Technical Document - Confidential

GSM FAX & DATA SERVICES

TEST SPECIFICATION

UDP

Document Number:	8411.400.00.001
Version:	0.3
Status:	Draft
Approval Authority:	Udo Sprute
Creation Date:	2000-June-30
Last changed:	2015-Mar-08 by Jacek Kwasnik
File Name:	udp.doc

Important Notice

Texas Instruments Incorporated and/or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and/or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and/or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and/or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and/or software may be based on or implement industry recognized standards and that certain third parties may claim intellectual property rights therein. The supply of products and/or the licensing of software does not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third–party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Change History

Date	Changed by	Approved by	Version	Status	Notes
2000-Jun-30	Jürgen Nickelsen		0.1		1
2003-Jun-03	XINTEGR		0.2	Draft	
2003-Aug-28	Jacek Kwasnik		0.3	Draft	2

Notes:

1. Initial version
2. DTI nomenclature claryfied, corrected

Table of Contents

List of Figures and Tables

List of References

[ISO 9000:2000] International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000

1.1 References

[1] Service Access Point ACI
8411.105.98.100; Condat GmbH
[2] Message Sequence Charts ACI
8411.205.98.100; Condat GmbH
[3] Technical Documentation ACI
8411.705.98.100; Condat GmbH

1.2 Abbreviations

ACI	AT Command Interpreter
AT	Attention sequence "AT" to indicate valid commands of the ACI
TAP	Test Application Program

1.3 Terms

Entity:	Program which executes the functions of a layer
Message:	A message is a data unit which is transferred between the entities of the same layer (peer-to-peer) of the mobile and infrastructure side. Message is used as a synonym to protocol data unit (PDU). A message may contain several information elements.
Primitive:	A primitive is a data unit which is transferred between layers on one component (mobile station or infrastructure). The primitive has an operation code which identifies the primitive and its parameters.
Service Access Point:	A Service Access Point is a data interface between two layers on one component (mobile station or infrastructure).

2 Overview

The Protocol Stacks are used to define the functionality of the GSM protocols for interfaces. The GSM specifications are normative when used to describe the functionality of interfaces, but the stacks and the subdivision of protocol layers does not imply or restrict any implementation. The information units passed via the SAPs are called primitives and consists of an operation code and several parameters. See the Users Guide for details.

2.1 RLP - Radio Link Protocol

This entity provides a Layer 2 protocol for asynchronous reliable data transfer as specified in GSM 04.22. It includes error correction, sequence numbers and a mechanism for repeating corrupted and lost messages.

2.2 L2R - Layer 2 Relay Functionality

The L2R provides relay functions in order to adapt the character-oriented data received from the TE via USART to the bit-oriented RLP protocol.

2.3 ACI - AT Command Interpreter

The ACI is specified in GSM 07.07. It is responsible for call establishment via the GSM voice protocol stack and terminal adaptation for asynchronous transparent character-oriented data transmission. The ACI is able to receive AT commands and send the replies over the USART driver to a remote PC. This makes it possible to control the voice and data protocol stack from a remote application running on a PC. The ACI also provides a unique interface for an internal MMI in the MS.

