Technical Document

LLD DTI CONNECTION MANAGER

Document Number:	8462.703.02.002
Version:	0.3
Status:	Draft
Approval Authority:	
Creation Date:	2002-Sep-10
Last changed:	2015-Mar-08 by XGUTTEFE
File Name:	LLD_DTI_Conn_Mng.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
2002-Sep-10	TLU		0.1	Planned	1
2003-Mar-12	TLU		0.2	Planned	
2003-May-19	XGUTTEFE		0.3	Draft	

Notes:

1. Initial version

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

A. References, Abbreviations, Terms

[TI 7010.801] 7010.801, References and Vocabulary, Texas Instruments.

1 Introduction

This document is a Low Level design for the DTI Manager, which is using DTI LIB Version 2. The document describes the establishing and releasing of DTI channels. It is assumed that the reader has a basic understanding about the Data Transmission Interface (DTI).

2 Overview

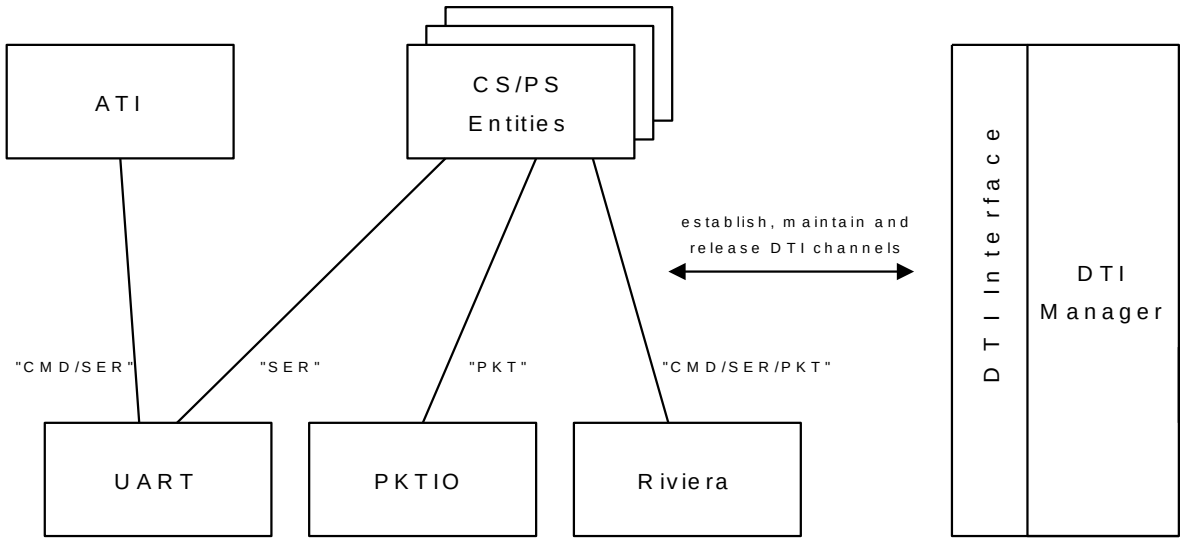

Figure 1: Overview DTI Manager and entities

3 DTI Manager

3.1 Common Design

For a clearer design the DTI Manager is divided in two parts, DTI Control Manger and DTI Connection Manager.

New AT commands will be introduced to configure the DTI channels from the extern side. Therefore an additional command handler will be added to handle these AT commands. A detailed description of the new AT commands can be found in an additional document.

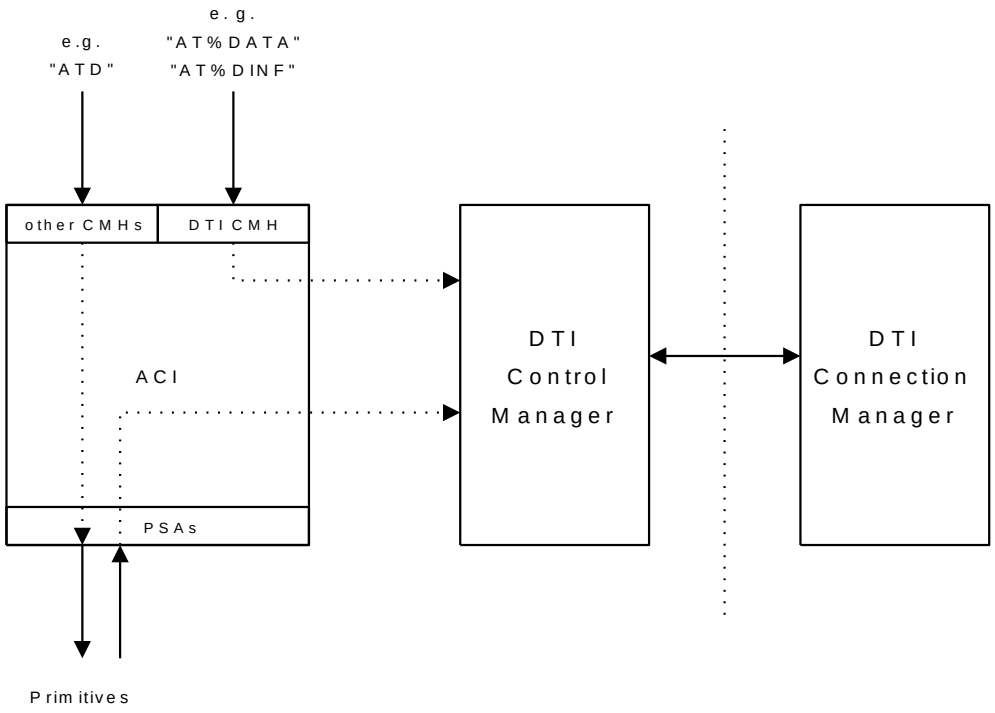

Figure 2: Division of the DTI Manager

3.2 DTI Control Manager

The DTI Control Manager manages the port- and profile data base. It has also knowledge about the ACI source Ids. This unit is a fixed part of the ACI. Detailed information about the DTI Control Manager can be found in an additional document.

