ENGLISH TRANSLATION

ITS APPLICATION SUB-LAYER SPECIFICATION GUIDELINE

ITS FORUM RC-014 Version 3.0

Established on May 25, 2017

Revised on October 10, 2019

Revised on December 4, 2020

ITS Info-communications Forum of Japan

General Notes to the English Translation of ITS Info-communications Forum Guidelines

1. Notes on Copyright

- The copyright of this document is ascribed to the ITS Info-communications Forum of Japan.

2. Notes on English Translation

- ITS Info-communications Forum Guidelines are usually written in Japanese. This document is a translation into English of the original document for the purpose of convenience of users. If there are any discrepancies in the content, expressions, etc. between the original document and this translated document, the original document shall prevail.

- ITS Info-communications Forum Guidelines, in the original language, are made publicly available through web posting. The original document of this translation may have been further revised and therefore users are encouraged to check the latest version at an appropriate page under the following URL:
http://www.itsforum.gr.jp/Public/Eguideline/index.html.

ITS APPLICATION SUB-LAYER SPECIFICATION GUIDELINE

ITS FORUM RC-014 Version 3.0

Established on May 25, 2017

Revised on October 10, 2019

Revised on December 4, 2020

ITS Info-communications Forum of Japan

Revision History

Version	Date	Chapter/ Section	Reason	Revised Content
1.0	May 25, 2017	Establishment	Newly established	
2.0	September 2, 2019	Introduction, Chapters 1 through 4, Annex D	Support for ARIB STD-T104	Included information in each chapter regarding use of ARIB STD-T104 for lower layer communications
3.0	December 4, 2020	Introduction, Chapters 1 through 4, Annexes A, E, F, and G	Support for ARIB STD-T109 and ARIB STD-T120	Added positioning of this guideline to the Introduction Included information in each chapter regarding use of ARIB STD-T109 and ARIB STD-T120 for lower layer communications

[BLANK]

Introduction

The existing communications technologies used in ITS and new communications technologies such as V2X are being used in Japan and abroad to rapidly implement and commercialize practical advanced driving assistance systems, which include automated driving systems.
This guideline is intended to achieve the following in order to promote further diffusion of existing wireless communication technologies used with ITS/DSRC and to support new wireless communication technologies such as V2X.

(1) Enhancement of user convenience by the ITS application sublayer
- Enhancement of the convenience of user-owned on-board equipment and transmission terminal services by extending the scope of the lower layers of the DSRC basic application interface
- By installing the ITS application sublayer applicable to DSRC, other narrow area communications, and wide area communications (e.g., ITS FORUM RC-005, IEEE 802.11p, ETSI EN 302 636, ARIB STD-T109, LTE, 4G, 5G.) used in ITS, it is possible to use the sublayer for various applications using the DSRC basic application sublayer and is easy to intelligentize.

(2) Construction of a platform not bound by the provisions of lower communication layers
- Multiple communication methods using existing wireless communication technologies and new wireless communication technologies, etc. are supported by constructing a flexible platform that can support future multi-access.
- The limitations imposed by wireless communication terminals are lifted, so interfaces with upper layers of applications, as seen from wireless communication terminals, can be shared, avoiding growing complexity during system construction.
- Even if service providers switch from the wireless communication terminals they are currently using to wireless communication terminals with new functions in order to provide richer services, it is easy to improve provided services, both in terms of functions and capabilities, without being concerned about the changed wireless communication terminals and without changing application interfaces.

[BLANK]

ITS APPLICATION SUB-LAYER SPECIFICATION GUIDELINE

Contents

Chapter 1 General Descriptions

1.1 **Overview**

This guideline defines the ITS application sublayer (ITS-ASL: ITS Application Sub-Layer). The ITS-ASL builds and extends ITS platforms for various applications and various wireless communication terminals by expanding the lower layer application scope of the basic application interface (basic API) on narrow area communications (DSRC) defined by the standard ARIB STD-T75 and using this basic API. The ITS-ASL also enables execution of applications using basic API by making it possible to complement the communication protocol functions of other narrow/wide area communications such as ITS FORUM RC-005, IEEE 802.11, ETSI EN 302 636, ARIB STD-T109, 4G, and 5G.

This will make it easier for the existing communications technologies used by ITS and new communications technologies such as V2X technologies to be used in automated driving systems, advanced driving systems, and next-generation satellite billing systems, for which practical implementation and product release are proceeding at a rapid rate both in Japan and overseas.

1.2 **Scope of Application and Positioning of This Document**

1.2.1 **Scope of Application**

ITS systems to which this guideline applies comprise a base station, a mobile station and test equipment as specified in ARIB STD-T75, ARIB STD-T88, ITS FORUM RC-005, IEEE 802.11, ARIB STD-T104, ARIB STD-T109, ARIB STD-T120, etc.

This guideline specifies the extended communication protocol, which interfaces between lower protocol stacks and Non-Network applications, and provides complementary communication functions for these lower layers.

1.2.2 **Position of ITS-ASL in Protocol Configuration**

Figure 1.2-1 shows the location of the ITS-ASL in the protocol stack.

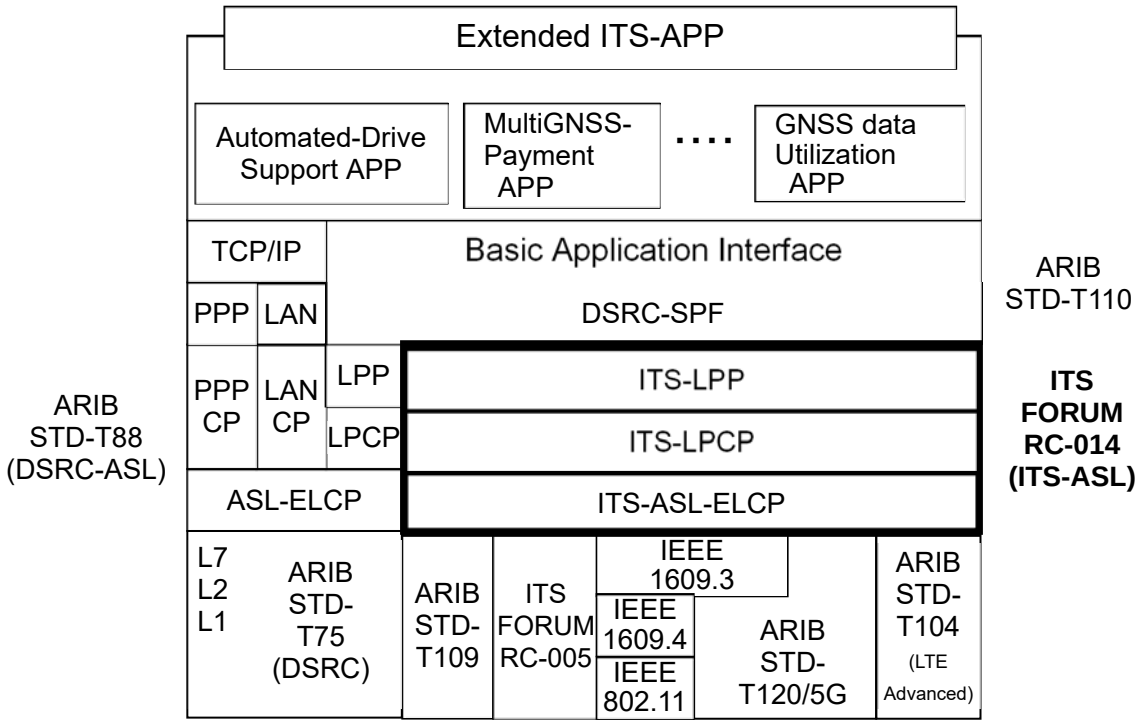


Figure 1.2-1 Protocol configuration of the extended ITS platform

1.3 **Reserve Field**

The variables or information filed as defined “reserve” in this guideline are reserved for future expansion capabilities. These variables or information filed as defined “reserve” in this guideline (version) may be defined as specific values or identifiers. The user of this guideline should take into account that these values or identifiers might be changed in the future version.

1.4 **Encoding Rule**

Variables specified in this guideline are described using Abstract Syntax Notation One :ISO/IEC 8824). The coding rules are a packed encoding rule (UNALIGNED PER (Packed Encoding Rule: ISO/IEC 8825-2).

1.5 **References**

Refer to the following standards and other materials for reference with regard to matters not stipulated in this Guideline. For undated references the latest edition of the publication referred to applies.

ARIB STD-T75	Dedicated Short-Range Communication (DSRC) System
ARIB STD-T88	Dedicated Short-range Communication (DSRC) Application Sub Layer
ARIB STD-T104	LTE-Advanced System
ARIB STD-T109	700 MHz Band Intelligent Transport Systems
ARIB STD-T110	Dedicated Short-Range Communication (DSRC) Basic Application Interface
ARIB STD-T120	IMT Systems based on 3GPP Specifications
ARIB TR-T16	Dedicated Short-range Communication (DSRC) System Test Items and Conditions for Land Mobile Station Compatibility Confirmation
ARIB TR-T17	Test Items And Conditions For Dedicated Short-range Communication (DSRC) Application Sub Layer Land Mobile Station Compatibility Confirmation
ARIB TR-T19	LTE-Advanced System
ARIB TR-T22	Dedicated Short-range Communication (DSRC) Basic Application Interface Test Items And Conditions For Land Mobile Station Compatibility Confirmation
JEITA TT-6001	Standard specification of ITS On-Board Unit
JEITA TT-6002	Standard specification for DSRC section of ITS On-Board Unit
JEITA TT-6003	Standard specification for Car Navigation System section of ITS On-Board Unit
JEITA TT-6004	Speech synthesizer symbols for ITS on-Board Unit
ISO/IEC 8824-1	Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation
ISO/IEC 8825-2	Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)
ISO15628:2007	Road transport and traffic telematics -- Dedicated short range communication (DSRC) -- DSRC application layer
ISO24103:2009	Intelligent transport systems -- Communications access for land mobiles (CALM) -- Media adapted interface layer (MAIL)
ISO29281:2011	Intelligent transport systems -- Communications access for land mobiles (CALM) -- Non-IP networking
IEEE 802.11-2012	IEEE Standard for Information technology— Telecommunications and information exchange between systems Local and metropolitan area networks— Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and

	Physical Layer (PHY) Specifications
IEEE 1609.3	IEEE Standard for Wireless Access in Vehicular Environments (WAVE)—Networking Services
ITS FORUM RC-005	Experimental Guideline for Inter-Vehicle Communications System using 5.8GHz-Band

Chapter 2 Overview of ITS Application Sublayer

2.1 **Service Interfaces and Protocol Stacks of the ITS Application Sublayer**

The protocol located in the Adapter layer (hereinafter referred to as “ITS-ASL-ELCP”) is an extended protocol for the ITS-LPCP to use services provided by the lower layers.

Figure 2.1-1 shows an overview of the service interfaces and the protocol stacks.

The ITS-ASL-ELCP exchanges protocol data units (PDU) with the peer ITS-ASL-ELCP by using the service interface that the lower layer provides. The ITS-ASL-ELCP conducts the communication procedures provided for in the ITS-ASL-ELCP. The ITS-ASL-ELCP also provides the ITS-LPCP the service interface of the equivalent service to the ASL-ELCP (the communication service for data transmission and the management service for management control).

The ITS-LPCP exchanges PDU with the peer ITS-LPCP by using the service interface that the ITS-ASL-ELCP provides. The ITS-LPCP conducts the communication procedures provided for in the ITS-LPCP.

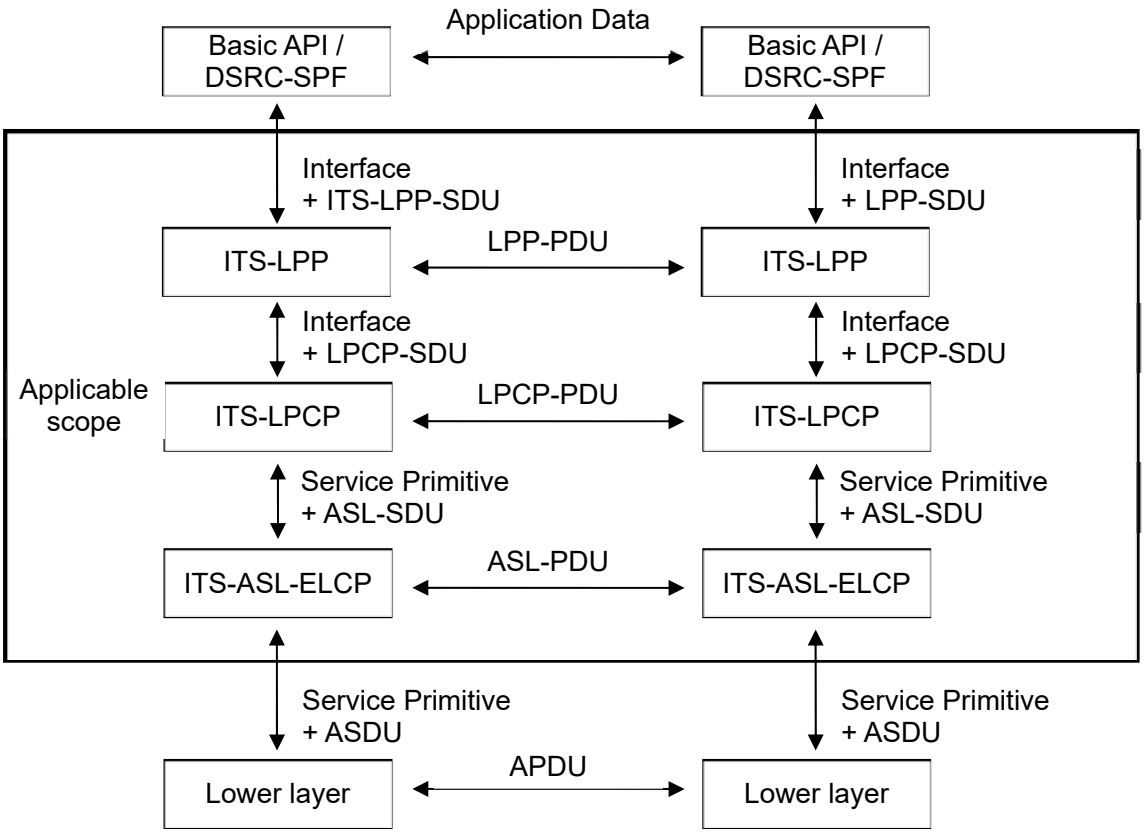


Figure 2.1-1 Overview of the Service Interface and the Protocols of the ITS-ASL

2.2 **Identification of Destination Application**

To send data units to the correct peer basic API, the ITS-ASL utilizes the application identification function provided by the lower layers.

2.2.1 **ITS FORUM RC-005**

In the case of ITS FORUM RC-005, as shown in Figure 2.2-1, identification using the Provider Service Identifier (PSID) based on the WAVE Short Message Protocol (WSMP) specification prescribed in IEEE 1609.3 is used. The transmitting application issues a request to WSMP to transmit a message to the remote station, specifying its own WSID. The receiving WSMP sends a message to the application specified by the WSID.

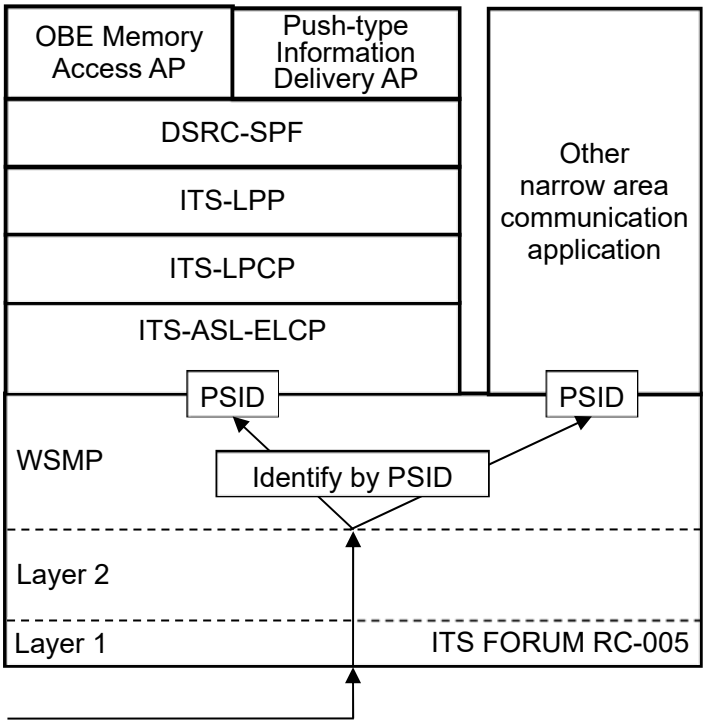


Figure 2.2-1 Identification of the Destination Application in ITS FORUM RC-005

2.2.2 **IEEE 802.11**

In the case of IEEE 802.11, as shown in Figure 2.2-2, identification using PSID provided by WSMP of IEEE 1609.3 is used. The transmitting application issues a request to WSMP to transmit a message to the remote station, specifying its own WSID. The receiving WSMP sends a message to the application specified by the WSID.

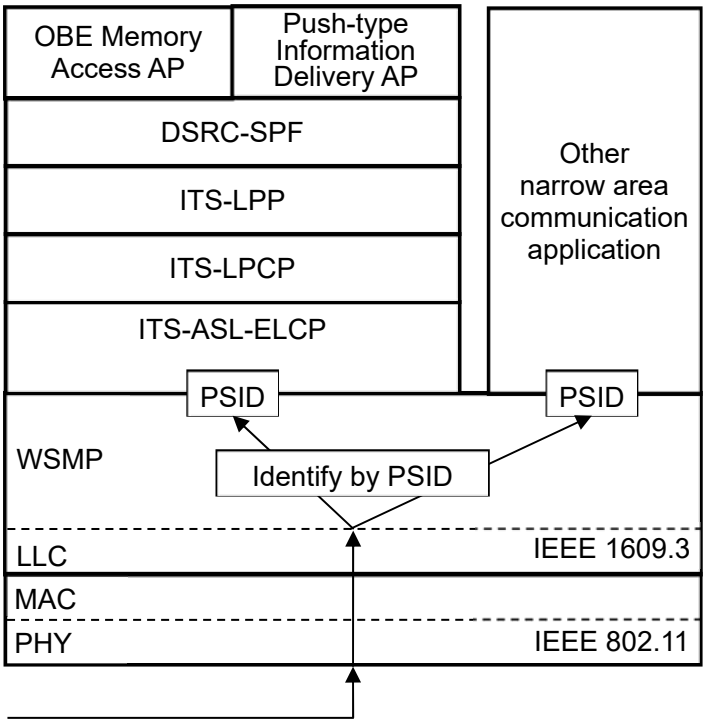


Figure 2.2-2 Identification of the Destination Application in IEEE 802.11

2.2.3 ARIB STD-T104/ARIB STD-T120 (Uu)

In the case of ARIB STD-T104 and ARIB STD-T120(Uu), as shown in Figure 2.2-2, the TCP/IP session port number is used to identify the destination application in this way.

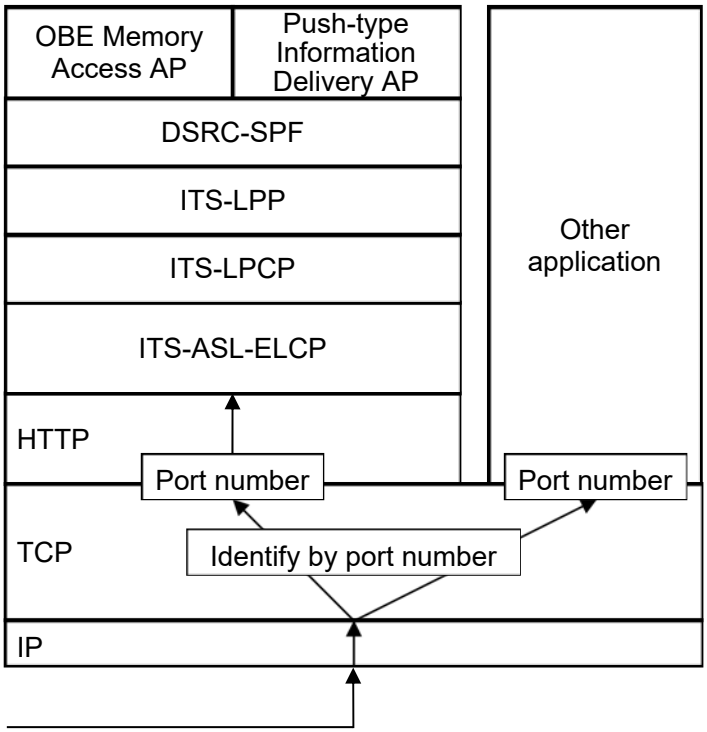


Figure 2.2-3 Identification of the Destination Application in ARIB STD-T104/ARIB STD-T120 (Uu)

The upper layer server and mobile station exchange messages as User Equipment (UE) in the LTE-Advanced network, as shown in Figure 2.2-4. For each UE device, ITS-ASL transmits data over HTTP/TCP/IP using LTE-Advanced U-plane. The mobile station application transmits messages in the form of HTTP requests to the upper layer server application. The upper layer server application transmits messages in the form of HTTP responses to the mobile station application.

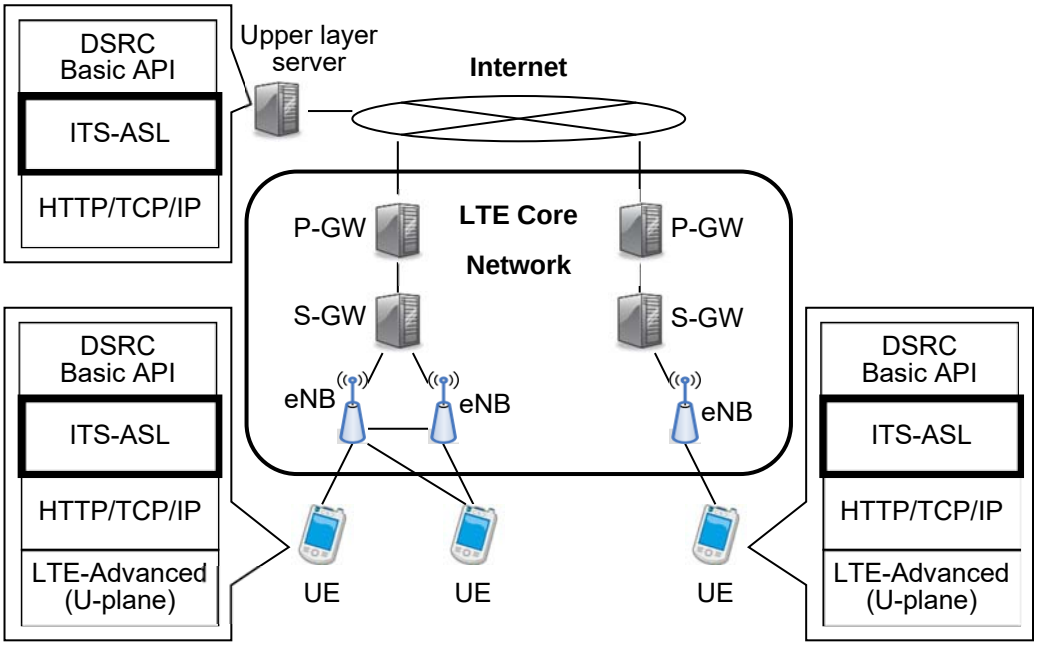


Figure 2.2-4 Protocol and Network Structure when Using ARIB STD-T104

2.2.4 ARIB STD-T120 (PC5)

In the case of ARIB STD-T120 (PC5), as shown in Figure 2.2-5, identification using the Provider Service Identifier (PSID) provided by WSMP of IEEE 1609.3 is used. The transmitting application issues a request to WSMP to transmit a message to the remote station, specifying its own WSID. The receiving WSMP sends a message to the application specified by the WSID.