3 Parameters

NOTE: Some of the primitive names described in this document carry "DTI" component in them, the other "DTI2" although actually the same "dti2" type of interface is meant and used. The same applies to parameter names where only "dti" occurs.

```
STRING (HL_NAME,"WAP")
STRING (LL_NAME,"IP")

/* declarations */
DECLARATION ( DTI_PARAMETER_WAP )
DECLARATION ( DTI_PARAMETER_WAP_SASB_OFF )
DECLARATION ( DTI_PARAMETER_WAP_SA_OFF )
DECLARATION ( DTI_PARAMETER_WAP_SB_OFF )

DECLARATION ( DTI_PARAMETER_IP )
DECLARATION ( DTI_PARAMETER_IP_SASB_OFF )
DECLARATION ( DTI_PARAMETER_IP_SA_OFF )
DECLARATION ( DTI_PARAMETER_IP_SB_OFF )

DECLARATION ( DTI_PARAMETER_UDP )
```

```
DECLARATION ( DTI_PARAMETER_UDP_SASB_OFF )
DECLARATION ( DTI_PARAMETER_UDP_SA_OFF )
DECLARATION ( DTI_PARAMETER_UDP_SB_OFF )

DECLARATION ( DTI_PARA_ST_LINES_SASB_ON)
DECLARATION ( DTI_PARA_ST_LINES_SASB_OFF)
DECLARATION ( DTI_PARA_ST_LINES_SA_OFF)
DECLARATION ( DTI_PARA_ST_LINES_SB_OFF)

/* IPGEN: processing input file "udp103.packets" */

FIELD(PKT_UDP_IN_103)
/* incoming UDP packet IP -> UDP */
                        0x60, 0x09, 0x00, 0x00,
                        /* IP proto 17 len 300 10.11.12.14 -> 10.11.12.13 */
                        0x45, 0x00, 0x01, 0x2c, 0x34, 0x56, 0x00, 0x00,
                        0x3a, 0x11, 0x1f, 0x3b, 0x0a, 0x0b, 0x0c, 0x0e,
                        0x0a, 0x0b, 0x0c, 0x0d,
                        /* UDP sport 9200 dport 1024 ulen 280 */
                        0x23, 0xf0, 0x04, 0x00, 0x01, 0x18, 0xa9, 0x9d,
                        /* data */
                        0x67, 0xc6, 0x69, 0x73, 0x51, 0xff, 0x4a, 0xec,
                        0x29, 0xcd, 0xba, 0xab, 0xf2, 0xfb, 0xe3, 0x46,
                        0x7c, 0xc2, 0x54, 0xf8, 0x1b, 0xe8, 0xe7, 0x8d,
                        0x76, 0x5a, 0x2e, 0x63, 0x33, 0x9f, 0xc9, 0x9a,
                        0x66, 0x32, 0x0d, 0xb7, 0x31, 0x58, 0xa3, 0x5a,
                        0x25, 0x5d, 0x05, 0x17, 0x58, 0xe9, 0x5e, 0xd4,
                        0xab, 0xb2, 0xcd, 0xc6, 0x9b, 0xb4, 0x54, 0x11,
                        0x0e, 0x82, 0x74, 0x41, 0x21, 0x3d, 0xdc, 0x87,
                        0x70, 0xe9, 0x3e, 0xa1, 0x41, 0xe1, 0xfc, 0x67,
                        0x3e, 0x01, 0x7e, 0x97, 0xea, 0xdc, 0x6b, 0x96,
                        0x8f, 0x38, 0x5c, 0x2a, 0xec, 0xb0, 0x3b, 0xfb,
                        0x32, 0xaf, 0x3c, 0x54, 0xec, 0x18, 0xdb, 0x5c,
                        0x02, 0x1a, 0xfe, 0x43, 0xfb, 0xfa, 0xaa, 0x3a,
                        0xfb, 0x29, 0xd1, 0xe6, 0x05, 0x3c, 0x7c, 0x94,
                        0x75, 0xd8, 0xbe, 0x61, 0x89, 0xf9, 0x5c, 0xbb,
                        0xa8, 0x99, 0x0f, 0x95, 0xb1, 0xeb, 0xf1, 0xb3,
                        0x05, 0xef, 0xf7, 0x00, 0xe9, 0xa1, 0x3a, 0xe5,
                        0xca, 0x0b, 0xcb, 0xd0, 0x48, 0x47, 0x64, 0xbd,
                        0x1f, 0x23, 0x1e, 0xa8, 0x1c, 0x7b, 0x64, 0xc5,
                        0x14, 0x73, 0x5a, 0xc5, 0x5e, 0x4b, 0x79, 0x63,
                        0x3b, 0x70, 0x64, 0x24, 0x11, 0x9e, 0x09, 0xdc,
                        0xaa, 0xd4, 0xac, 0xf2, 0x1b, 0x10, 0xaf, 0x3b,
                        0x33, 0xcd, 0xe3, 0x50, 0x48, 0x47, 0x15, 0x5c,
                        0xbb, 0x6f, 0x22, 0x19, 0xba, 0x9b, 0x7d, 0xf5,
                        0x0b, 0xe1, 0x1a, 0x1c, 0x7f, 0x23, 0xf8, 0x29,
                        0xf8, 0xa4, 0x1b, 0x13, 0xb5, 0xca, 0x4e, 0xe8,
                        0x98, 0x32, 0x38, 0xe0, 0x79, 0x4d, 0x3d, 0x34,
                        0xbc, 0x5f, 0x4e, 0x77, 0xfa, 0xcb, 0x6c, 0x05,
                        0xac, 0x86, 0x21, 0x2b, 0xaa, 0x1a, 0x55, 0xa2,
                        0xbe, 0x70, 0xb5, 0x73, 0x3b, 0x04, 0x5c, 0xd3,
                        0x36, 0x94, 0xb3, 0xaf, 0xe2, 0xf0, 0xe4, 0x9e,
                        0x4f, 0x32, 0x15, 0x49, 0xfd, 0x82, 0x4e, 0xa9,
                        0x08, 0x70, 0xd4, 0xb2, 0x8a, 0x29, 0x54, 0x48,
                        0x9a, 0x0a, 0xbc, 0xd5, 0x0e, 0x18, 0xa8, 0x44
ENDFIELD(PKT_UDP_IN_103, 304)


/* IPGEN: processing input file "udp202.packets" */
```

```
FIELD(PKT_UDP_IN_202)
/* incoming UDP packet IP -> UDP */
                        0x60, 0x09, 0x00, 0x00,
                        /* IP proto 17 len 300 10.11.12.14 -> 10.11.12.13 */
                        0x45, 0x00, 0x01, 0x2c, 0x34, 0x56, 0x00, 0x00,
                        0x3a, 0x11, 0x1f, 0x3b, 0x0a, 0x0b, 0x0c, 0x0e,
                        0x0a, 0x0b, 0x0c, 0x0d,
                        /* UDP sport 9200 dport 1024 ulen 280 */
                        0x23, 0xf0, 0x04, 0x00, 0x01, 0x18, 0xa9, 0x9d,
                        /* data */
                        0xac, 0x5b, 0xf3, 0x8e, 0x4c, 0xd7, 0x2d, 0x9b,
                        0x09, 0x42, 0xe5, 0x06, 0xc4, 0x33, 0xaf, 0xcd,
                        0xa3, 0x84, 0x7f, 0x2d, 0xad, 0xd4, 0x76, 0x47,
                        0xde, 0x32, 0x1c, 0xec, 0x4a, 0xc4, 0x30, 0xf6,
                        0x20, 0x23, 0x85, 0x6c, 0xfb, 0xb2, 0x07, 0x04,
                        0xf4, 0xec, 0x0b, 0xb9, 0x20, 0xba, 0x86, 0xc3,
                        0x3e, 0x05, 0xf1, 0xec, 0xd9, 0x67, 0x33, 0xb7,
                        0x99, 0x50, 0xa3, 0xe3, 0x14, 0xd3, 0xd9, 0x34,
                        0xf7, 0x5e, 0xa0, 0xf2, 0x10, 0xa8, 0xf6, 0x05,
                        0x94, 0x01, 0xbe, 0xb4, 0xbc, 0x44, 0x78, 0xfa,
                        0x49, 0x69, 0xe6, 0x23, 0xd0, 0x1a, 0xda, 0x69,
                        0x6a, 0x7e, 0x4c, 0x7e, 0x51, 0x25, 0xb3, 0x48,
                        0x84, 0x53, 0x3a, 0x94, 0xfb, 0x31, 0x99, 0x90,
                        0x32, 0x57, 0x44, 0xee, 0x9b, 0xbc, 0xe9, 0xe5,
                        0x25, 0xcf, 0x08, 0xf5, 0xe9, 0xe2, 0x5e, 0x53,
                        0x60, 0xaa, 0xd2, 0xb2, 0xd0, 0x85, 0xfa, 0x54,
                        0xd8, 0x35, 0xe8, 0xd4, 0x66, 0x82, 0x64, 0x98,
                        0xd9, 0xa8, 0x87, 0x75, 0x65, 0x70, 0x5a, 0x8a,
                        0x3f, 0x62, 0x80, 0x29, 0x44, 0xde, 0x7c, 0xa5,
                        0x89, 0x4e, 0x57, 0x59, 0xd3, 0x51, 0xad, 0xac,
                        0x86, 0x95, 0x80, 0xec, 0x17, 0xe4, 0x85, 0xf1,
                        0x8c, 0x0c, 0x66, 0xf1, 0x7c, 0xc0, 0x7c, 0xbb,
                        0x22, 0xfc, 0xe4, 0x66, 0xda, 0x61, 0x0b, 0x63,
                        0xaf, 0x62, 0xbc, 0x83, 0xb4, 0x69, 0x2f, 0x3a,
                        0xff, 0xaf, 0x27, 0x16, 0x93, 0xac, 0x07, 0x1f,
                        0xb8, 0x6d, 0x11, 0x34, 0x2d, 0x8d, 0xef, 0x4f,
                        0x89, 0xd4, 0xb6, 0x63, 0x35, 0xc1, 0xc7, 0xe4,
                        0x24, 0x83, 0x67, 0xd8, 0xed, 0x96, 0x12, 0xec,
                        0x45, 0x39, 0x02, 0xd8, 0xe5, 0x0a, 0xf8, 0x9d,
                        0x77, 0x09, 0xd1, 0xa5, 0x96, 0xc1, 0xf4, 0x1f,
                        0x95, 0xaa, 0x82, 0xca, 0x6c, 0x49, 0xae, 0x90,
                        0xcd, 0x16, 0x68, 0xba, 0xac, 0x7a, 0xa6, 0xf2,
                        0xb4, 0xa8, 0xca, 0x99, 0xb2, 0xc2, 0x37, 0x2a,
                        0xcb, 0x08, 0xcf, 0x61, 0xc9, 0xc3, 0x80, 0x5e
ENDFIELD(PKT_UDP_IN_202, 304)


/* IPGEN: processing input file "udp305.packets" */

FIELD(PKT_UDP_IN_305)
/* incoming UDP packet to unbound port */
                        0x20, 0x01, 0x00, 0x00,
                        /* IP proto 17 len 36 10.11.12.14 -> 10.11.12.13 */
                        0x45, 0x00, 0x00, 0x24, 0x11, 0xc6, 0x00, 0x00,
                        0x40, 0x11, 0x3c, 0xd3, 0x0a, 0x0b, 0x0c, 0x0e,
                        0x0a, 0x0b, 0x0c, 0x0d,
                        /* UDP sport 1201 dport 9 ulen 16 */
                        0x04, 0xb1, 0x00, 0x09, 0x00, 0x10, 0xc2, 0xd3,
```

```
                        /* data */
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07
ENDFIELD(PKT_UDP_IN_305, 40)

FIELD(PKT_ICMP_OUT_305)
/* outgoing ICMP packet due to port unreachable */
                        0xc0, 0x01, 0x00, 0x00,
                        /* IP proto 1 len 56 10.11.12.13 -> 10.11.12.14 */
                        0x45, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00,
                        0x40, 0x01, 0x4e, 0x95, 0x0a, 0x0b, 0x0c, 0x0d,
                        0x0a, 0x0b, 0x0c, 0x0e,
                        /* ICMP type 3 code 3 */
                        0x03, 0x03, 0x35, 0x5f, 0x00, 0x00, 0x00, 0x00,
                        /* data */
                        0x45, 0x00, 0x00, 0x24, 0x11, 0xc6, 0x00, 0x00,
                        0x40, 0x11, 0x3c, 0xd3, 0x0a, 0x0b, 0x0c, 0x0e,
                        0x0a, 0x0b, 0x0c, 0x0d, 0x04, 0xb1, 0x00, 0x09,
                        0x00, 0x10, 0xc2, 0xd3
ENDFIELD(PKT_ICMP_OUT_305, 60)


/* IPGEN: processing input file "udp306.packets" */

FIELD(PKT_UDP_IN_306)
/* incoming UDP packet with UDP checksum error */
                        0x20, 0x01, 0x00, 0x00,
                        /* IP proto 17 len 36 10.11.12.14 -> 10.11.12.13 */
                        0x45, 0x00, 0x00, 0x24, 0x11, 0xc6, 0x00, 0x00,
                        0x40, 0x11, 0x3c, 0xd3, 0x0a, 0x0b, 0x0c, 0x0e,
                        0x0a, 0x0b, 0x0c, 0x0d,
                        /* UDP sport 1201 dport 9 ulen 16 */
                        0x04, 0xb1, 0x00, 0x09, 0x00, 0x10, 0x04, 0xc7,
                        /* data */
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07
ENDFIELD(PKT_UDP_IN_306, 40)


/* IPGEN: processing input file "udp307.packets" */

FIELD(PKT_UDP_IN_307)
/* incoming UDP packet to bound port */
                        0x20, 0x01, 0x00, 0x00,
                        /* IP proto 17 len 36 10.11.12.14 -> 10.11.12.13 */
                        0x45, 0x00, 0x00, 0x24, 0x11, 0xc6, 0x00, 0x00,
                        0x40, 0x11, 0x3c, 0xd3, 0x0a, 0x0b, 0x0c, 0x0e,
                        0x0a, 0x0b, 0x0c, 0x0d,
                        /* UDP sport 9200 dport 80 ulen 16 */
                        0x23, 0xf0, 0x00, 0x50, 0x00, 0x10, 0xa3, 0x4d,
                        /* data */
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07
ENDFIELD(PKT_UDP_IN_307, 40)

/* Downlink packet to WAP */
FIELD(SDU_UDP_IN_307)
                        0xa0, 0x00, 0x00, 0x00,
                        0x0e, 0x0c, 0x0b, 0x0a, /* Source Address */
                        0x0d, 0x0c, 0x0b, 0x0a, /* Destination Address */
                        0xf0, 0x23,     /* Source Port */
                        0x50, 0x00,     /* Destination Port */
```

```
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07
ENDFIELD(SDU_UDP_IN_307, 24)

/* IPGEN: processing input file "udp308.packets" */

FIELD(PKT_ICMP_OUT_308)
/* incoming ICMP packet due to port unreachable */
                        0xc0, 0x01, 0x00, 0x00,
                        /* IP proto 1 len 56 10.11.12.14 -> 10.11.12.13 */
                        0x45, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00,
                        0x40, 0x01, 0x4e, 0x95, 0x0a, 0x0b, 0x0c, 0x0e,
                        0x0a, 0x0b, 0x0c, 0x0d,
                        /* ICMP type 3 code 3 */
                        0x03, 0x03, 0x29, 0x47, 0x00, 0x00, 0x00, 0x00,
                        /* data */
                        0x45, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00,
                        0x40, 0x11, 0x4e, 0x3d, 0x0a, 0x0b, 0x0c, 0x0d,
                        0x0a, 0x0b, 0x0c, 0x0e, 0x00, 0x50, 0x23, 0xf0,
                        0x00, 0x08, 0xaf, 0x6d
ENDFIELD(PKT_ICMP_OUT_308, 60)


/* IPGEN: processing input file "udp309.packets" */

FIELD(PKT_UDP_OUT_2_309)
/* outgoing UDP packet UDP -> IP */
                        0x00, 0x04, 0x00, 0x00,
                        /* IP proto 17 len 128 10.11.12.13 -> 10.11.12.14 */
                        0x45, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00,
                        0x00, 0x11, 0x00, 0x00, 0x0a, 0x0b, 0x0c, 0x0d,
                        0x0a, 0x0b, 0x0c, 0x0e,
                        /* UDP sport 80 dport 9200 ulen 108 */
                        0x00, 0x50, 0x23, 0xf0, 0x00, 0x6c, 0x86, 0x02,
                        /* data */
                        0x6e, 0x03, 0x28, 0xda, 0x4c, 0xd7, 0x6a, 0x19,
                        0xed, 0xd2, 0xd3, 0x99, 0x4c, 0x79, 0x8b, 0x00,
                        0x22, 0x56, 0x9a, 0xd4, 0x18, 0xd1, 0xfe, 0xe4,
                        0xd9, 0xcd, 0x45, 0xa3, 0x91, 0xc6, 0x01, 0xff,
                        0xc9, 0x2a, 0xd9, 0x15, 0x01, 0x43, 0x2f, 0xee,
                        0x15, 0x02, 0x87, 0x61, 0x7c, 0x13, 0x62, 0x9e,
                        0x69, 0xfc, 0x72, 0x81, 0xcd, 0x71, 0x65, 0xa6,
                        0x3e, 0xab, 0x49, 0xcf, 0x71, 0x4b, 0xce, 0x3a,
                        0x75, 0xa7, 0x4f, 0x76, 0xea, 0x7e, 0x64, 0xff,
                        0x81, 0xeb, 0x61, 0xfd, 0xfe, 0xc3, 0x9b, 0x67,
                        0xbf, 0x0d, 0xe9, 0x8c, 0x7e, 0x4e, 0x32, 0xbd,
                        0xf9, 0x7c, 0x8c, 0x6a, 0xc7, 0x5b, 0xa4, 0x3c,
                        0x02, 0xf4, 0xb2, 0xed
ENDFIELD(PKT_UDP_OUT_2_309, 132)

FIELD(SDU_UDP_OUT_101)
                        0xe0, 0x00, 0x00, 0x00,
                        0x0a, 0x0b, 0x0c, 0x0d,
                        0x0a, 0x0b, 0x0c, 0x0e,
                        0x04, 0x00, 0xf0, 0x23,
                        0x62, 0x6c, 0x75, 0x62, 0x62, 0x65, 0x72, 0x66,
                        0x61, 0x73, 0x65, 0x62, 0x65, 0x69, 0x65, 0x6e
ENDFIELD(SDU_UDP_OUT_101, 32)

FIELD(SDU_UDP_OUT_203)
```

```
                    0xe0, 0x00, 0x00, 0x00,
                    0x0a, 0x0b, 0x0c, 0x0d,
                    0x0a, 0x0b, 0x0c, 0x0e,
                    0x00, 0x04, 0xf0, 0x23,
                    0x62, 0x6c, 0x75, 0x62, 0x62, 0x65, 0x72, 0x66,
                    0x61, 0x73, 0x65, 0x62, 0x65, 0x69, 0x65, 0x6e
ENDFIELD(SDU_UDP_OUT_203, 32)


FIELD(SDU_UDP_OUT_1_309)
                    0x80, 0x03, 0x00, 0x00,
                    0x0d, 0x0c, 0x0b, 0x0a,
                    0x0e, 0x0c, 0x0b, 0x0a,
                    0x50, 0x00, 0xf0, 0x23,
                    0x6e, 0x03, 0x28, 0xda, 0x4c, 0xd7, 0x6a, 0x19,
                    0xed, 0xd2, 0xd3, 0x99, 0x4c, 0x79, 0x8b, 0x00,
                    0x22, 0x56, 0x9a, 0xd4, 0x18, 0xd1, 0xfe, 0xe4,
                    0xd9, 0xcd, 0x45, 0xa3, 0x91, 0xc6, 0x01, 0xff,
                    0xc9, 0x2a, 0xd9, 0x15, 0x01, 0x43, 0x2f, 0xee,
                    0x15, 0x02, 0x87, 0x61, 0x7c, 0x13, 0x62, 0x9e,
                    0x69, 0xfc, 0x72, 0x81, 0xcd, 0x71, 0x65, 0xa6,
                    0x3e, 0xab, 0x49, 0xcf, 0x71, 0x4b, 0xce, 0x3a,
                    0x75, 0xa7, 0x4f, 0x76, 0xea, 0x7e, 0x64, 0xff,
                    0x81, 0xeb, 0x61, 0xfd, 0xfe, 0xc3, 0x9b, 0x67,
                    0xbf, 0x0d, 0xe9, 0x8c, 0x7e, 0x4e, 0x32, 0xbd,
                    0xf9, 0x7c, 0x8c, 0x6a, 0xc7, 0x5b, 0xa4, 0x3c,
                    0x02, 0xf4, 0xb2, 0xed

ENDFIELD(SDU_UDP_OUT_1_309, 116)

FIELD(SDU_UDP_OUT_310)
                    0x50, 0x03, 0x00, 0x00,
                    0x0d, 0x0c, 0x0b, 0x0a,
                    0x00, 0x00, 0x00, 0x00,
                    0x50, 0x00, 0xf0, 0x23,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03
ENDFIELD(SDU_UDP_OUT_310, 116)

FIELD(SDU_UDP_OUT_311)
                    0x50, 0x03, 0x00, 0x00,
                    0x0d, 0x0c, 0x0b, 0x1a,
                    0x0e, 0x0c, 0x0b, 0x0a,
                    0xf0, 0x23, 0x50, 0x00,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
```

```
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                        0x00, 0x01, 0x02, 0x03
ENDFIELD(SDU_UDP_OUT_311, 116)

/* UDP and IP – Datagram */
FIELD(UDP_IP_BRODCAST_DST_ADDR)
   0x70, 0x02, 0x00, 0x00,
   0x45,0x00,0x00,0x4e,0x00,0x00,0x00,0x00,0x00,0x11,0x00,0x00,
   /* Source address */
   0xc6,0xa8,0x01,0x05,
  /* Destination address */
  0xff,0xff,0xff,0xff,
  /* Source port and destination port */
   0x00,0x89, 0x00,0x89,
  /* Length and UDP checksum */
   0x00,0x3a,0x53,0x16,
   0x7a,0x2e,0x01,0x10,
   0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x45,0x47,0x45,0x4d,0x45,0x42,0x46,
   0x45,0x45,0x49,0x46,0x46,0x46,0x44,0x43,0x41,0x43,0x41,0x43,0x41,0x43,0x41,0x43,
   0x41,0x43,0x41,0x43,0x41,0x43,0x41,0x42,0x4e,0x00,0x00,0x20,0x00,0x01
ENDFIELD(UDP_IP_BRODCAST_DST_ADDR, 82)

FIELD(WAP_BRODCAST_DST_ADDR)
    0xf0, 0x01, 0x00, 0x00,
   /* Source address */
  0x05,0x01,0xa8,0xc6,
  /* Destination address */
  0xff,0xff,0xff,0xff,
  /* Source port and destination port */
   0x89,0x00, 0x89,0x00,
   0x7a,0x2e,0x01,0x10,
   0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x45,0x47,0x45,0x4d,0x45,0x42,0x46,
   0x45,0x45,0x49,0x46,0x46,0x46,0x44,0x43,0x41,0x43,0x41,0x43,0x41,0x43,0x41,0x43,
   0x41,0x43,0x41,0x43,0x41,0x43,0x41,0x42,0x4e,0x00,0x00,0x20,0x00,0x01
ENDFIELD(WAP_BRODCAST_DST_ADDR, 66)

BYTE   ERROR_MSG_NUMBER        3
LONG   OWN_IP_ADDRESS     0x0a0b0c0d          /* 10.11.12.13 */
LONG   PEER_IP_ADDRESS    0x0a0b0c0e          /* 10.11.12.14 */

LONG   DST_BROADCAST_ADDR   0xffffffff
LONG   OWN_IP_ADDRESS_1 0xc6a80105

LONG   INVALID_IP_ADDRESS      0x0                   /* 0.0.0.0 */

BYTE   UDP_P_ID           0x21

LONG   IP_LINK_ID         3
BYTE   IP_P_ID                 0

LONG   WAP_LINK_ID        5
BYTE   WAP_P_ID           0
```

```
SHORTWAP_PORT           80

SHORTZERO_PORT          0

SHORTUDP_BIND_PORT              1234

SHORTFIRST_FREE_PORT    1025
SHORTFIXED_PORT_1               80
SHORTFIXED_PORT_2               0x0050
SHORTFIXED_PORT_3               0x2947
SHORTFIXED_PORT_4               0x0303
SHORTFIXED_PORT_5               0x23f0
SHORTFIXED_PORT_6               0x0089

/* Parameters for sending DTI2 primitives */

BEGIN_PSTRUCT ("st_lines", DTI_PARA_ST_LINES_SASB_ON)
        SET_COMP ("st_flow",            DTI_FLOW_ON)
        SET_COMP ("st_line_sa",                 DTI_SA_ON)
        SET_COMP ("st_line_sb",                 DTI_SB_ON)
        SET_COMP ("st_break_len",       DTI_BREAK_OFF)
ENDSTRUCT
BEGIN_PSTRUCT ("st_lines", DTI_PARA_ST_LINES_SASB_OFF)
        SET_COMP ("st_flow",            DTI_FLOW_ON)
        SET_COMP ("st_line_sa",                 DTI_SA_OFF)
        SET_COMP ("st_line_sb",                 DTI_SB_OFF)
        SET_COMP ("st_break_len",       DTI_BREAK_OFF)
ENDSTRUCT
BEGIN_PSTRUCT ("st_lines", DTI_PARA_ST_LINES_SA_OFF)
        SET_COMP ("st_flow",            DTI_FLOW_ON)
        SET_COMP ("st_line_sa",                 DTI_SA_OFF)
        SET_COMP ("st_line_sb",                 DTI_SB_ON)
        SET_COMP ("st_break_len",       DTI_BREAK_OFF)
ENDSTRUCT
BEGIN_PSTRUCT ("st_lines", DTI_PARA_ST_LINES_SB_OFF)
        SET_COMP ("st_flow",            DTI_FLOW_ON)
        SET_COMP ("st_line_sa",                 DTI_SA_ON)
        SET_COMP ("st_line_sb",                 DTI_SB_OFF)
        SET_COMP ("st_break_len",       DTI_BREAK_OFF)
ENDSTRUCT

BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_WAP)
        SET_COMP ("p_id",       WAP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SASB_ON)
ENDSTRUCT
BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_WAP_SASB_OFF)
        SET_COMP ("p_id",       WAP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SASB_OFF)
ENDSTRUCT
BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_WAP_SA_OFF)
        SET_COMP ("p_id",       WAP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SA_OFF)
ENDSTRUCT
BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_WAP_SB_OFF)
        SET_COMP ("p_id",       WAP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SB_OFF)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_IP)
        SET_COMP ("p_id",      IP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SASB_ON)
ENDSTRUCT

BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_IP_SASB_OFF)
        SET_COMP ("p_id",      IP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SASB_OFF)
ENDSTRUCT

BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_IP_SA_OFF)
        SET_COMP ("p_id",      IP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SA_OFF)
ENDSTRUCT

BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_IP_SB_OFF)
        SET_COMP ("p_id",      IP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SB_OFF)
ENDSTRUCT

BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_UDP)
        SET_COMP ("p_id",      UDP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SASB_ON)
ENDSTRUCT

BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_UDP_SASB_OFF)
        SET_COMP ("p_id",      UDP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SASB_OFF)
ENDSTRUCT

BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_UDP_SA_OFF)
        SET_COMP ("p_id",      UDP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SA_OFF)
ENDSTRUCT

BEGIN_PSTRUCT ("parameters", DTI_PARAMETER_UDP_SB_OFF)
        SET_COMP ("p_id",      UDP_P_ID)
        SET_COMP ("st_lines", DTI_PARA_ST_LINES_SB_OFF)
ENDSTRUCT
```

4 TEST CASES

4.1 Routing (internal) (UDP001)

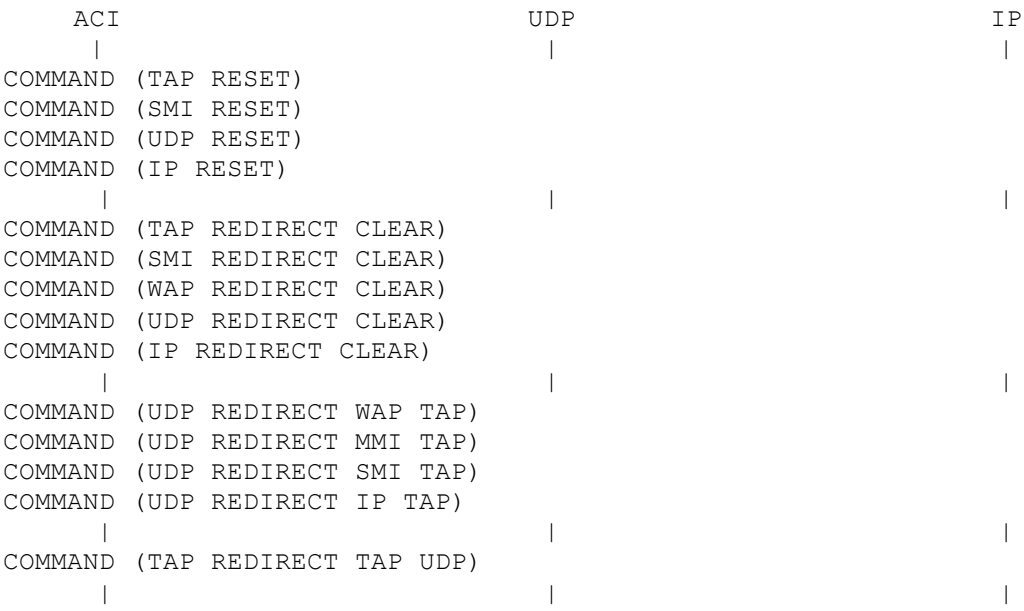
4.1.1 UDP101: Setup the Routing for the UDP test

Description:

Routings for the UDP tests are set.