3.3 DTI Connection Manager

The DTI Connection Manager is responsible for providing information about all available DTI channels and their connection states. It is also responsible for the establishing and releasing of DTI channels.

3.3.1 Connection States

Every end-to-end connection, every entity-to-entity connection (tuple connection) and every peer has one of the following states.

- CONNECTING : one entity has confirmed the DTI-connect request to the DTI Manager
- CONNECTED : both entities have confirmed the DTI-connect request to the DTI Manager
- DISCONNECTING : one entity has confirmed the DTI-disconnect request to the DTI Manager
- DISCONNECTED : both entities have confirmed the DTI-disconnect request to the DTI Manager

The following figures describe the state transitions of the DTI Connection Manager.

3.3.2 Establish a DTI connection

The following figure describes the establishing of an end-to-end DTI connection (e.g. UART-PPP-SNDCP). An end-to-end connection may consist of multiple tuple (entity-to-entity) connection (e.g. UART-PPP, PPP-SNDCP). The connection state of the end-to-end connection is CONNECTED when all tuple connections are in the state CONNECTED.

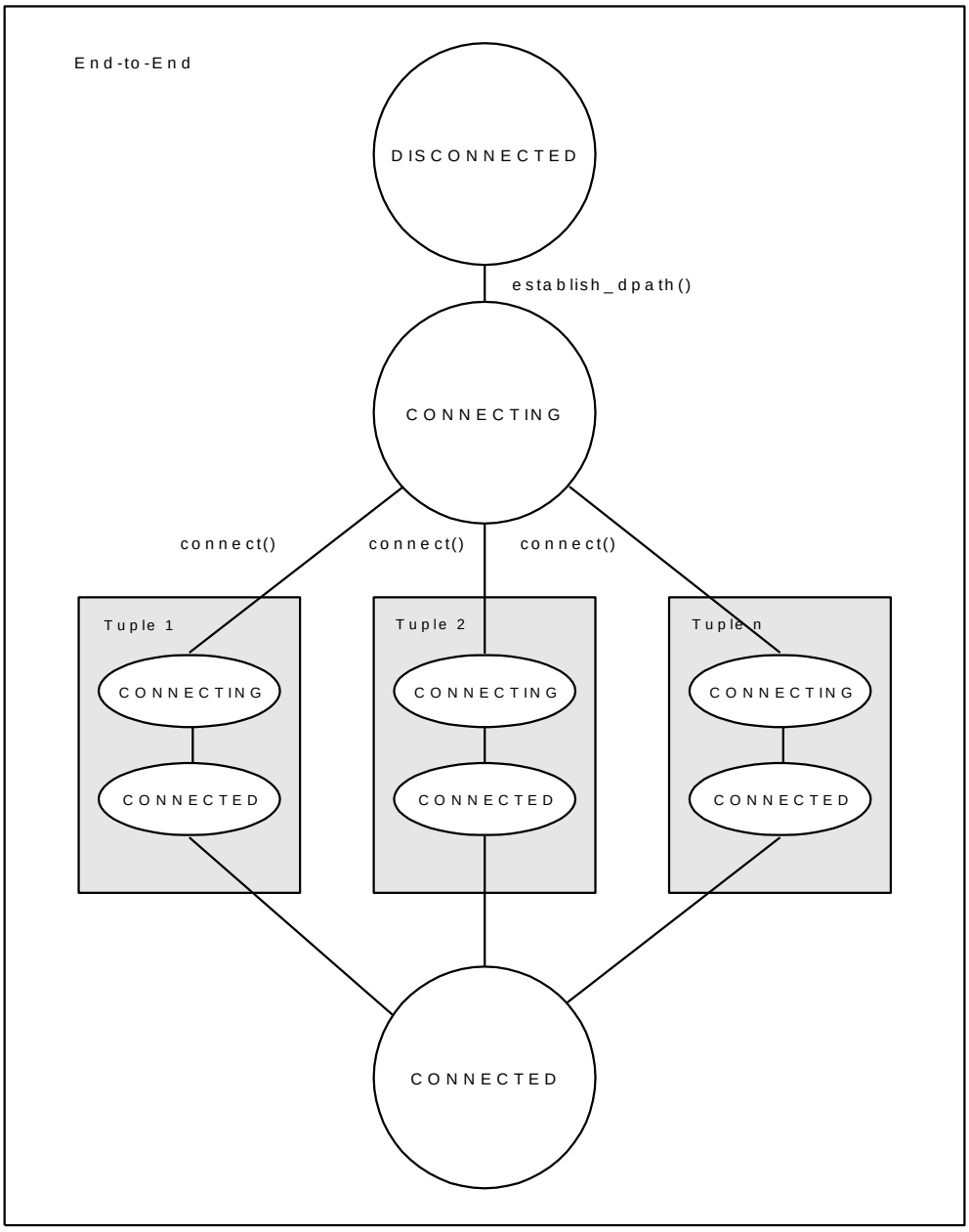

Figure 3: Establish an end-to-end connection

The following figure describes the establishing of a tuple (entity-to-entity) connection (e.g. UART-PPP). The connection state of the tuple is CONNECTED when both peer-connection-states are CONNECTED.