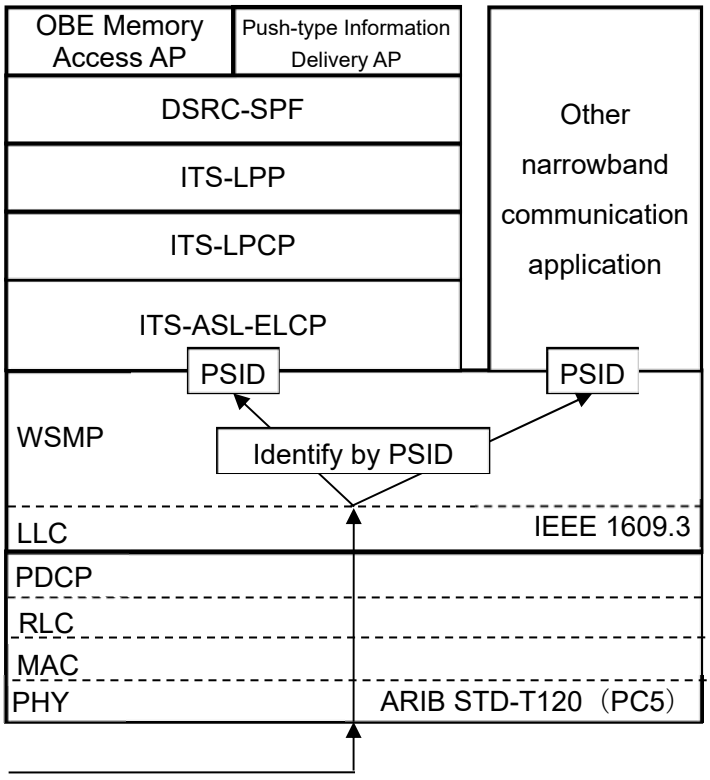


Figure 2.2-5 Identification of the Destination Application in ARIB STD-T120 (PC5)

2.2.5 ARIB STD-T109

In the case of ARIB STD-T109, as shown in Figure 2.2-2, there are no Layer 7 parameters that identify destination applications. When there are multiple applications, an ARIB STD-T109 Layer 7 header ApplicationAssociatedInformation field or reserved field must be used.

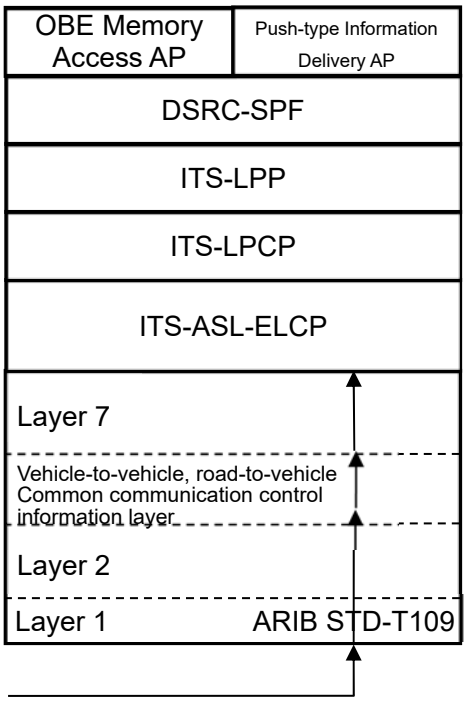


Figure 2.2-6 Identification of the Destination Application in ARIB STD-T109

2.3 **Numbering Plan (Link Address)**

It is assumed that Link addresses for communication and the numbers of each type of equipment are different. In order to protect communication privacy, an address with a length of 4 octets, chosen by Mobile Station at random, is used as the link address. This address is shared with the Base Station and Mobile Station, and the same value is used while communications are continued.

2.3.1 **Link AddressType and Configuration**

Link addresses are categorized into the following two types, each having a length of 4 octets.

(1) A broadcast (group) link address for the transmission of data, etc., from a base station to specified groups of mobile stations.

(2) A private link address for point-to-point two-way communications between a mobile station and a base station.

The broadcast (group) link address shall be the 1 octet address (1xxx xxxx) created with the first bit as "1" and the next 7 bits. After the second octet, all are "0".

The private link address shall be the 4 octet address created with the first bit as “0” and the next 31 bits.

2.3.2 **Generation Method of a Private Link Address**

The mobile station transmits a private link address as a connection response message to the base station.

(1) When generated

In principle, a mobile station shall generate a new random number as a private link address when the operation starts and shall retain the value until the operation is completed.

(2) Generation Algorithm

Private link addresses are not to overlap among mobile stations within the same communication zone at the same time. Therefore, a generation algorithm which has a low probability of generating duplicate private link addresses among mobile stations shall be adopted in consideration of not only the randomness of private link addresses within a mobile station but also the randomness between mobile stations.

(3) Handling Procedures for Base Stations with Duplicate Private Link Addresses

Base stations shall consider that there is some probability of duplicate link addresses. Examination of duplicate link addresses and the procedures are defined as follows and are applied to ITS-ASL-ELCP.

(a) A base station shall examine the duplication of link addresses after receiving the connection response message from mobile stations.

(b) In the case where duplicate link addresses are detected, the base station shall immediately transmit a release using the link address and shall release the connection with the mobile station that has the duplicate link address.

[BLANK]

Chapter 3 ITS Application Sub-Layer Specification

3.1 **Extended Link Control Protocol (ITS-ASL-ELCP)**

3.1.1 **Overview**

3.1.1.1 **Functions**

The ITS-ASL-ELCP has the following functions, in order to complement the communication facility of the lower layer, and it provides the communication services for data transmission and the management service to control the ITS-LPCP.

(1) Multi-protocol correspondence
(2) Bulk transmission control
(3) Broadcast mode control
(4) Communication connection management

3.1.1.2 **Configuration**

Figures 3.1-1 and 3.1-2 show the ITS-ASL-ELCP configuration of the upper layer server and mobile station. As extended communication control entities, they perform data transfer service processes, bulk transmission control, broadcast mode control, and area determination. Furthermore, as communication control management entities, they perform event notification and transmission connection management. For LTE support, the upper layer server transmission connection management entity determines if the other party's mobile station is within the service provision area.

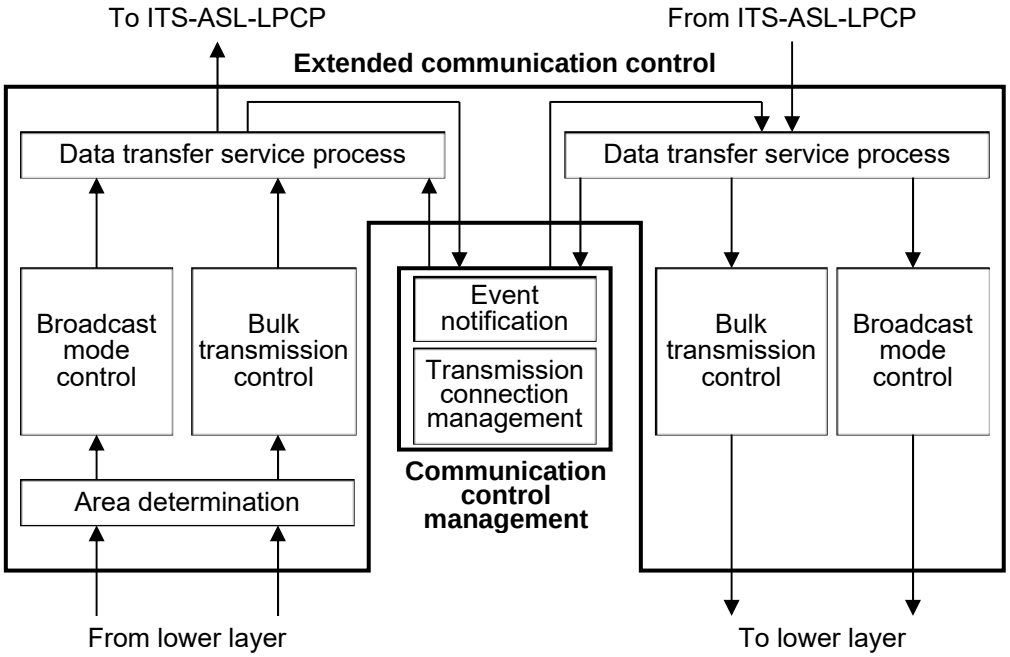


Figure 3.1-1 ITS-ASL-ELCP Entity Configuration of Upper Layer Servers

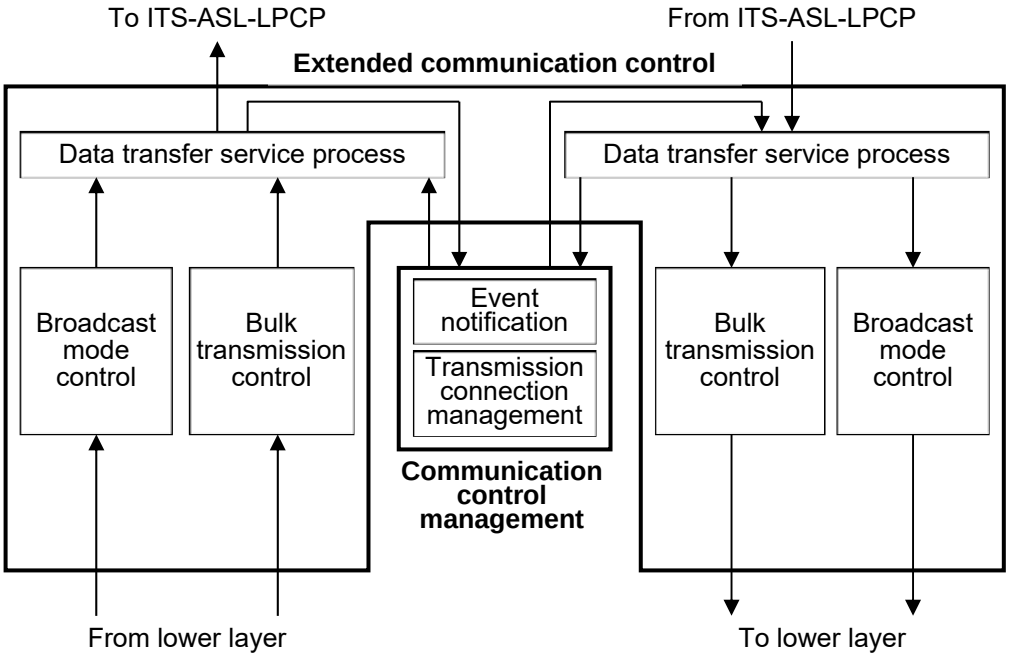


Figure 3.1-2 ITS-ASL-ELCP Entity Configuration of Mobile Stations

3.1.2 **Extended Link Control**

3.1.2.1 **Communication Service Interface**

3.1.2.1.1 **Overview of Primitive Relationship**

The Communication control of the ITS-ASL-ELCP provides the ITS-LPCP with the following primitives as communication service.

SendDataUnit.request
SendDataUnit.indication
SendDataUnitForArea.request
StopDataUnit.request

The SendDataUnit.request and SendDataUnitForArea.request are sent to the ITS-ASL-ELCP from the ITS-LPCP, in order to request that the ASL-SDU passed from the ITS-ASL-LPCP is transmitted to the remote station. The SendDataUnit.indication is sent to the ITS-LPCP from the ITS-ASL-ELCP, in order to show the arrival of the ASL-SDU. The StopDataUnit.request is sent to the ITS-ASL-ELCP from the ITS-LPCP in order to request that transmission of the ASL-SDU passed from the ITS-LPCP is stopped.

3.1.2.1.2 **Service Contents Specification**

In this sub-clause, primitives and parameters about the communication service are specified. Parameters as an interface are described abstractly. And the information, which is needed for a receiving peer entity, is specified. However, the concrete realization methods, which provide with this information, are not specified.

The logical relationship between communication service primitives, which are provided by the ITS-ASL-ELCP to the ITS-LPCP, is shown in Figure 3.1-3.

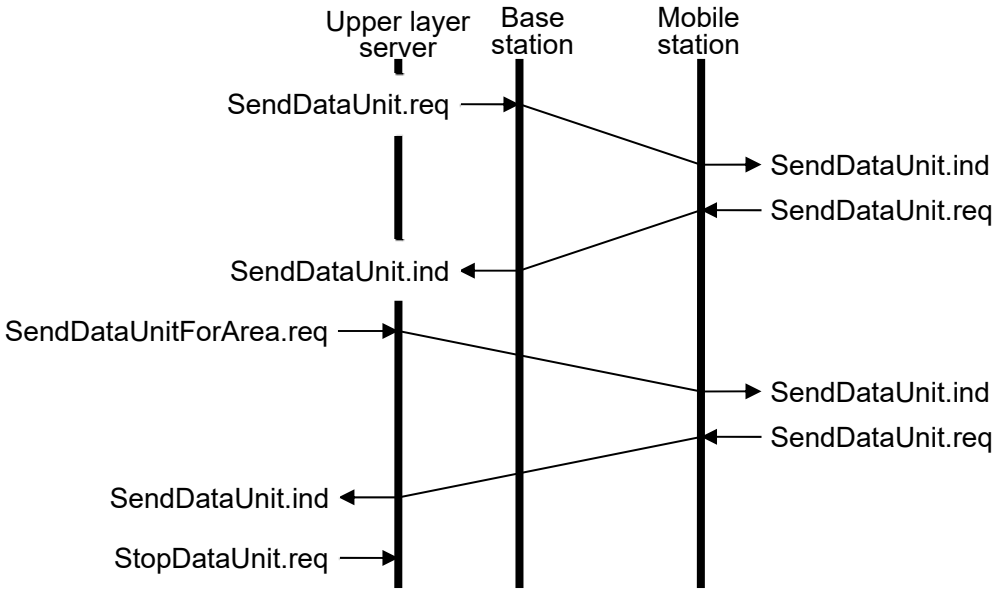


Figure 3.1-3 Logical Relationships between Communication Service Primitives

3.1.2.1.2.1 **Data Transfer Request Primitive**

(1) Function

This primitive is service primitive which request that the ASL-SDU is transmitted to remote station.

(2) When generated

The ITS-LPCP always generates this primitive.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

SendDataUnit.request (link Address, parameter)

The parameter “linkAddress” stores the link address, which is used in the ITS-ASL-ELCP. The private link address or the group broadcast link address is specified. In addition, when the group broadcast link address is specified, the ASL-SDU is delivered as the broadcast mode. Link address is defined at 2.3.

The parameter “parameter” stores the ASL-SDU passed from the ITS-LPCP in the transmitting station.

3.1.2.1.2.2 **Data Arrival Notify Primitive**

(1) Function

This primitive is a service primitive, which notifies of the arrival of the ASL-SDU from the remote station.

(2) When generated

When the arrival of the ASL-SDU is shown, the ITS-ASL-ELCP generates this primitive.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

SendDataUnit.indication (linkAddress, parameter)

The parameter “linkAddress” stores the link address, which is used in the ITS-ASL-ELCP. The private link address or the group broadcast link address is specified.

The parameter “parameter” stores the arrival of the ASL-SDU.

3.1.2.1.2.3 **Area Specification Data Transfer Request Primitive**

(1) Function

This primitive is a service primitive that requests that the ASL-SDU be transmitted to a mobile station in the target area.

(2) When generated

The upper layer server's ITS-LPCP always generates this primitive.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

SendDataUnitForArea.request (linkAddress, parameter, areaList, messageId)

The parameter “linkAddress” contains the link address, which is used in the ITS-ASL-ELCP. The private link address or the group broadcast link address can be specified. In addition, when the group broadcast link address is specified, the ASL-SDU is delivered via broadcast mode. Link address is defined in 2.3.

The parameter “parameter” contains the ASL-SDU passed from the ITS-LPCP in the transmitting station.

The parameter "areaList" contains a list of area identifiers that indicate transmission target areas.

The parameter "messageId" contains the identifier that identifies the transmitted message.

3.1.2.1.2.4 **Data Stop Request Primitive**

(1) Function

This primitive is a service primitive that requests that transmission of the ASL-SDU be stopped.

(2) When generated

The upper layer server's ITS-LPCP always generates this primitive.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

StopDataUnit.request (messageId)

The parameter "messageId" contains the identifier that identifies the message for which to stop transmission.

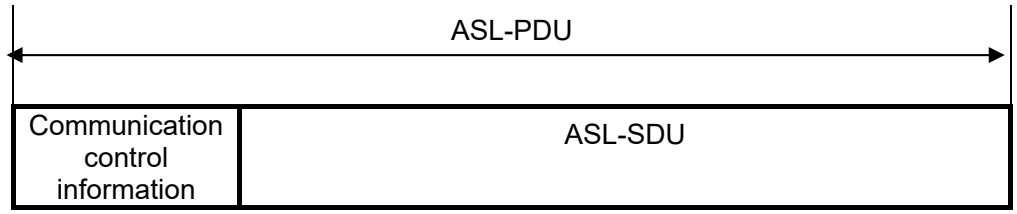
3.1.2.2 **Protocol Data Unit (PDU)**

3.1.2.2.1 **PDU Format**

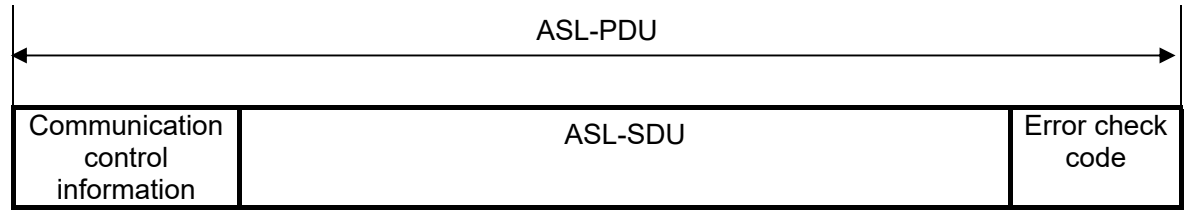
The Protocol data unit of communication control (ASL-PDU: ASL Protocol Data Unit) shown below is the ITS-ASL PDU. The PDU consists of the control field, which controls the control information for directing the procedure of the ITS-ASL-ELCP, and the information field, which contains the ASL-SDU passed from the ITS-LPCP. The content of the control field varies depending on the lower layer.

3.1.2.2.1.1 **ITS FORUM RC-005**

The configuration of the PDU in the case of ITS FORUM RC-005 is shown in Figure 3.1. The PDU has communication control information as the control field.



(a) PDU Format for Individual Communication

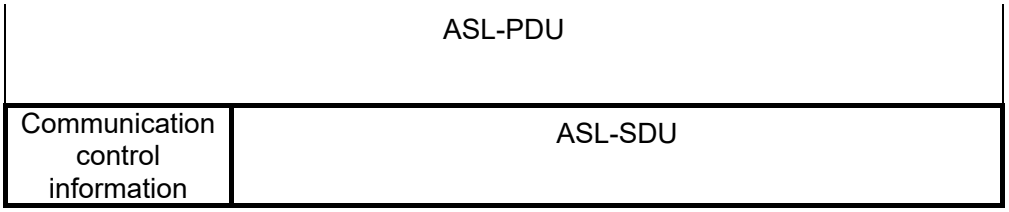


(b) PDU Format for Broadcast Communication

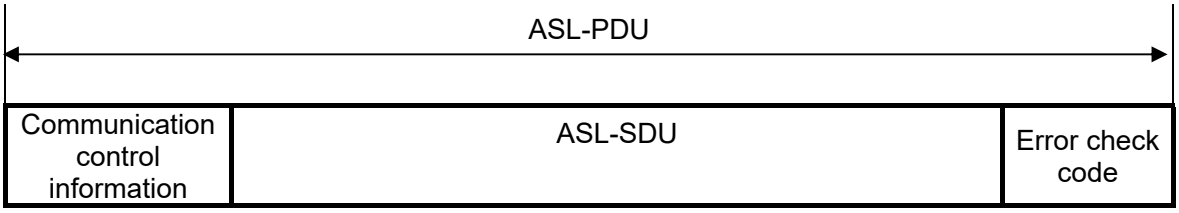
Figure 3.1 Configuration of the ASL-PDU (ITS FORUM RC-005)

3.1.2.2.1.2 **IEEE 802.11**

The configuration of the PDU in the case of IEEE 802.11 is shown in Figure 3.1. The PDU has communication control information as the control field.



(a) PDU Format for Individual Communication

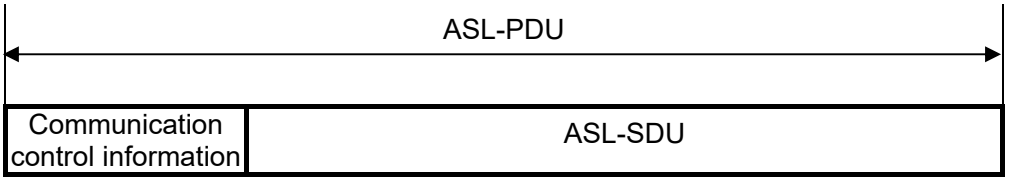


(b) PDU Format for Broadcast Communication

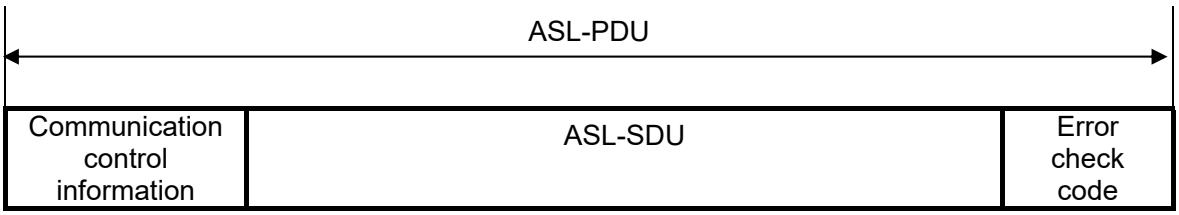
Figure 3.1 Configuration of the ASL-PDU (IEEE 802.11)

3.1.2.2.1.3 **ARIB STD-T104**

The configuration of the PDU in the case of ARIB STD-T104 is shown in Figure 3.1-6. The PDU has communication control information as the control field.



(a) PDU Format for Individual Communication

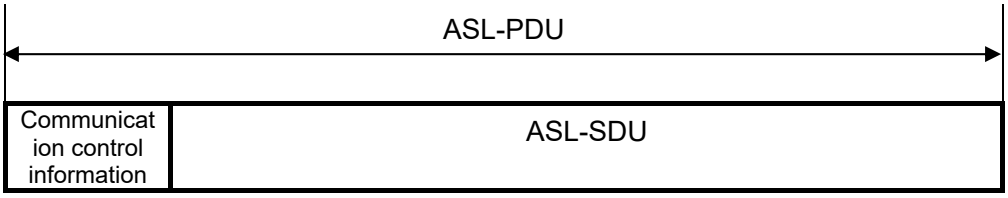


(b) PDU Format for Broadcast Communication

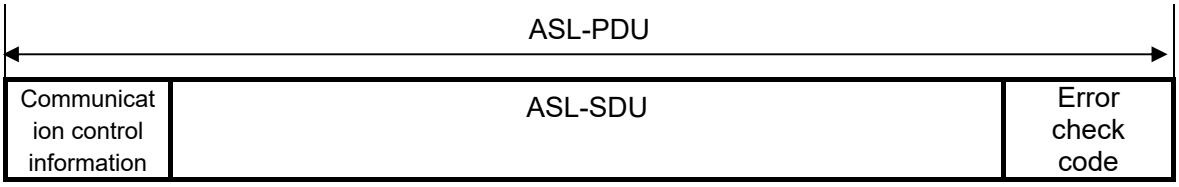
Figure 3.1-6 Configuration of the ASL-PDU (ARIB STD-T104)

3.1.2.2.1.4 **ARIB STD-T109**

The configuration of the PDU in the case of ARIB STD-T109 is shown in Figure 3.1-7. There is communication control information in the control field.