Preamble:

None



Parametrization:

Primitive	Parameter	Value

History:

28 June 2000	NI	initial

4.2 Deactivated State

4.2.1 UDP201: Activation #1

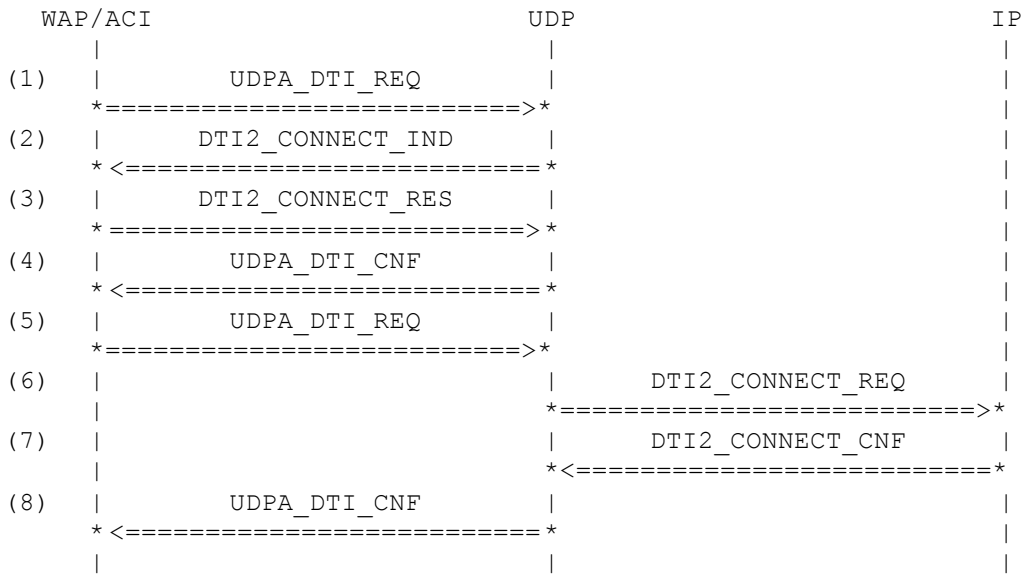
Description:

The UDP entity receives the UDPA_DTI_REQ and is switched to the activated state. UDP informs ACI that the entity is activated.

Variants: <A>..<D>

Preamble:

```
<A>UDP101 /* Setup the Routing for the UDP test
*/
<B>UDP204 /* Ignore reception of
DTI2_DATA_TEST_REQ */
<C>UDP205 /* Ignore reception of
DTI2_READY_IND */
<D>UDP206 /* Ignore reception of
DTI2_DATA_TEST_IND */
```



Parametrization:

Primitive	Parameter	Value
(1) UDPA_DTI_REQ	dti_conn	UDPA_CONNECT_DTI
	entity_name	HL_NAME
	link_id	WAP_LINK_ID
	dti_direction	UDPA_DTI_TO_HIGHER_LAYER
(2) DTI2_CONNECT_IND	link_id	WAP_LINK_ID
	version	DTI_VERSION_10
(3) DTI2_CONNECT_RES	link_id	WAP_LINK_ID

	version	DTI_VERSION_10
(4) UDPA_DTI_CNF	dti_conn	UDPA_CONNECT_DTI
	link_id	WAP_LINK_ID
(5) UDPA_DTI_REQ	dti_conn	UDPA_CONNECT_DTI
	entity_name	LL_NAME
	link_id	IP_LINK_ID
	dti_direction	
	UDPA_DTI_TO_LOWER_LAYER	
(6) DTI2_CONNECT_REQ	link_id	IP_LINK_ID
	version	DTI_VERSION_10
(7) DTI2_CONNECT_CNF	link_id	IP_LINK_ID
	version	DTI_VERSION_10
(8) UDPA_DTI_CNF	dti_conn	UDPA_CONNECT_DTI
	link_id	IP_LINK_ID

History:

28 June 2000	NI	initial
25 Oct 2002	KJF	new UDPA SAP

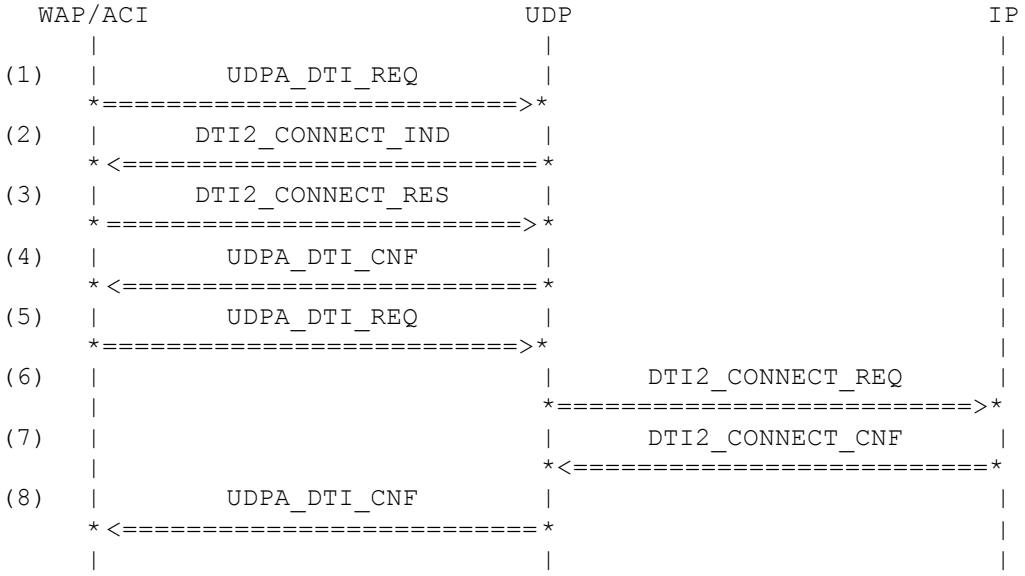
4.2.2 UDP202: Activation #2

Description:

The UDP entity receives the UDPA_DTI_REQ and is switched to the activated state. UDP informs ACI that the entity is activated.

Preamble:

UDP311A



Parametrization:

Primitive	Parameter	Value

(1) UDPA_DTI_REQ	dti_conn	UDPA_CONNECT_DTI
	entity_name	HL_NAME
	link_id	WAP_LINK_ID
	dti_direction	
	UDPA_DTI_TO_HIGHER_LAYER	
(2) DTI2_CONNECT_IND	link_id	WAP_LINK_ID
	version	DTI_VERSION_10
(3) DTI2_CONNECT_RES	link_id	WAP_LINK_ID
	version	DTI_VERSION_10
(4) UDPA_DTI_CNF	dti_conn	UDPA_CONNECT_DTI
	link_id	WAP_LINK_ID
(5) UDPA_DTI_REQ	dti_conn	UDPA_CONNECT_DTI
	entity_name	LL_NAME
	link_id	IP_LINK_ID
	dti_direction	
	UDPA_DTI_TO_LOWER_LAYER	
(6) DTI2_CONNECT_REQ	link_id	IP_LINK_ID
	version	DTI_VERSION_10
(7) DTI2_CONNECT_CNF	link_id	IP_LINK_ID
	version	DTI_VERSION_10
(8) UDPA_DTI_CNF	dti_conn	UDPA_CONNECT_DTI
	link_id	IP_LINK_ID

History:

28 June 2000	NI	initial
25 Oct 2002	KJF	new UDPA SAP

4.2.3 UDP203: Activation #3

Description:

The UDP entity receives the UDPA_DTI_REQ and is switched to the activated state. UDP informs ACI that the entity is activated.

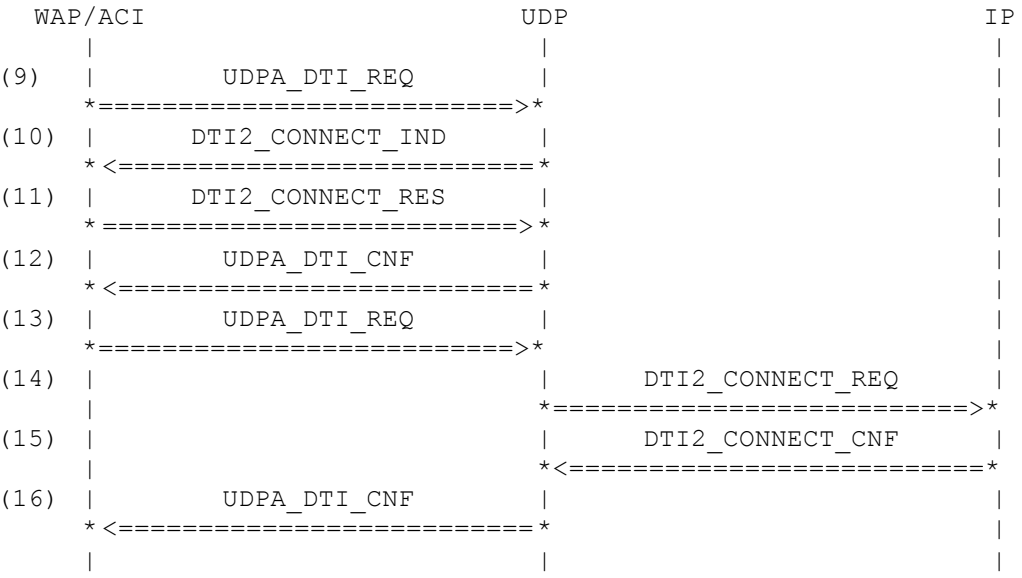
Variants: <A>..<C>

Preamble:

<A> UDP310A

<B> UDP311B

<C> UDP310C



Parametrization:

Primitive	Parameter	Value
(9) UDPA_DTI_REQ	dti_conn	UDPA_CONNECT_DTI
	entity_name	HL_NAME
	link_id	WAP_LINK_ID
	dti_direction	UDPA_DTI_TO_HIGHER_LAYER
(10) DTI2_CONNECT_IND	link_id	WAP_LINK_ID
	version	DTI_VERSION_10
(11) DTI2_CONNECT_RES	link_id	WAP_LINK_ID
	version	DTI_VERSION_10
(12) UDPA_DTI_CNF	dti_conn	UDPA_CONNECT_DTI
	link_id	WAP_LINK_ID
(13) UDPA_DTI_REQ	dti_conn	UDPA_CONNECT_DTI
	entity_name	LL_NAME
	link_id	IP_LINK_ID
	dti_direction	

	UDPA_DTI_TO_LOWER_LAYER	
(14) DTI2_CONNECT_REQ	link_id	IP_LINK_ID
	version	DTI_VERSION_10
(15) DTI2_CONNECT_CNF	link_id	IP_LINK_ID
	version	DTI_VERSION_10
(16) UDPA_DTI_CNF	dti_conn	UDPA_CONNECT_DTI
	link_id	IP_LINK_ID

History:

28 June 2000	NI	initial
25 Oct 2002	KJF	new UDPA SAP

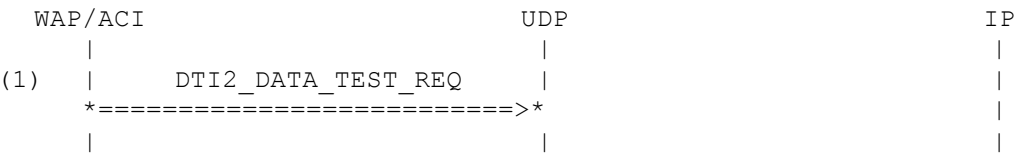
4.2.4 UDP204: Ignore reception of DTI2_DATA_TEST_REQ

Description:

The UDP entity receives DTI2_DATA_TEST_REQ from the transport layer and ignores it.

Preamble:

UDP101



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_101

History:

09 Nov 2001	TVO	initial

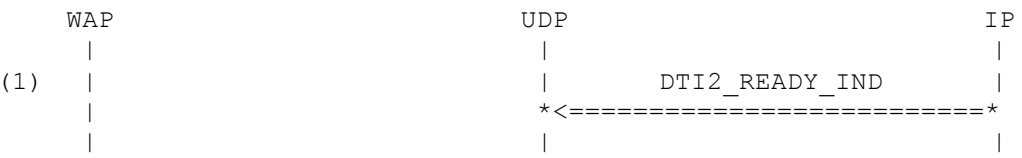
4.2.5 UDP205: Ignore reception of DTI2_READY_IND

Description:

The UDP entity receives a DTI2_READY_IND from an interface layer and ignores it.

Preamble:

UDP101



Parametrization:

Primitive	Parameter	Value
(1) DTI2_READY_IND	link_id	IP_LINK_ID

History:

09 Nov 2001	TVO	initial

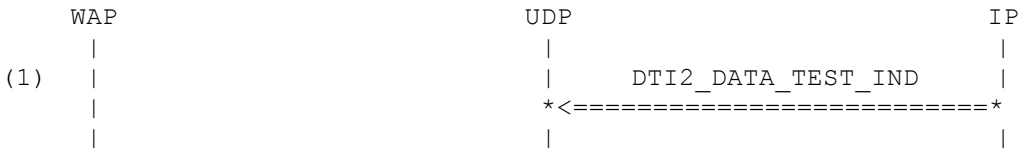
4.2.6 UDP206: Ignore reception of DTI2_DATA_TEST_IND

Description:

The UDP entity receives a DTI2_DATA_TEST_IND from the interface layer and ignores it.

Preamble:

UDP101



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_103

History:

09 Nov 2001	TVO	initial

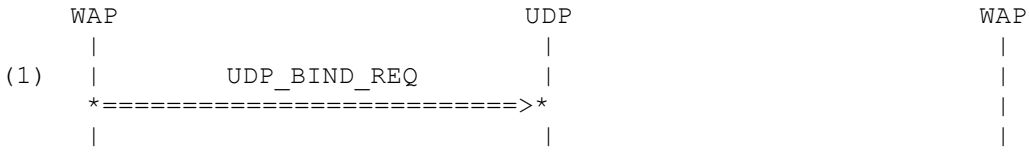
4.2.7 UDP207: Ignore reception of UDP_BIND_REQ

Description:

The UDP entity receives a UDP_BIND_REQ from an application and ignores it.

Preamble:

UDP101



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	UDP_BIND_PORT

History:

30 June 2000	NI	initial
19 July 2000	xof	UDP_BIND_UDPDOWN

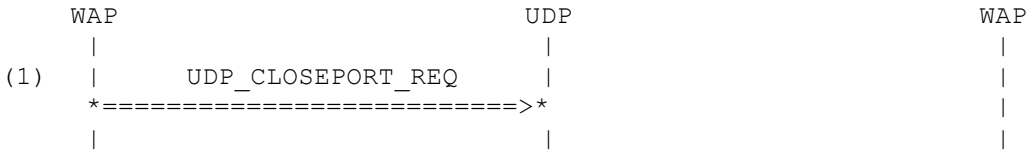
4.2.8 UDP208: Ignore reception of UDP_CLOSEPORT_REQ

Description:

The UDP entity receives a UDP_CLOSEPORT_REQ from an application and ignores it.

Preamble:

UDP101



Parametrization:

Primitive	Parameter	Value
(1) UDP_CLOSEPORT_REQ	port	UDP_BIND_PORT

History:

30 June 2000	NI	initial

4.3 Activated but Non-configured State

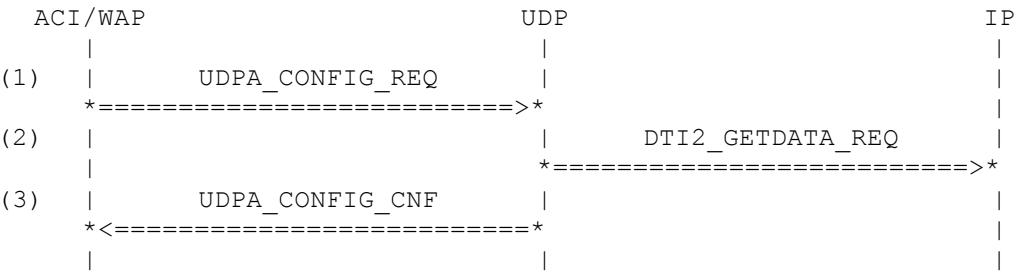
4.3.1 UDP301: Configuration #1

Description:

Upon Reception of UDPA_CONFIG_REQ the UDP entity sends a DTI2_GETDATA_REQ to IP
and confirms to ACI with UDPA_CONFIG_CNF.
It is now able to receive data from IP or WAP and to relay it if appropriate.