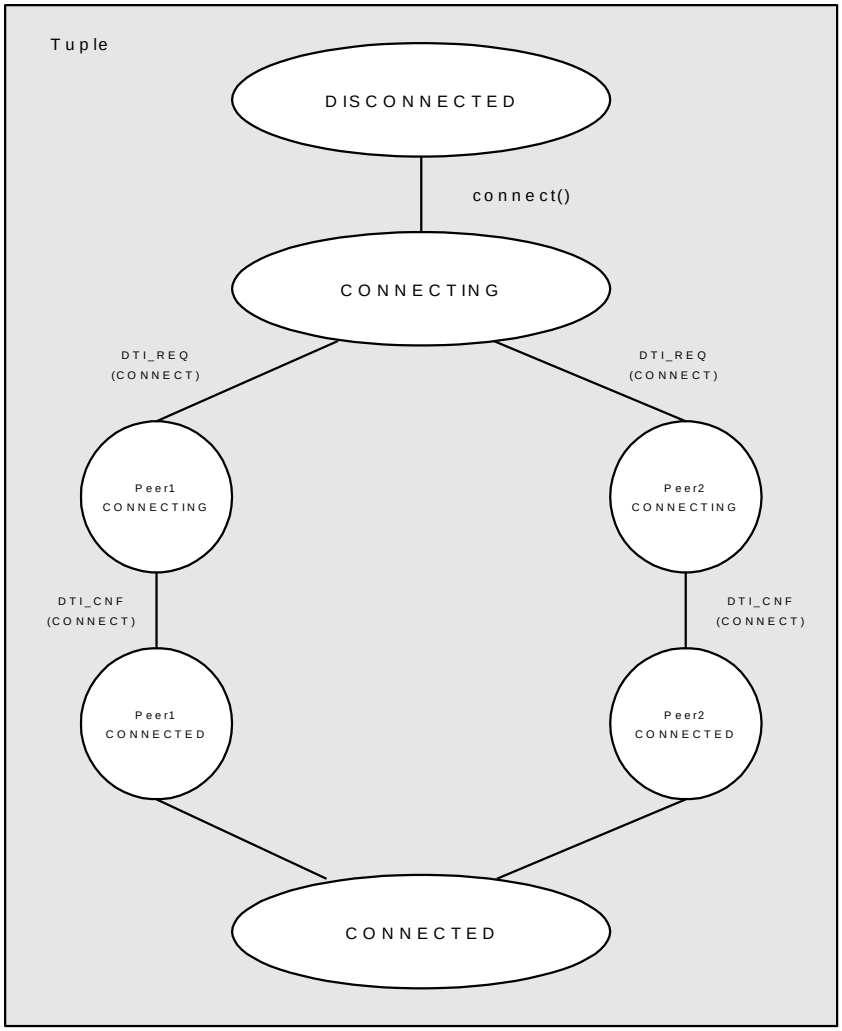

Figure 4: Establish a tuple connection

3.3.3 Close a DTI Connection

The following figure describes the closing of an end-to-end DTI connection (e.g. UART-PPP-SNDCP). The connection state of the end-to-end connection is DISCONNECTED when all tuple connections are in the state DISCONNECTED.

