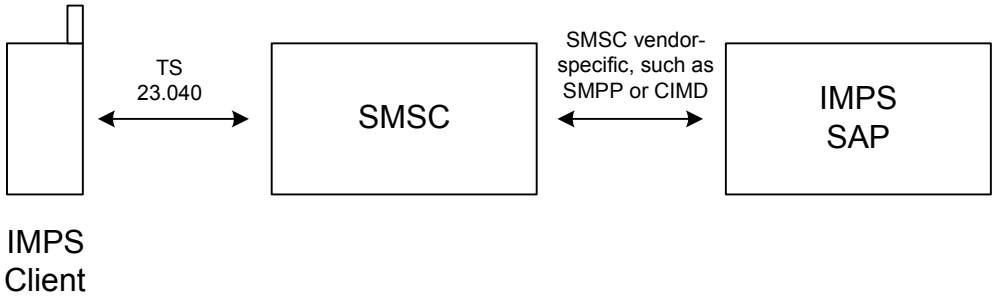


Figure 6: Architecture for SMS transport binding

The protocol between SMSC and IMPS SAP is one of the SMSC-vendor specific protocols, such as SMPP or CIMD.

The IMPS SAP is addressed by the recipient address in short message as encoded in TP-DA field in SMS-SUBMIT TPDU [TS 23.040] or as a Destination Address as described in [TIAEIA-637]. The SMSC MUST be able to recognize this number and route it towards the IMPS SAP. The receiving SMSC is addressed in the RP-DA field in RP-DATA RPDU [TS 24.011].

10.2 Access point to IMPS SAP

The access point definition towards IMPS SAP requires normal SMSC access point definition as well as the special IN-type number identifying the IMPS SAP.

10.3 Encoding of Short Messages

The encoding of SMS messages has two alternatives: 7-bit encoding using the SMS default character set [TS 23.038] and 8-bit encoding using UTF-8 character encoding scheme. With 8-bit encoding, the minimum supported set of characters is ISO-8859-1 (Latin 1).

Each IMPS short message MAY be either textual or contain a User Data Header (UDH) with the TP-UDHI value set to 1. In case of UDH, the detection of the IMPS-primitives is based on a registered IMPS application port identified in the headers. In case of textual short message, the detection of IMPS-primitives is based on detection of the message

preamble. The User Data Header contains a 16-bit application port number identifying the source port and destination port. The structure of the short message with UDH is defined as follows [TS 23.040]:

The destination port number is assigned in the CSP transport-binding document [CSP Trans]. When the session is started with a login primitive containing the UDH, the IMPS server and client must continue the session using the messages encoded with UDH throughout the session. Similarly, when the session is started with a login primitive without UDH, the IMPS server and client must continue the session without UDH.

- UDL Length of the message
- UDHL Length of User Data Header (7 for IMPS message)
- IEID Information Element Identifier (05_{hex} = 16-bit application port)
- IEDL Length of IED (4 = four octets for ports)
- IED Port numbers (octet 1,2 = destination port, octet 3,4=originator port)
- UD User Data (IMPS primitive)

The source and destination port have same port numbers, assigned in 10.4 Transaction over Multiple Short Messages.

Servers MAY support either SMS with UDH or textual SMS, but do not have to support both. Servers can indicate that they do not support the form of SMS received from the client using status code 550 (Header encoding not supported). See [CSP].

10.4 Transaction over Multiple Short Messages

A transaction over SMS bearer MAY be split to more than one short message when parameters exceed 160 characters. To accomplish this, there are two basic techniques available:

1. Use of concatenated short messages (SM-TP level concatenation).
2. Use concatenation identifier (***DD***) identifier in a short message text.

Alternative 1) is based on SMS technology, which is optionally supported in the SMSC and terminal product. It does not require support in text level.

Alternative 2) needs to be used when SMS concatenation is not available. The rules are:

- If the short message does not need text-level concatenation (single SMS or SM-TP level concatenation), the ***DD*** identifier is not present.
- If the short message needs text-level concatenation, the **DD** identifier is present (starting from 'a') and the first letter identifies the current SMS within concatenation sequence and the second letter identifies the last SMS within the sequence.
- While transferring the multiple short messages concatenated in a text level, the transaction identifier will be the same in all of the short messages.
- The request or response is incomplete until all short messages within the same transaction are received.
- The receiver MUST reassemble the IMPS message based on the MSISDN of the sender since the Session-ID is not available in all parts and MAY even be split up between two SMS messages.
- In case of UTF-8 encoding the split MAY occur on a character boundary OR within a multi-byte character. The receiver MUST NOT assume that a whole UTF-8 encoded character has been received at the end of each SMS message.