Preamble:

UDP201A



Parametrization:

Primitive	Parameter	Value
(1) UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_UP
(2) DTI2_GETDATA_REQ	link_id	IP_LINK_ID
(3) UDPA_CONFIG_CNF		

History:

30 June 2000	NI	initial
02 Oct 2000	xof	revised
15 Nov 2001	TVO	added variants
25 Oct 2002	KJF	new UDPA SAP

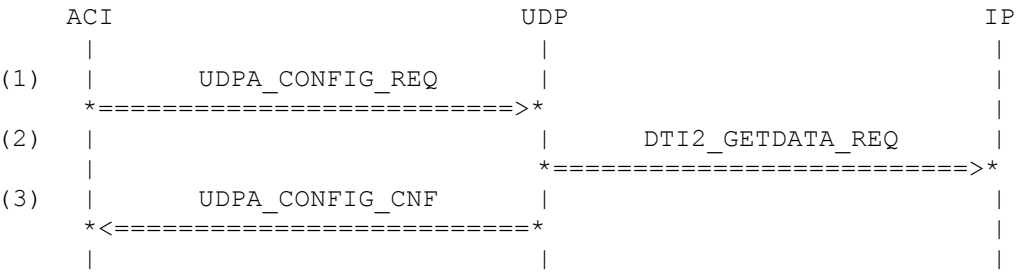
4.3.2 UDP302: Configuration #2

Description:

Upon Reception of UDPA_CONFIG_REQ the UDP entity sends a DTI2_GETDATA_REQ to IP
and confirms to ACI with UDPA_CONFIG_CNF.
It is now able to receive data from IP or WAP and to relay it if appropriate.

Preamble:

UDP202



Parametrization:

Primitive	Parameter	Value
(1) UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_UP
(2) DTI2_GETDATA_REQ	link_id	IP_LINK_ID
(3) UDPA_CONFIG_CNF		

History:

30 June 2000	NI	initial
02 Oct 2000	xof	revised

4.3.3 UDP303: Configuration #3

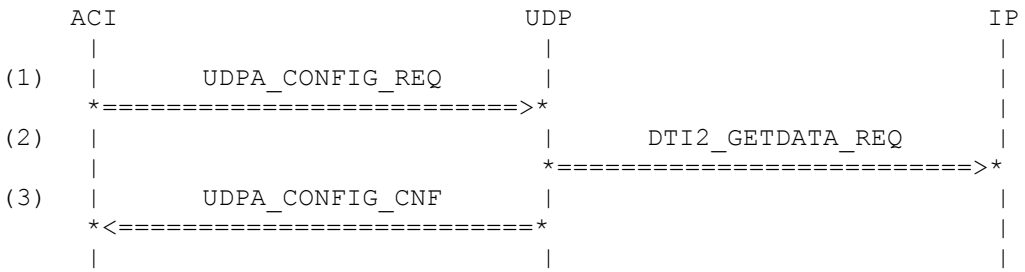
Description:

Upon Reception of UDPA_CONFIG_REQ the UDP entity sends a DTI2_GETDATA_REQ to IP
and confirms to ACI with UDPA_CONFIG_CNF.
It is now able to receive data from IP or WAP and to relay it if appropriate.

Variants: <A>..<C>

Preamble:

<A>UDP203A
<B>UDP203B
<C>UDP203C



Parametrization:

Primitive	Parameter	Value
(1) UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_UP
(2) DTI2_GETDATA_REQ	link_id	IP_LINK_ID
(3) UDPA_CONFIG_CNF		

History:

30 June 2000	NI	initial
02 Oct 2000	xof	revised
15 Nov 2001	TVO	added variants

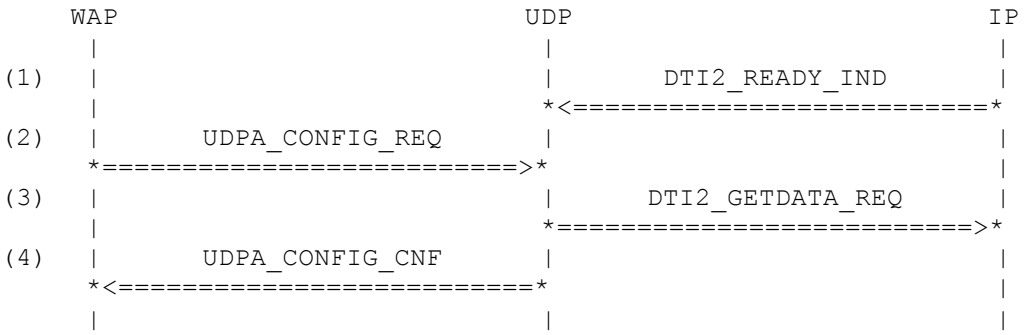
4.3.4 UDP304: Reception of DTI2_READY_IND and then configure

Description:

The UDP entity receives a DTI2_READY_IND from the network layer and stores it. Following configuration completes normally.

Preamble:

UDP201A



Parametrization:

	Primitive	Parameter	Value
(1)	DTI2_READY_IND	link_id	IP_LINK_ID
(2)	UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_UP
(3)	DTI2_GETDATA_REQ	link_id	IP_LINK_ID
(4)	UDPA_CONFIG_CNF		

History:

30 June 2000	NI	initial

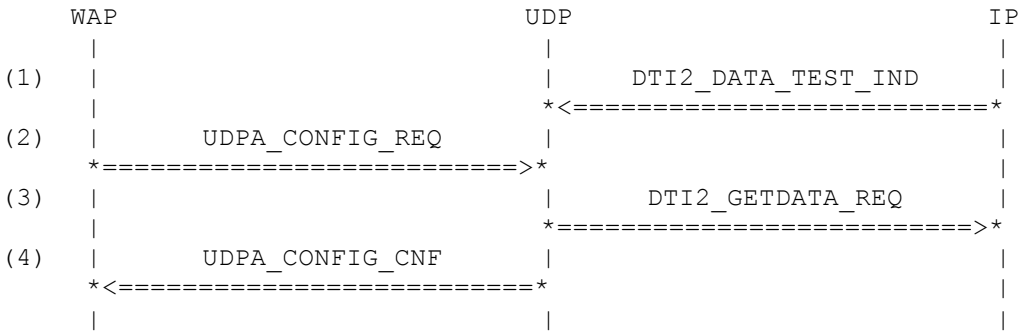
4.3.5 UDP305: Ignore reception of DTI2_DATA_TEST_IND and then configure

Description:

The UDP entity receives a DTI2_DATA_TEST_IND from the network layer and ignores it. Following configuration completes normally.

Preamble:

UDP201A



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_202
(2) UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_UP
(3) DTI2_GETDATA_REQ	link_id	IP_LINK_ID
(4) UDPA_CONFIG_CNF		

History:

28 June 2000	NI	initial

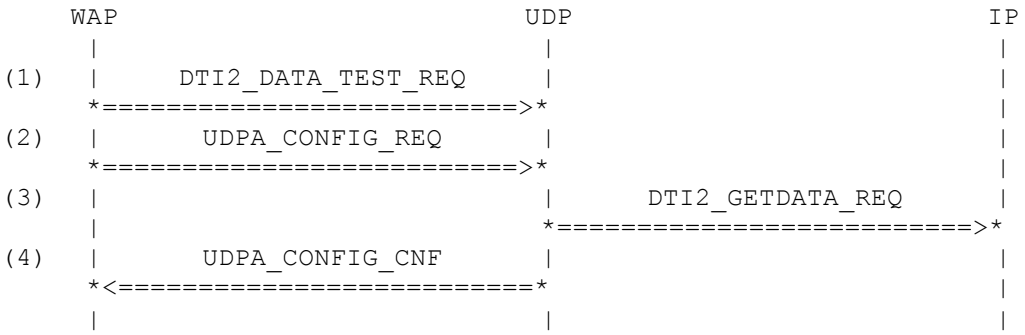
4.3.6 UDP306: Ignore reception of DTI2_DATA_TEST_REQ and then configure

Description:

The UDP entity receives a DTI2_DATA_TEST_REQ from an application and ignores it. Following configuration completes normally.

Preamble:

UDP201A



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_203
(2) UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_UP
(3) DTI2_GETDATA_REQ	link_id	IP_LINK_ID
(4) UDPA_CONFIG_CNF		

History:

30 June 2000	NI	initial

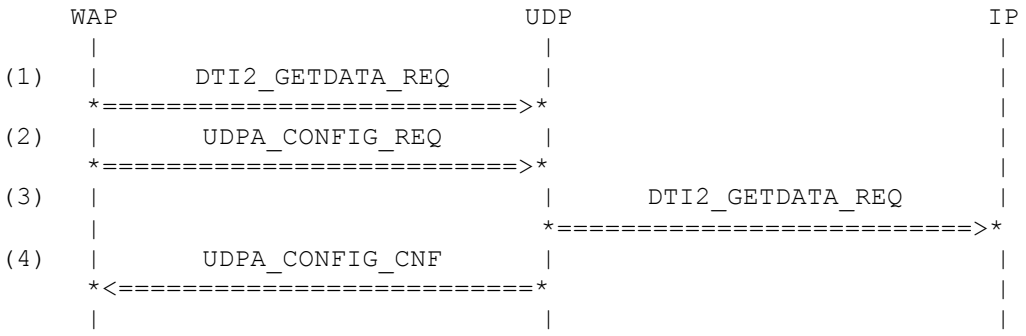
4.3.7 UDP307: Ignore reception of DTI2_GETDATA_REQ and then configure

Description:

The UDP entity receives a DTI2_DATA_TEST_REQ from the transport layer and ignores it.
Following configuration completes normally.

Preamble:

UDP201A



Parametrization:

Primitive	Parameter	Value
(1) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(2) UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_UP
(3) DTI2_GETDATA_REQ	link_id	IP_LINK_ID
(4) UDPA_CONFIG_CNF		

History:

30 June 2000	NI	initial

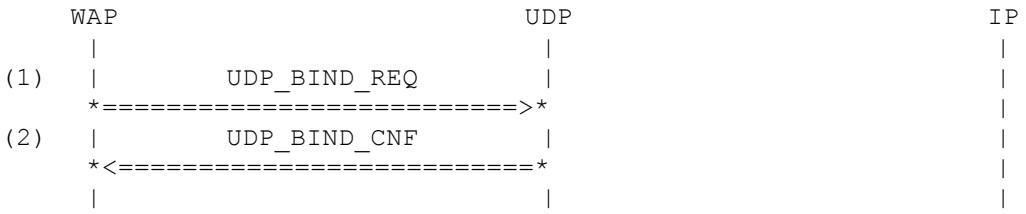
4.3.8 UDP308: Reception of UDP_BIND_REQ – error report

Description:

The UDP entity receives an UDP_BIND_REQ from an application.
UDP returns an UDP_BIND_CNF with the err field set to UDP_BIND_UDPDOWN.

Preamble:

UDP201A



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	UDP_BIND_PORT
(2) UDP_BIND_CNF	port	UDP_BIND_PORT
	err	UDP_BIND_UDPDOWN

History:

30 June 2000	NI	initial

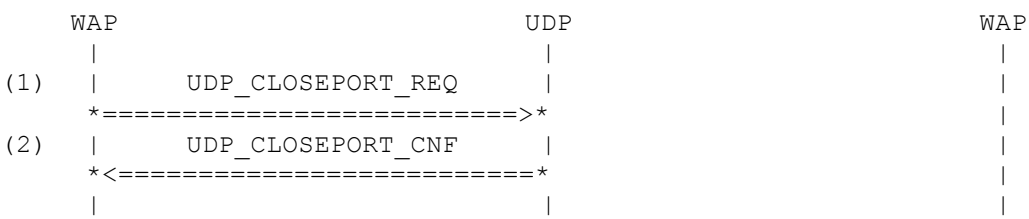
4.3.9 UDP309: Reception of UDP_CLOSEPORT_REQ - confirm

Description:

The UDP entity receives a UDP_CLOSEPORT_REQ from an application.
UDP confirms the request.

Preamble:

UDP201A



Parametrization:

Primitive	Parameter	Value
(1) UDP_CLOSEPORT_REQ	port	UDP_BIND_PORT
(2) UDP_CLOSEPORT_CNF		

History:

30 June 2000	NI	initial

4.3.10 UDP310: Deactivation by UDPA_DTI_REQ (lower layer)

Description:

The UDP entity receives the UDPA_DTI_REQ(UDPA_DISCONNECT_DTI) primitive for the lower layer.

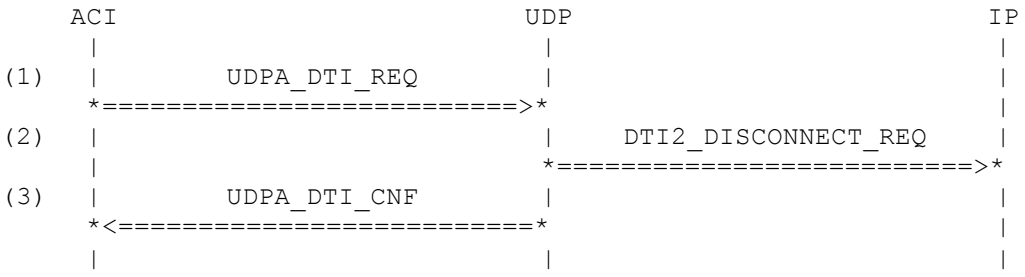
Variants: <A>..<C>

Preamble:

<A>UDP311C

<B>UDP420A

<C>UDP424B



Parametrization:

Primitive	Parameter	Value
(1) UDPA_DTI_REQ	dti_conn	UDPA_DISCONNECT_DTI
	entity_name	LL_NAME
	link_id	IP_LINK_ID
	dti_direction	UDPA_DTI_TO_LOWER_LAYER
(2) DTI2_DISCONNECT_REQ	link_id	IP_LINK_ID
	cause	DTI_CAUSE_NORMAL_CLOSE
(3) UDPA_DTI_CNF	dti_conn	UDPA_DISCONNECT_DTI
	link_id	IP_LINK_ID

History:

28 June 2000	NI	initial
25 Oct 2002	KJF	new UDPA SAP

4.3.11 UDP311: Deactivation by UDPA_DTI_REQ (higher layer)

Description:

The UDP entity receives the UDPA_DTI_REQ(UDPA_DISCONNECT_DTI) primitive for the higher layer.

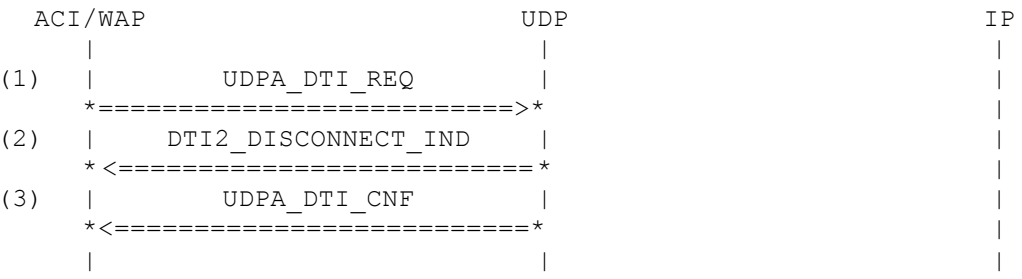
Variants: <A>..<C>

Preamble:

<A> UDP310B

<B>UDP422B

<C>UDP420B



Parametrization:

Primitive	Parameter	Value
(4) UDPA_DTI_REQ	dti_conn	UDPA_DISCONNECT_DTI
	entity_name	HL_NAME
	link_id	WAP_LINK_ID
	dti_direction	UDPA_DTI_TO_HIGHER_LAYER
(5) DTI2_DISCONNECT_IND	link_id	WAP_LINK_ID
	cause	DTI_CAUSE_NORMAL_CLOSE
(6) UDPA_DTI_CNF	dti_conn	UDPA_DISCONNECT_DTI
	link_id	WAP_LINK_ID

History:

28 June 2000	NI	initial
25 Oct 2002	KJF	new UDPA SAP

4.3.12 UDP312: Deactivation after reception of DTI2_DISCONNECT_IND

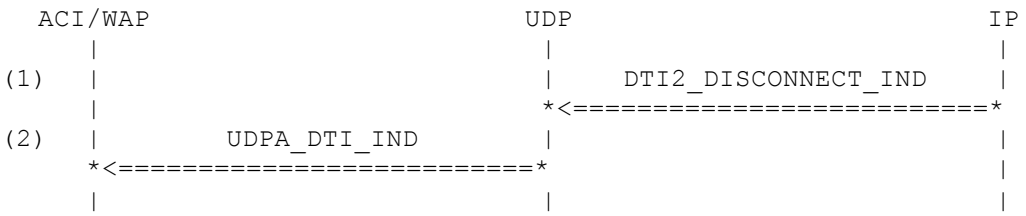
Description:

The UDP entity receives a DTI2_DISCONNECT_IND
UDP switches to deactivated state.