(a) PDU Format for Individual Communication

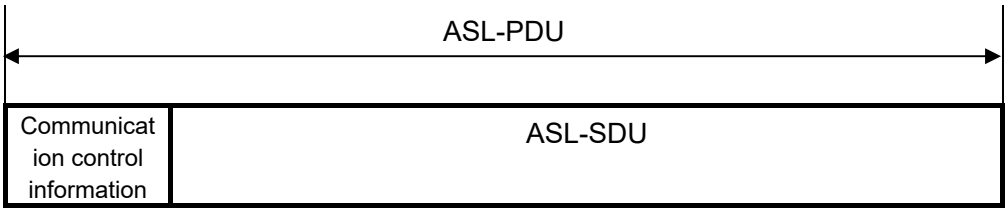


(b) PDU Format for Broadcast Communication

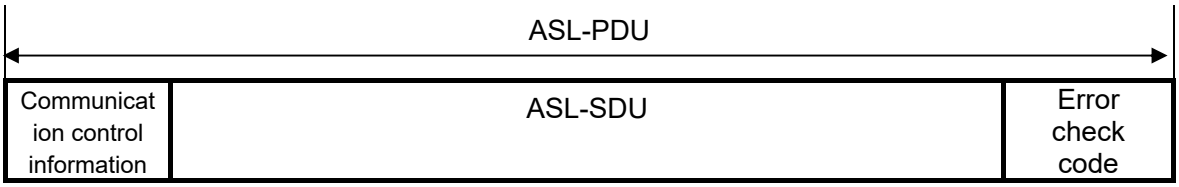
Figure 3.1 Configuration of the ASL-PDU (ARIB STD-T109)

3.1.2.2.1.5 **ARIB STD-T120**

3.1.2.2.1.5 The configuration of the PDU in the case of ARIB STD-T120 is shown in Figure 3.1-8. There is communication control information in the control field.



(a) PDU Format for Individual Communication



(b) PDU Format for Broadcast Communication

Figure 3.1 Configuration of the ASL-PDU (ARIB STD-T120)

3.1.2.2.2 **PDU Elements**

3.1.2.2.2.1 **Connection Identification**

The link address, the element identifier EID (Element Identifier), and so on for identification of connections is transferred as a parameter of the service primitive or content of ASL-SDU according to the lower layer.

3.1.2.2.2.2 **Format of the Control Field**

3.1.2.2.2.2.1 **ITS FORUM RC-005**

The communication control information is contained in the control field for directing the procedure of the ITS-ASL-ELCP. This content is specified in sub-clause 3.1.2.3.1.

3.1.2.2.2.2.2 **IEEE 802.11**

The communication control information is contained in the control field for directing the procedure of the ITS-ASL-ELCP. This content is specified in sub-clause 3.1.2.3.1.

3.1.2.2.2.2.3 **ARIB STD-T104/ARIBSTD-T120**

The communication control information is contained in the control field for directing the procedure of the ITS-ALS-ELCP. This content is specified in sub-clause 3.1.2.3.1.

3.1.2.2.2.2.4 **ARIB STD-T109**

The communication control information is contained in the control field for directing the procedure of the ITS-ASL-ELCP. This content is specified in sub-clause 3.1.2.3.1.

3.1.2.2.2.3 **Format of the Information Field**

The ASL-SDU passed from the ITS-LPCP is divided into bulk segments or is penetrated and stored in the ASL-SDU field.

3.1.2.2.2.4 **Format of the Error Check Code (Checksum)**

In the case of broadcast communications, a 32-bit checksum is added to the end of the ASL-SDU as an error check code. The checksum is a 32-bit one complement sum for the ASL-SDU. If the end of ASL-SDU is less than the check bit length (32 bits), it is calculated by inserting 0 into the lower digits.

3.1.2.3 **Procedure Elements of the Extended Communication Control**

3.1.2.3.1 **Communication Control Information Format**

3.1.2.3.1.1 **ITS FORUM RC-005, IEEE802.11, and ARIB STD-T104/ARIB STD-T120**

The communication control information is contained the control field for directing the procedure of the ITS-ASL-ELCP, and it is held in common between the base station and the mobile station in order to perform procedure. The format of this communication control information field is shown in Table 3.1.

In addition, this communication control information is defined by the parameter “aslLinkProtocol” of the “AslPDU” type, which define the ASL-PDU format of the ITS-ASL-ELCP.

Table 3.1 Communication Control Information (aslLinkProtocol) Field Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Preamble	bulk Enable	bulkTer mination	pduGroup				
2	SegmentNumber							
3	broadcastParameter (optional)							
	dummy				serviceTime			
4	serviceTime							

(1) Bulk transmission valid identifier (bulkEnable)

This identifier indicates that bulk transmission is valid or invalid.

When an ASL-PDU is bulk segment, which is bulk transferred, this identifier of the ASL-PDU shall be true.

When the ASL-PDU shall not process segmentation, this identifier is false.

(2) Valid identifier of the bulk transmission (bulkTermination)

This identifier indicates the last segment of bulk transmission.

When an ASL-PDU is the last segment, which is bulk transferred, this identifier of the ASL-PDU shall be true.

When the ASL-PDU shall not process segmentation, this identifier is false.

(3) PDU group number (pduGroup)
This number is the identification information of bulk segment.

A PDU group number is assigned to the ASL-SDU before division processing for the bulk transmission. The number shall be assigned by modulo “32” for each sending queue. And a number shall be also assigned to the ASL-SDU, which is not adapted in the bulk transmission.

When the received bulk segment assembles the ASL-SDU, the segment of the same PDU group number shall be used.

(4) Segment number (segmentNumber)
This number is a serial number, which shows the division order of a bulk segment.

The segment number of first divided bulk segment shall be set to “0”, and the serial number shall be set to the value which is incremented in order. When the bulk segment is assembled, the order of the segment number is guaranteed.

When a bulk transmission is not applied, the segment number shall be set to “0”.

(5) Auxiliary parameter for the broadcast mode (broadcastParameter)
The Auxiliary parameter for the broadcast mode is an option. When the ASL-SDU is segmented on the broadcast mode processing, the following parameter is added to the communication control information of the bulk segment.

(a) Value of connection guard timer (serviceTime)
This value shall be set to the connection timer of the mobile station (CTO: Connection Timer for OBU) in applicable communication area. When the point-to-point communication is not performed (does not process association procedure), the mobile station manages the validity time of broadcast service using this value. A unit shall be millisecond, and the value shall be set to range of “0” to “4095”. In addition, when it shows that the time is infinite, the value shall be set to “0”.

However, setting "0" is only for testing, and a mobile station that received “0” in actual operation state should discard the ASL-PDU.

NOTE: When the point-to-point communication is performed in parallel with broadcast communication, the mobile station uses the timer value given by the ASL base station profile.

3.1.2.3.1.2 **ARIB STD-T109**

The communication control information is contained in the control field for directing the procedure of the ITS-ASL-ELCP and it is held in common between the base station and the mobile station in order to perform procedures. The format of this communication control information field is shown in Table 3.1. Explanations for the same parameters in Table 3.1 and Table 3.1 have been omitted.

In addition, this communication control information is defined by the parameter “aslLinkProtocol” of the “AslPDU” type, which defines the ASL-PDU format of the ITS-ASL-ELCP.

Table 3.1 Communication Control Information (aslLinkProtocol) Field Format

<table>
<tr><td></td><td>7 (MSB)</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0 (LSB)</td></tr>
<tr><td>1</td><td>Preamble</td><td>bulk Enable</td><td>bulkTer mination</td><td colspan="5">pduGroup</td></tr>
<tr><td>2</td><td colspan="8">SegmentNumber</td></tr>
<tr><td>3</td><td colspan="8">broadcastParameter (optional)</td></tr>
<tr><td></td><td colspan="4">Reserved (0)</td><td colspan="4">serviceTime</td></tr>
<tr><td>4</td><td colspan="8">serviceTime</td></tr>
<tr><td>5</td><td colspan="8" rowspan="6">destinationLinkAddress</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
</table>

(6) Destination link address (destinationLinkAddress)

This indicates the destination link address.

It contains the value stored in the linkAddress parameter of the data transmission request primitive (SendDataUnit.request) sent from ITS-LPCP.

For private addresses, it is used to determine if the destination link address matches the transmitting station’s link address.

3.1.2.3.2 **Communication Control Parameters**

3.1.2.3.2.1 **Maximum Receiving Data Unit Length of the ITS-ASL-ELCP (MRU)**

The MRU is the maximum receiving length of the data when the ASL-ELCP receives from the ITS-LPCP. The MRU of the ITS-ASL-ELCP shall be the larger value in the maximum transmission unit (MTU: Maximum Transmission Unit), which is specified by the implemented ITS-LPCP.

3.1.2.3.2.2 **Segment Unit for Unicast of Bulk Transmissions (SUU)**

The SUU is a unit that segments the ASL-SDU when bulk transmission is performed. This unit shall be set to the following value according to the selected profile. The value of the SUU shall be set according to the lower layer.

3.1.2.3.2.3 **Segment Unit of the Broadcast Mode Control (SUM)**

The SUM is a unit that segments the ASL-SDU when bulk transmission is performed on broadcast mode control. This unit is set to the following value according to the selected profile. The value of the SUM shall be set according to the lower layer.

3.1.2.3.2.4 **Repetition Transmitting Number of the Broadcast Mode Control (k)**

The k is a number of times, which the ASL-PDU sends repeatedly on the broadcast mode control. Since the receiving error rate of the data, which is sent by broadcast, is different value according to the value of k, it needs to be decided in consideration of the reliability of a system.

3.1.2.3.3 **Communication Services from Lower layer Interface**

The specification of the primitives provided for by the lower layer used by ITS-ASL-ELCP is shown in this sub-clause.

3.1.2.3.3.1 **Communication Interface with ITS FORUM RC-005**

(1) WSM-WaveShortMessage.request

The upper layer entity requests transmission of WAVE Short Message.

WSM-WaveShortMessage.request (Channel Identifier, DataRate,
Transmit Power Level, ProviderServiceIdentifier, User Priority, WsmExpiryTime,
Length, Data, Peer MAC address, WSMP header extensions, WAVE Element ID)

The parameter Channel Identifier indicates the channel number to be used for this transmission.

The parameter DataRate indicates the data rate to be used for this transmission.

The parameter Transmit Power Level indicates the power level in dBm to be used for this transmission.

The parameter ProviderServiceIdentifier is used to construct the WSM.

The parameter User Priority indicates priority specified in IEEE 802.11.

The parameter WsmExpiryTime indicates the time at which the message is no longer valid.

The parameter Length indicates the length of WSM data.

The parameter Data indicates the content of WSM data.

The parameter Peer MAC Address indicates destination MAC address specified in IEEE 802.11.

The parameter WSMP header extensions indicates the description contained in the WSMP header extension fields.

The parameter WAVE Element ID indicates the type of WSM data contained, to assist receive-side processing.

Table 3.1 shows the types, valid ranges and stored values of each parameter.

Table 3.1 Description of parameter (WSM-WaveShortMessage.request)

Name	Type	Valid range	Value
Channel Identifier	INTEGER	0 to 200	See IEEE 1609.3
DataRate	INTEGER	2 to127	See IEEE 1609.3
Transmit Power Level	SIGNED INTEGER	-127 to 127	See IEEE 1609.3
ProviderServiceIdentifier	OCTET STRING	-	See IEEE 1609.3
User Priority	INTEGER	0 to 7	See IEEE 802.11-2012
WsmExpiryTime	INTEGER	0 to 2^{64}-1	See IEEE 1609.3
Length	INTEGER	1 to 65535	length of “parameter”
Data	OCTET STRING	-	“parameter”
Peer MAC Address	MACAddress	-	“linkAddress”
WSMP header extensions	BIT STRING	-	See IEEE 1609.3
WAVE Element ID	INTEGER	128 to 255	128 (WAVE Short Message)

(2) WSM-WaveShortMessage.indication

The upper layer entity receives notification of reception of WAVE Short Message.

WSM-WaveShortMessage.indication (WsmpVersion, Channel Number, DataRate,
Transmit Power Used, ProviderServiceIdentifier, User Priority, Length, Data,
Peer MAC address)

The parameter WsmpVersion is extracted from the WSMP header

The parameter Channel Number is extracted from the WSMP header.

The parameter DataRate is extracted from the WSMP header.

The parameter Transmit Power Used is extracted from the WSMP header if present.

The parameter ProviderServiceIdentifier is extracted from the WSMP header.

The parameter User Priority is extracted from the DLUNITDATA.indication priority.

The parameter Length is extracted from the WSMP header.

The parameter Data is extracted from the WSMP header.

The parameter Peer MAC Address is extracted from the DLUNITDATA.indication source_address.

Table 3.1 shows the types, valid ranges and stored values of each parameter.

Table 3.1 Description of parameter (WSM-WaveShortMessage.indication)

Name	Type	Valid range	Value
WsmpVersion	INTEGER	0 to 15	See IEEE 1609.3
Channel Number	INTEGER	0 to 200	See IEEE 1609.3
DataRate	INTEGER	2 to 127	See IEEE 1609.3
Transmit Power Level	SIGNED INTEGER	-127 to 127	See IEEE 1609.3
ProviderServiceIdentifier	OCTET STRING	-	See IEEE 1609.3
User Priority	INTEGER	0 to 7	See IEEE 802.11-2012
Length	INTEGER	1 to 65535	length of “parameter”
Data	OCTET STRING	-	“parameter”
Peer MAC Address	MACAddress	-	“linkAddress”

3.1.2.3.3.2 **Communication Interface with IEEE 802.11/ARIB STD-T120(PC5)**

(1) WSM-WaveShortMessage.request

The upper layer entity requests transmission of WAVE Short Message.

WSM-WaveShortMessage.request (Channel Identifier, DataRate,
Transmit Power Level, ProviderServiceIdentifier, User Priority, WsmExpiryTime,
Length, Data, Peer MAC address, WSMP header extensions, WAVE Element ID)

The parameter Channel Identifier indicates the channel number to be used for this transmission.

The parameter DataRate indicates the data rate to be used for this transmission.

The parameter Transmit Power Level indicates the power level in dBm to be used for this transmission.

The parameter ProviderServiceIdentifier is used to construct the WSM.

The parameter User Priority indicates priority specified in IEEE 802.11.

The parameter WsmExpiryTime indicates the time at which the message is no longer valid.

The parameter Length indicates the length of WSM data.

The parameter Data indicates the content of WSM data.

The parameter Peer MAC Address indicates destination MAC address specified in IEEE 802.11.

The parameter WSMP header extensions indicates the description contained in the WSMP header extension fields.

The parameter WAVE Element ID indicates the type of WSM data contained, to assist receive-side processing.

Table 3.1 shows the types, valid ranges and stored values of each parameter.

Table 3.1 Description of parameter (WSM-WaveShortMessage.request)

Name	Type	Valid range	Value
Channel Identifier	INTEGER	0 to 200	See IEEE 1609.3
DataRate	INTEGER	2 to127	See IEEE 1609.3
Transmit Power Level	SIGNED INTEGER	-127 to 127	See IEEE 1609.3
ProviderServiceIdentifier	OCTET STRING	-	See IEEE 1609.3
User Priority	INTEGER	0 to 7	See IEEE 802.11-2012
WsmExpiryTime	INTEGER	0 to 2^{64}-1	See IEEE 1609.3
Length	INTEGER	1 to 65535	length of “parameter”
Data	OCTET STRING	-	“parameter”
Peer MAC Address	MACAddress	-	“linkAddress”
WSMP header extensions	BIT STRING	-	See IEEE 1609.3
WAVE Element ID	INTEGER	128 to 255	128 (WAVE Short Message)

(2) WSM-WaveShortMessage.indication

The upper layer entity receives notification of reception of WAVE Short Message.

WSM-WaveShortMessage.indication (WsmpVersion, Channel Number, DataRate, Transmit Power Used, ProviderServiceIdentifier, User Priority, Length, Data, Peer MAC address)

The parameter WsmpVersion is extracted from the WSMP header

The parameter Channel Number is extracted from the WSMP header.

The parameter DataRate is extracted from the WSMP header.

The parameter Transmit Power Used is extracted from the WSMP header if present.

The parameter ProviderServiceIdentifier is extracted from the WSMP header.

The parameter User Priority is extracted from the DLUNITDATA.indication priority.

The parameter Length is extracted from the WSMP header.

The parameter Data is extracted from the WSMP header.

The parameter Peer MAC Address is extracted from the DLUNITDATA.indication source_address.

Table 3.1 shows the types, valid ranges and stored values of each parameter.

Table 3.1 Description of parameter (WSM-WaveShortMessage.indication)

Name	Type	Valid range	Value
WsmpVersion	INTEGER	0 to 15	See IEEE 1609.3
Channel Number	INTEGER	0 to 200	See IEEE 1609.3
DataRate	INTEGER	2 to 127	See IEEE 1609.3
Transmit Power Level	SIGNED INTEGER	-127 to 127	See IEEE 1609.3
ProviderServiceIdentifier	OCTET STRING	-	See IEEE 1609.3
User Priority	INTEGER	0 to 7	See IEEE 802.11-2012
Length	INTEGER	1 to 65535	length of “parameter”
Data	OCTET STRING	-	“parameter”
Peer MAC Address	MACAddress	-	“linkAddress”

3.1.2.3.3.3 **Communication Interface with ARIB STD-T104/ARIB STD-T120(Uu)**

(1) HTTP request

The mobile station upper layer entity requests transmission of an HTTP request and issues notification of reception of an HTTP request to the upper layer server's upper layer entity. Either the GET or POST methods are used, and the request contains the following parameters.

(a) Latitude

This value indicates the latitude of the mobile station. It is expressed in units of 0.0000001 degrees (10^{-7} degrees), with a value between -900,000,000 and 900,000,000, indicating the range between -90 degrees (90 degrees south latitude) and 90 degrees (90 degrees north latitude). If the latitude is unknown, the value is set to -2,147,483,648 (0x80000000).

(b) Longitude

This value indicates the longitude of the mobile station. It is expressed in units of 0.0000001 degrees (10^{-7} degrees), with a value between -1,800,000,000 and 1,800,000,000, indicating the range between -180 degrees (180 degrees west longitude) and 180 degrees (180 degrees east longitude). If the longitude is unknown, the value is set to -2,147,483,648 (0x80000000).

(c) Elevation

This value indicates the elevation of the mobile station. It is expressed in units of 0.1m, with a value between -4,095 and 61,439, indicating the range between -409.5m and 6143.9m. If the elevation is unknown, the value is set to -4096 (0xF000).

(d) Speed

This value indicates the speed of the mobile station. It is expressed in units of 0.01m/s, with a value between 0 and 16,383, indicating the range between 0.00m/s and 163.83m/s. If the speed is unknown, the value is set to 65535 (0xFFFF).

(e) Heading

This value indicates the heading of the mobile station. It is expressed in units of 0.0125 degrees, with a value between 0 and 28,799, indicating the range between 0 degrees, indicating north, and 359.9875 degrees, rotating clockwise from 0 degrees. If the heading is unknown, the value is set to 65535 (0xFFFF).

(f) Data

This contains binary data indicating the content of the ASL-PDU.

(2) HTTP response

The upper layer server's upper layer entity requests the transmission of an HTTP response, and issues notification of reception of an HTTP response to the mobile station's upper layer entity. It has the following parameters.

(a) Data

This contains binary data indicating the content of the ASL-PDU. Multiple ASL-PDU data can be contained in a single HTTP response by setting the Content-Type to "multipart/mixed."

3.1.2.3.3.4 **Communication interface with ARIB STD-T109**

(1) Request

The upper layer entity requests transmission of application data.

(a) Base station

BaseSationBroadcastData request (
SequenceNumber, ControlInformation, SecurityClassification,
SecurityInformation, ApplicationAssociatedInformation,
ApplicationDataLength, ApplicationData, LinkAddress)

(b) Mobile station

MobileSationBroadcastData request (
ControlInformation, SecurityClassification,
SecurityInformation, ApplicationAssociatedInformation,
ApplicationDataLength, ApplicationData, LinkAddress)

The parameter SequenceNumber indicates the total number of packets and order of packets generated by the base station application each time.

The parameter ControlInformation indicates wireless communication parameter control information.

The parameter SecurityClassification indicates security classification information.

The parameter SecurityInformation indicates security-related information.

The parameter ApplicationAssociatedInformation indicates application-related information.

The parameter ApplicationDataLength indicates the application data length.

The parameter ApplicationData indicates the application data.

The parameter LinkAddress indicates the link address of the destination terminal.

Table 3.1 shows the types, valid ranges and stored values of each parameter.

Table 3.1 Description of parameters (BaseSationBroadcastData request/MobileSationBroadcastData request)

Name	Type	Valid range	Value
SequenceNumber	INTEGER	—	See ARIB STD-T109
ControlInformation	INTEGER	0-15	See ARIB STD-T109
SecurityClassification	INTEGER	0-1	See ARIB STD-T109
SecurityInformation	OCTET STRING	—	See ARIB STD-T109
ApplicationAssociatedInformation	OCTET STRING	—	See ARIB STD-T109
ApplicationDataLength	INTEGER	0-65535	See ARIB STD-T109
ApplicationData	OCTET STRING	—	See ARIB STD-T109
LinkAddress	MACAddress	—	linkAddress

(2) Notification

The upper layer entity receives notification of reception of application data.

(a) Base station

BaseSationBroadcastData notification (
SecurityClassification, SecurityInformation, ApplicationAssociatedInformation,
ApplicationDataLength, ApplicationData, LinkAddress)

(b) Mobile station

MobileSationBroadcastData notification (
SecurityClassification SecurityInformation, ApplicationAssociatedInformation,
ApplicationDataLength, ApplicationData, LinkAddress)

The parameter SecurityClassification indicates security classification information.

The parameter SecurityInformation indicates security-related information.

The parameter ApplicationAssociatedInformation indicates application-related information.

The parameter ApplicationDataLength indicates the application data length.

The parameter ApplicationData indicates the application data.

The parameter LinkAddress indicates information regarding the sender’s ID code.

Table 3.1 shows the types, valid ranges and stored values of each parameter.

Table 3.1 Description of parameters (BaseSationBroadcastData notification/MobileSationBroadcastData notification)

Name	Type	Valid range	Value
SecurityClassification	INTEGER	0-1	See ARIB STD-T109
SecurityInformation	OCTET STRING	—	See ARIB STD-T109
ApplicationAssociatedInformation	OCTET STRING	—	See ARIB STD-T109
ApplicationDataLength	INTEGER	0-65535	See ARIB STD-T109
ApplicationData	OCTET STRING	—	See ARIB STD-T109
LinkAddress	MACAddress	—	linkAddress

3.1.2.4 **ITS-ASL-ELCP Procedure**

3.1.2.4.1 **ITS FORUM RC-005・IEEE 802.11/ARIB STD-T120(PC5)**

3.1.2.4.1.1 **Data Transfer Service Process Procedure**

(1) Data transfer

When the request primitive of data transmitting (SendDataUnit.request) is called from the ITS-LPCP, the ITS-ASL-ELCP gets the ASL-SDU from the parameter “parameter”.

When the content of the parameter “linkAddress” is a private link address, the ITS-ASL-ELCP applies the bulk mode control procedure of the sending side to the acquired ASL-SDU as specified in sub-clause 0(1), and the generated ASL-PDU is added to the transmission queue.

When the content of the parameter “linkAddress” is a broadcast link address, the ITS-ASL-ELCP applies the broadcast mode control procedure to the acquired ASL-SDUas specified by sub-clause 0(1), and the generated ASL-PDU is added to the broadcast transmission queue. All ASL-PDU stored in the transmission queue for the broadcast are transmitted in order of the PDU group number; this processing is repeated “k” times. When "k" repetitions are completed, all segments of this PDU group number are discarded.

However, in the following case, the ASL-SDU is assumed to be invalid and data processing is not performed.

(a) When the size of the ASL-SDU passed by the parameter “parameter” exceeds the MRU of the ASL-ELCP, the request primitive is discarded. Notice of the condition that “the size of data exceeded the maximum of a buffer” is provided by the notice event primitive (EventInfomation.indication) to the ASL-NCP which requested the transmission.

(b) When the transmission queue is full, the ASL-SDU is discarded. Notice of the condition that “the transmission queue is full, the request of transmission is failed” is provided by the notice event primitive (EventInfomation.indication) to the ASL-NCP which requested the transmission.

(2) Data reception

When the ASL-SDU is obtained from the result of the bulk transmitting processing of the receiving side shown in sub-clause 3.1.2.4.1.2 (2) or the broadcast control mode processing of the receiving side shown in sub-clause 3.1.2.4.1.3 (2), the ASL-SDU is distributed to the ITS-LPCP using the notice primitive of the data arrival (SendDataUnit.indication).