The text-level concatenation identifier does not require that the messages will be received in order. An example of concatenation could be the terminal sending a message:

```
WV13NM23ac MC="This is a very
WV13NM23cc very long textual content..."
```

```
WV13NM23bc very long message, and it has very
```

And the server responding:

```
WV13ST23 ST=200
```

A single short message MAY also contain messages from multiple transactions. In such cases, messages are separated by an ampersand (&) parameter according to the parameter syntax (e.g. space-ampersand-space). Note that the preamble MUST NOT be fragmented, thus each IMPS message preamble and an extra space character MUST fit as a whole into the SMS message.

11.Registered Identifiers

11.1 Content Type

The IMPS content types for textual and binary XML, SMS Bindings as well as for CIR content type are registered through IANA.

The following content types have been registered with IANA:

application/vnd.wv.csp+xml

application/vnd.wv.csp+wbxml

application/vnd.wv.csp.cir (for PAP push submission)

application/vnd.wv.csp.sms

Servers that provide communication means for clients/servers that are using earlier version(s) of the IMPS specification MUST apply the appropriate content type before sending.

11.2 WAP Push Application Id

The push application id is registered with the Open Mobile Alliance OMNA registry.

The following application id has been registered with OMNA:

0x0A - x-wap-application:wv.ua

Servers that provide communication means for clients/servers that are using earlier version(s) of the IMPS specification MUST apply the appropriate WAP Push Application ID before sending.

11.3 Port Number for Standalone UDP/IP CIR Channel

The following port number has been registered with IANA for standalone UDP/IP CIR Channel:

3717

11.4 Port Number for SMS Binding

The following port number has been registered with IANA for SMS binding:

3590

11.5 Port Number for Standalone SMS CIR Channel

The following port number has been registered with IANA for SMS CIR Channel:

3716

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-IMPS_CSP_Transport-V1_3	23 Jan 2007	Status changed to Approved by TP TP Doc ref# OMA-TP-2006-0453R02

Appendix B. Static Conformance Requirements (Normative)

The notation used in this section is specified in [IOPPROC].

B.1 Transport Requirements

B.1.1 Clients

Item	Function	Reference	Status	Requirement
IMPS-CSP-TRANSP-C-001	Support for transport binding for data channel		M	IMPS-CSP-TRANSP-C-003 OR IMPS-CSP-TRANSP-C-004 OR IMPS-CSP-TRANSP-C-005 OR IMPS-CSP-TRANSP-C-006 OR IMPS-CSP-TRANSP-C-007
IMPS-CSP-TRANSP-C-002	Support for transport binding for CIR channel.		O	IMPS-CSP-TRANSP-C-003 OR IMPS-CSP-TRANSP-C-004 OR IMPS-CSP-TRANSP-C-005 OR IMPS-CSP-TRANSP-C-006 OR IMPS-CSP-TRANSP-C-015 OR IMPS-CSP-TRANSP-C-016
IMPS-CSP-TRANSP-C-003	Support for HTTP binding in data channel		O	IMPS-CSP-TRANSP-C-013
IMPS-CSP-TRANSP-C-004	Support for HTTP/S binding in data channel		O	IMPS-CSP-TRANSP-C-013
IMPS-CSP-TRANSP-C-005	Support for WSP 1.2 binding in data channel		O	IMPS-CSP-TRANSP-C-013
IMPS-CSP-TRANSP-C-006	Support for WSP 2.0 binding in data channel		O	IMPS-CSP-TRANSP-C-013
IMPS-CSP-TRANSP-C-007	Support for SMS binding in data channel		O	IMPS-CSP-SMS-C-001 OR IMPS-CSP-SMS-C-003
IMPS-CSP-TRANSP-C-008	Support for WAP push SMS binding in CIR channel		O	IMPS-CSP-TRANSP-C-012 AND IMPS-CSP-TRANSP-C-014
IMPS-CSP-TRANSP-C-009	Support for WAP push UDP/IP binding in CIR channel		O	IMPS-CSP-TRANSP-C-012 AND IMPS-CSP-TRANSP-C-014
IMPS-CSP-TRANSP-C-010	Support for standalone UDP/IP binding in CIR channel		O	IMPS-CSP-TRANSP-C-014
IMPS-CSP-TRANSP-C-011	Support for standalone TCP/IP binding in CIR channel.		O	IMPS-CSP-TRANSP-C-014
IMPS-CSP-TRANSP-C-012	With WSP 1.2 or WSP 2.0 bindings for data channel, only WAP SMS binding or WAP UDP binding is used in CIR channel.		O	IMPS-CSP-TRANSP-C-008 OR IMPS-CSP-TRANSP-C-009
IMPS-CSP-TRANSP-C-013	Sending of Poll request when poll request is received in IMPS message inside the WSP/HTTP(S) POST response.		O	IMPS-CSP-TRANSP-C-003 OR IMPS-CSP-TRANSP-C-004 OR IMPS-CSP-TRANSP-C-005 OR IMPS-CSP-TRANSP-C-006
IMPS-CSP-TRANSP-C-014	Sending of PollingRequest when CIR is received		O	IMPS-CSP-TRANSP-C-008 OR IMPS-CSP-TRANSP-C-009 OR IMPS-CSP-TRANSP-C-010 OR IMPS-CSP-TRANSP-C-011 OR IMPS-CSP-TRANSP-C-015 OR IMPS-CSP-TRANSP-C-016
IMPS-CSP-TRANSP-C-015	Support standalone SMS binding for CIR channel		O	IMPS-CSP-TRANSP-C-014