Variants: <A>..<B>

Preamble:

```
UDP201A /* Activation #1*/
UDP506  /* Outgoing UDP message with only
IP_ADDR_REQ */
```



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DISCONNECT_IND	link_id	IP_LINK_ID
	cause	
	DTI_CAUSE_NORMAL_CLOSE	
(2) UDPA_DTI_IND	link_id	IP_LINK_ID

History:

13 Nov 2001	TVO	initial
30-Oct-2002	KJF	new UDPA SAP

4.3.13 UDP313: Deactivation after reception of DTI2_DISCONNECT_REQ

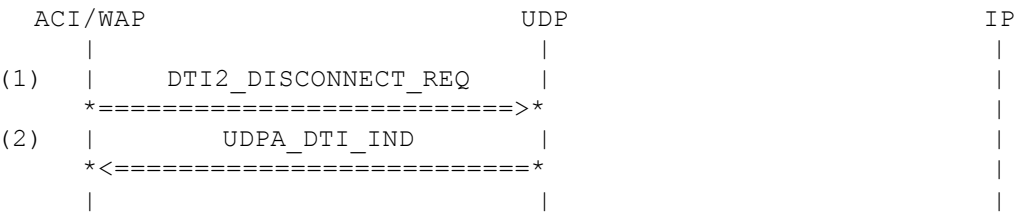
Description:

The UDP entity receives a DTI2_DISCONNECT_REQ.
UDP confirms its deactivation and switches to deactivated state.

Variants: <A>..<B>

Preamble:

```
UDP201A /* Activation #1*/
UDP506  /* Outgoing UDP message with only
IP_ADDR_REQ */
```



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DISCONNECT_REQ	link_id	WAP_LINK_ID
	cause	DTI_CAUSE_NORMAL_CLOSE
(2) UDPA_DTI_IND	link_id	WAP_LINK_ID

History:

13 Nov 2001	TVO	initial
25 Oct 2002	KJF	new UDPA SAP

4.4 Configured State

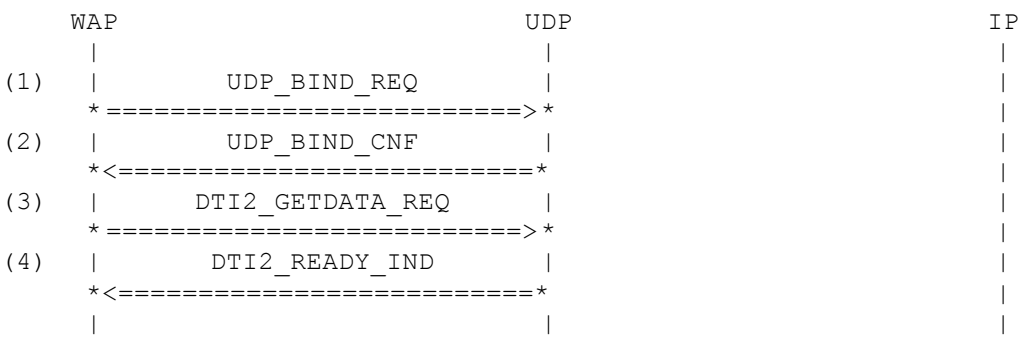
4.4.1 UDP401: Reception of UDP_BIND_REQ (any port)

Description:

(1) The UDP entity receives a UDP_BIND_REQ from an application with an unspecified port number (port field is zero).
(2) UDP binds the application to a free port and returns the port number in the UDP_BIND_CNF.
(3) The application requests data from the transport layer with DTI2_GETDATA_REQ.
(4) UDP indicates that it is ready to send data on behalf of the application with DTI2_READY_IND.

Preamble:

UDP301 /* Configuration #1 */



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	ZERO_PORT
(2) UDP_BIND_CNF	port	FIRST_FREE_PORT
	err	UDP_BIND_NOERROR
(3) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(4) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

30 June 2000	NI	initial

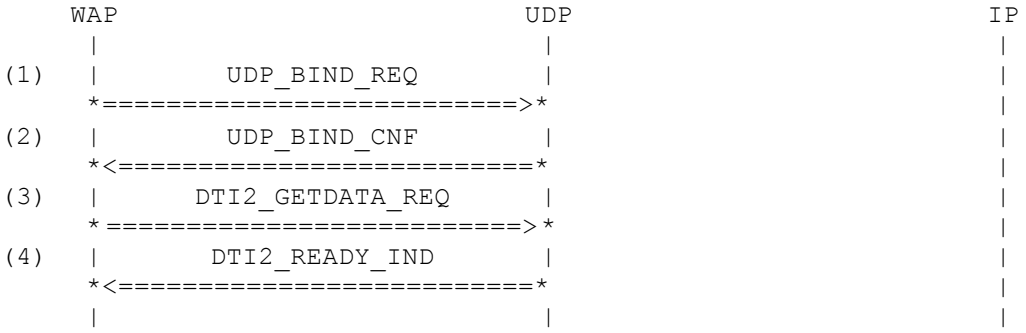
4.4.2 UDP402: Reception of UDP_BIND_REQ (fixed port, success)

Description:

(1) The UDP entity receives a UDP_BIND_REQ from an application (e. g. WAP) with a fixed port number.
(2) UDP binds the application to a the port and returns the port number in the UDP_BIND_CNF.
(3) The application requests data from the transport layer with DTI2_GETDATA_REQ.
(4) UDP indicates that it is ready to send data on behalf of the application with DTI2_READY_IND.

Preamble:

UDP301 /* Configuration #1 */



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	FIXED_PORT_1
(2) UDP_BIND_CNF	port	FIXED_PORT_1
	err	UDP_BIND_NOERROR
(3) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(4) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

30 June 2000	NI	initial

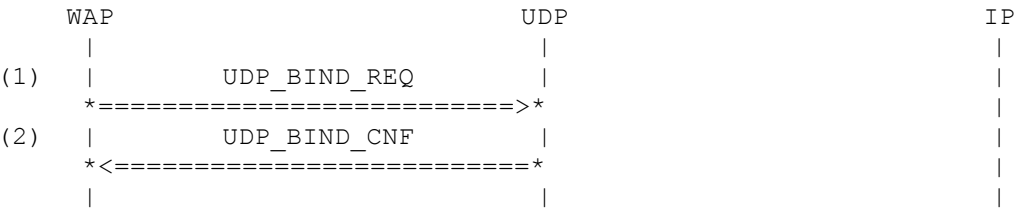
4.4.3 UDP403: Reception of UDP_BIND_REQ (fixed port, failure)

Description:

The UDP entity receives a UDP_BIND_REQ from an application (e. g. WAP) with a fixed port number that is already in use.
UDP sends a UDP_BIND_CNF with the err field set to UDP_BIND_PORTINUSE.

Preamble:

UDP402



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	FIXED_PORT_1
(2) UDP_BIND_CNF	port	FIXED_PORT_1
	err	UDP_BIND_PORTINUSE

History:

30 June 2000	NI	initial

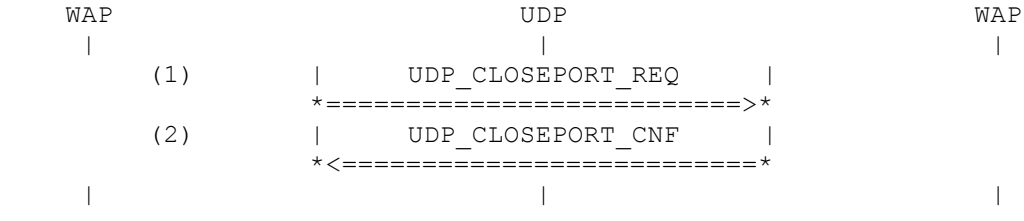
4.4.4 UDP404: Reception of UDP_CLOSEPORT_REQ

Description:

(1) The UDP entity receives a UDP_CLOSEPORT_REQ from an application.
(2) UDP confirms the request.

Preamble:

UDP401



Parametrization:

Primitive	Parameter	Value
(1) UDP_CLOSEPORT_REQ	port	FIRST_FREE_PORT
(2) UDP_CLOSEPORT_CNF		

History:

30 June 2000	NI	initial

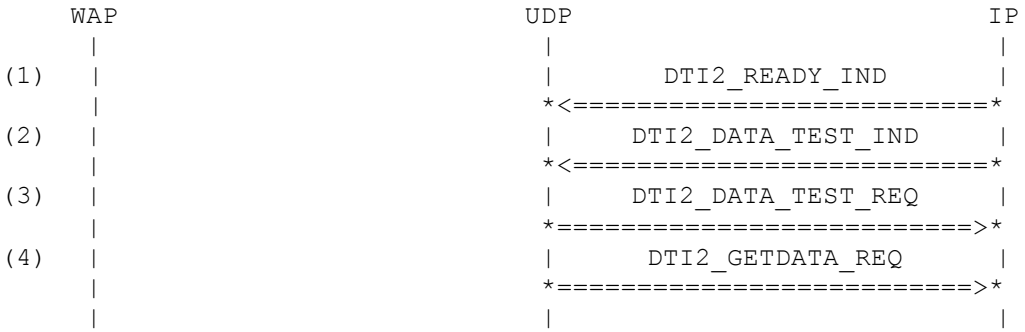
4.4.5 UDP405: UDP Message to an Unbound Port

Description:

(1) UDP receives a DTI2_READY_IND from IP.
(2) The UDP entity receives a UDP message destined to an unbound port.
(3) UPD sends an ICMP message (type Destination Unreachable, code Port Unreachable)
via DTI2_DATA_TEST_REQ.
(4) UDP requests the next data primitive from IP via DTI2_GETDATA_REQ.

Preamble:

UDP301 /* Configuration #1 */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_READY_IND	link_id	IP_LINK_ID
(2) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_305
(3) DTI2_DATA_TEST_REQ	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	PKT_ICMP_OUT_305
(4) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

30 June 2000	NI	initial
13 July 2000	NI	DTI2_READY_IND at the beginning added

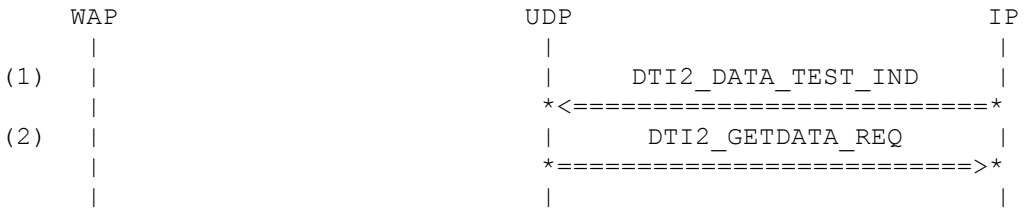
4.4.6 UDP406: UDP Message with Checksum Error

Description:

(1) The UDP entity receives a UDP message with checksum error from IP, which is silently discarded.
(2) UDP answers to IP with DTI2_GETDATA_REQ.

Preamble:

UDP301 /* Configuration #1 */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_306
(2) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

30 June 2000	NI	initial

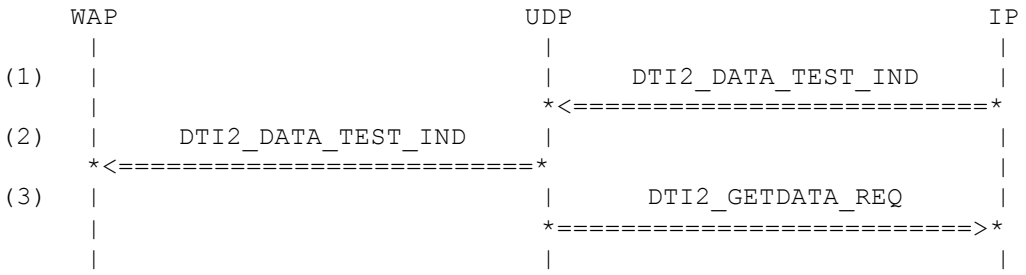
4.4.7 UDP407: UDP Message to a Bound Port

Description:

(1) The UDP entity receives a UDP message destined to a bound port.
(2) After appropriate processing, UDP relays the packet to WAP.
(3) UDP answers to IP with DTI2_GETDATA_REQ.

Preamble:

UDP412 /* Reception of UDP_BIND_REQ (fixed port - 2, success) */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_307
(2) DTI2_DATA_TEST_IND	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	SDU_UDP_IN_307
(3) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

30 June 2000	NI	initial
13 July 2000 added	NI	DTI2_GETDATA_REQ at the beginning
19 July 2000	xof	UDP301 as startcase
12 Nov 2001 the beginning, UDP412	TVO	removed DTI2_GETDATA_REQ from since it has already been sent within

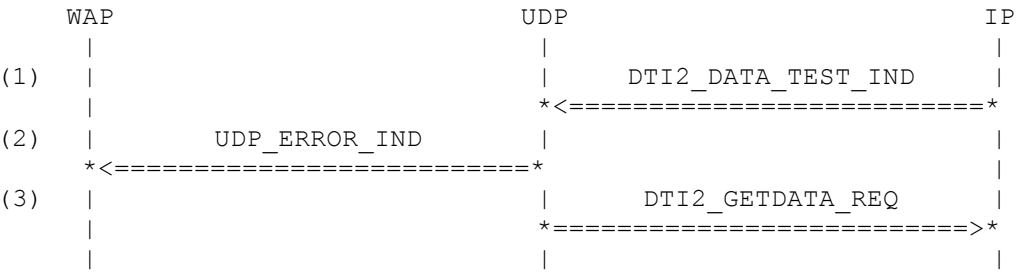
4.4.8 UDP408: Incoming ICMP error message

Description:

(1) The UDP entity receives an incoming ICMP error message from IP.
(2) UDP relays the error message in appropriate form to the concerning application (WAP).
(3) UDP answers to IP with DTI2_GETDATA_REQ.

Preamble:

UDP301



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_ICMP_OUT_308
(2) UDP_ERROR_IND	dst_port	FIXED_PORT_3
	src_port	FIXED_PORT_4
	err_msg	ERROR_MSG_NUMBER
	src_addr	PEER_IP_ADDRESS
	dst_addr	OWN_IP_ADDRESS
(3) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

30 June 2000 NI initial

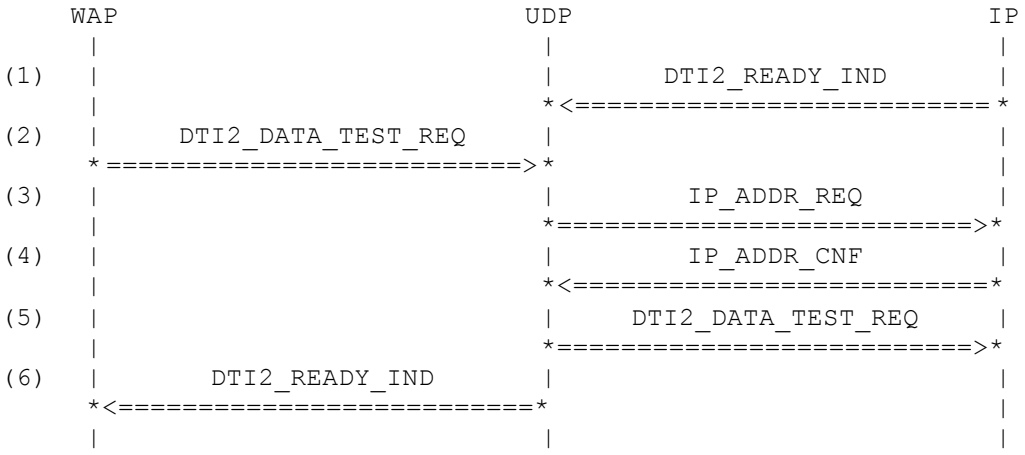
4.4.9 UDP409: Outgoing UDP message (success)

Description:

(1) UDP receives a DTI2_READY_IND from IP.
(2) The UDP entity receives a UDP message from WAP via DTI2_DATA_TEST_REQ
(3) UDP sends an IP_ADDR_REQ to IP to determine the outgoing source address.
(4) IP sends an IP_ADDR_CNF with the source IP address
(5) UDP sends the outgoing message to IP with DTI2_DATA_TEST_REQ
(6) UDP indicates to WAP with DTI2_READY_IND that it is ready to accept the next message.