The ASL-SDU, which is passed to the ITS-LPCP, is stored in the parameter “parameter” deleting communication control information, peer station identifier and sending source identifier.

The link address of the ASL-SDU is stored in the parameter “linkAddress”.

3.1.2.4.1.2 Bulk Transmission Control Procedure

(1) Bulk Transmission Process of the Sending Side

When the size of the ASL-SDU is smaller than the SUU, the ITS-ASL-ELCP adds the communication control information of the invalid bulk transmission mode to the acquired ASL-SDU and generates the ASL-PDU according to the rules in sub-clause 3.1.2.3.1.

When the size of the ASL-SDU is larger than the SUU, the ASL-SDU is divided into bulk segments according to the size of SUU sequentially from the beginning. The ITS-ASL-ELCP adds the communication control information to each bulk segment, and the ITS-ASL-ELCP generates the ASL-PDU according to the rules in 3.1.2.3.1.

(2) Bulk Transmission Process of the Receiving Side

In the bulk transmission processing of the receiving side, the following processes are performed with reference to the communication control information in the ASL-PDU stored in the receiving queue.

When the bulk transmission valid identifier in the bulk segments is true, all segments have the same PDU group number and source identifier, and all ASL-PDU from the segment for which the segment number is “0” to the segment of which the bulk transmission termination identifier is true are stored in the receiving queue, and the ITS-ASL-ELCP connects the segments in order of the segment numbers and reproduces the ASL-SDU.

When the bulk transmission valid identifier is false, the ITS-ASL-ELCP deletes the communication control information from the ASL-PDU and reproduces the ASL-SDU.

3.1.2.4.1.3 Broadcast Mode Control Procedure

(1) Broadcast Mode Control Process of the Sending Side

When the link address in the ASL-SDU is the link address of the group broadcast, the ITS-ASL-ELCP performs the following processes using the broadcast mode control.

The ITS-ASL-ELCP adds the error check code shown in sub-clause 3.1.2.2.2, and segments the ASL-SDU into the ASL-PDU according to the procedure of the bulk transmission process on the sending side shown in sub-clause 3.1.2.4.2(1). However, the SUM is applied to the unit of segmentation.

In this case, the PDU group number of the communication control information is added to the segment assign the consecutive numbers by modulo 32. The option field of the communication control information is valid and the auxiliary parameter for broadcast mode is added to the communication control information of the bulk segment.

Further, the link address of the ASL-PDU is set to the broadcast link address value of “0” (the first octet is “1000 0000”).

(2) Broadcast Mode Control Process of the Receiving Side

The following processes are conducted according to the communication control information

in the ASL-PDU stored in the receiving queue for the broadcast communication. However, the ASL-PDU of the broadcast link address value of “0” (the first octet is “1000 0000”) is processed, and the ASL-PDU that not satisfy this requirement are discarded.

When the bulk transmission valid identifier indicates false, the ITS-ASL-ELCP deletes the communication control information and the error check code from the ASL-PDU and reproduces the ASL-SDU. However, if the error check code is wrong, the ITS-ASL-ELCP discards the receive data.

When the bulk transmission valid identifier in the bulk segments is true, all segments are the same PDU group number, and all ASL-PDUs from the segment for which the segment number is “0” to the segment of which the bulk transmission termination identifier is true are stored in the receiving queue, the ITS-ASL-ELCP unifies the PDUs in order of the segment number and reproduces the ASL-SDU. However, if the error check code is wrong, the ITS-ASL-ELCP doesn’t reproduce the ASL-SDU and discards the received data. After unification is reproduced, processing is not conducted until the bulk segment of a different ASL-PDU group number is received. After the bulk segment of a different ASL-PDU group number is received, processing is re-started according to the content of communication control information.

3.1.2.4.2 **ARIB STD-T104/ARIB STD-T120(Uu)**

3.1.2.4.2.1 **Mobile Station to Upper Layer Server Data Transfer Service Process Procedure**

Figure 3.1-9 shows an example of the data transfer service process procedure for transferring data from a mobile station to an upper layer server.

(1) Data transfer from mobile station

Mobile station extended communication control is performed as follows. The mobile station generates the ASL-PDU according to the procedure in 3.1.2.4.1.1 (1) and sends it, together with the mobile station's location information, to the upper layer server via an HTTP request.

(2) Reception of data by upper layer server

Upper layer server extended communication control is performed as follows. The service provision area linked to the link address is looked up using communication control management and compared against the mobile station's location information. If the location information is within the service provision area, the received ASL-PDU is used to generate the ASL-SDU and the ASL-SDU is distributed to the ITS-LPCP using the data arrival notify primitive (SendDataUnit.indication).

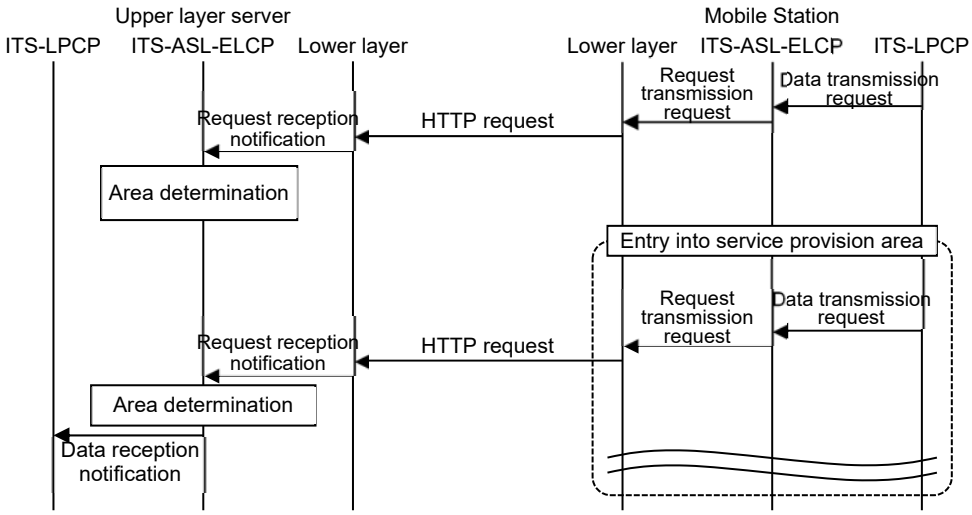


Figure 3.1-9 Example of Mobile Station to Upper Layer Server Data Transfer Service Process Procedure

3.1.2.4.2.2 **Upper Layer Server to Mobile Station Data Transfer Service Process Procedure**

Figure 3.1-10 shows an example of the data transfer service process procedure for transferring data from an upper layer server to a mobile station.

(1) Upper layer server data transmission request

When the request primitive of data transmitting (SendDataUnit.request) is called from the ITS-LPCP, the extended communication control gets the ASL-SDU from the parameter “parameter”.

When the content of the parameter “linkAddress” is a private link address, the bulk mode control procedure of the sending side is applied to the acquired ASL-SDU. This procedure is specified in 3.1.2.4.1.2(1). The generated ASL-PDU is linked to the area identifier specified in the parameter "areaList" and stored in the transmission list.

When the content of the parameter “linkAddress” is a broadcast link address, broadcast mode control processing is applied to the acquired ASL-SDU. This procedure is specified in 3.1.2.4.1.3(1). The generated ASL-PDU is linked to the area identifier specified in the parameter "areaList" and stored in the broadcast transmission list.

However, in the following case, the ASL-SDU is assumed to be invalid and data processing is not performed.

(a) When the size of the ASL-SDU passed by the parameter “parameter” exceeds the MRU of the ASL-ELCP, the request primitive is discarded. Notice of the condition that “the size of data exceeded the maximum of a buffer” is provided by the notice event primitive (EventInfomation.indication) to the ASL-NCP which requested the transmission.

(b) When the transmission queue is full, the ASL-SDU is discarded. Notice of the condition that “the transmission queue is full, the request of transmission failed” is provided by the notice event

primitive (EventInfomation.indication) to the ASL-NCP that requested the transmission.

(2) Transmission of service request message at the mobile station

When receiving a service request primitive from ITS-LPCP or an application, the communication control management on the mobile station generates a service request message and transmits the message to the upper layer server together with the mobile station's location information as an HTTP request according to a schedule decided in advance.

The specific service request message transmission schedule is an implementation issue and is not specified in this specification.

(3) Reception of service response message at the upper layer server

When receiving a service request message, the extended communication control on the upper layer server refers to the transmission list and checks the service provision area information specified when the ASL-PDU was specified.

If there is no service provision area that matches the location information included in the service request message, a service response message without an ASL-PDU stored in the transmission list is generated and sent as an HTTP response.

If there is a service provision area that matches the location information included in the service request message, service response messages are generated for all ASL-PDUs linked to the service provision area that are stored in the transmission list. These messages are sent as HTTP responses.

(4) Reception of data by mobile station

The mobile station extended communication control receives data using the procedure shown in 3.1.2.4.1.1(2). One ASL-PDU is extracted at a time, and delivery to the ITS-LPCP is performed for each reproduced ASL-SDU.

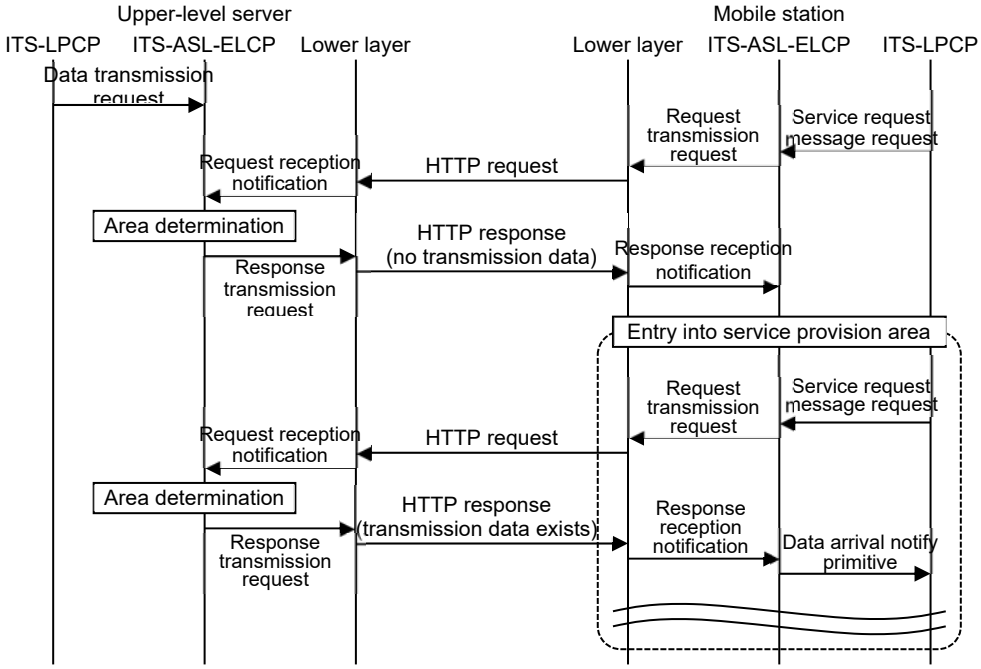


Figure 3.1-10 Example of Upper Layer Server to Mobile Station Data Transfer Service Process Procedure

3.1.2.4.3 **ARIB STD-T109**

3.1.2.4.3.1 **Data Transfer Service Process Procedure**

(1) Data transmission

When the request primitive of data transmitting (SendDataUnit.request) is called from the ITS-LPCP, the ASL-SDU is acquired from the parameter “parameter”.

When the content of the parameter “linkAddress” is a private link address, the bulk transmission process of the sending side is applied to the acquired ASL-SDU. This is specified in 3.1.2.4.1.2 (1). The generated ASL-PDU is stored in the transmission queue.

When the content of the parameter “linkAddress” is a broadcast link address, broadcast mode control processing is applied to the acquired ASL-SDU. This procedure is specified in 3.1.2.4.1.3 (1). The generated ASL-PDU is stored in the broadcast transmission queue. ASL-PDU stored in the broadcast transmission queue are transmitted in PDU order for each PDU group number. This process is repeated k times. After being repeated k times, all corresponding PDU group number segments are discarded.

However, in the following case, the ASL-SDU is assumed to be invalid and data processing is not performed.

(a) When the size of the ASL-SDU passed by the parameter “parameter” exceeds the MRU of the ASL-ELCP, the request primitive is discarded. Notice of the condition that “the size of data exceeded the maximum of a buffer” is provided by the notice event primitive

(EventInfomation.indication) to the ASL-NCP which requested the transmission.

(b) When the transmission queue is full, the ASL-SDU is discarded. Notice of the condition that "the transmission queue is full, the request of transmission failed" is provided by the notice event primitive (EventInfomation.indication) to the ASL-NCP that requested the transmission.

(2) Data reception

When the ASL-SDU is obtained from the result of the bulk transmitting processing of the receiving side shown in sub-clause 3.1.2.4.3.2 (2) or the broadcast control mode processing of the receiving side shown in sub-clause 3.1.2.4.3.3 (2), the ASL-SDU is distributed to the ITS-LPCP using the notice primitive of the data arrival (SendDataUnit.indication).

The ASL-SDU, which is passed to the ITS-LPCP, is stored in the parameter "parameter", deleting communication control information, peer station identifier and sending source identifier.

The link address of the ASL-SDU is stored in the parameter "linkAddress".

3.1.2.4.3.2 **Bulk Transmission Control Procedure**

(1) Bulk Transmission Process of the Sending Side

When the size of the ASL-SDU is smaller than the SUU, the ITS-ASL-ELCP adds the communication control information of the invalid bulk transmission mode to the acquired ASL-SDU and generates the ASL-PDU according to the rules in sub-clause 3.1.2.3.1.

When the size of the ASL-SDU is larger than the SUU, the ASL-SDU is divided into bulk segments according to the size of SUU sequentially from the beginning. The ITS-ASL-ELCP adds the communication control information to each bulk segment, and the ITS-ASL-ELCP generates the ASL-PDU according to the rules in 3.1.2.3.1.

(2) Bulk Transmission Process of the Receiving Side

In the bulk transmission processing of the receiving side, the following processes are performed with reference to the communication control information in the ASL-PDU stored in the receiving queue. The link address of the ASL-PDU to be processed is considered identical to the destination link address of the communication control information and ASL-PDU that do not match are discarded.

When the bulk transmission valid identifier in the bulk segments is true, all segments have the same PDU group number and source identifier, and all ASL-PDU from the segment for which the segment number is "0" to the segment of which the bulk transmission termination identifier is true are stored in the receiving queue, and the ITS-ASL-ELCP connects the segments in order of the segment numbers and reproduces the ASL-SDU.

When the bulk transmission valid identifier is false, the ITS-ASL-ELCP deletes the communication control information from the ASL-PDU and reproduces the ASL-SDU.

3.1.2.4.3.3 **Broadcast Mode Control Procedure**

(1) Broadcast Mode Control Process of the Sending Side

When the link address in the ASL-SDU is the link address of the group broadcast, the ITS-ASL-ELCP performs the following processes using the broadcast mode control.

The ITS-ASL-ELCP adds the error check code shown in sub-clause 3.1.2.2.2, and segments the ASL-SDU into the ASL-PDU according to the procedure of the bulk transmission process on the sending side shown in sub-clause 3.1.2.4.2(1). However, the SUM is applied to the unit of segmentation.

In this case, the PDU group number of the communication control information is added to the segment assign the consecutive numbers by modulo 32. The option field of the communication control information is valid and the auxiliary parameter for broadcast mode is added to the communication control information of the bulk segment.

Further, the link address of the ASL-PDU is set to the broadcast link address value of “0” (the first octet is “1000 0000”).

(2) Broadcast Mode Control Process of the Receiving Side

The following processes are conducted according to the communication control information in the ASL-PDU stored in the receiving queue for the broadcast communication. However, the ASL-PDU of the broadcast link address value of “0” (the first octet is “1000 0000”) is processed, and the ASL-PDU that not satisfy this requirement are discarded.

When the bulk transmission valid identifier indicates false, the ITS-ASL-ELCP deletes the communication control information and the error check code from the ASL-PDU and reproduces the ASL-SDU. However, if the error check code is wrong, the ITS-ASL-ELCP does not reproduce the ASL-SDU and discards the receive data.

After unification is reproduced, processing is not conducted until the bulk segment of a different ASL-PDU group number is received. After the bulk segment of a different ASL-PDU group number is received, processing is resumed according to the content of the communication control information.

3.1.3 **Communication Control Management**

3.1.3.1 **Management Service Interface**

3.1.3.1.1 **Primitive Relationship Overview**

The communication control management of the ITS-ASL-ELCP provides the following management services to the ITS-LPCP.

(1) Event notify service

The event notify service provides the following primitive.

EventInformation.indication

To notify the events such as errors, etc. occurred within the ITS-ASL-ELCP, the “EventInformation.indication” is passed from the ITS-ASL-ELCP to the ITS-LPCP of the other party station or its own station.

(2) Connection management service

The connection management service provides the following primitive.

Connection.request
SetConnectionStatus.request
ConnectionForArea.request
Service.request
AreaSetting.request
AreaSetting.indication

To request the start of connections with surrounding stations, the “Connection.request” is passed from the ITS-LPCP or application to the ITS-ASL-ELCP. To request setting of the connection status, the “SetConnectionStatus.request” is passed from the ITS-LPCP or application to the ITS-ASL-ELCP.

To request the start of connections with mobile stations, the “ConnectionForArea.request” is passed from the ITS-LPCP or application to the ITS-ASL-ELCP for the designated service provision area. To request the transmission of service requests from the mobile station to the upper layer server, the “Service.request” is passed from the ITS-LPCP or application to the ITS-ASL-ELCP. "AreaSetting.request" is used to request the setting or updating of service provision areas. "AreaSetting.indication" is used to provide notification of area identifiers as the result of the setting or updating of service provision areas.

3.1.3.1.2 **Service Content Specification**

This sub-clause specifies the primitives and parameters related with the management services. The parameters as “interface” (interfacing factors) are described conceptually and the information needed for the receiving end entity is defined. But the detailed realization method to pass this information is not restricted.

The logical relationship among management service primitives that the ITS-ASL-ELCP provides for the ITS-LPCP is shown in Figure 3.1.

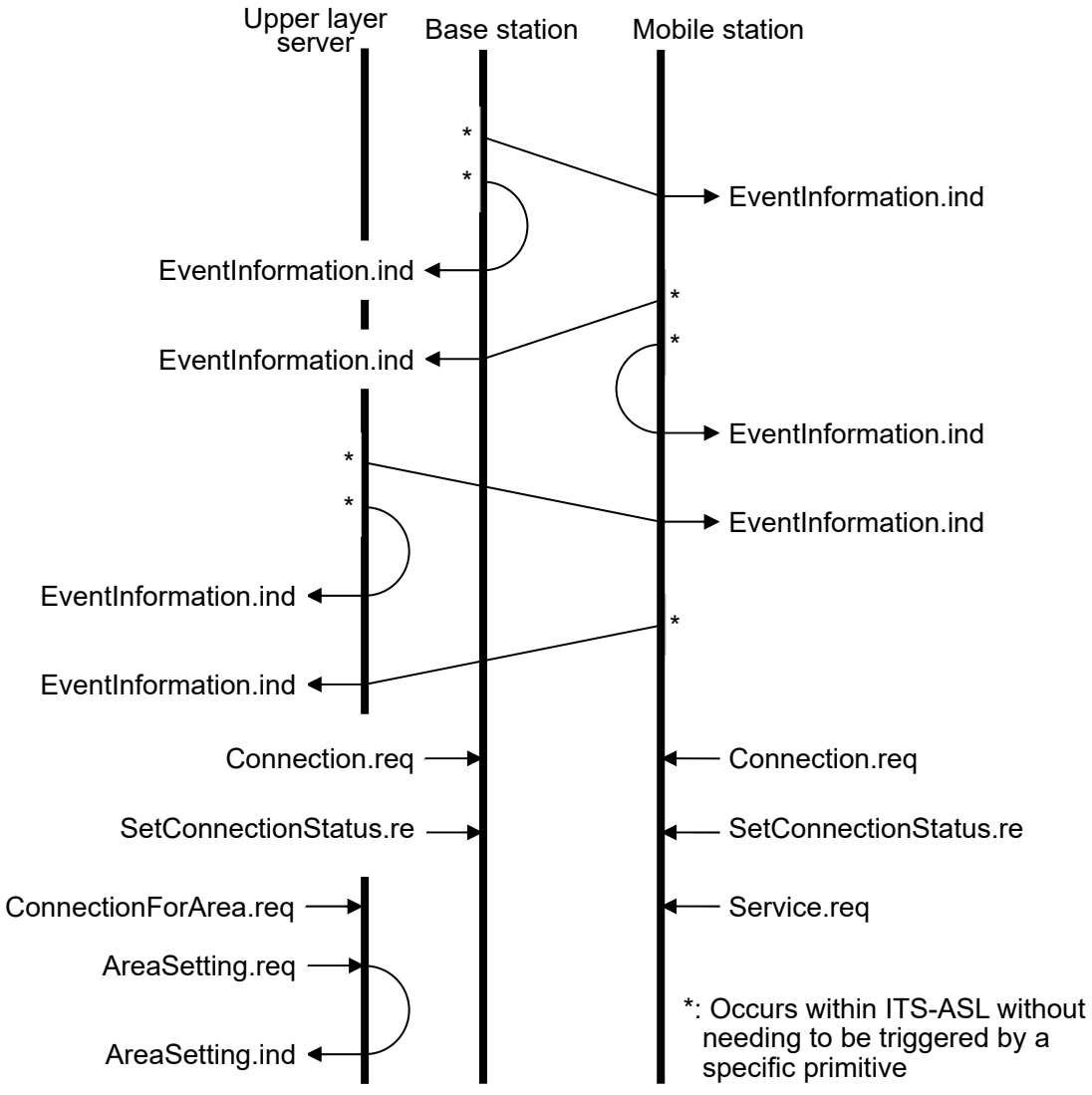


Figure 3.1-11 Logical Relationship among Management Service Primitives

3.1.3.1.3 **Event Notify Services**

These services notify events such as errors, etc. occurred within the ITS-ASL-ELCP, and the ITS-ASL-ELCP provide the following primitives.

NOTE: This primitive process such events occurred within the ASL-ELCP, therefore, the definition of both “PDU coding according to the DsrcControlPDU type in the event notification service” and “the access control information (Member access control according to the DsrcControlPDU type)” is given in the definition of ITS-ASL-ELCP.

3.1.3.1.3.1 **Event Notify Primitive**

(1) Function

This primitive is a service primitive that notifies events such as errors, etc. occurred within the ITS-ASL-ELCP.

(2) When generated

The ITS-ASL-ELCP generates this primitive when an event such as error, etc. occurred within the ITS-ASL-ELCP is notified.

(3) Parameters of service primitive

This primitive shall provide the following parameter.

EventInformation.indication (linkAddress, status, [extensionParameter])

The parameter “linkAddress” contains the link address to be used to identifier peer station in the ITS-ASL-ELCP.

The parameter “status” contains the code to indicate an event occurred.

The parameter “extensionParameter” contains a piece of information to supplement the contents of the parameter “status” as needed. This parameter is optional.

3.1.3.1.4 **Connection management service**

These services manage connection status among surrounding stations, and the ITS-ASL-ELCP provide the following primitives.

3.1.3.1.4.1 **Connection Request Primitive**

(1) Function

This primitive is a service primitive that requests the start of connections with surrounding stations.

(2) When generated

This primitive is always generated by the upper protocol.

(3) Parameters of service primitive

This primitive provides the following parameter.

Connection.request (linkAddress)

The parameter “linkAddress” contains the link address to be used to identify the peer station in the ITS-ASL-ELCP.

3.1.3.1.4.2 **Connection Status Containing Request Primitive**

(1) Function

This primitive is a service primitive that requests that the connection status be set.

(2) When generated

This primitive is always generated by the upper protocol.

(3) Parameters of service primitive

This primitive provides the following parameter.

SetConnectionStatus.request (portNo, linkAddress, status)

The parameter “portNumber” contains the identifier to identify requested application.
The parameter “linkAddress” contains the link address to be used to identify the peer station in the ITS-ASL-ELCP.

As table 3.1 shows,the parameter “status” contains the code to indicate that an event occurred.