Item	Function	Reference	Status	Requirement
IMPS-CSP-TRANSP-C-016	Support for Standalone HTTP binding in CIR channel.		O	IMPS-CSP-TRANSP-C-014
IMPS-CSP-TRANSP-C-017	Support for Offline notification.		O	(IMPS-CSP-TRANSP-C-007 OR IMPS-CSP-TRANSP-C-008 OR IMPS-CSP-TRANSP-C-015) AND ([CSP]:IMPS-CSP-SAP-C-014 OR [CSP]:IMPS-CSP-SAP-C-015)

B.1.2 Servers

Item	Function	Reference	Status	Requirement
IMPS-CSP-TRANSP-S-001	Support for transport binding for data channel		M	IMPS-CSP-TRANSP-S-003 OR IMPS-CSP-TRANSP-S-004 OR IMPS-CSP-TRANSP-S-005 OR IMPS-CSP-TRANSP-S-006 OR IMPS-CSP-TRANSP-S-007
IMPS-CSP-TRANSP-S-002	Support for transport binding for CIR channel.		O	IMPS-CSP-TRANSP-S-003 OR IMPS-CSP-TRANSP-S-004 OR IMPS-CSP-TRANSP-S-005 OR IMPS-CSP-TRANSP-S-006 OR IMPS-CSP-TRANSP-S-015 OR IMPS-CSP-TRANSP-S-016
IMPS-CSP-TRANSP-S-003	Support for HTTP binding in data channel		O	IMPS-CSP-TRANSP-S-014
IMPS-CSP-TRANSP-S-004	Support for HTTP/S binding in data channel		O	IMPS-CSP-TRANSP-S-014
IMPS-CSP-TRANSP-S-005	Support for WSP 1.2 binding in data channel		O	IMPS-CSP-TRANSP-S-014
IMPS-CSP-TRANSP-S-006	Support for WSP 2.0 binding in data channel		O	IMPS-CSP-TRANSP-S-014
IMPS-CSP-TRANSP-S-007	Support for SMS binding in data channel		O	IMPS-CSP-SMS-S-001 OR IMPS-CSP-SMS-S-003
IMPS-CSP-TRANSP-S-008	Support for WAP push SMS binding in CIR channel		O	IMPS-CSP-TRANSP-S-012
IMPS-CSP-TRANSP-S-009	Support for WAP push UDP/IP binding in CIR channel		O	IMPS-CSP-TRANSP-S-012
IMPS-CSP-TRANSP-S-010	Support for standalone UDP/IP binding in CIR channel		O	
IMPS-CSP-TRANSP-S-011	Support for standalone TCP/IP binding in CIR channel.		O	IMPS-CSP-TRANSP-S-013
IMPS-CSP-TRANSP-S-012	With WSP 1.2 or WSP 2.0 bindings for data channel, only WAP SMS binding or WAP UDP binding is used in CIR channel.		O	IMPS-CSP-TRANSP-S-008 OR IMPS-CSP-TRANSP-S-009
IMPS-CSP-TRANSP-S-013	If the server discovers that the CIR channel over standalone TCP/IP is disconnected, the server-originated primitive contains the CIR element with the value ‘F’ to notify the client.		O	IMPS-CSP-TRANSP-S-011
IMPS-CSP-TRANSP-S-014	If a session has expired, and server originated notifications are delivered to the client		O	