Preamble:

UDP401 /* Reception of UDP_BIND_REQ (any port) */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_READY_IND	link_id	IP_LINK_ID
(2) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_1_309
(3) IP_ADDR_REQ	dst_addr	PEER_IP_ADDRESS
	trans_prot	UDP_PROTOCOL
(4) IP_ADDR_CNF	src_addr	OWN_IP_ADDRESS
	err	IP_ADDR_NOERROR
	trans_prot	UDP_PROTOCOL
(5) DTI2_DATA_TEST_REQ	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	PKT_UDP_OUT_2_309
(6) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

30 June 2000	NI	initial
13 July 2000	NI	DTI2_READY_IND at the beginning added
19 July 2000	xof	Frame corrected
12 Nov 2001	TVO	changed preamble from UDP301 to UDP301

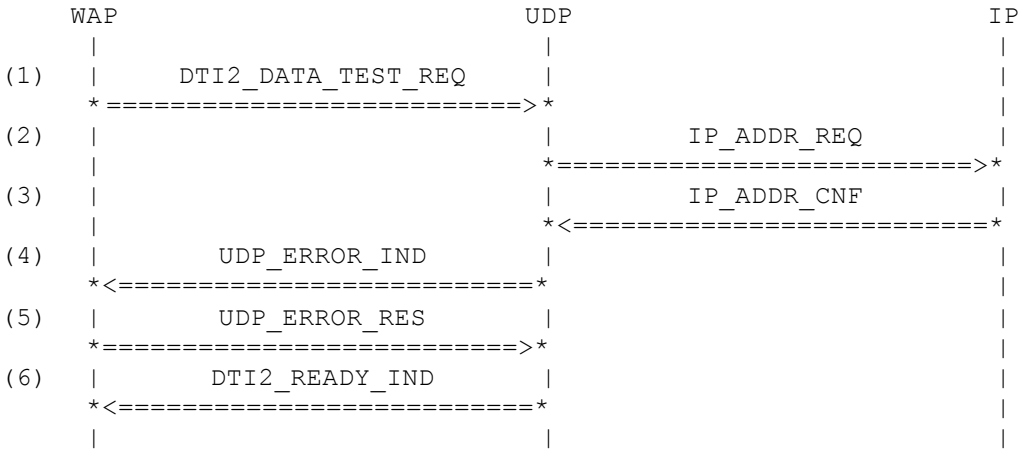
4.4.10 UDP410: Outgoing UDP message (no route to destination)

Description:

(1) The UDP entity receives an UDP message with an invalid destination address from WAP via DTI2_DATA_TEST_REQ.
(2) UDP sends an IP_ADDR_REQ to IP to determine the outgoing source address.
(3) IP sends an IP_ADDR_CNF with the error IP_ADDR_NOROUTE.
(4) UDP sends an appropriate error indication to WAP
(5) WAP responds to the error indication.
(6) UDP indicates to WAP with DTI2_READY_IND that it is ready to accept the next message.

Preamble:

UDP401 /* Reception of UDP_BIND_REQ (any port) */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_310
(2) IP_ADDR_REQ	dst_addr	INVALID_IP_ADDRESS
	trans_prot	UDP_PROTOCOL
(3) IP_ADDR_CNF	src_addr	OWN_IP_ADDRESS
	err	IP_ADDR_NOROUTE
	trans_prot	UDP_PROTOCOL
(4) UDP_ERROR_IND	dst_port	FIXED_PORT_5
	src_port	FIXED_PORT_2
	err_msg	IP_ADDR_NOROUTE
	src_addr	OWN_IP_ADDRESS
	dst_addr	INVALID_IP_ADDRESS
(5) UDP_ERROR_RES		
(4) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

30 June 2000	NI	initial
12 Nov 2001	TVO	changed preamble from UDP301 to UDP301

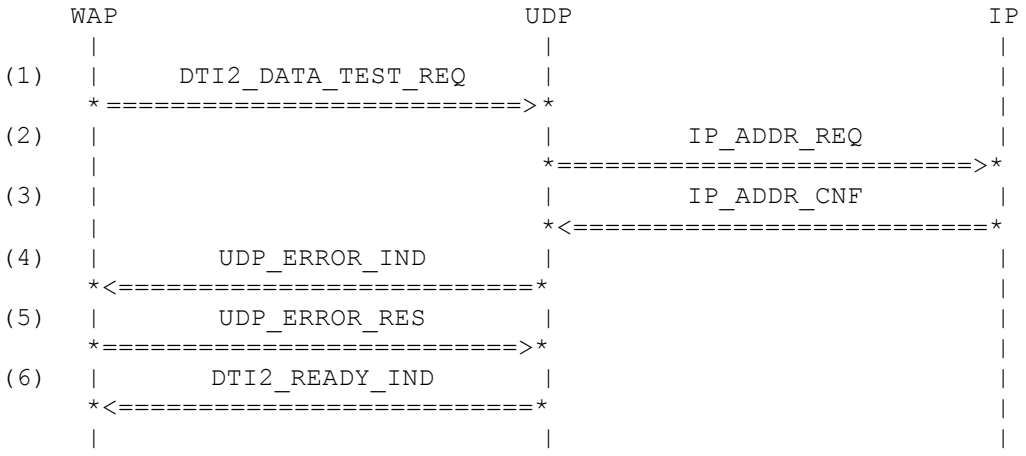
4.4.11 UDP411: Outgoing UDP message (source address conflict)

Description:

(1) The UDP entity receives an UDP message with an invalid source address from WAP via DTI2_DATA_TEST_REQ.
(2) UDP sends an IP_ADDR_REQ to IP to determine the outgoing source address.
(3) IP sends an IP_ADDR_CNF with the source address.
(4) UDP sends an error indication to WAP with the correct source address. (unequal to the address in the DTI2_DATA_TEST_REQ).
(5) WAP responds to the error indication.
(6) UDP indicates to WAP with DTI2_READY_IND that it is ready to accept the next message.

Preamble:

UDP401 /* Reception of UDP_BIND_REQ (any port) */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_311
(2) IP_ADDR_REQ	dst_addr	PEER_IP_ADDRESS
	trans_prot	UDP_PROTOCOL
(3) IP_ADDR_CNF	src_addr	OWN_IP_ADDRESS
	err	IP_ADDR_NOERROR
	trans_prot	UDP_PROTOCOL
(4) UDP_ERROR_IND	dst_port	WAP_PORT
	src_port	FIXED_PORT_5
	err_msg	IP_ADDR_NOROUTE
	src_addr	OWN_IP_ADDRESS
	dst_addr	PEER_IP_ADDRESS
(5) UDP_ERROR_RES		
(6) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

30 June 2000	NI	initial

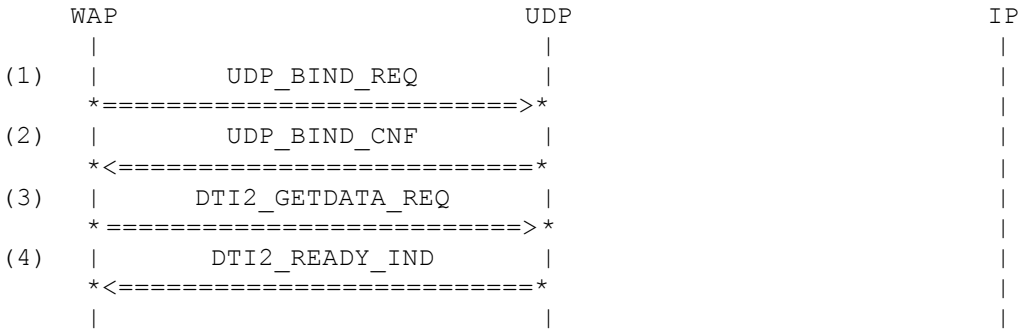
4.4.12 UDP412: Reception of UDP_BIND_REQ (fixed port - 2, success)

Description:

(1) The UDP entity receives a UDP_BIND_REQ from an application (e. g. WAP) with a fixed port number.
(2) UDP binds the application to a the port and returns the port number in the UDP_BIND_CNF.

Preamble:

```
UDP301 /* Configuration #1 */
```



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	FIXED_PORT_2
(2) UDP_BIND_CNF	port	FIXED_PORT_2
	err	UDP_BIND_NOERROR
(3) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(4) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

02 September 2000	OFL	initial

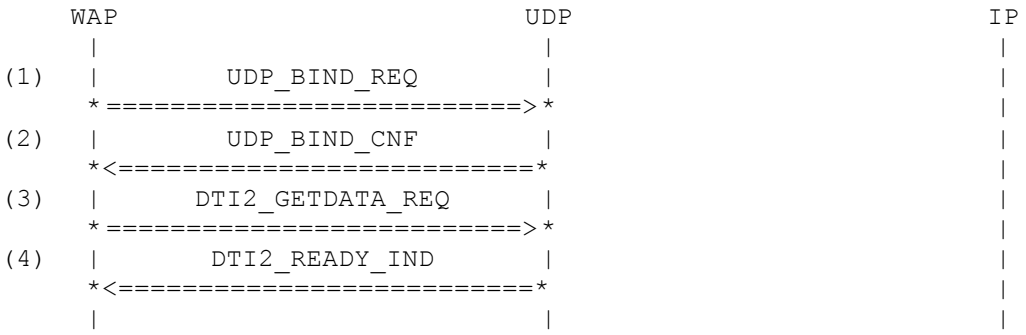
4.4.13 UDP413: Reception of UDP_BIND_REQ – port number 6

Description:

The UDP entity receives a UDP_BIND_REQ from an application (e. g. WAP) with a fixed port number.
UDP binds the application to a free port and returns the port number in the UDP_BIND_CNF.
The application requests data from the transport layer with DTI2_GETDATA_REQ.
UDP indicates that it is ready to send data on behalf of the application with DTI2_READY_IND.

Preamble:

UDP301 /* Configuration #1 */



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	FIXED_PORT_6
(2) UDP_BIND_CNF	port	FIXED_PORT_6
	err	UDP_BIND_NOERROR
(3) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(4) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

30 June 2000	NI	initial

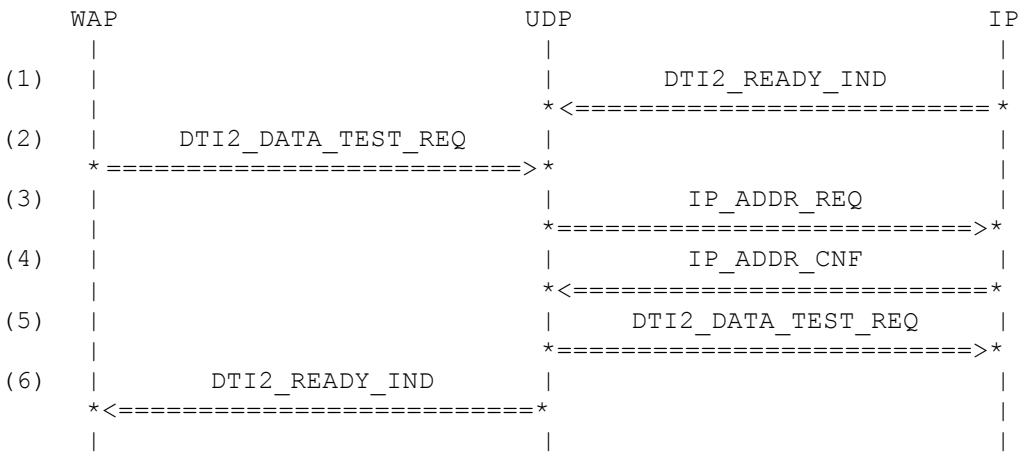
4.4.14 UDP414: Outgoing UDP message (success), UDP packet from Linux workstation

Description:

(1) UDP receives a DTI2_READY_IND from IP.
(2) The UDP entity receives a UDP message from WAP via DTI2_DATA_TEST_REQ
(3) UDP sends an IP_ADDR_REQ to IP to determine the outgoing source address.
(4) IP sends an IP_ADDR_CNF with the source IP address
(5) UDP sends the outgoing message to IP with DTI2_DATA_TEST_REQ
(6) UDP indicates to WAP with DTI2_READY_IND that it is ready to accept the next message.

Preamble:

```
UDP413 /* Reception of UDP_BIND_REQ – port number 6 */
```



Parametrization:

Primitive	Parameter	Value
(1) DTI2_READY_IND	link_id	IP_LINK_ID
(2) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	
	WAP_BRODCAST_DST_ADDR	
(3) IP_ADDR_REQ	dst_addr	DST_BROADCAST_ADDR
	trans_prot	UDP_PROTOCOL
(4) IP_ADDR_CNF	src_addr	OWN_IP_ADDRESS_1
	err	IP_ADDR_NOERROR
	trans_prot	UDP_PROTOCOL
(5) DTI2_DATA_TEST_REQ	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	
	UDP_IP_BRODCAST_DST_ADDR	
(6) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

01 October 2000	xof	Initial

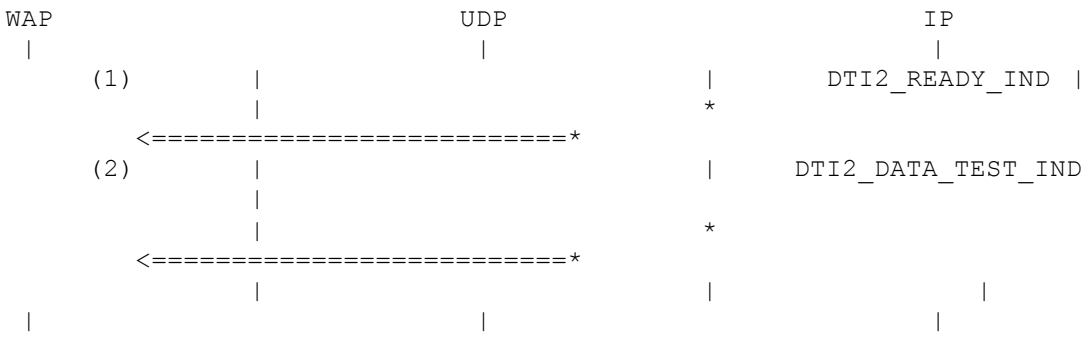
4.4.15 UDP415: UDP Message to an Unbound Port, prepare send ICMP message

Description:

UDP receives a DTI2_READY_IND from IP.
The UDP entity receives a UDP message destined to an unbound port.
UDP set up a ICMP message: port unreachable

Preamble:

UDP301 /* Configuration #1 */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_READY_IND	link_id	IP_LINK_ID
(2) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_305

History:

30 June 2000	NI	initial
13 July 2000	NI	DTI2_READY_IND at the beginning added

4.4.16 UDP416: Reception of UDP_BIND_REQ – port number 2

Description:

The UDP entity receives a UDP_BIND_REQ from an application (e. g. WAP) with a fixed port number.
UDP binds the application to a the port and returns the port number in the UDP_BIND_CNF.
The application requests data from the transport layer with DTI2_GETDATA_REQ.
UDP indicates that it is ready to send data on behalf of the application with DTI2_READY_IND.