Table 3.1 Status Identifier (status) Contents

Value	Meaning
0	Needs confirmation of connection with mobile station
1	Done confirmation of connection with mobile station
2-255	ReservedForFutureUse

3.1.3.1.4.3 **Area Specification Connection Request Primitive**

(1) Function

This primitive is a service primitive that requests the start of connections with mobile stations in a specified service provision area.

(2) When generated

This primitive is always generated by the upper protocol.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

ConnectionForArea.request (area)

The parameter "area" contains the service provision area.

3.1.3.1.4.4 **Service Request Message Request Primitive**

(1) Function

This primitive is a service primitive used by mobile stations that requests that upper layer servers transmit service requests.

(2) When generated

This primitive is always generated by the upper protocol.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

Service.request (linkAddress)

The parameter “linkAddress” contains the link address, which is used in the ITS-ASL-ELCP.

3.1.3.1.4.5 **Area Setting Request Primitive**

(1) Function

This primitive is a service primitive that requests the setting or updating of service provision areas.

(2) When generated

This primitive is always generated by the upper protocol.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

AreaSetting.request (type, [point], [radius], [upperPoint], [lowerPoint], [upperDist], [lowerDist], [leftOffset], [rightOffset], [area])

3.1.3.1.4.6 **Area Setting Notification Primitive**

(1) Function

This primitive is a service primitive that provides notification of an area identifier as the result of the setting or updating of a service provision area.

(2) When generated

This primitive is generated by ITS-ASL-ELCP when a service provision area is set up updated.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

AreaSetting.indication (area)

3.1.3.2 **Protocol Data Unit (PDU)**

3.1.3.2.1 **PDU Format**

The communication control management to manage the ITS-ASL-ELCP assigns the peer layer access point identifiers as that of the ITS-ASL-ELCP in the communication control management in order to formulate the peer layer protocol among management entities, as a result, the data transfer and reception among management entities are possible by utilizing the communication service interface of the ITS-ASL-ELCP. For this purpose, the PDU format of communication control management is defined with the network control protocol data unit (NCP-PDU) that is the same as that of the PDU of the ASL-NCP.

The format of NCP-PDU is shown in Figure Figure 3.1. The NCP-PDU shall consist of the control field (access control information) that contains the control information to instruct the procedure of ASL-NCP and the information field (network control service unit (NCP-SDU: NCP Service Data Unit)) that contains the PDU of the upper layer protocol.

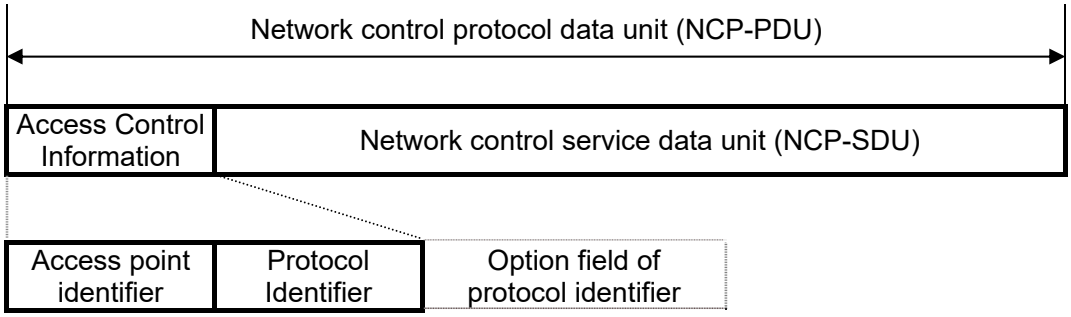


Figure 3.1 Communication Control Management PDU Format

3.1.3.2.2 **PDU Element**

3.1.3.2.2.1 **Connection Identification**

The access point identifier to identify the communication control management is given or received in the control field of an NCP-PDU. In addition, the link address to identify connections, etc. is given and received as the service primitive’s parameter that is provided by the ITS-ASL-ELCP.

3.1.3.2.2.2 **Control Filed Format**

The access control information that is consisted of both access point identifier sub-field and protocol identifier sub-field shall be contained in the control field to indicate the procedure of

communication control management.

The field format of the above-mentioned access control information shall be shown in Table 3.1.

Table 3.1 Access Control Information Field Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Access point identifier				Protocol identifier			
2	Optional field identifier							

(1) Access point identifier sub-field

The value to identify the access point of communication control management shall be contained in the sub-field of access point identifier. The access point identifier of communication control management is 0.

(2) Protocol identifier sub-field

The control information to instruct the procedure of communication control management shall be contained in the sub-field of protocol identifier. In addition, the sub-field of protocol identifier has an optional field for protocol identifier to multiplex additional information that supplements the control contents, and this optional field shall be used for communication control management as needed.

In the communication control management, the identification information shown in

Table 3.1 is defined and also the messages that are commonly owned by the base station and the mobile station are defined. Protocol identifier 3, 6, 7, 8, 9, 10 shown in

Table 3.1 are used in this guideline.

Table 3.1 Communication Control Management Protocol identifier

Protocol identifier	Description	Type of optional field	Used in this guideline
0	Disabled to use		Not used
1	Echo process (Transmission)	MsEchoParameter type	Not used
2	Echo process (Response)	MsEchoParameter type	Not used
3	Event process	MsEventParameter type	Used
4	Access management (Transmission of random numbers)	MsAuthCodeChallenge type	Not used
5	Access management (Response to digital signature)	MsAuthCodeSignature type	Not used
6	Connection request	MsConnectRequest type	Used
7	Connection response	MsConnectResponse type	Used
8	Connection confirmation	NULL	Used
9	Connection keep request	NULL	Used
10	Connection keep response	NULL	Used
11-15	Reserved		Not used

3.1.3.2.2.3 **Information Field Format**

The information field shall be NULL (data of zero length).

3.1.3.3 **Communication Connection Management**

3.1.3.3.1 **Communication Connection Management Procedure Element**

3.1.3.3.1.1 **Access Control Information Format**

The control information to specify the procedure for communication connection management is contained in the access control information and the message possessed commonly by a base station and a mobile station is defined.

3.1.3.3.1.1.1 **connection keep request message**

Table 3.1 shows the format of the connect request message to transmit a request of connection.

Table 3.1 connection keep request message Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Access point identifier linkControlManagement (0)				Protocol identifier ConnectRequest (6)			
2	Version information				Value of connection control timer for the mobile stations			
3	Value of connection control timer for the mobile stations							

(1) Access point identifier

This identifier contains an identifier “linkControlManagement” (0) which indicates the communication control management.

(2) Protocol identifier

This identifier contains an identifier “ConnectRequest” (6) which indicates the connection keep request message.

(3) Option field of protocol identifier

This identifier contains the following contents as a result of encoding with an MsConnectRequest type.

(a) Version information (versionIndex)

This information indicates the version information of the ITS-ASL (ITS-LPCP) on base stations. The version index is an integer from “0” to “15”, the first edition is defined as “0”. When it is revised, “1” is added to the edition before revision.

(b) Value of connection control timer for mobile stations (serviceTime)

This timer value refers to the “T1 max” that a mobile station sets in the CTO that controls

the connection status between the mobile station and the base station concerned. The time value shall be set to “0” through “4095” with a millisecond unit.

Further, when the time value is set to “0” (zero), the time length is infinite. However, setting "0" is only for testing, and a mobile station that receives “0” in actual operation state will discard the initial connection process.

3.1.3.3.1.1.2 **Connection Response Message**

Table 3.1 shows the format of the connection response message to transmit a response to received connection request.

Table 3.1 Connection Response Message Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Access point identifier linkControlManagement (0)				Protocol identifier ConnectResponse (7)			
2	Link Address				Version information			

(1) Access point identifier

This identifier contains an identifier “linkControlManagement” (0) which indicates the communication control management.

(2) Protocol identifier

This identifier contains an identifier “ConnectResponse” (7) which indicates the connection response message.

(3) Option field of protocol identifier

This identifier contains the following contents as a result of encoding with an MsConnectResponse type.

(a) Link Address

This information indicates the link address generated randomly by the mobile station.

(b) Version information (versionIndex)

This information indicates the version information of the ITS-ASL (ITS-LPCP) on base stations. The version index is an integer from “0” to “15”, the first edition is defined as “0”. When it is revised, “1” is added to the edition before revision.

3.1.3.3.1.1.3 **Connection Confirm Message**

Table 3.1 shows the format of the connection confirm message to transmit a confirmation to received connection response.

Table 3.1 Connection Confirmation Message Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Access point identifier linkControlManagement (0)				Protocol identifier ConnectConfirm (8)			

(1) Access point identifier

This identifier contains an identifier “linkControlManagement” (0) which indicates the communication control management.

(2) Protocol identifier

This identifier contains an identifier “ConnectConfirm” (8) which indicates the connection confirm message.

(3) Option field of protocol identifier

The information field shall be NULL (data of zero length).

3.1.3.3.1.1.4 **connection keep request message**

Table 3.1 shows the format of the connection keep request message to transmit a request of keeping the connection.

Table 3.1 connection keep request message Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Access point identifier linkControlManagement (0)				Protocol identifier ConnectKeepRequest (9)			

(1) Access point identifier

This identifier contains an identifier “linkControlManagement” (0) which indicates the communication control management.

(2) Protocol identifier

This identifier contains an identifier “ConnectKeepRequest” (9) which indicates the connection

keep request message.

(3) Option field of protocol identifier

The information field shall be NULL (data of zero length).

3.1.3.3.1.1.5 **Connection Keep Response Message**

Table 3.1 shows the format of the connection keep response message to transmit a response to received connection keep request.

Table 3.1 Connection Keep Response Message Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Access point identifier linkControlManagement (0)				Protocol identifier ConnectKeepResponse (10)			

(1) Access point identifier

This identifier contains an identifier “linkControlManagement” (0) which indicates the communication control management.

(2) Protocol identifier

This identifier contains an identifier “ConnectKeepResponse” (10) which indicates the connection keep request message.

(3) Option field of protocol identifier

The information field shall be NULL (data of zero length).

3.1.3.3.1.2 **Management Control Variables**

3.1.3.3.1.2.1 **Communication Connection Management Timer (CTR, CTO, T1max, T2max)**

The communication connection management timer (CTR: Connection Timer for RSU) is the timer to monitor the communication connection status between base station and mobile station. The CTR shall be generated at every communication connection established between a base station and a mobile station and deleted when the communication with the mobile station is terminated.

The CTO is the timer to monitor the communication connection status between a base station and a mobile station. The CTO shall be generated at every communication connection established with the base station and deleted when the CTO expired.

The T1max is the timer value to be set in the CTO that is passed from a base station in the form of the parameter “serviceTime” in the ASL base station profile or the parameter “serviceTime” defined in the broadcast transmission mode supplementary parameter in communication control information.

The T2max shall be the timer value to be set in the CTR.

Further, the values of the T1max and the T2max is specified in accordance with the base station installation condition, etc.

3.1.3.3.1.2.2 **Watchdog Timer for Transmission Schedule (WTTS)**

The WTTS is the timer of a base station to monitor whether the periodical transmission of “connection keep request message” is ensured for mobile stations.

The WTTS shall be generated at every communication connection established with a mobile station and deleted when the communication with the mobile station is terminated.

The T1max to be set in the CTO shall be set in the WTTS.

3.1.3.3.2 Communication Connection Management Procedure

3.1.3.3.2.1 ITS FORUM RC-005,IEEE 802.11, ARIB STD-T109 and ARIB STD-T120(PC5)

3.1.3.3.2.1.1 Communication Connection Procedure

A base station and a mobile station perform the following communication connection procedure when a communicable version is selected based on the version information of a connection keep request message and a connection response message.

The communication connection procedure of the base station and mobile stations are shown below. In addition, Figure 3.1 shows an example of the procedure of the base station and mobile stations specified in this subclause.

(1) Transmission of connection keep request message at the base station

When receiving a connection request primitive from ITS-LPCP or an application, the communication control management on the base station generates a connection keep request message and transmits the message to communication control management on the mobile station by broadcast communication according to a schedule decided in advance.

The specific transmission schedule is an implementation issue and is not specified in this specification.

(2) Reception of connection keep request message and transmission of connection response message at the mobile station

When receiving a connection keep request message, the communication control management on the mobile station refers to the connection management table and checks the connection status. In the case of “connected”, the communication control management finishes the communication connection procedure. In the case of “not connected”, the communication control management transmits a connection response message to the base station.

(3) Reception of connection response message and transmission of connection confirm message at the base station

When receiving a connection response message, the communication control management on the base station refers to the connection management table and checks the connection status with the peer station indicated in the connection response message. In the case of “connected”, the communication control management finishes the communication connection procedure. In the case of “not connected”, the communication control management adds the station as “connected” to the connection management table.

Then, the base station enables an ITS-LPCP to be usable for communications with the mobile station concerned and issues a status “Communication connection notice” by using an event notice primitive in the management service to the ITS-LPCP concerned.

Furthermore, the communication control management transmits a connection confirmation message to the mobile station and finishes the communication connection procedure.

(4) Reception of connection confirm message at the mobile station

When receiving a connection confirmation message, the communication control management on the mobile station refers to the connection management table and checks connection status. In the case of “connected”, the communication control management finishes the communication connection procedure. In the case of “not connected”, the communication control management adds the station as “connected” to the connection management table.

Then, the mobile station enables an ITS-LPCP to be usable for communications with the mobile station concerned and issues a status “Communication connection notice” by using an event notice primitive in the management service to the ITS-LPCP concerned.

Thus, the communication control management finishes the communication connection procedure.

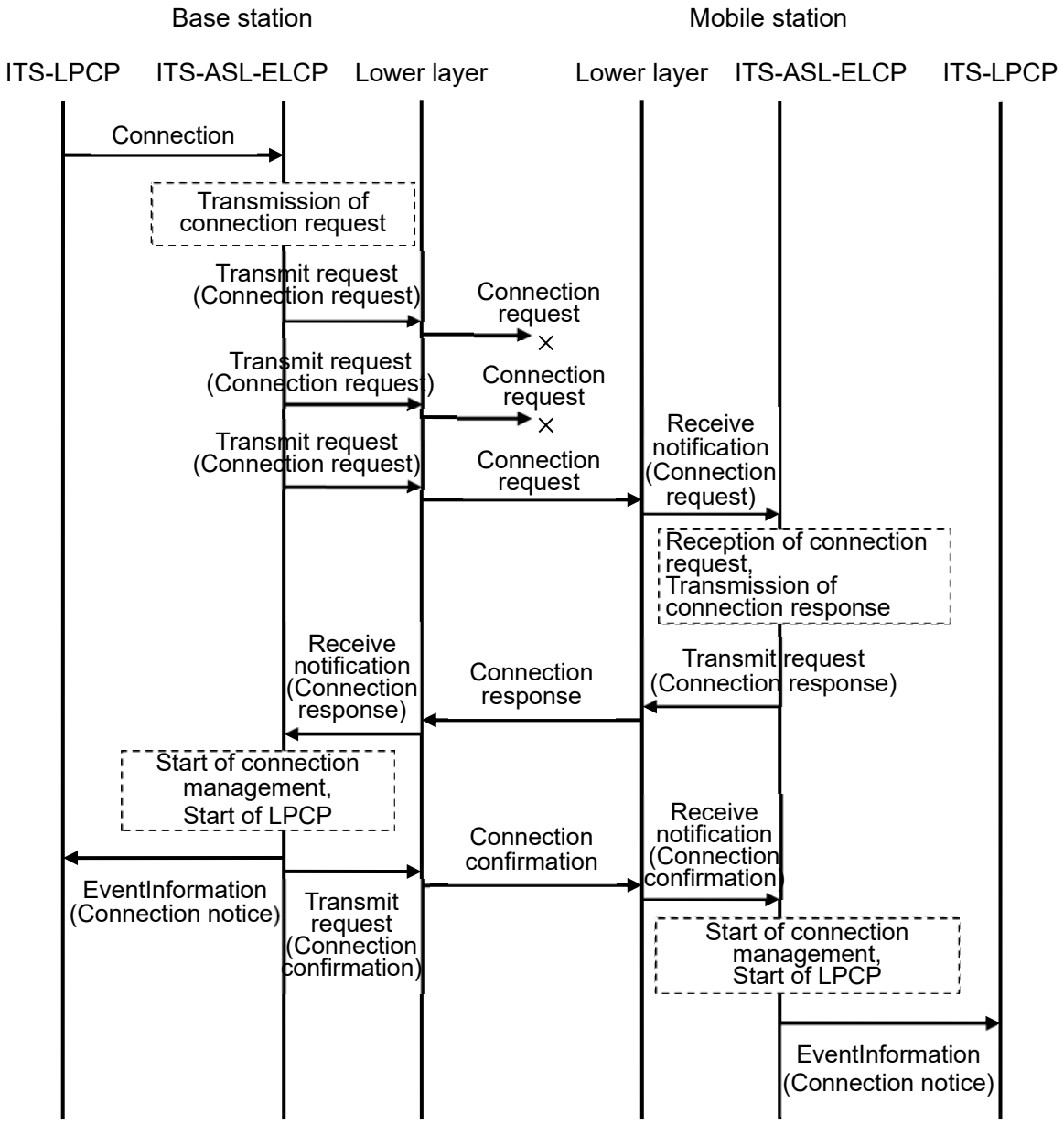


Figure 3.1 Example of Communication Connection Procedure

3.1.3.3.2.1.2 **Communication Connection Management Procedure**

The procedure to maintain and manage the communication management of a base station and mobile stations is described below. In addition, Figure 3.1 shows the connection management procedure of base stations and mobile stations specified in this subclause.

(1) Communication connection management procedure of the base station

In the communication control management of base station, the CTR and the WTTS are generated for each mobile station at the beginning of communication connections. At that time, T2 max is set in the CTR and T1 max set in the WTTS.

In the communication control management of the base station, the CTR and the WTTS shall start to be triggered by transmitting a connection keep request message to manage the communication connection status with a mobile station.

If the lower layer does not provide a resend function as shown in Figure 3.1, the connection keep request message is repeatedly transmitted with a period of resend time T3, and the transmission is terminated when a connection keep response message is received.

The CTR is stopped when a connection keep response message that corresponds to the above request primitive is received.

The WTTS is restarted when transmitting a next connection keep request message.

When a timeout of the CTR occurs owing to a receiving-disable of the connection keep response message, or when a timeout of the WTTS occurs owing to a transmission disable of connection keep request, a “Communication disconnection notice” status is issued to the ITS-LPCP being enabled by using an Event notice primitive of the management service in order to terminate the use of ITS-LPCP for the mobile station concerned.

When receiving a message without going through the ITS-ASL-ELCP, as shown in Figure 3.1, the base station application sets “1” to parameter “status” of the connection status containing the request primitive and transmits the primitive to the communication control management of the own station. In the communication control management receiving this, "1" is set in the internal connection status management variable. In case where the connection status management variable is "1" at the next polling cycle, the communication control management restarts the WTTS without transmitting the connection keep request message, and sets the connection status management variable to "0". See Table 3.1 for the value to be set for parameter “status” of connection status containing request primitive and the internal connection status management variable.mmunication control management completes the transmission of the service primitive at least once for those mobile stations until the WTTS times out. The schedule of a concrete transmitting schedule of the service primitive is an implementation issue and is not specified in this specification.

(2) Communication connection management procedure of the mobile station

In the communication connection management of mobile station, the CTO shall be generated at the beginning of communication connections. By referring to the communication connection timer value of a mobile station from the base station's profile received by the connection keep request message, the timer value shall be set in the CTO and shall be started.

From this process onward, the CTO shall be restarted every time when notice is provided of a valid service primitive arrival from the base station.

When a timeout of the CTR occurs owing to not receiving a valid service primitive, a "Communication disconnection notice" status shall be issued to the ITS-LPCP by using an "event notice" primitive of management service in order to terminate the use of the ITS-LPCP for the mobile station concerned.

Further, the processes and the CTO performed with the base station are terminated, and a new connection notice or waiting for broadcast receiving is enabled.