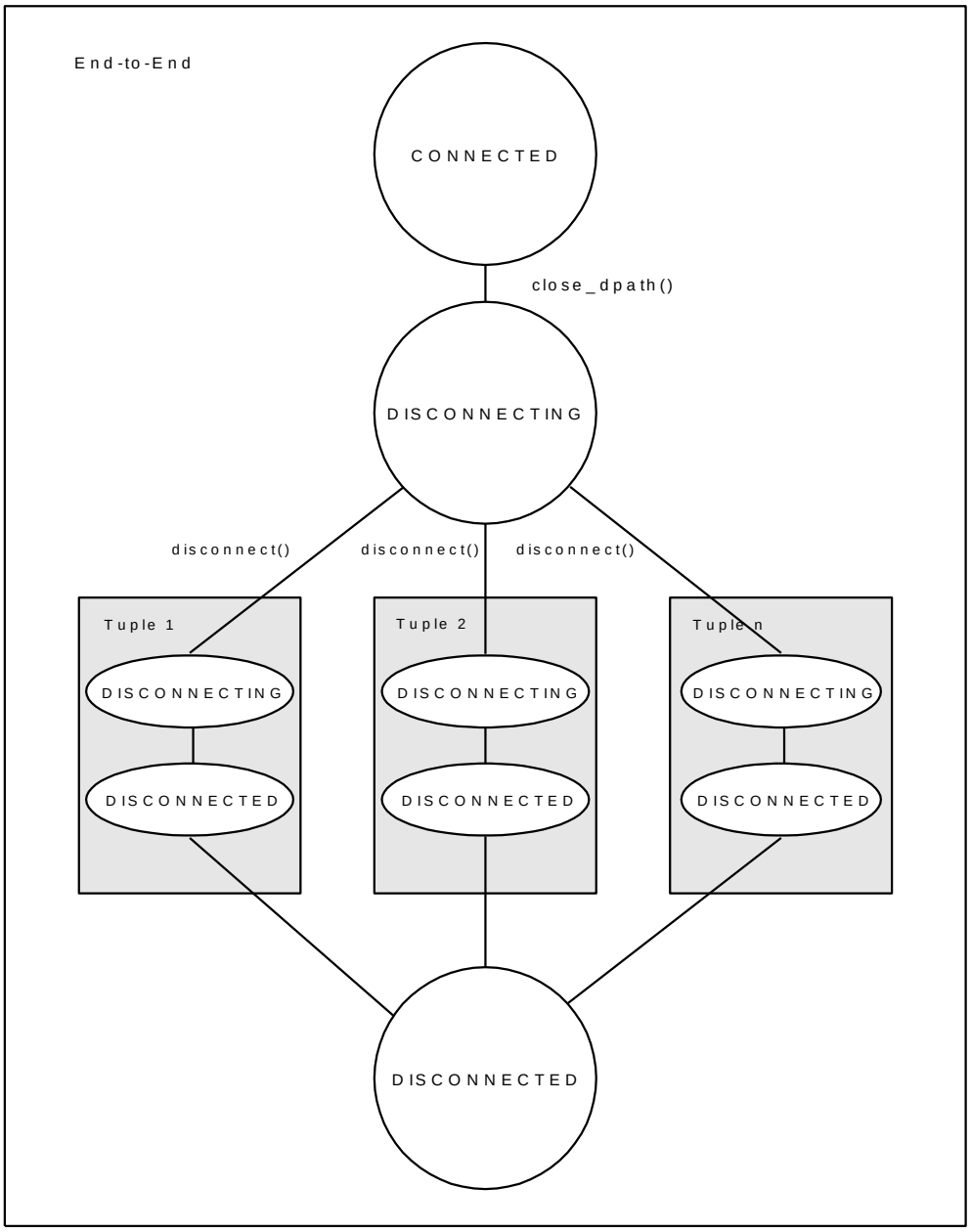

Figure 5: Disconnect end-to-end connection

The following figure describes the closing of a tuple (entity-to-entity) connection (e.g. UART-PPP). The connection state of the tuple is DISCONNECTED when both peer-connection-states are DISCONNECTED.

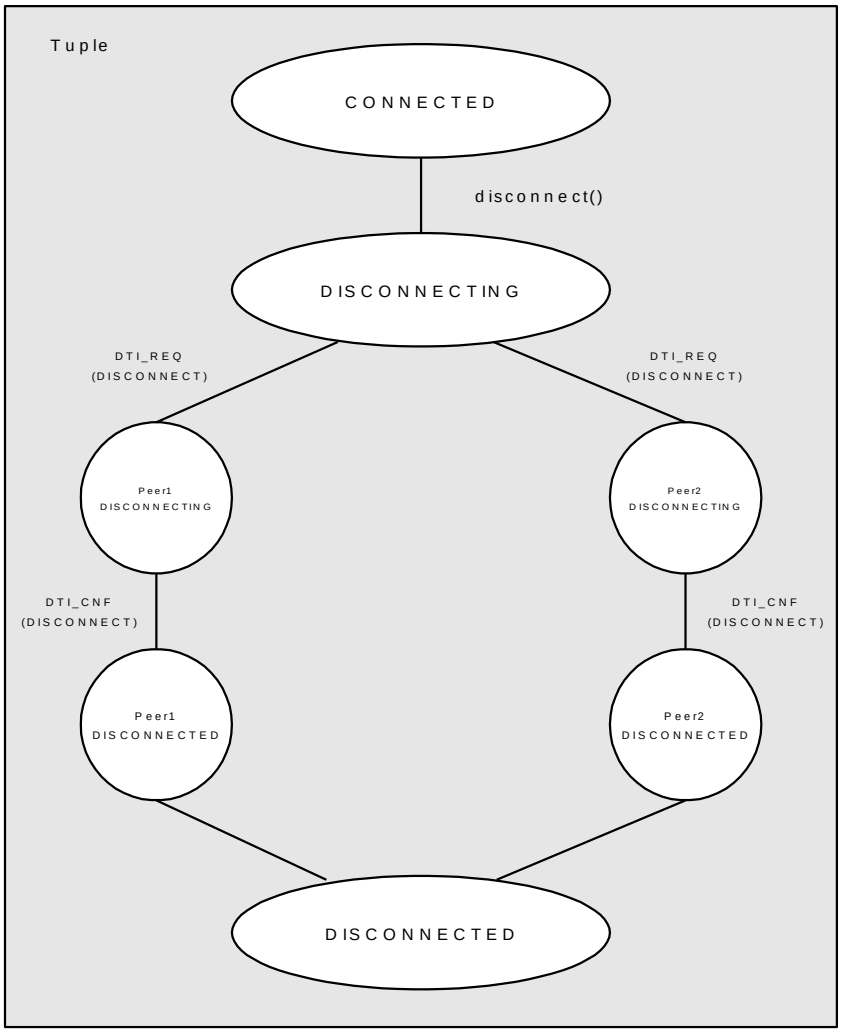

Figure 6: Close a tuple connection

3.3.4 Cutting and appending a DTI connection

It is possible to replace an existing DTI connection by a new DTI connection. Therefore the old connection must be cut. The old and the new connection peers are stored in the DTI channel parameters. The DTI Connection manager establishes the new end-to-end connection when the old connection is disconnected. For example, if a default data call is processed then a DTI connection between ATI and UART will be replaced by a DTI connection between L2R and UART. The next figure shows the cutting and the establishing of a DTI connection.

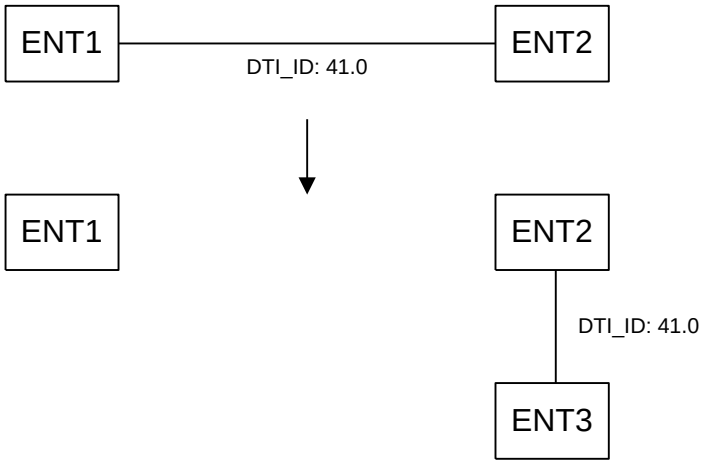

Figure 7: Cutting and establishing a DTI connection

It is also possible to append a new tuple connection to an existing tuple connection. An example for this is to append an IP-SNDCP connection to an UDP-IP connection. The next figure shows the appending of a DTI connection.