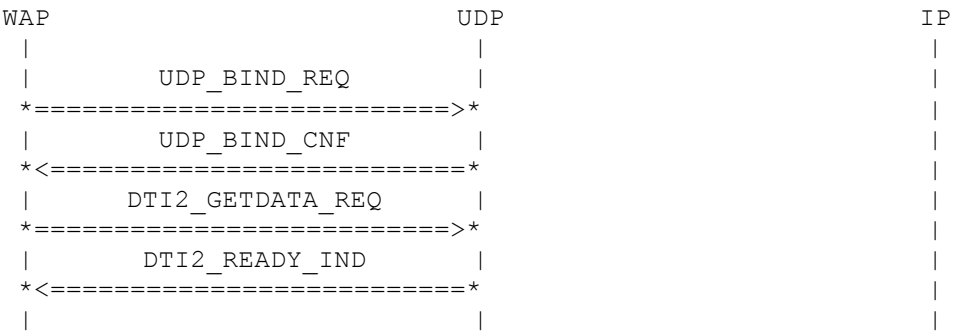
Variants: <A>..<C>

Preamble:

<A>UDP302

<B>UDP303B

<C>UDP303C



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	FIXED_PORT_2
(2) UDP_BIND_CNF	port	FIXED_PORT_2
	err	UDP_BIND_NOERROR
(3) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(4) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

02 October 2000	OFL	initial
15 November 2001	TVO	added variants

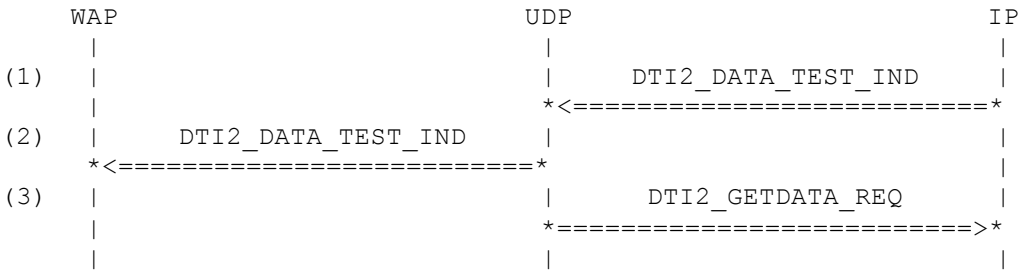
4.4.17 UDP417: UDP Message to a Bound Port

Description:

The UDP entity receives a UDP message destined to a bound port.
After appropriate processing, UDP relays the packet to WAP.
UDP answers to IP with DTI2_GETDATA_REQ.

Preamble:

UDP416A /* Reception of UDP_BIND_REQ – port number 2 */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_307
(2) DTI2_DATA_TEST_IND	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	SDU_UDP_IN_307
(3) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

02 October 2000	OFL	initial
12 Nov 2001	TVO	removed DTI2_GETDATA_REQ from WAP to avoid sending of two DTI2_GETDATA_REQ in a line (UDP319/UDP320)

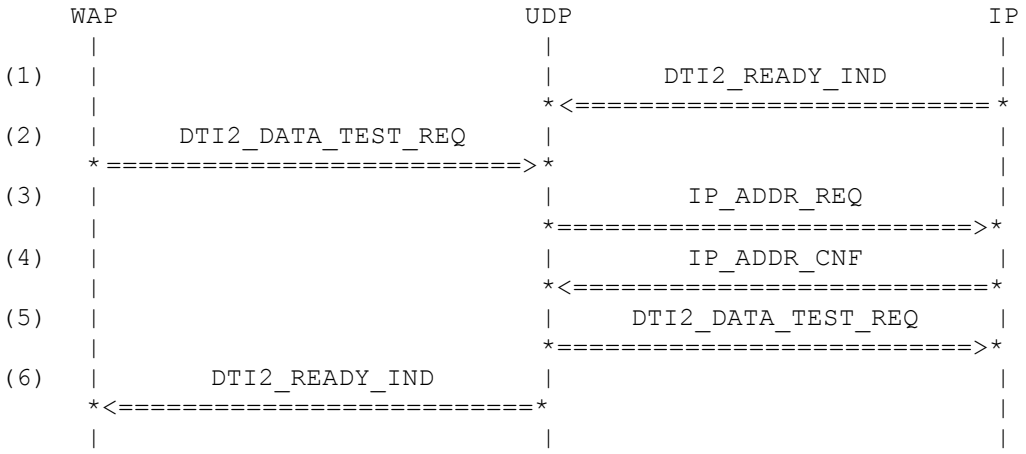
4.4.18 UDP418: Outgoing UDP message (success)

Description:

UDP receives a DTI2_READY_IND from IP.
The UDP entity receives a UDP message from WAP via DTI2_DATA_TEST_REQ
UDP sends an IP_ADDR_REQ to IP to determine the outgoing source address.
IP sends an IP_ADDR_CNF with the source IP address
UDP sends the outgoing message to IP with DTI2_DATA_TEST_REQ
UDP indicates to WAP with DTI2_READY_IND that it is ready to accept the next message.

Preamble:

UDP417



Parametrization:

Primitive	Parameter	Value
(1) DTI2_READY_IND	link_id	IP_LINK_ID
(2) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_1_309
(3) IP_ADDR_REQ	dst_addr	PEER_IP_ADDRESS
	trans_prot	UDP_PROTOCOL
(4) IP_ADDR_CNF	src_addr	OWN_IP_ADDRESS
	err	IP_ADDR_NOERROR
	trans_prot	UDP_PROTOCOL
(5) DTI2_DATA_TEST_REQ	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	PKT_UDP_OUT_2_309
(6) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

02 October 2000	OFL	initial

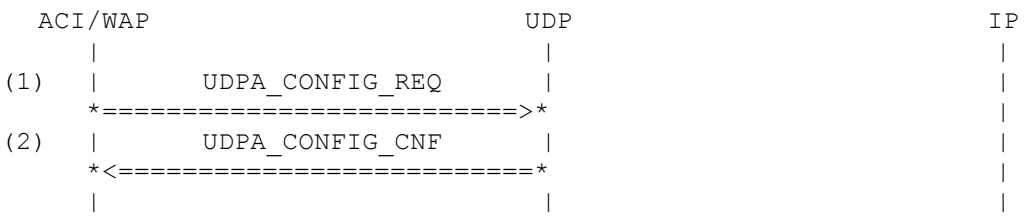
4.4.19 UDP419: Disconnection by UDPA_CONFIG_REQ

Description:

The UDP entity receives UDPA_CONFIG_REQ(UDPA_CONFIG_DOWN).
UDP confirms its deactivation and switches to deactivated state.

Preamble:

UDP301 /* Configuration #1 */



Parametrization:

Primitive	Parameter	Value
(1) UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_DOWN
(2) UDPA_CONFIG_CNF		

History:

28 June 2000	NI	initial
25 Oct 2002	KJF	new UDPA SAP

4.4.20 UDP420: Disconnection by UDPA_CONFIG_REQ – Port bound

Description:

The UDP entity receives UDPA_CONFIG_REQ(UDPA_CONFIG_DOWN).
UDP indicates its deactivation to WAP, which is bound to a port.
UDP confirms its deactivation and switches to deactivated state.

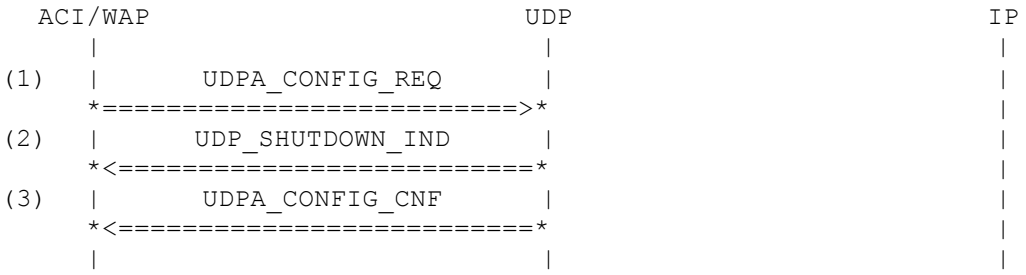
Variants: <A>..<B>

Preamble:

<A>UDP401
<B>UDP506

Preamble:

UDP401 /* Reception of UDP_BIND_REQ (any port) */



Parametrization:

Primitive	Parameter	Value
(1) UDPA_CONFIG_REQ	cmd	UDPA_CONFIG_DOWN
(2) UDP_SHUTDOWN_IND		
(3) UDPA_CONFIG_CNF		

History:

28 June 2000	NI	initial
25 Oct 2002	KJF	new UDPA SAP

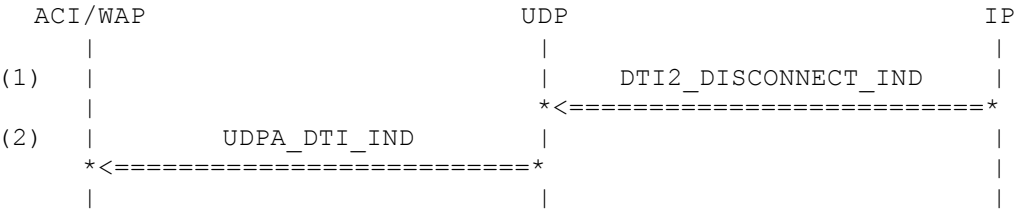
4.4.21 UDP421: Deactivation after reception of DTI2_DISCONNECT_IND

Description:

The UDP entity receives a DTI2_DISCONNECT_IND.
UDP switches to deactivated state.

Preamble:

UDP424A /* Deactivation after reception of DTI2_DISCONNECT_REQ – Port bound */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DISCONNECT_IND	link_id	IP_LINK_ID
	cause	DTI_CAUSE_NORMAL_CLOSE
(2) UDPA_DTI_IND	link_id	IP_LINK_ID

History:

13 Nov 2001	TVO	initial
30-Oct-2002	KJF	new UDPA SAP

4.4.22 UDP422: Deactivation after reception of DTI2_DISCONNECT_IND – Port bound

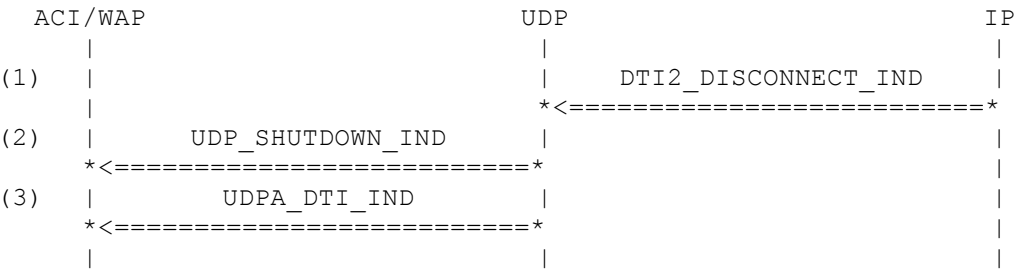
Description:

The UDP entity receives a DTI2_DISCONNECT_IND
UDP indicates its deactivation to WAP, which is bound to a port.
UDP switches to deactivated state.

Variants: <A>..<B>

Preamble:

<A> UDP401 /* Reception of UDP_BIND_REQ (any port) */
<B> UDP506 /* Outgoing UDP message with only IP_ADDR_REQ */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DISCONNECT_IND	link_id	IP_LINK_ID
	cause	
	DTI_CAUSE_NORMAL_CLOSE	
(2) UDP_SHUTDOWN_IND		
(3) UDPA_DTI_IND	link_id	IP_LINK_ID

History:

13 Nov 2001	TVO	initial
30-Oct-2002	KJF	new UDPA SAP

4.4.23 UDP423: Deactivation after reception of DTI2_DISCONNECT_REQ

Description:

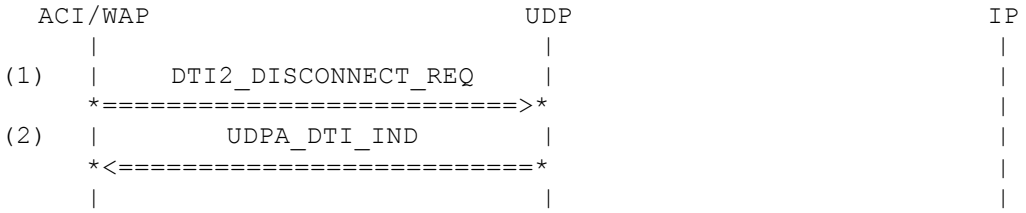
The UDP entity receives a DTI2_DISCONNECT_REQ.
UDP confirms its deactivation and switches to deactivated state.
In variant B, UDPA is deactivated before an IP_ADDR_CNF is received

Variants: <A>..<B>

Preamble:

```
<A> UDP301 /* Configuration #1 */
<B> UDP422A /* Deactivation after reception of DTI2_DISCONNECT_IND –
Port bound */
```

Preamble:



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DISCONNECT_REQ	link_id	WAP_LINK_ID
	cause	DTI_CAUSE_NORMAL_CLOSE
(2) UDPA_DTI_IND	link_id	WAP_LINK_ID

History:

13 Nov 2001	TVO	initial – adapted from UDP300

4.4.24 UDP424: Deactivation after reception of DTI2_DISCONNECT_REQ – Port bound

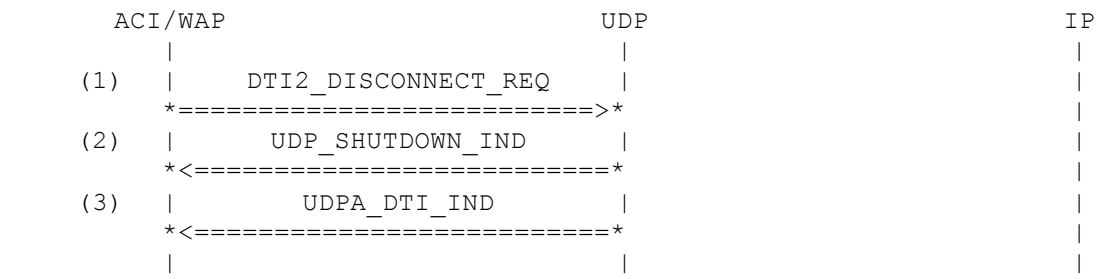
Description:

The UDP entity receives a DTI2_DISCONNECT_REQ, followed by a UDPA_DEACTIVATE_REQ
UDP indicates its deactivation to WAP, which is bound to a port.
UDP confirms its deactivation and switches to deactivated state.

Variants: <A>..<B>

Preamble:

<A> UDP401 /* Reception of UDP_BIND_REQ (any port) */
<B> UDP506 /* Outgoing UDP message with only IP_ADDR_REQ */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DISCONNECT_REQ	link_id	WAP_LINK_ID
	cause	
	DTI_CAUSE_NORMAL_CLOSE	
(2) UDP_SHUTDOWN_IND		
(3) UDPA_DTI_IND	link_id	WAP_LINK_ID

History:

13 Nov 2001	TVO	initial – adapted from UDP300

4.5 Fault tolerance

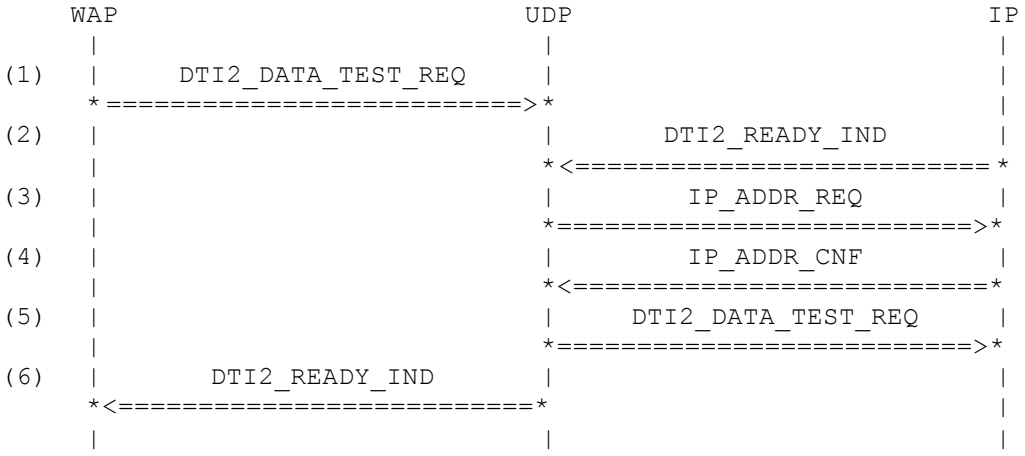
4.5.1 UDP501: Outgoing UDP message with DTI2_READY_IND after reception of DTI2_DATA_TEST_REQ

Description:

UDP receives a DTI2_READY_IND from IP.
The UDP entity receives a UDP message from WAP via DTI2_DATA_TEST_REQ
UDP sends an IP_ADDR_REQ to IP to determine the outgoing source address.
IP sends an IP_ADDR_CNF with the source IP address
UDP sends the outgoing message to IP with DTI2_DATA_TEST_REQ
UDP indicates to WAP with DTI2_READY_IND that it is ready to accept the next message.