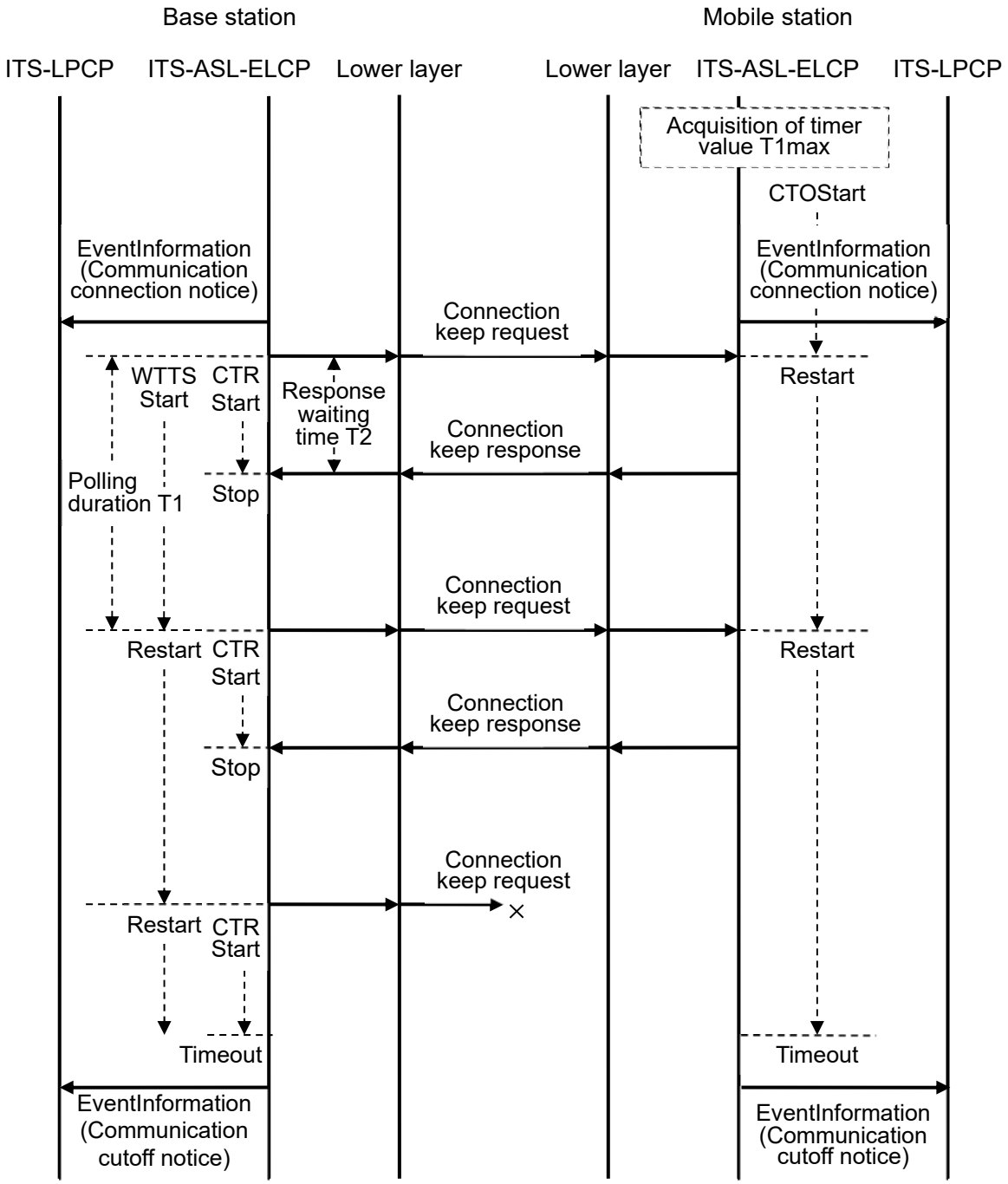


Figure 3.1 Example of Communication Connection Management Procedure

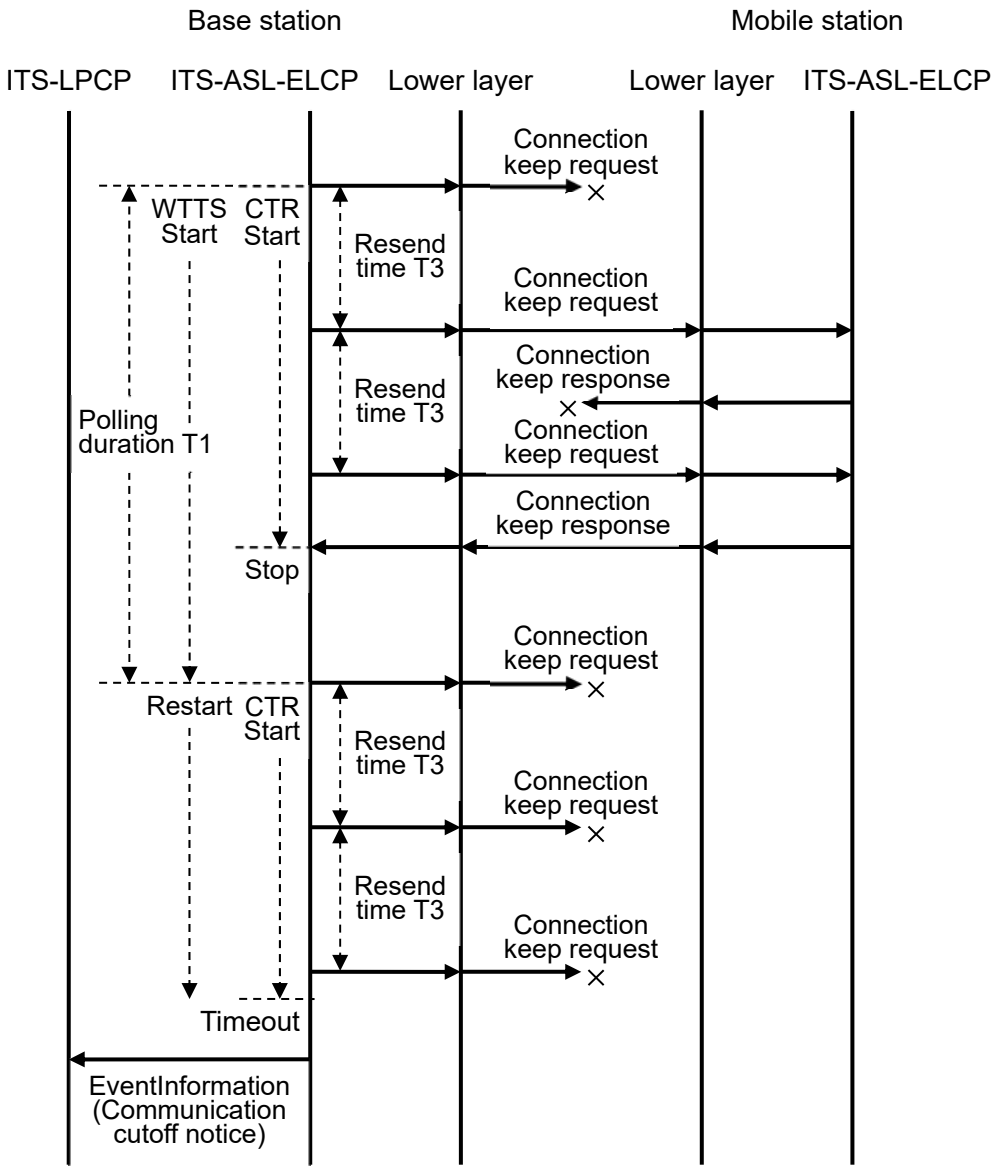


Figure 3.1 Example of Resend Control Procedure of Connection Keep Request

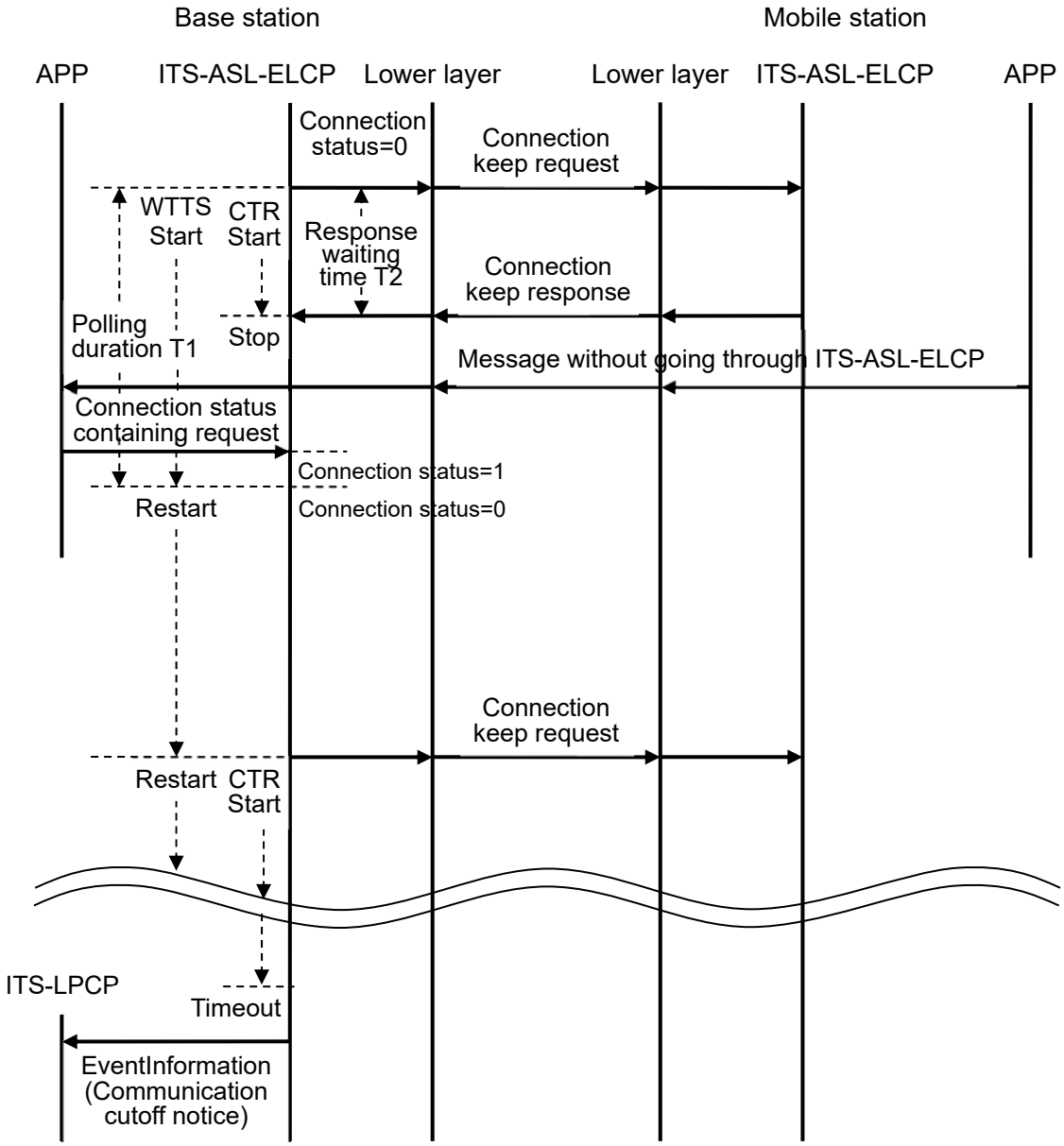


Figure 3.1 Example of Communication Connection Management Procedure Set Connection Status by Application

3.1.3.3.2.2 **ARIB STD-T104 / ARIB STD-T120(Uu)**

3.1.3.3.2.2.1 **Area Management Procedure**

Figure 3.1-17 shows an example of the procedure for managing a service provision area.

(1) Setting the area on the upper layer server

When the primitive from the application requesting the setting of the area (AreaSetting.request) is called, the communication control management on the upper layer server updates the area management table with the specified location information. If no area identifier is specified, an area identifier is assigned and newly registered. If an area identifier is specified, the existing setting is updated.

After updating the area management table, the communication control management on the upper layer server notifies the application of the area identifier using the area setting notification primitive (AreaSetting.indication). The upper layer server application manages area identifiers and services by associating them with each other.

(2) Upper-level server connection requests

When the area specification connection request primitive (ConnectionForArea.request) is called from the ITS-LPCP, the communication control management of the upper layer server changes the state of the specified service provision area in the area management table to "Connection standby."

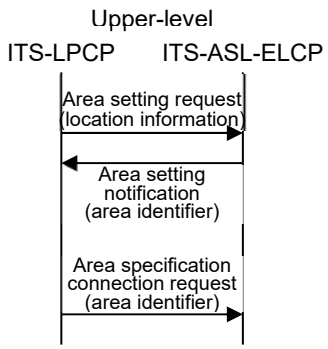


Figure 3.1 Example of Area Setting Procedure

3.1.3.3.2.2.2 **Communication Connection Procedure**

The communication connection procedures of the base station and mobile stations are shown below. In addition, Figure 3.1-18 shows an example of the procedures of the base station and mobile stations specified in this subclause.

(1) Transmission of service request message at the mobile station

When a service request primitive is received from ITS-LPCP or an application, the communication control management on the mobile station generates a service request message specifying a link address and transmits the message to the communication control management of the upper layer server together with the mobile station's location information as an HTTP request according to a schedule decided in advance. In the first service request, the link address is a group broadcast link address. In subsequent service requests, the link addresses assigned to each mobile station are used.

The specific service request message transmission schedule is an implementation issue and is not specified in this specification.

(2) Transmission of service response message at the upper layer server

When a service request message is received, the extended communication control on the upper layer server refers to the communication control management area management table and extracts "Connection standby" service provision area information. From the extracted information, service provision areas that match the location information in the service request message are identified and the link management table is updated. If the specified link address is a group broadcast link address, a link address number is assigned to the mobile station from which the service request was sent and the link address is registered in the link management table.

If a service provision area has been added to the link management table, an ASL-PDU is created indicating a communication connection to the added service provision area. If a service provision area has been deleted from the link management table, an ASL-PDU is created indicating a communications disconnection to the deleted service provision area. If there is no service provision area for the newly numbered link address, an ASL-PDU is created indicating communication registration. If none of the above conditions apply, an ASL-PDU is created indicating that the communication connection is maintained. A service response message containing the created ASL-PDU is generated and sent as an HTTP response.

If an ASL-PDU indicating a communication connection is sent, the base station enables an ITS-LPCP to be usable for communications with the mobile station concerned and uses a management service event notice primitive to issue a status “Communication connection notice” to the ITS-LPCP concerned and completes the communication connection procedure. If an ASL-PDU indicating a communications registration is sent, a management service event notice primitive is used to issue a status “Communications registration notice” to the ITS-LPCP concerned. If an ASL-PDU indicating a communications disconnection is sent, a management service event notice primitive is used to issue a status “Communications disconnection notice” to the ITS-LPCP concerned and the use of the ITS-LPCP for the mobile station is terminated.

(3) Reception of service response message at the mobile station

The communication control management of the mobile station updates the link management table based on the content of the service response message.

If an ASL-PDU indicating a communication connection is received, the ITS-LPCP is enabled to be usable for communications with the mobile station concerned and a management service event notice primitive is used to issue a status “Communication connection notice” to the ITS-LPCP concerned. If an ASL-PDU indicating a communications registration is received, a management service event notice primitive is used to issue a status “Communications registration notice” to the ITS-LPCP concerned. If an ASL-PDU indicating a communications disconnection is received, a management service event notice primitive is used to issue a status “Communications disconnection notice” to the ITS-LPCP concerned and the use of the ITS-LPCP for the mobile station is terminated.

This completes the communication connection procedure.

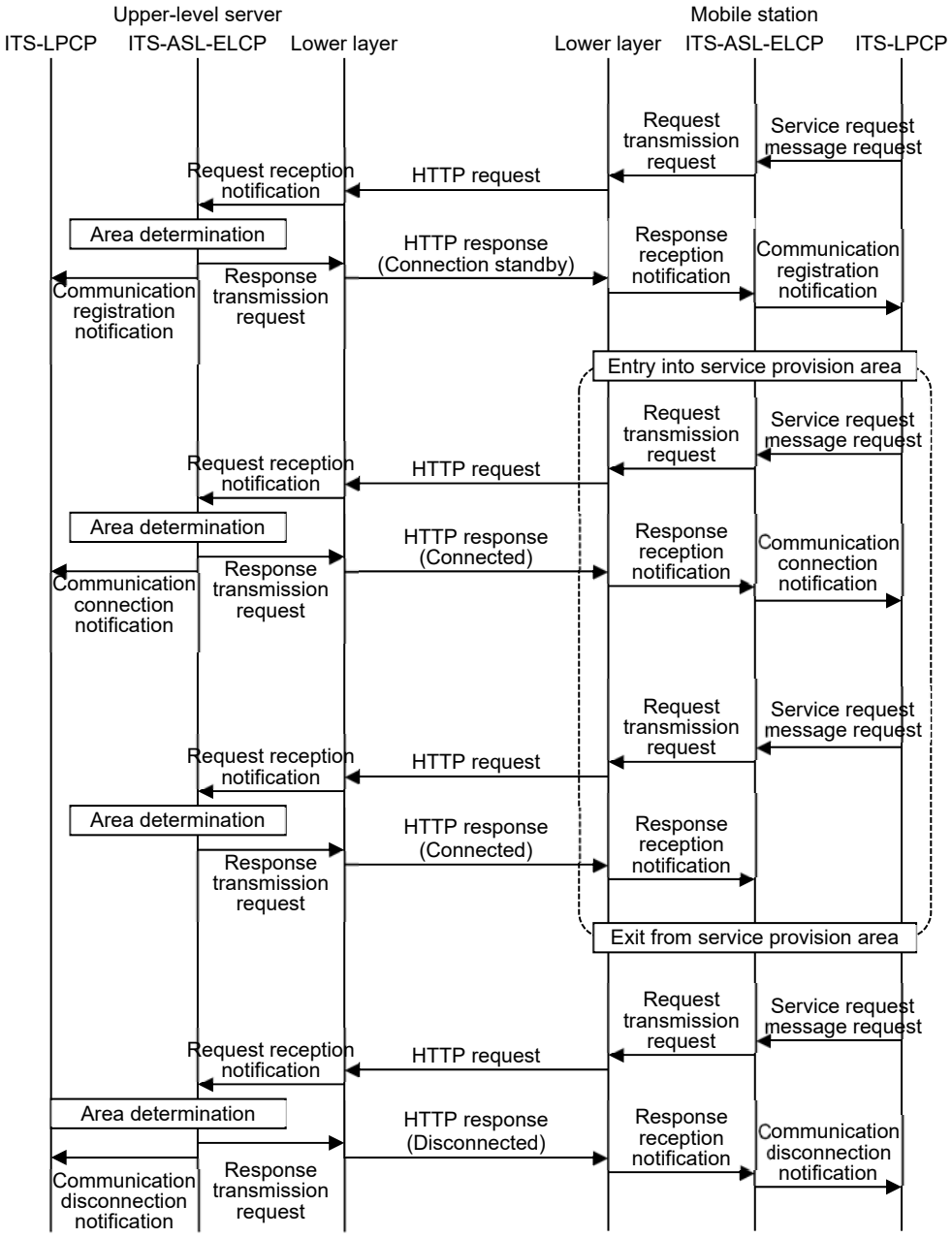


Figure 3.1 Example of Communication Connection Procedure

3.1.3.4 **Management Service Process**

3.1.3.4.1 **Management Service Process Procedure Element**

3.1.3.4.1.1 **Access Control Information Format**

In the management service process, the control information to specify the management service procedure is contained in the access control information, and the message owned by a base station and a mobile station commonly is defined.

3.1.3.4.1.1.1 **Event Notify Message**

This message is a message that shall be notified to the peer entity when an event such as error, etc. occurred in the ITS-ASL-ELCP is recognized. The format of “event notify message” shall be shown in Table 3.1.

Table 3.1 Event Notify Message Format

<table>
<tr><td></td><td>7 (MSB)</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0 (LSB)</td></tr>
<tr><td>1</td><td colspan="4">Access point identifier
linkControlManagement (0)</td><td colspan="4">Protocol identifier
eventReport (3)</td></tr>
<tr><td>2</td><td>Option
Index</td><td colspan="8">Status identifier “status”</td></tr>
<tr><td rowspan="2">3</td><td colspan="8">extentionParameter length</td></tr>
<tr><td colspan="8">extentionParameter contents</td></tr>
</table>

(1) Access point identifier

This identifier shall contain an identifier “linkControlManagement” “0” to indicate the communication control management.

(2) Protocol identifier

This identifier shall contain an identifier “evenReport” “3”.

(3) Option field of protocol identifier

This identifier shall contain the following contents as the encoding result of MsEventParameter type.

(a) Option identifier (Optional Index)

This identifier identifies whether or not event additional information (extensionParameter) is added.

(b) Status identifier (status)

This identifier indicates occurred event contents. Refer to Table 3.1 for details.

(c) Length identifier of event additional information

This identifier shall indicate the data length of succeeding event additional information. The unit is octet. The area size of this length identifier shall be expanded complying with ASN.1 encoding rule.

(d) Event additional information contents

The Event additional information contents contain an inconstant length data.

Table 3.1 Status Identifier (status) Contents

Value	Meaning	Notice destination	Contents of ExtentionParameter
0	Disabled to use	-	-
1	No access points existed	The other party station	Option
2-3	Disabled to use	-	-
4	Data size exceeded the upper limit	Own station	Contains all parameters delivered by Service request primitive (See NOTE below.)
5	No vacancy in transmission queue; Required service cancelled	Own station	Contains all parameters delivered by Service request primitive (See NOTE below.)
6	Disabled to use	-	-
7	Not corresponds to the specified version.	Base station	None
8-93	reservedForFutureUse	-	-
94-95	Disabled to use	-	-
96	Communication connection notice	Own station	Contains UserProfile type parameter
97	Communication connection notice	Own station	Contains UserProfile type parameter
98-127	Reserved for a future use	-	-

NOTE: In practical service applications, it is assumed that appropriate countermeasures will be provided such as that “the service primitive issuing origin (in own station)” holds data transmissions by returning the primitive to the service primitive issuing origin. Further, the ITS-LPCP side treatments to correspond to such countermeasures comply with the specifications in each ITS-LPCP.

3.1.3.4.2 **Management Service Processing Procedure**

3.1.3.4.2.1 **Event Process**

The communication control management shall perform an event process to notify event contents and the status to its own station or the other peer station when an event such as error, etc., occurred in the ITS-ASL-ELCP.

When the event contents and status are notified to the peer entity, the “event notify” message shall be used.

When event contents and the status are notified to its own entity, event contents and the status shall be notified complying with the specification of the event notify primitive (EventInfromation.indication).

3.2 **Local Port Control Protocol (ITS-LPCP)**

3.2.1 **Overview**

The ITS-LPCP is a control protocol that provides the data transfer service to the upper layer protocol such as applications and the management service to provide communication means to non-network type applications whose model type typically is the client/server type or peer-to-peer type.

3.2.2 **Implementation Scope**

3.2.2.1 **Equipment Configuration**

The ITS-LPCP specifications are for an integrated-type configuration (in which all software including non-network type applications are implemented in the base station/mobile stations). The separated-type configuration for the base station is specified as LPPoverUDP.

3.2.2.2 **Service Classification**

Services provided using the ITS-LPCP are classified into the following two types:

(1) Broadcast type service

Broadcast service is performed from the base station to mobile stations using the broadcast mode control function of the ITS-ASL-ELCP.

(2) Bidirectional type service

This service performs individual bidirectional communication using the client/server type communication control function of the ITS-ASL-ELCP.

3.2.2.3 **Implementation Scope**

Table 3.21 shows the implementation scope of each function of the LPCP corresponding to the classified services.

Table 3.21 Implementation Scope of Local Port Control Protocol

		Broadcast type Service	Bidirectional type service
Local port No.	Default NCP (0x0801)	-	M
	Echo (0x0802)	-	M
	Other ports	See NOTE below.	See NOTE below.
LPCP function	TransferData	M	M
	EventReport (connection, disconnection and accept port list)	-	M
	EventReport (other than above)	O	O
	OpenPort	M (mobile)	M
	ClosePort	M (mobile)	M

NOTE: Other port numbers are opened in accordance with mounted applications.

Symbol	Description
M	Mandatory
M (mobile/base)	Mandatory only in mobile/base station
O	Option
-	Not applicable

3.2.3 **Local Port**

In order to perform multiple non-network type applications, the LPCP defines the identification information (local port) to identify connection for the upper layer protocol. In order to send data correctly from a sending source application to a destination application, the LPCP identifies connection of each application using local ports that identify the sending destination application and sending source application and a link address that identifies the counterpart station.

3.2.3.1 **Local Port Numbers**

Numbers given for identification of application (local port numbers) are used as the connection identifier in non-network type applications. There are two kinds of local port numbers, reserved port and private port. In this specification, local port numbers are specified as shown in the table below. “1” to“0x0FFF” are reserved port numbers, and “0x1000” to “0xFFFF” are private port numbers.

Table 3.2-2 Local Port Number Classification

Port No.	Application	Remarks
0	Unused	
1 to 0x07FF	Same as definition of UDP port number	RFC3232
0x0800	Unused	
0x0801	Default NCP	
0x0802	Echo application for ITS-LPCP wrap test	
0x0803 to 0x0FEE	Ports for server/peer-to-peer application	
0x0FEF	ITS-Echo application for LPP wrap test	
0x0FF0 to 0x0FFE	Application ports for test/trial system	[0x0FF0] to [0x0FF8] are used in compatibility confirmation test for mobile stations.
0x0FFF	Local port management entity	

3.2.3.1.1 **Relationship between Applications and Local Port Numbers**

The application models are supposed to be the client/server model and peer-to-peer model. In the client/server model, reserved port numbers are used for server processes, and private local port numbers are used for client processes in general. In the peer-to-peer model, reserved port numbers are used in bidirectional processes in general. Reserved port numbers shall not be used without the management of port numbers.

When any local port other than local ports with reserved port numbers are used as receiving ports in the server processes or peer-to-peer model, local ports duplicate use problem may be caused. If erroneous connection occurs due to overlapped use of a local port, the application receiving erroneous data shall set the corresponding local port as a rejected reception port.

3.2.3.1.2 **Management of Local port Numbers**

Although assignment and management of reserved port numbers are outside the range of this standard, consideration for management such as registration is required in using them.

3.2.3.1.3 **Setting of Local Port Numbers**

(1) A reserved port number shall be assigned to an application without overlap globally.
(2) An application can have two or more local ports.
(3) Each application shall use receiving port numbers, which do not overlap within each station.
(4) When determination of the sending source is unnecessary or when the sending source is known, the sending source port can be omitted.

3.2.4 **LPCP Interface Service Specification**

3.2.4.1 **Interaction Overview**

3.2.4.1.1 **Data Transfer Service Interface**

The LPCP provides the following primitives as the data transfer service to the upper layer protocol.

TransferData.request

TransferData.indication

The “TransferData.request” is passed from the upper layer protocol to the LPCP to request to transfer the NCP-SDU passed from the upper layer protocol to the remote station. The “TransferData.indication” is passed from the LPCP to the upper layer protocol to indicate arrival of the NCP-SDU.

3.2.4.1.2 **Management Service Interface**

The ITS-LPCP provides the following management services to the upper layer protocol.

(1) Event notify service

In the event notify service, the ITS-LPCP provides the following primitive to the upper layer protocol.

EventReport.indication

The “EventReport.indication” is passed from the ITS-LPCP to the upper layer protocol in the local station to notify of an event notified by the event notification service of the ITS-ASL-ELCP, or passed from the ITS-LPCP to the upper layer protocol in the remote station or local station to provide notification that an event such as error occurred in the LPCP.

(2) Local port management services

In the local port management service, the LPCP provides the following primitives to the upper layer protocol.

OpenPort.request

OpenPort.confirm

ClosePort.request

The “OpenPort.request” is passed from the upper layer protocol to the ITS-LPCP to request the opening of a local port. The “OpenPort.confirm” is passed from the ITS-LPCP to the upper layer protocol to provide notification of the opened local port number. The “ClosePort.request” is passed from the upper layer protocol to the ITS-LPCP to request the closing of a local port.

3.2.4.2 **Service Content Specification**

This subclause specifies primitives and parameters related to the data transfer service and management service. Parameters are described abstractly as interfaces, and the information required for the receiving entity are specified. A specific implementation is not constrained in the method of making this information available.