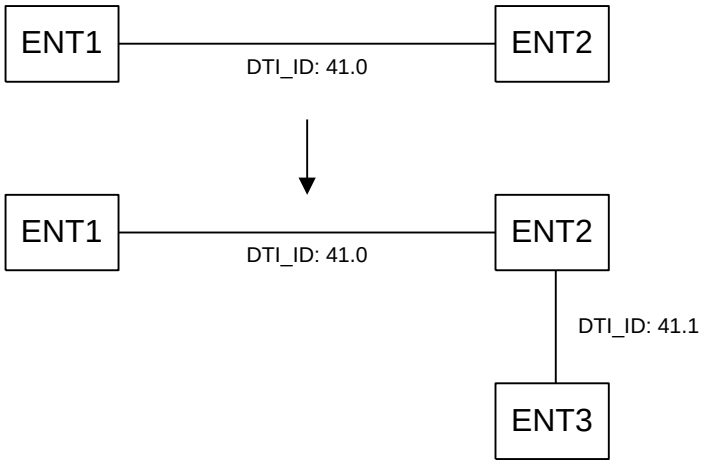

Figure 8: Appending a DTI connection

4 Identifiers

4.1 DTI Link ID

The DTI Link ID is a unique identifier of a DTI connection between two entities. It is composed of 4 bytes. Every byte has a different meaning. The first byte is reserved for future additions. The second byte describes the association of two data paths. This will be realised for future purposes. The third byte is the DTI ID that is the identifier of an end-to-end connection of one data stack. The fourth byte is the Tuple Number, which describes a part of the end-to-end connection between two entities.

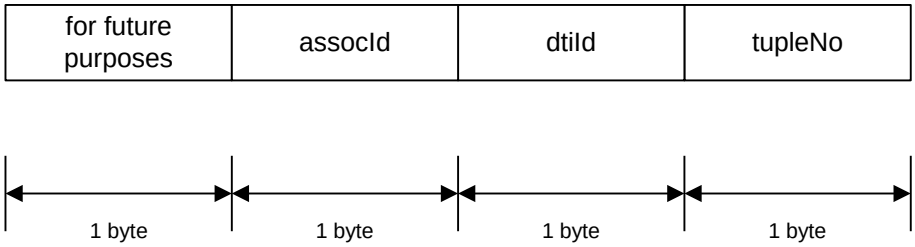

Figure 9: Composition of Link ID

Example:

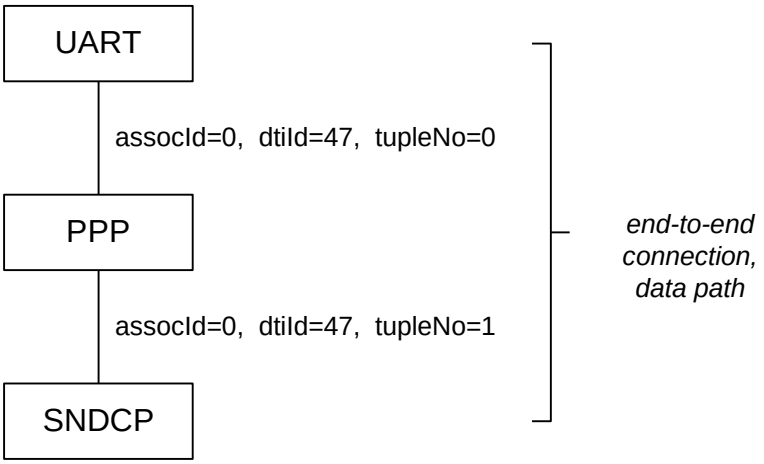

Figure 10: Example Link ID

5 Integration with existing software

5.1 Functions

5.1.1 DTI Connection Manager Functions

5.1.1.1 dti_conn_init

Return:

void

Parameter:

- *mng_ent_cb /* DTI connect/disconnect function */*

Definition:

void dti_conn_init(T_DTI_CONN_MNG_ENT_CB mng_ent_cb)*

Description:

This function is called only once at the start of the protocol stack. It serves for initialising the DTI Connection Manager.

5.1.1.2 dti_conn_new

Return:

UBYTE

Parameter:

- *dtiId /* DTI ID */*

Definition:

UBYTE dti_conn_new(UBYTE dtiId)

Description:

This function registers a new DTI connection. It returns the DTI ID of the new connection. If the parameter dtiId has the value DTI_DTI_ID_NOTPRESENT then a new DTI ID is created.

5.1.1.3 dti_conn_compose_link_id

Return:

T_DTI_CONN_LINK_ID

Parameter:

- *dummy /* for future purposes */*
- *assoc /* association of two data paths, for future purposes */*
- *dti_id /* DTI ID */*
- *tuple_no /* tuple number */*

Definition:

```
T_DTI_CONN_LINK_ID dti_conn_compose_link_id(  UBYTE dummy,
                                             UBYTE assoc,
                                             UBYTE dti_id,
                                             UBYTE tuple_no)
```

Description:

This function composes a link ID using the given parameters.

5.1.1.4 dti_conn_est_dpath

Return:

BOOL

Parameter:

- *dtiId /* DTI ID */*
- *entityList /* array of entities to connect */*
- *numEntities /* number of entities to connect (in entityList) */*
- *mode /* mode: APPEND or SPLIT */*
- *cb /* call-back function to inform the PSA that the connection is established */*

Definition:

```
BOOL dti_conn_est_dpath(    UBYTE dtiId,
                            T_DTI_ENTITY_ID * entityList,
                            UBYTE numEntities,
                            T_DTI_CONN_MODE mode,
                            T_DTI_CONN_CB* cb )
```

Description:

This function establishes a data path for the given entities. If *APPEND* was set, then a existing DTI channel would not released and new entities will be appended. If SPLIT was set, then the existing connection is replaced by the new connection. This function is called when a connection was requested (e. g. by ATD or AT+CGACT). When the end-to-end connection is connected then the call-back function cb is called.