Preamble:

UDP418 /* Outgoing UDP message (success) */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_1_309
(2) DTI2_READY_IND	link_id	IP_LINK_ID
(3) IP_ADDR_REQ	dst_addr	PEER_IP_ADDRESS
	trans_prot	UDP_PROTOCOL
(4) IP_ADDR_CNF	src_addr	OWN_IP_ADDRESS
	err	IP_ADDR_NOERROR
	trans_prot	UDP_PROTOCOL
(5) DTI2_DATA_TEST_REQ	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	PKT_UDP_OUT_2_309
(6) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

02 October 2000	OFL	initial

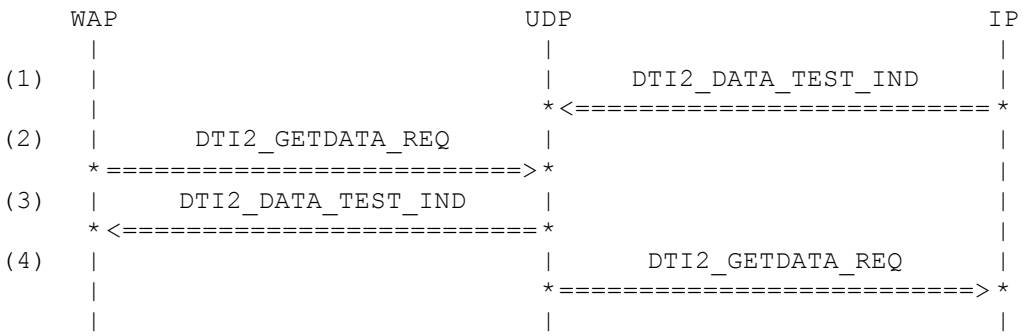
4.5.2 UDP502: UDP Message to a Bound Port, get DTI2_GETDATA_REQ after DTI2_DATA_TEST_IND

Description:

The UDP entity receives a UDP message destined to a bound port.
After appropriate processing, UDP relays the packet to WAP.
UDP answers to IP with DTI2_GETDATA_REQ.

Preamble:

UDP501 /* Outgoing UDP message with DTI2_READY_IND after reception of DTI2_DATA_TEST_REQ */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_307
(2) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(3) DTI2_DATA_TEST_IND	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	SDU_UDP_IN_307
(4) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

02 October 2000 OFL initial

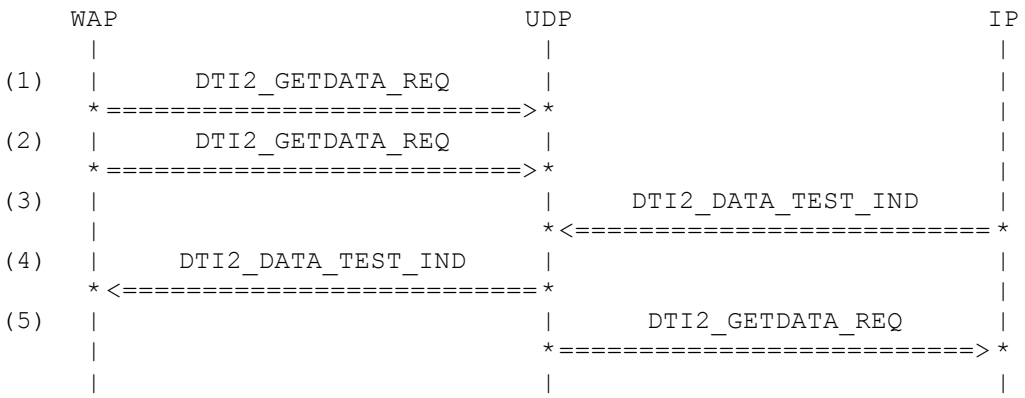
4.5.3 UDP503: Receiving two DTI2_GETDATA_REQ after DTI2_DATA_TEST_IND

Description:

Receiving two DTI2_GETDATA_REQ.
Ignore the second.
After appropriate processing, UDP relays the packet to WAP.
UDP answers to IP with DTI2_GETDATA_REQ.

Preamble:

UDP501 /* Outgoing UDP message with DTI2_READY_IND after reception of DTI2_DATA_TEST_REQ */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(2) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(3) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_307
(4) DTI2_DATA_TEST_IND	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	SDU_UDP_IN_307
(5) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

02 October 2000	OFL	initial

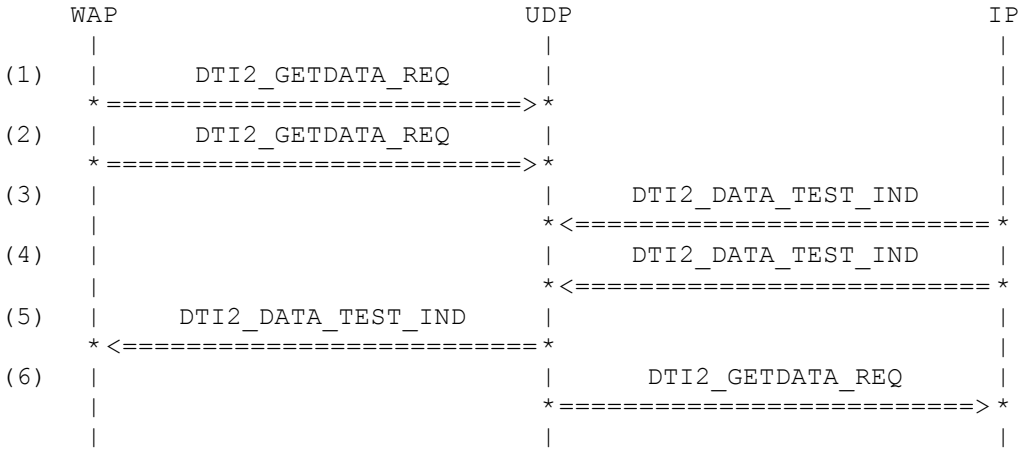
4.5.4 UDP504: Receiving two DTI2_GETDATA_REQ and two DTI2_DATA_TEST_IND

Description:

Receiving two DTI2_DATA_TEST_IND.
Ignore the second.
After appropriate processing, UDP relays the packet to WAP.
UDP answers to IP with DTI2_GETDATA_REQ.

Preamble:

UDP503



Parametrization:

Primitive	Parameter	Value
(1) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(2) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(3) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_307
(4) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	PKT_UDP_IN_307
(5) DTI2_DATA_TEST_IND	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	SDU_UDP_IN_307
(6) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

02 October 2000 OFL initial

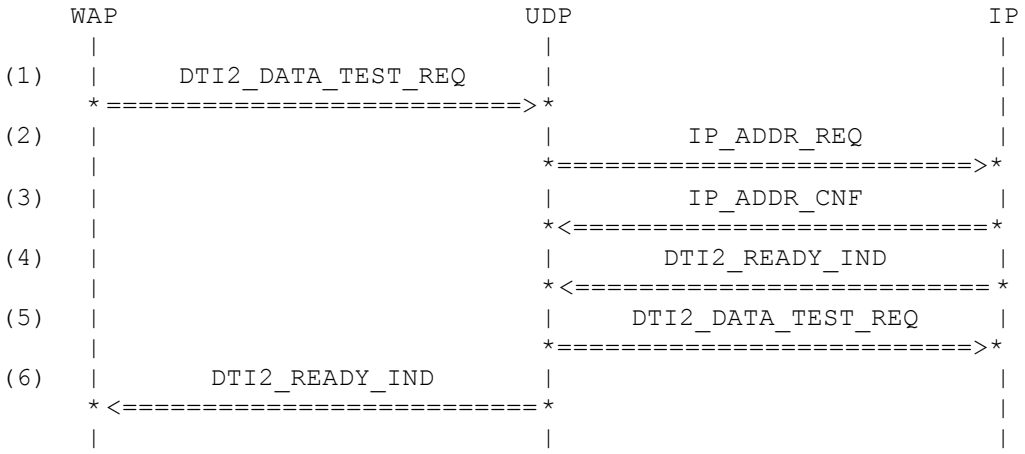
4.5.5 UDP505: Outgoing UDP message with receiving a DTI2_DATA_TEST_REQ

Description:

UDP receives a DTI2_DATA_TEST_REQ.
After receiving the DTI2_READY_IND, UDP sends the UDP packet to the lower entity.

Preamble:

UDP504



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_1_309
(2) IP_ADDR_REQ	dst_addr	PEER_IP_ADDRESS
	trans_prot	UDP_PROTOCOL
(3) IP_ADDR_CNF	src_addr	OWN_IP_ADDRESS
	err	IP_ADDR_NOERROR
	trans_prot	UDP_PROTOCOL
(4) DTI2_READY_IND	link_id	IP_LINK_ID
(5) DTI2_DATA_TEST_REQ	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	PKT_UDP_OUT_2_309
(6) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

02 October 2000 OFL initial

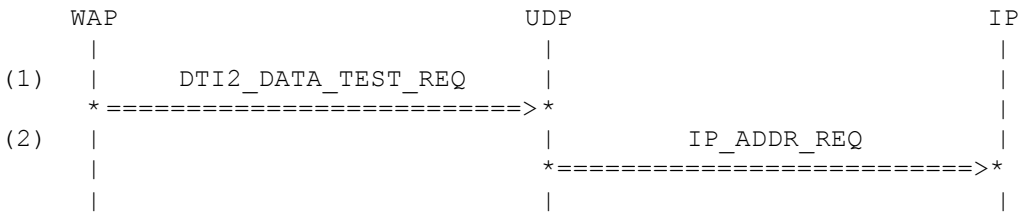
4.5.6 UDP506: Outgoing UDP message with IP_ADDR_REQ only

Description:

UDP receives a outgoing massage and a IP_ADDR_REQ.

Preamble:

UDP501 /* Outgoing UDP message with DTI2_READY_IND after reception of DTI2_DATA_TEST_REQ */



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_1_309
(2) IP_ADDR_REQ	dst_addr	PEER_IP_ADDRESS
	trans_prot	UDP_PROTOCOL

History:

02 October 2000	OFL	initial

4.5.7 UDP507: Reception of UDP_BIND_REQ

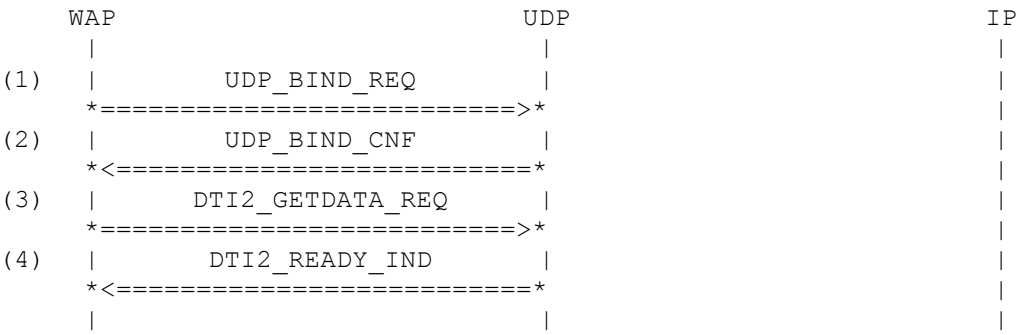
Description:

The UDP entity receives a UDP_BIND_REQ from an application (e. g. WAP) with a fixed port number.
UDP binds the application to a the port and returns the port number in the UDP_BIND_CNF.
The application requests data from the transport layer with DTI2_GETDATA_REQ.
UDP indicates that it is ready to send data on behalf of the application with DTI2_READY_IND.

Variants: <A>..<C>

Preamble:

<A>UDP303A
<B>UDP303B
<C>UDP303C



Parametrization:

Primitive	Parameter	Value
(1) UDP_BIND_REQ	port	FIXED_PORT_2
(2) UDP_BIND_CNF	port	FIXED_PORT_2
	err	UDP_BIND_NOERROR
(3) DTI2_GETDATA_REQ	link_id	WAP_LINK_ID
(4) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

02 October 2000	OFL	initial

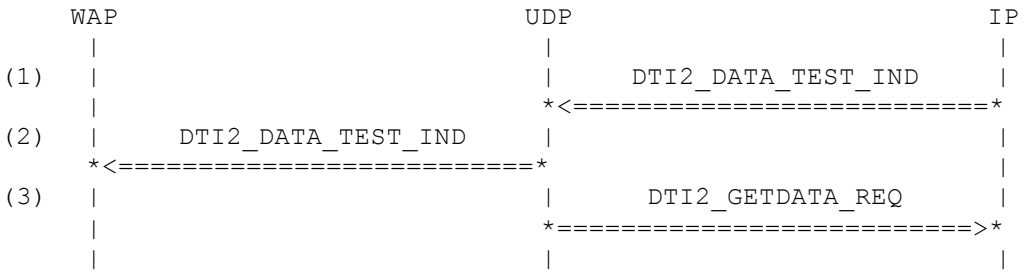
4.5.8 UDP508: UDP Message to a Bound Port

Description:

The UDP entity receives a UDP message destined to a bound port.
After appropriate processing, UDP relays the packet to WAP.
UDP answers to IP with DTI2_GETDATA_REQ.

Preamble:

UDP507A



Parametrization:

Primitive	Parameter	Value
(1) DTI2_DATA_TEST_IND	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_IP
	sdu	PKT_UDP_IN_307
(2) DTI2_DATA_TEST_IND	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	SDU_UDP_IN_307
(3) DTI2_GETDATA_REQ	link_id	IP_LINK_ID

History:

02 October 2000	OFL initial
12 November 2001	TVO DTI2_GETDATA_REQ removed due to duplicity.

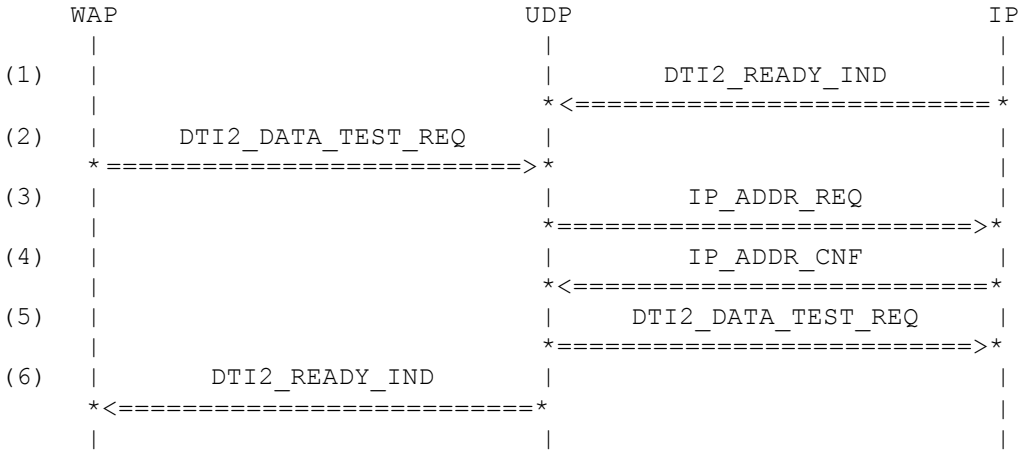
4.5.9 UDP509: Outgoing UDP message (success)

Description:

UDP receives a DTI2_READY_IND from IP.
The UDP entity receives a UDP message from WAP via DTI2_DATA_TEST_REQ.
UDP sends an IP_ADDR_REQ to IP to determine the outgoing source address.
IP sends an IP_ADDR_CNF with the source IP address
UDP sends the outgoing message to IP with DTI2_DATA_TEST_REQ
UDP indicates to WAP with DTI2_READY_IND that it is ready to accept the next message.

Preamble:

UDP508



Parametrization:

Primitive	Parameter	Value
(1) DTI2_READY_IND	link_id	IP_LINK_ID
(2) DTI2_DATA_TEST_REQ	link_id	WAP_LINK_ID
	parameters	DTI_PARAMETER_WAP
	sdu	SDU_UDP_OUT_1_309
(3) IP_ADDR_REQ	dst_addr	PEER_IP_ADDRESS
	trans_prot	UDP_PROTOCOL
(4) IP_ADDR_CNF	src_addr	OWN_IP_ADDRESS
	err	IP_ADDR_NOERROR
	trans_prot	UDP_PROTOCOL
(5) DTI2_DATA_TEST_REQ	link_id	IP_LINK_ID
	parameters	DTI_PARAMETER_UDP
	sdu	PKT_UDP_OUT_2_309
(6) DTI2_READY_IND	link_id	WAP_LINK_ID

History:

02 October 2000	OFL	initial

Appendices

A. Acronyms

DS-WCDMA Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000) Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: http://www.imt-2000.org/>