3.2.4.2.1 **Data Transfer Service Interface**

The parameter “destinationPort” indicates the local port number (application), which is the data sending destination, and together with the parameter “linkAddress” identifies the access point of the local port control protocol.

The parameter “sourcePort” indicates the local port number (application) that is the data sending source. When a response from the sending destination is required, it is used as the default access point for giving the response.

NOTE: When any access point (local port number) other than the default access point is used, the application specifies it respectively. For details, refer to 3.2.3.1.

The parameter “userData” is provided by the actual NCP-SDU itself or by passing a pointer to the NCP-SDU or by other means.

The ITS-LPCP handles these parameters from the upper layer protocol in the following specification.

Table 3.2-3 Data Transfer Service Parameters

Parameter name	ASN.1 type	Remarks
linkAddress	DsrcLID	
sourcePort	PortNo	
destinationPort	PortNo	
userData	OCTET STRING	

Figure 3.21 shows the logical relationship among primitives in the data transfer service provided by the ITS-LPCP to the upper layer protocol.

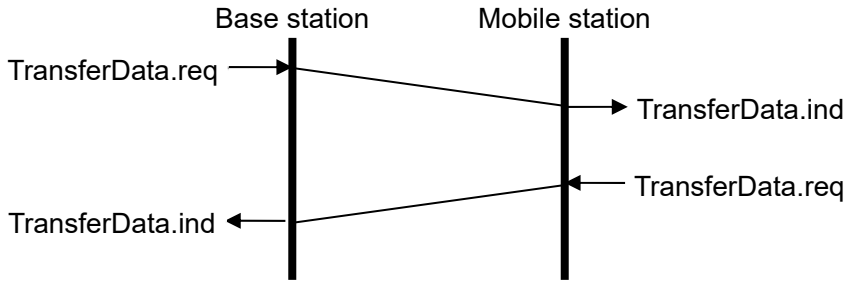


Figure 3.21 Logical Relationship between Data Transfer Service Primitives

3.2.4.2.2 **Management Service Interface**

The parameter “destinationPort” indicates the local port number (application) that is the data sending destination, and identifies the access point of the ITS-LPCP together with the parameter“linkAddress”.

The parameter “eventCode” indicates the type of event that occurred.

The parameter “extensionParameter” indicates the additional event information corresponding to each event code, and may be provided by the actual data itself or by passing pointer to the data or by other means.

The parameter “openPort” indicates the number of an opened local port.

The parameter “primitiveType” indicates the type of the indication primitive received by the opened local port.

The parameter “recvEventCode” indicates the type of event received by the opened local port.

Table 3.2-4 Management Service Parameters

Parameter name	ASN.1 type	Remarks
linkAddress	DsrcLID	
destinationPort	PortNo	
eventCode	LpcpEventCode	
extentionParameter	OCTET STRING	
openPort	PortNo	
primitiveType	LpcpPrimitiveType	
recvEventCode	LpcpEventCode	
closePort	PortNo	

Figure 3.2-2 shows the logical relationship among primitives in the management service provided by the LPCP to the upper layer protocol.

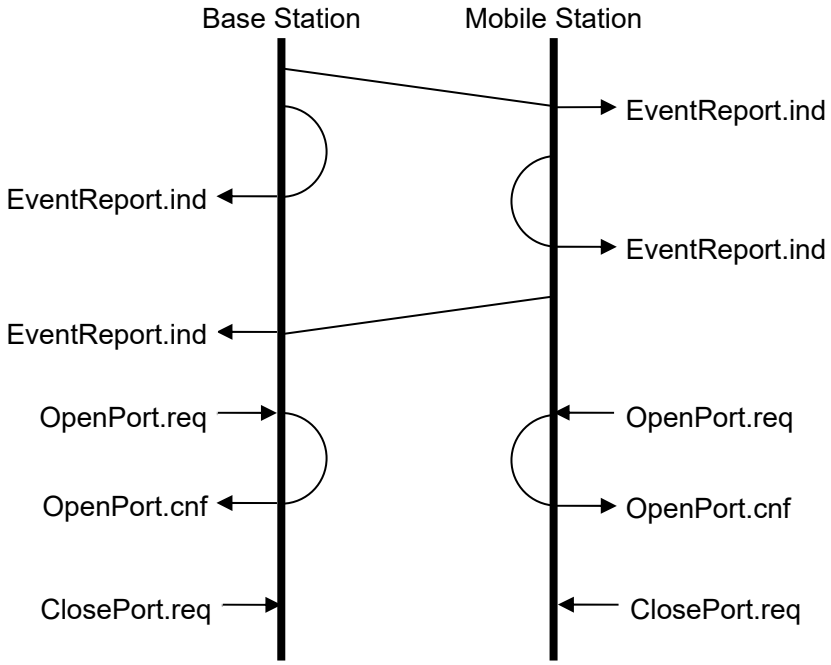


Figure 3.2-2 Logical Relationship between Management Service Primitives

3.2.4.3 **Data Transfer Service**

3.2.4.3.1 **Data Transfer Request Service Primitive (TransferData.request)**

(1) Function

This primitive is to request transfer of the NCP-SDU to the remote station.

(2) When generated

This primitive is generated by the upper layer protocol.

(3) Parameters of service primitive

This primitive shall provide parameters as follows.

TransferData.request (linkAddress, sourcePort, destinationPort, userData)

The parameter “linkAddress” is the link address used in the ITS-ASL-ELCP. The “linkAddress” parameter in the mobile station is a private link address. The parameter “linkAddress” in the base station is a private link address or groupbroadcast link address. When the parameter “linkAddress” is a broadcast link address, the NCP-SDU is delivered in the broadcast mode.

The parameter “sourcePort” indicates the local port number of the sending source.

The parameter “destinationPort” indicates the local port number of the sending destination.

The parameter “userData” indicates the NCP-SDU passed from the upper layer protocol in the local station.

3.2.4.3.2 **Data Transfer Indication Service Primitive (TransferData.indication)**

(1) Function

This service primitive provides notification of reception of the NCP-SDU from the remote station.

(2) When generated

This primitive is generated by the ITS-LPCP to indicate reception of the NCP-SDU.

(3) Parameters of service primitive

This primitive shall provide parameters as follows:

TransferData.indication (linkAddress, sourcePort, destinationPort, userData)

The parameter “linkAddress” is the link address used in the DSRC. The “linkAddress” parameter in the mobile station is a private link address. The parameter “linkAddress” in the base station is a private link address or group broadcast link address.

The parameter “sourcePort” indicates the local port number of the sending source.

The parameter “destinationPort” indicates the local port number of the sending destination.

The parameter “userData” indicates the received NCP-SDU.

3.2.4.4 **Management Services**

3.2.4.4.1 **Event Notify Service**

This service provides notification of an event notified by the event notification service of the ITS-

ASL-ELCP or provides notification that an event such as an error occurred in the ITS-LPCP. The ITS-LPCP provides the following service primitive.

3.2.4.4.1.1 **Event Notify Indication Service (EventReport.indication)**

(1) Function

This primitive provides notification of an event from the event notification service of the ITS-ASL-ELCP or provides notification that an event such as an error occurred in the ITS-LPCP.

(2) When generated

This primitive is generated by the ITS-LPCP when notification of an event such as an error is provided by the event notification service of the ITS-ASL-ELCP or when an event such as an error occurred in the ITS-LPCP is provided.

(3) Parameters of service primitive

This primitive shall provide parameters as follows:

EventReport.indication (linkAddress, destinationPort,
eventCode, [extentionParameter])

The parameter “linkAddress” indicates the link address used in the DSRC.

The parameter “destinationPort” indicates the local port number of the application that will be notified of the event.

The parameter “eventCode” indicates the code indicating the event that occurred.

The parameter “extensionParameter” indicates the information to supplement the contents of the parameter “eventCode” when necessary. This parameter is omitted for NULL data (data whose length is “0”).

3.2.4.4.2 **Local Port Management Service**

This service opens or closes local ports for transferred data and occurred events. The ITS-LPCP provides the following service primitives.

3.2.4.4.2.1 **Local Port Open Request Service Primitive (OpenPort.request)**

(1) Function

This service primitive is to request the opening of a local port for receiving data or events.

(2) When generated

This primitive is generated by the upper layer protocol.

(3) Parameters of service primitive

This primitive shall provide parameters as follows:

OpenPort.request ([openPort], [primitiveType], [recvEventCode])

The parameter “openPort” is indicates the local port number to be opened. This parameter is optional. When this parameter is omitted, LPCP assigns a local port number.

The parameter “primitiveType” indicates the identifier to specify the indication primitive type received by the opened local port. This parameter is optional. When this parameter is omitted, the request is to receive all indication primitives. For details, refer to Table 3.2-5

The parameter “recvEventCode” indicates the identifier to specify the event type received by the opened local port. This parameter is optional. When this parameter is omitted, the request is to receive all events. The content is equivalent to those of the event code (eventCode). (Refer to Table 3.2-9)

NOTE: When the upper layer protocol uses LPCP; it is required to give the request for communication connection notice by using the local port open primitive in advance.

Table 3.2-5 Primitive Type Identifier (primitiveType) Description

Primitive type	Description	Remarks
0	All primitives	
1	Data transfer indication primitive	
2	Event notification primitive	

3.2.4.4.2.2 **Local Port Open Confirmation Service Primitive (OpenPort.confirm)**

(1) Function

This service primitive provides notification of the local port number opened to receive data or events.

(2) When generated

This primitive is generated by the ITS-LPCP when the OpenPort.request primitive is issued.

(3) Parameters of service primitive

This primitive shall provide parameters as follows:

OpenPort.confirm ([openPort])

The parameter “openPort” indicates the opened local port number. This parameter is optional. When this parameter is omitted, it provides notification that opening of the specified local port has failed.

3.2.4.4.2.3 **Local Port Close Request Service Primitive (ClosePort.request)**

(1) Function

This service primitive is to request the closing of an opened local port.

(2) When generated

This primitive is generated by the upper layer protocol.

(3) Parameters of service primitive

This primitive shall provide parameters as follows:

ClosePort.request (closePort)

The parameter “closePort” indicates the local port number to be closed.

3.2.5 **ITS-LPCP Procedure Element**

3.2.5.1 **Maximum Transfer Unit (MTU) of ITS-LPCP**

The MTU is the maximum length of data, which can be passed from the ITS-LPCP to the ITS-ASL-ELCP. The MTU of the ITS-LPCP shall be “1400” octets (including the access control information).

3.2.5.2 **Access Control Information Format**

ITS-LPCP shall store a protocol identifier of LPCP and an option field type as shown in Table 3.2-6 in the access control information to define the messages shared by the base station and the mobile station.

Table 3.2-6 Protocol Identifier of ITS-LPCP

Protocol identifier	Description	Option field type
0	Event notification (eventReport)	LpcpEventParameter type
1	Data Transfer (message)	LpcpTransferDataPDU type
2-15	Reserved	

3.2.5.2.1 **Data Transfer Message**

This message is provided to transfer the PDU of the upper layer protocol. Table 3.2-7 shows the data transfer message format.

Table 3.2-7 Data Transfer Message Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Access point identifier “localPortControl (1)” or “localPortControl2 (14)”				Protocol identifier “message (1)”			
2	Source local port number (high-order)							
3	Source local port number (low-order)							
4	Destination local port number (high-order)							
5	Destination local port number (low-order)							

(1) Access point identifier

This field shall set the identifier “localPortControl (1)” or “localPortControl2 (14)” indicating the ITS-LPCP.

(2) Protocol identifier

This field shall set the identifier “message (1)” indicating the data transfer message.

(3) Optional fields

This field shall set the following contents as the result of the “LpcpTransferDataPDU” type coding.

(a) Source local port number

This field shall set the local port number of the sending source.

(b) Destination local port number

This field shall set the local port number of the sending destination.

3.2.5.2.2 **Event Notify Message**

This message is provided to notify of an event by the event notification service of the ITS-ASL-ELCP or an event occurred in the ITS-LPCP. Table 3.2-8 shows the event notification message format.

Table 3.2-8 Event Notify Message Format

	7 (MSB)	6	5	4	3	2	1	0 (LSB)
1	Access point identifier “localPortControl (1)” or “localPortControl2 (14)”				Protocol identifier “eventReport(0)”			
2	Event code “eventCode”							
3	Length of “extensionParameter”							
	Contents of “extensionParameter							

(1) Access point identifier

This field shall set the identifier “localPortControl (1)” or “localPortControl2 (14)” indicating the ITS-LPCP.

(2) Protocol identifier

This field shall set the identifier “eventReport (0)” indicating the event notification message.

(3) Optional fields

This field shall set the following contents as the result of the “LpcpEventParameter” type coding.

(a) Event code

This identifier shall specify the details of the occurred event. The codes from “0” to “127” identify the ITS-ASL-ELCP status. The codes from “128” to “255” identify the ITS-LPCP status. For details, refer to Table 3.2-9.

(b) Length of extensionParameter

This field shall set the data length of the following “extensionParameter”. The unit is octet. The size of this field expands according to the ASN.1 encoding rule. If no event information follows (that is, in the case of null), “0” is set in this field.

(c) Contents of extensionParameter

This field shall set the contents of “extensionParameter”.

Table 3.2-9 Descriptions of "eventCode"

eventCode	Description	Notification Target	Content of “extentionParameter”
0	Use prohibited.		None
1-3	Not used.		None
4	The data size exceeds the upper limit value.	Local (own) Station	
5	The sending service is aborted because of a sending queue overflow.	Local (own) Station	
6	The group broadcast link address is invalid.	Local (own) Station	
7-93	Reserved For Future Use		None
94-95	Not used.		None

96	Connection notice	Local (own) Station	The “UserProfile” type parameter is stored.
97	Disconnection notice	Local (own) Station	The “UserProfile” type parameter is stored.
98-127	Reserved For Future Use		
128	The communication is not connected.	Local (own) Station	
129	The destination port is invalid.	Remote (peer) Station	The “InvalidPort” type parameter is stored.
130	Acceptable port list	Remote (peer) Station	The “PortList” type parameter is stored.
131-255	Reserved For Future Use		

3.2.5.3 **Control Information of ITS-LPCP**

3.2.5.3.1 **Accept Local Port List**

The “accept (able) local port list” consists of the local port No. (openPort), notification primitive type (primitiveType) and notification event type (reverence) passed from OpenPort.request, and is used to identify whether or not the destination port of receiving data is opened.

The ITS-LPCP adds a local port to the list when it receives the “OpenPort.request”, and deletes a local port from the list when it receives the “ClosePort.request”.

3.2.5.3.2 **Communication Control Information List**

The “communication control information list” consists of the link address and the“UserProfile” type parameter passed from the status “connection notice” through the “EventInformation.indication” from ITS-ASL-ELCP, And is used to indicate whether or not the communication is connected when the “TransferData.request” is received.

The ITS-LPCP adds the information to the list when it receives the status “connection notice” from the ITS-ASL-ELCP through the “EventInformation.indication” in the management service, and deletes the information from the list when it receives the status “disconnection notice” through the “EventInformation.indication” from ITS-ASL-ELCP.

3.2.6 **ITS-LPCP Procedure**

3.2.6.1 **Local Port Management Procedure**

3.2.6.1.1 **Local Port Open Processing**

When the “OpenPort.request” in which the parameter “openPort” is specified is received from the upper layer protocol, the ITS-LPCP confirms the “accept local port list”. If the specified local port (openPort) does not overlap an existing port, the ITS-LPCP registers the local port Number. (openPort), notification primitive type (primitiveType) and notification event type (recvEventCode) in the accept local port list, and then notifies the upper layer protocol of the opened local port number through the “OpenPort.confirm”. If the specified local port (openPort) does overlap an existing port, the ITS-LPCP does not register the contents above, and notifies the upper layer

protocol that opening of the local port has failed through the “OpenPort.confirm” with the “openPort” parameter omitted. When receiving an “OpenPort.request” in which the parameter “openPort” is omitted, the local port control protocol shall assign a local port number to the requested process, register the assignment result, notification primitive type (primitiveType) and notification event type (recvEventCode) in the accept local port list, and then notify the upper layer protocol of the opened local port number through the “OpenPort.confirm”. At the time of number assignment, the local port control protocol shall check the “accept local port list”, and assign a non-overlapping number.

3.2.6.1.2 **Local Port Close Processing**

When receiving the “ClosePort.request” from the upper layer protocol, the LPCP deletes the information on the requested local port number from the “accept local port list”, and will not perform the notification of the received message after that for the deleted local port.

3.2.6.2 **Connection Processing Procedure**

3.2.6.2.1 **Communication Connection Process**

When receiving the “connection notice” through the “EventInformation.indication” in the management service of the ASL-ELCP, the LPCP registers the received link address and “UserProfile” type parameter in the communication control information list. And then the LPCP notifies the local port of a “connection notice” through the“EventReport.indication”. This notification procedure is applied to a local port whose contents in the “accept local port list” are either of the following.

(1) When the content of “primitiveType” is omitted or “all primitives (0)”.
(2) When the content of “primitiveType” is “event notification service (2)”, and“recvEventCode” is omitted.
(3) When the content of “primitiveType” is “event notification service (2)”, and the content of “recvEventCode” is “connection notice (96)”.

After that, the LPCP refers to the accept local port list, generates the event notification message whose event code is “accept port list (130)” and whose additional event information is the accept local port list in the local station, and then sends the message to the remote (peer) station.
When receiving the event notification message (“accept port list (130)”) sent from the remote (peer) station, the LPCP notifies the local port of the event “accept port list (130)” through the “EventReport.indication”. This notification procedure is applied to a local port whose contents in the “accept local port list” are either of the following.

(1) When the content of “primitiveType” is omitted or “all primitives (0)”.

(2) When the content of “primitiveType” is “event notification service (2)”, and“recvEventCode” is omitted.

(3) When the content of “primitiveType” is “event notification service (2)”, and the content of “recvEventCode” is “accept port list (130)”.

Figure 3.2-3 shows the outline of the Initial setup procedure of the LPCP according to the procedure above.

This procedure is performed only in the point-to-point type communication.

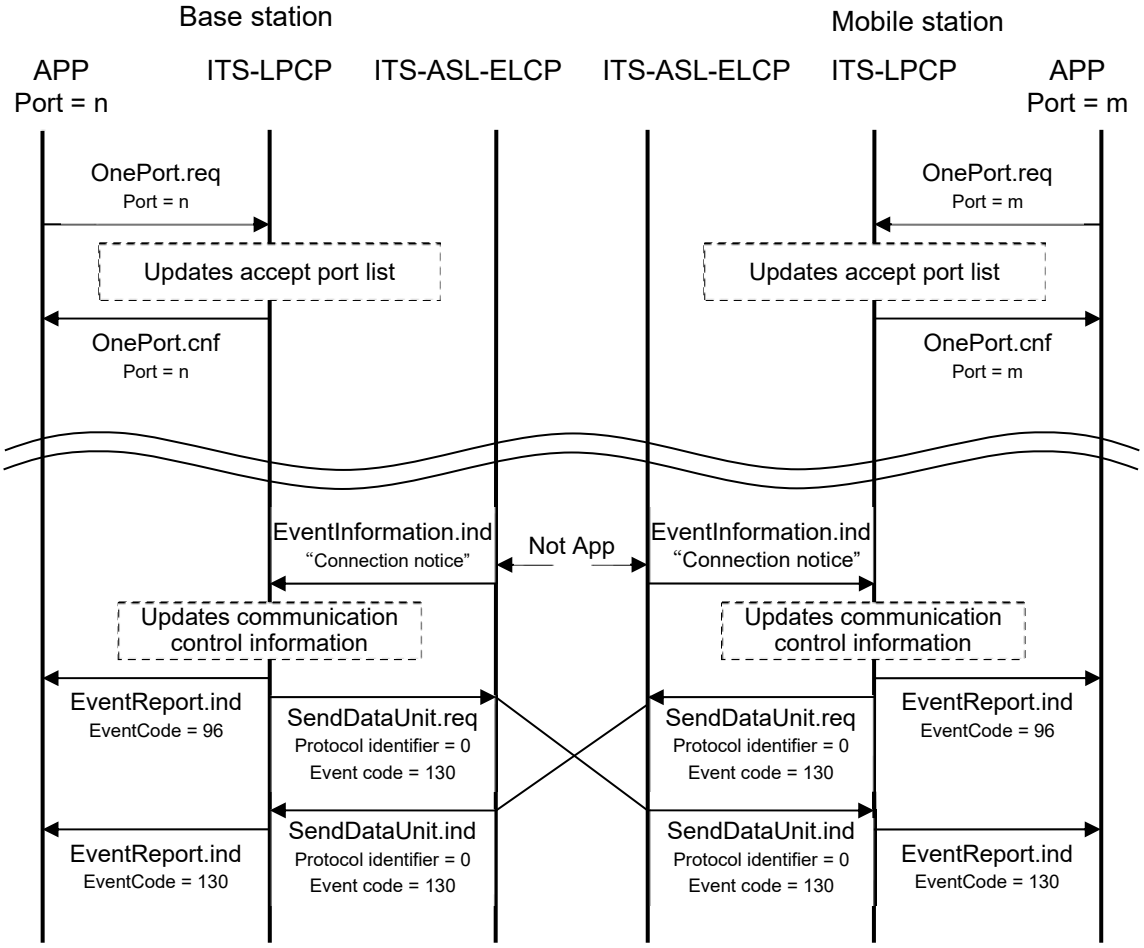


Figure 3.2-3 Initial Setup Process Outline

3.2.6.2.2 **Communication End Process**

When receiving “disconnection notice” through the EventInformation.indication in the management service of the ASL-ELCP, the LPCP deletes the information on the received link address from the communication control information list.

And then the LPCP notifies the local port of a “disconnection notice” through the

EventReport.indication. This notification procedure is applied to a local port whose contents in the accept local port list are either of the following.

(1) When the content of “primitiveType” is omitted or “all primitives (0)”.
(2) When the content of “primitiveType” is “event notification service (2)”, and “recvEventCode” is omitted.
(3) When the content of “primitiveType” is “event notification service (2)”, and the content of “recvEventCode” is “communication disconnection notice (97)”.

Figure 3.2-4 shows the outline of the communication end procedure of the LPCP according to the procedure above.
This procedure is performed only in the point-to-point type communication.

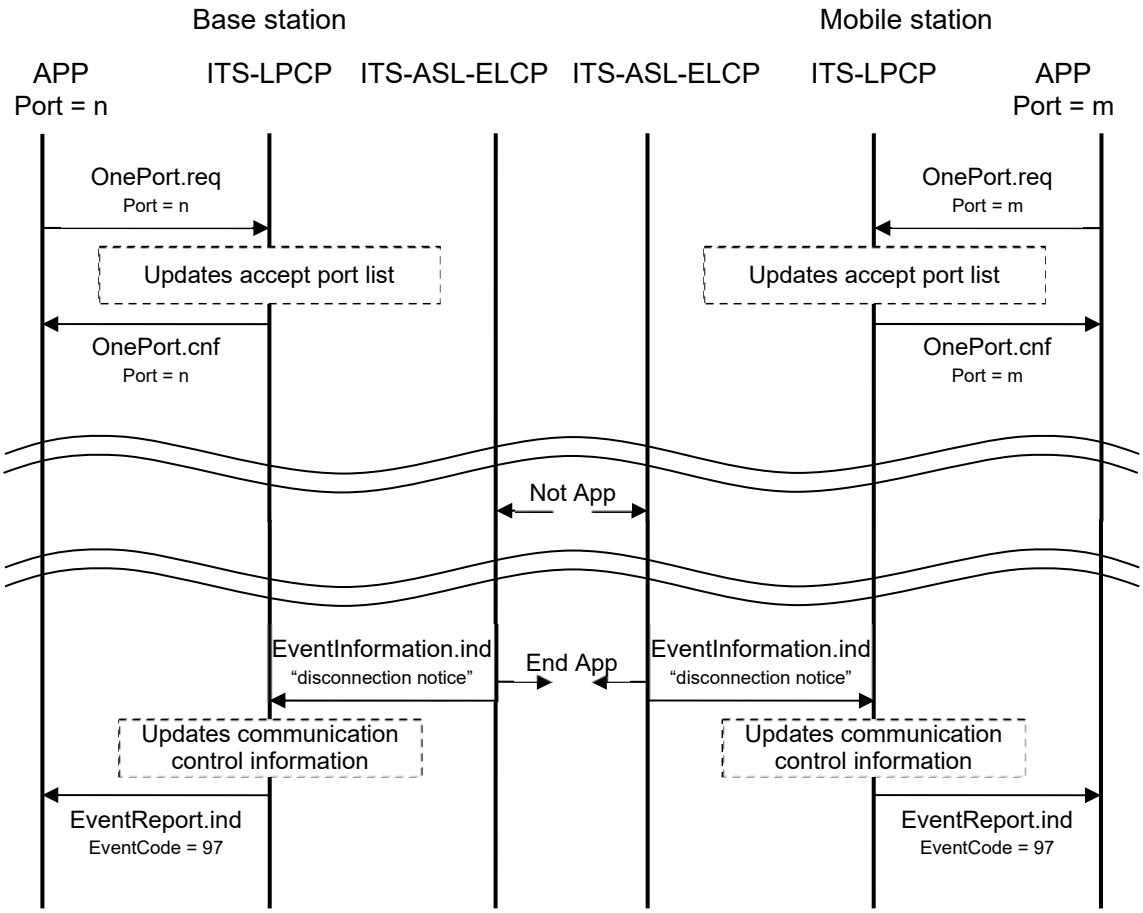


Figure 3.2-4 Communication End Process Outline

3.2.6.3 **Data Transfer Procedure**

Figure 3.2-5 shows the outline of the data transfer procedure.