5.1.1.5 dti_conn_close_dpath

Return:

BOOL

Parameter:

- *dtiId /* DTI ID */*

Definition:

```
BOOL dti_conn_close_dpath(    UBYTE dtiId )
```

Description:

This function closes a data path for the given dtiId.

5.1.1.6 dti_conn_erase_entry

Return:

void

Parameter:

- *dti_id /* DTI ID */*

Definition:

```
void dti_conn_erase_entry(    UBYTE dti_id)
```

Description:

This function removes a DTI channel from the channel list, which is held in the DTI Connection Manager. Note that this function does not close a DTI channel.

5.1.1.7 dti_conn_is_dti_channel_connected

Return:

BOOL

Parameter:

- *dti_id /* DTI ID */*
- *ent_id /* entityID */*

Definition:

```
BOOL dti_conn_is_dti_channel_connected(  T_DTI_ENTITY_ID ent_id,
                                         UBYTE dti_id )
```

Description:

This function returns TRUE if the DTI Connection given by dti_id is connected and the entity with the given ent_id is part of the connection.

5.1.1.8 dti_conn_is_dti_channel_disconnected

Return:

BOOL

Parameter:

- *dti_id /* DTI ID */*

Definition:

```
BOOL dti_conn_is_dti_channel_disconnected(  UBYTE dti_id )
```

Description:

This function returns TRUE if the DTI Connection given by dti_id is disconnected.

5.1.1.9 dti_conn_entity_connected

Return:

void

Parameter:

- *link_id /* Link ID */*
- *ent_id /* EntityID */*
- *result /* result: DTI_OK or DTI_ERROR */*

Definition:

```
void dti_conn_entity_connected(  T_DTI_CONN_LINK_ID link_id,
                                 T_DTI_ENTITY_ID ent_id,
                                 T_DTI_CONN_RESULT result )
```

Description:

This function must be called to inform the DTI Connection Manager that a connection is established. If result equals DTI_ERROR then the whole data path is closed.

5.1.1.10 dti_conn_entity_disconnected

Return:

void

Parameter:

- *link_id /* Link ID */*
- *ent_id /* EntityID */*

Definition:

```
void dti_conn_entity_disconnected( T_DTI_CONN_LINK_ID link_id,
                                   T_DTI_ENTITY_ID ent_id )
```

Description:

This function must be called to inform the DTI Connection Manager that a connection is closed.

5.1.1.11 dti_conn_close_all_connections

Return:

void

Parameter:

Definition:

void dti_conn_close_all_connections()

Description:

This function closes all established DTI channels and erases its entries from DTI channel list. Further all other DTI channels are removed from DTI channel list.

5.1.1.12 dti_conn_close_all_connections

Return:

void

Parameter:

Definition:

void dti_conn_close_all_connections()

Description:

This function closes all established DTI channels and erases its entries from DTI channel list. Further all other DTI channels are removed from DTI channel list.

5.2 Data Structures

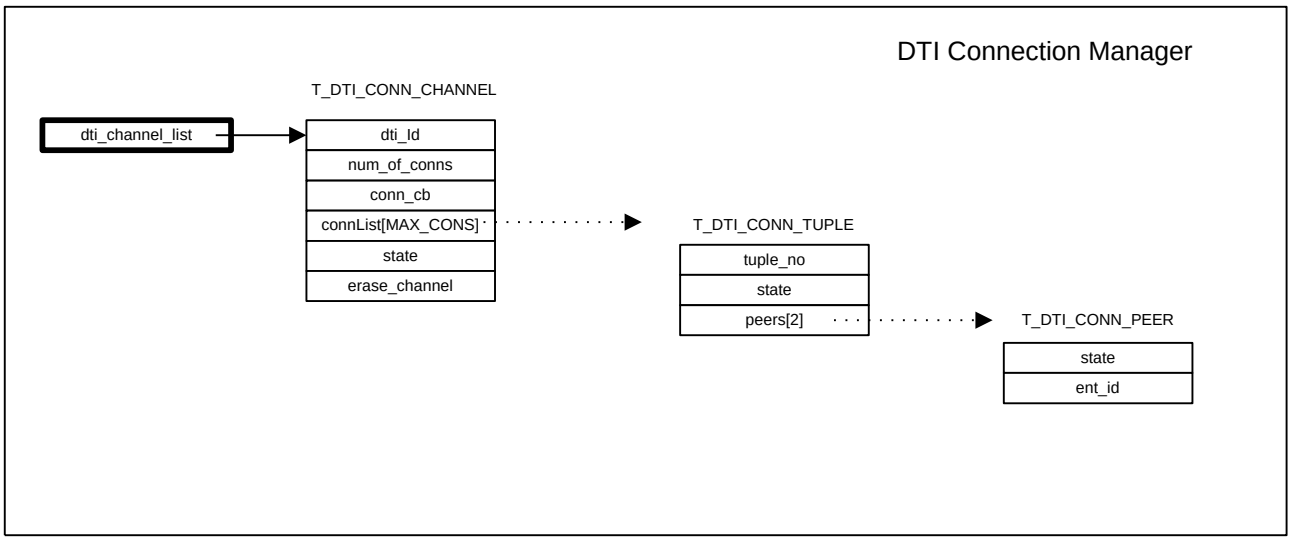

Figure 11: Data structure for one DTI channel

The data structure in the shown figure is used by the DTI Connection Manager to maintain a DTI channel. It contains all information about a DTI channel. An instance of this structure is created by a function call of dti_conn_new() and it is erased by dti_conn_erase_entry() or dti_conn_close_all_connections().

6 Message Sequence Charts

Only the most important function parameters are shown in the following figures.