Item	Function	Reference	Status	Requirement
	through a PollingRequest, the server delivers a Disconnect notification to the client before closing any session related resources.			
IMPS-CSP-TRANSP-S-015	Support for Standalone SMS binding in CIR channel.		O	
IMPS-CSP-TRANSP-S-016	Support for Standalone HTTP binding in CIR channel.		O	
IMPS-CSP-TRANSP-S-017	Support for Offline notification.		O	(IMPS-CSP-TRANSP-C-008 OR IMPS-CSP-TRANSP-C-009 OR IMPS-CSP-TRANSP-C-015) AND [CSP]:IMPS-CSP-CCAPAB-S-002

B.2 SMS and PTS Encoding Requirements

B.2.1 Clients

Item	Function	Reference	Status	Requirement
IMPS-CSP-SMS-C-001	Support for SMS encoded with UDH	10	O	IMPS-CSP-TRANSP-C-007 AND IMPS-CSP-SMS-C-002
IMPS-CSP-SMS-C-002	When session is started with UDH and the server supports it, all primitives are encoded with UDH during the session.	10	O	IMPS-CSP-TRANSP-C-007 AND IMPS-CSP-SMS-C-001
IMPS-CSP-SMS-C-003	Support for SMS encoded without UDH (textual)	10	O	IMPS-CSP-TRANSP-C-007 AND IMPS-CSP-SMS-C-004
IMPS-CSP-SMS-C-004	When session is started without UDH and the server supports it, all primitives are encoded without UDH during the session	10	O	IMPS-CSP-TRANSP-C-007 AND IMPS-CSP-SMS-C-003
IMPS-CSP-SMS-C-005	Support for one PTS message to contain multiple IMPS messages	10	O	IMPS-CSP-TRANSP-C-007
IMPS-CSP-SMS-C-006	Support for Plain Text Syntax over SMS transport	10	O	IMPS-CSP-TRANSP-C-007
IMPS-CSP-SMS-C-007	Support for Plain Text Syntax over HTTP transport	10	O	IMPS-CSP-TRANSP-C-003
IMPS-CSP-SMS-C-008	Support for Plain Text Syntax over HTTPS transport	10	O	IMPS-CSP-TRANSP-C-004
IMPS-CSP-SMS-C-009	Support for Plain Text Syntax over WSP transport	10	O	IMPS-CSP-TRANSP-C-005 OR IMPS-CSP-TRANSP-C-006
IMPS-CSP-SMS-C-010	Support for SMS encoded without UDH (textual) over a non-SMS transport	10	O	IMPS-CSP-TRANSP-C-005 OR IMPS-CSP-TRANSP-C-006

B.2.2 Servers

Item	Function	Reference	Status	Requirement
IMPS-CSP-SMS-S-001	Support for SMS encoded with UDH	10	O	IMPS-CSP-TRANSP-S-007 AND IMPS-CSP-SMS-S-002
IMPS-CSP-SMS-S-002	When session is started with UDH and the server supports it,	10	O	IMPS-CSP-TRANSP-S-007 AND IMPS-CSP-SMS-S-001

Item	Function	Reference	Status	Requirement
	all primitives are encoded with UDH during the session			
IMPS-CSP-SMS-S-003	Support for SMS encoded without UDH (textual)	10	O	IMPS-CSP-TRANSP-S-007 AND IMPS-CSP-SMS-S-004
IMPS-CSP-SMS-S-004	When session is started without UDH and the server supports it (CSPSMS-3), all primitives are encoded without UDH during the session	10	O	IMPS-CSP-TRANSP-S-007 AND IMPS-CSP-SMS-S-003
IMPS-CSP-SMS-S-005	Support for one PTS message to contain multiple IMPS messages	10	O	IMPS-CSP-TRANSP-S-007
IMPS-CSP-SMS-S-006	Support for Plain Text Syntax over a SMS transport	10	O	IMPS-CSP-TRANSP-S-007
IMPS-CSP-SMS-S-007	Support for Plain Text Syntax over HTTP transport	10	O	IMPS-CSP-TRANSP-S-003
IMPS-CSP-SMS-S-008	Support for Plain Text Syntax over HTTPS transport	10	O	IMPS-CSP-TRANSP-S-004
IMPS-CSP-SMS-S-009	Support for Plain Text Syntax over WSP transport	10	O	IMPS-CSP-TRANSP-S-005 OR IMPS-CSP-TRANSP-S-006
IMPS-CSP-SMS-S-010	Support for SMS encoded without UDH (textual) over a non-SMS transport	10	O	IMPS-CSP-TRANSP-S-005 OR IMPS-CSP-TRANSP-S-006