6.1 Setup and connect a DTI Channel

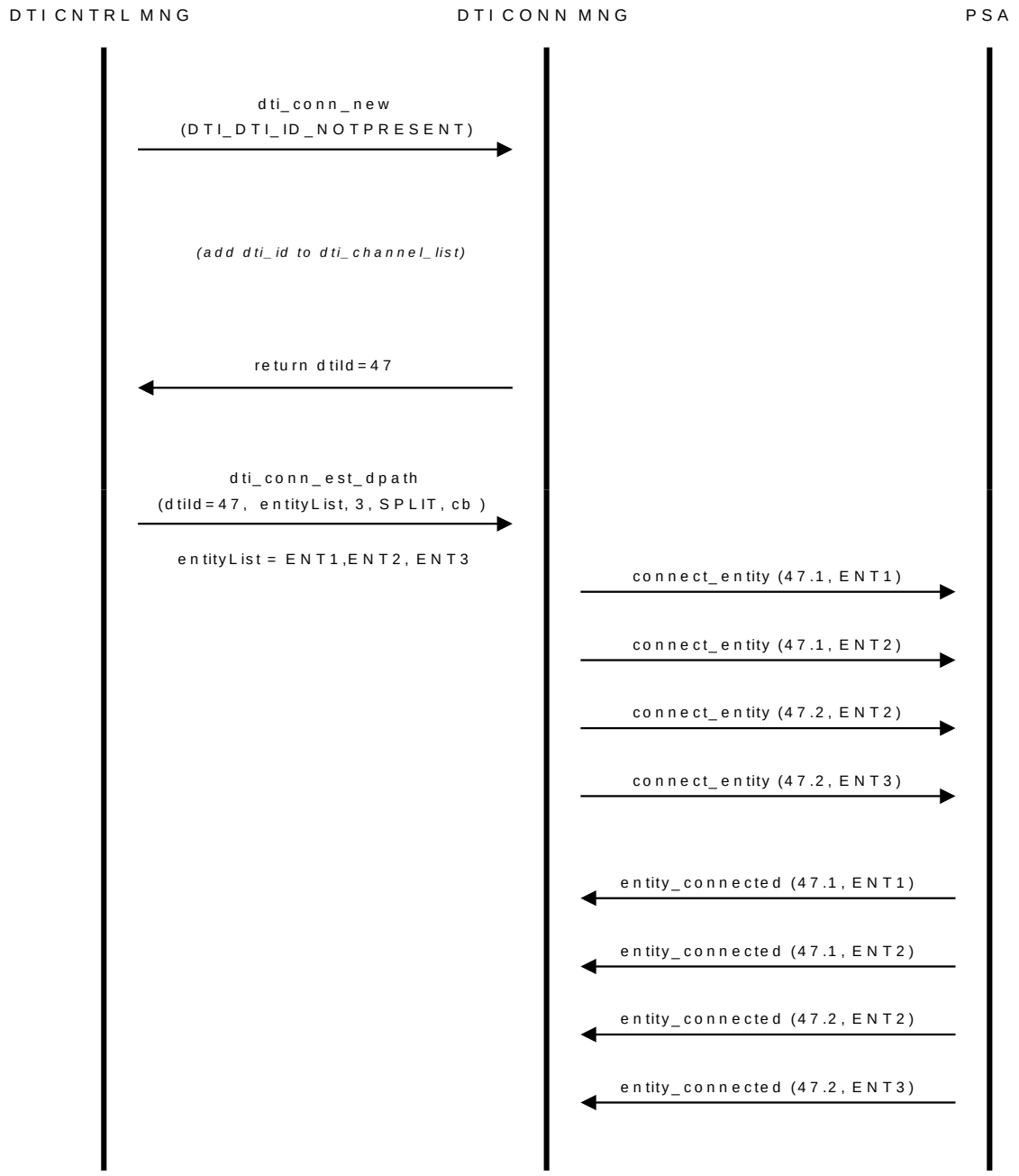

Figure 12: Setup and connect a DTI Channel

6.2 Close two DTI Channels

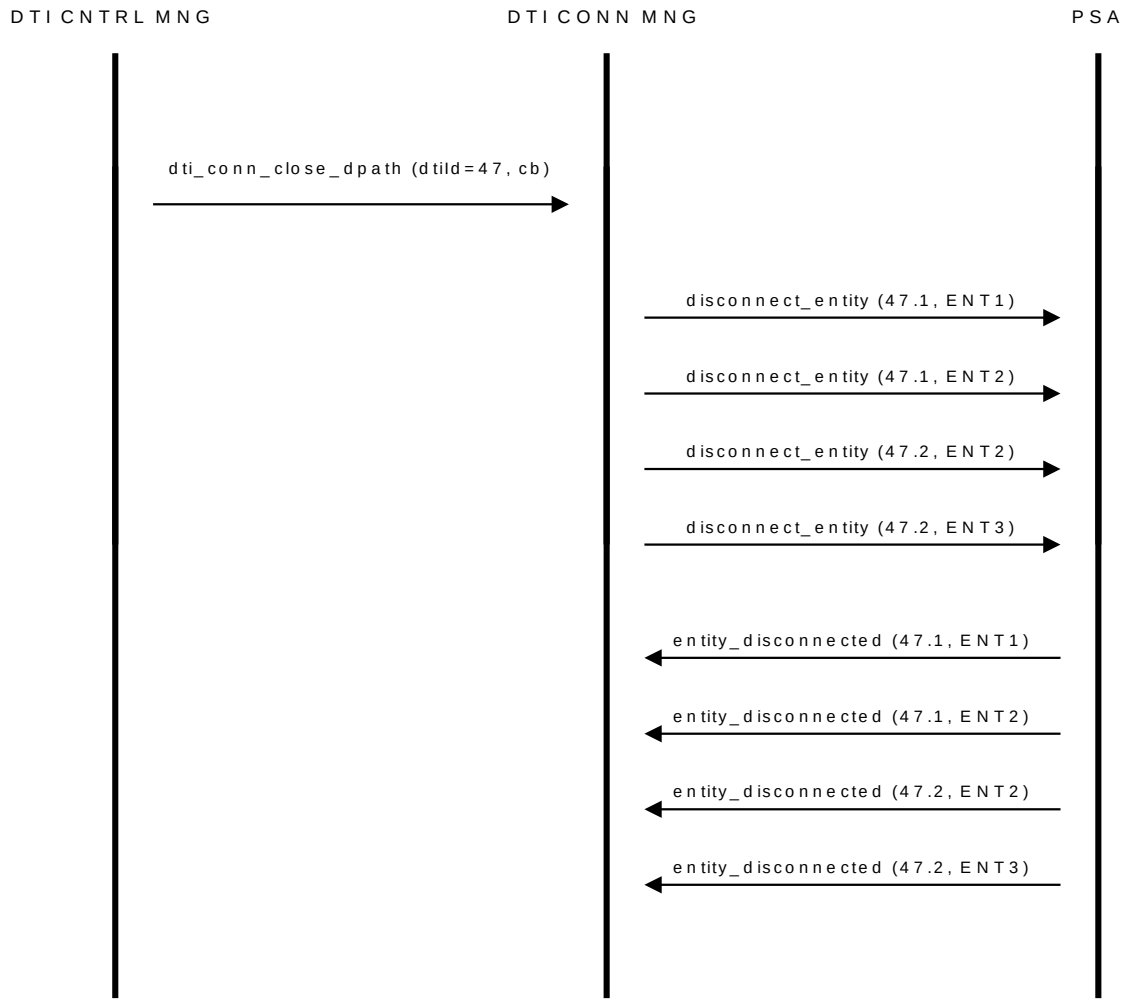

Figure 13: Close two DTI Channels

7 Appendix

7.1 Example: Creating of a new ATI channel (initial) and switching to data channel

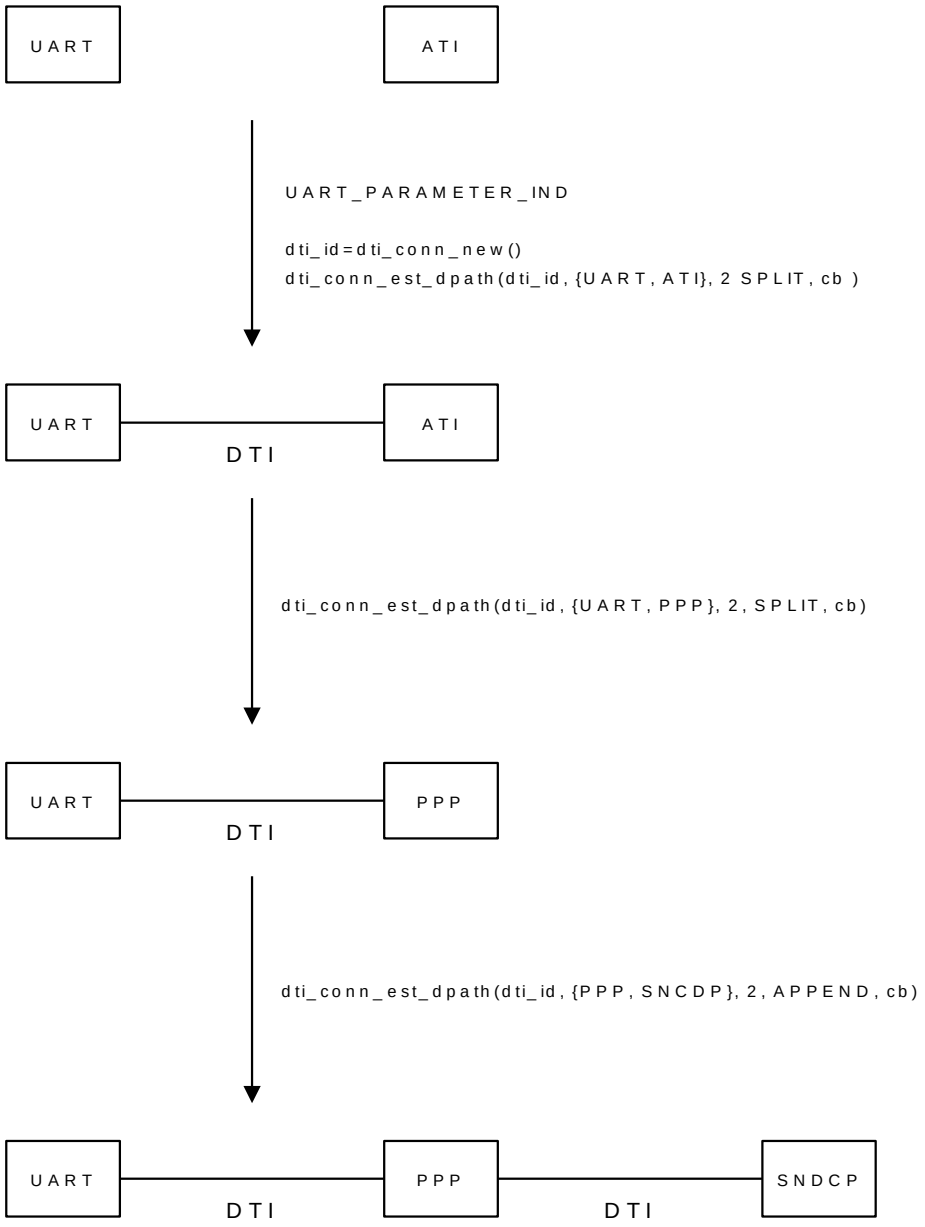

Figure 14: Creating of a new ATI channel (initial) and switching to data channel

Appendices

B. Acronyms

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

C. 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/>