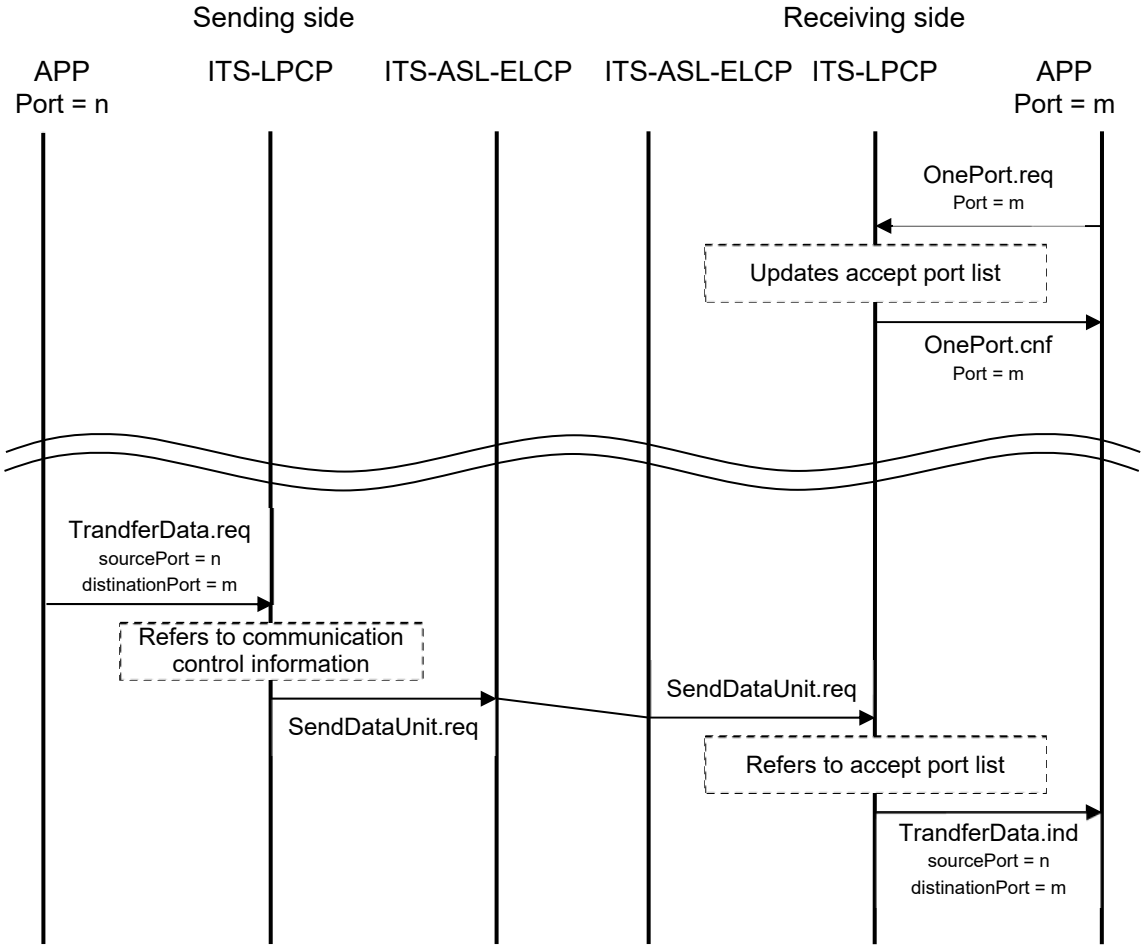


Figure 3.2-5 Data Sending Process Procedure Outline

3.2.6.3.1 **Data Sending Process**

When receiving the “TransferData.request” primitive from the upper layer protocol, the LPCP refers to the communication control information list, and then sends the NCP-SDU passed from the primitive through the data transfer message if the link address of the primitive is valid.

At this time, the contents passed from the primitive shall be set in the sending source local port number and sending destination local port number in the data transfer message.

When receiving the “TransferData.request” from the upper layer protocol, if the specified link address is a private one and the DSRC is not connected, the LPCP sends the source local port number specified by the “TransferData.request” a “DSRC not connected” through the “EventReport.indication”.

However, this event notification procedure is performed only when the event notification request was passed from the “OpenPort.request” primitive. If the event notification request was not given, the LPCP does not notify the upper layer protocol of the event. The necessity of event notification is judged based on the “accept local port list”.

Figure 3.2-6 shows the outline of the processing procedure when the communication is not connected.

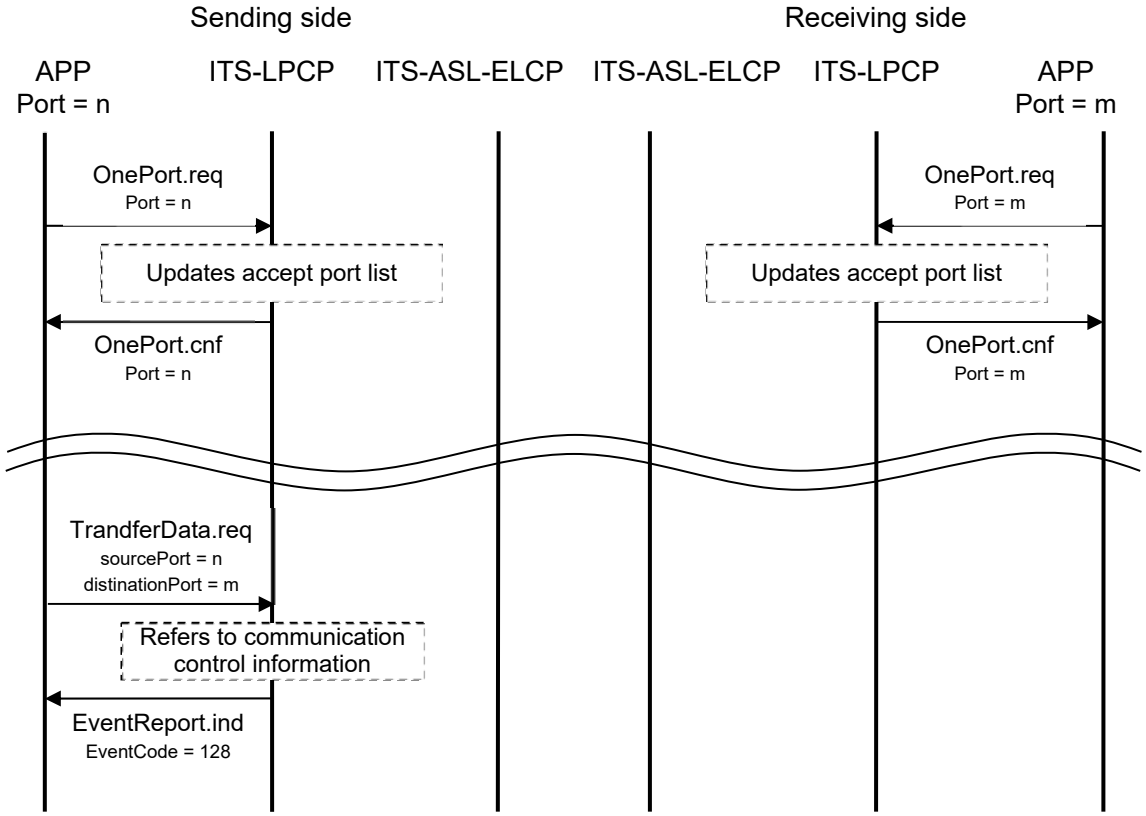


Figure 3.2-6 Outline of Processing Procedure When DSRC Is Not Connected

When receiving the event notification message from the remote station to which the data transfer message was sent, the LPCP confirms the event code given in the message.

When the contents of the event code are “The sending destination local port is invalid.” the LPCP notifies the source local port specified in the “extensionParameter” of a “The destination local port is invalid.” through “EventReport.indication”.

However, this event notification procedure is performed only when the event notification request was passed from the “OpenPort.request” primitive. If the event notification request was not given, the LPCP does not notify the upper layer protocol of the event. The necessity of event notification is judged based on the accepted local port list.

3.2.6.3.2 **Data Reception Process**

When receiving a data transfer message, the LPCP extracts the protocol identifier, destination local port number, source local port number and NCP-SDU from the message.

The LPCP refers to the “accept local port list”, and then notifies the upper layer protocol specified by the destination local port number that the NCP-SDU be received from the remote station through the “TransferData.indication” if the destination local port number is valid.

This data receiving processing is applied to a local port whose contents in the “accept local port list” are either of the following:

(1) When the contents of “primitiveType” are omitted or “all primitives (0)”.
(2) When the contents of “primitiveType” are “data transfer indication primitive (1)”.

When the link address is a private one and the destination local port number specified in the received data transfer message is invalid, the LPCP sends back the event notification message whose event code indicates “The destination local port is invalid.”

When the link address is a group broadcast one and the destination local port number specified in the received data transfer message is invalid, the LPCP aborts the received data.

Figure 3.2-7 shows the outline of the procedure when the destination local port number is invalid.

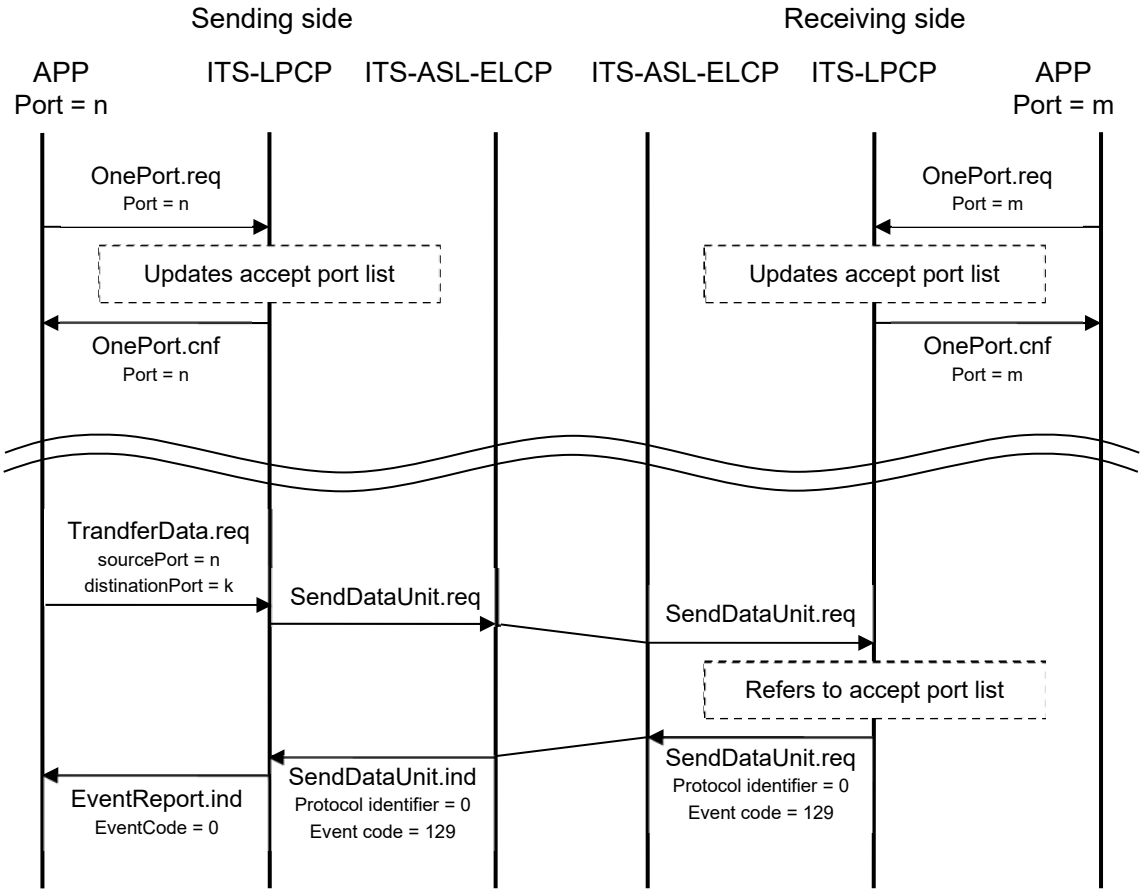


Figure 3.2-7 Outline of Reception Procedure in Case of the Invalid Destination Local Port Number

3.2.6.4 **Process Procedure Related to Local Port Numbers**

As examples of processing procedure related to the local port numbers, the processing procedure for peer to peer communication type service (client/server model), peer to peer communication type service (peer-to-peer model), and broadcast type service are described below.

3.2.6.4.1 **Peer to Peer Communication Type Service (Client/Server Model) Process Procedure**

Figure 3.2-8 to Figure 3.2-13 show echo application procedure as examples of processing procedure in the client/server model.

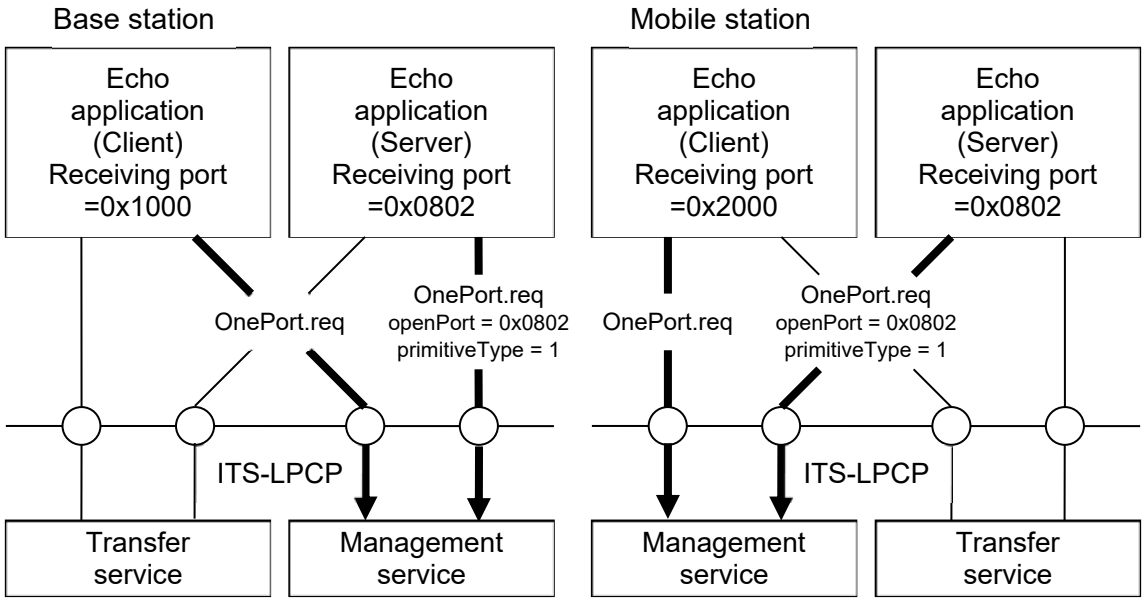


Figure 3.2-8 Example of Application Registration Operation (1)

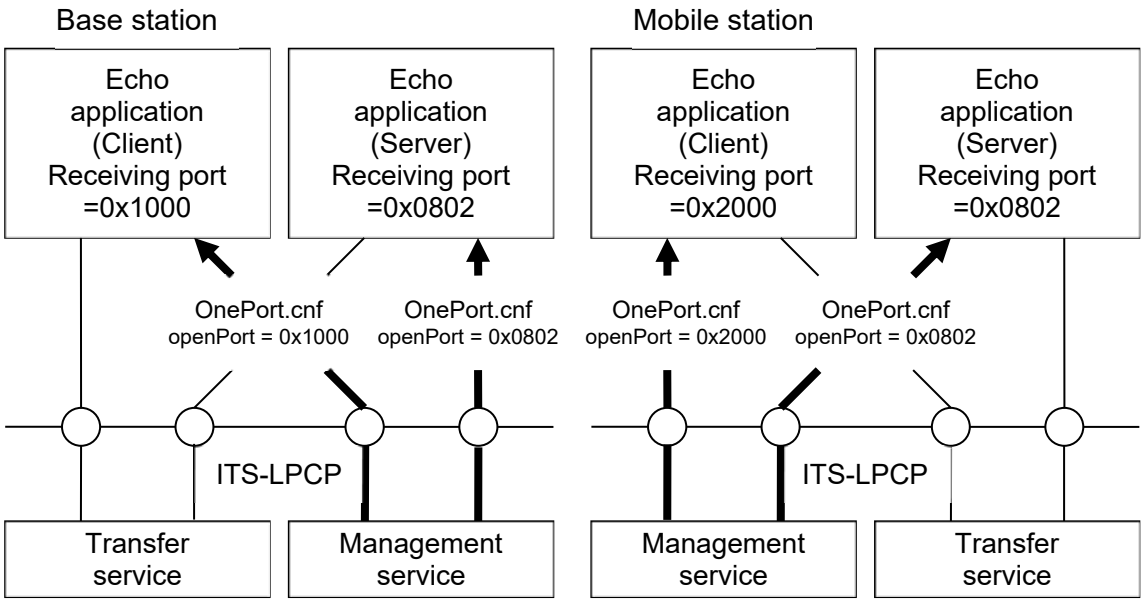


Figure 3.2-9 Example of Application Registration Operation (2)

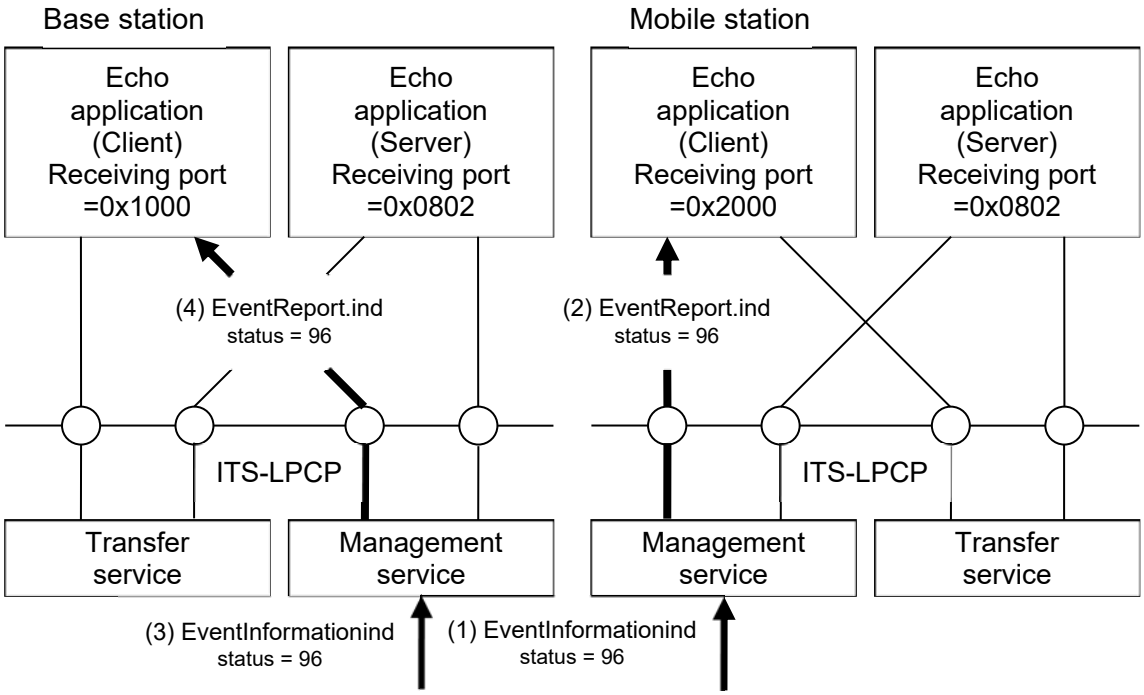


Figure 3.2-10 Initial Set up Operation (connection notice)

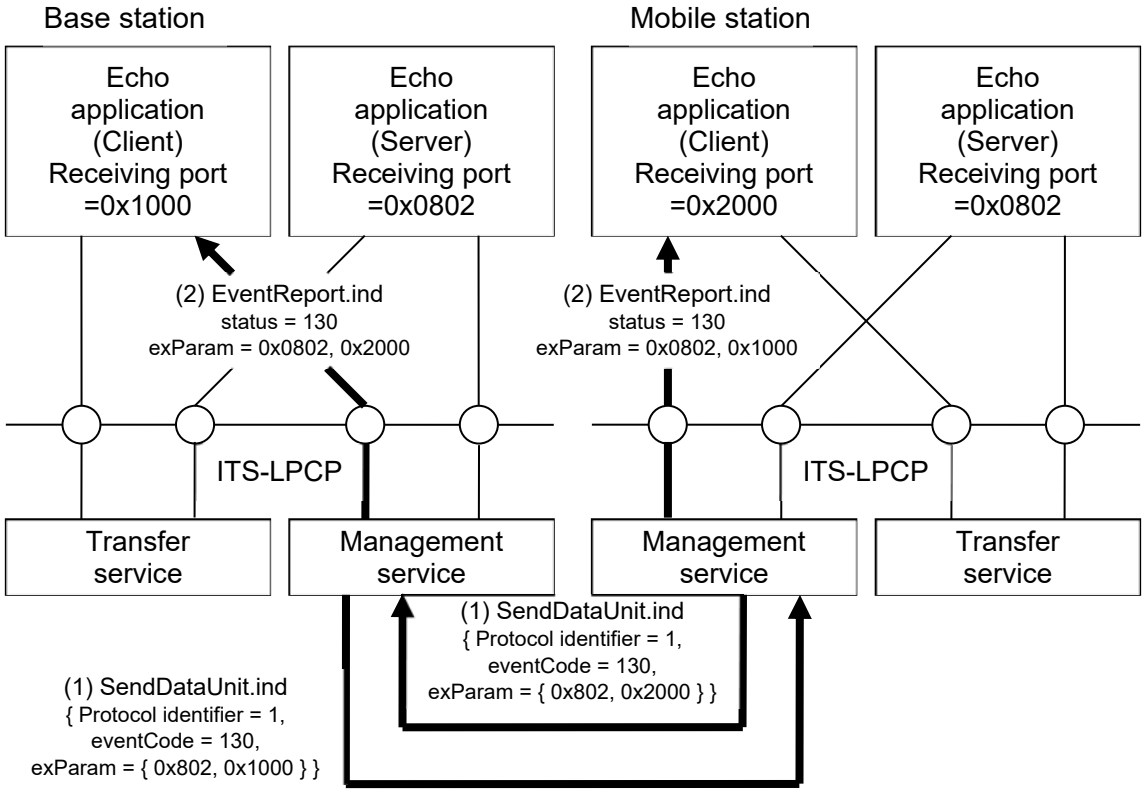


Figure 3.2-11 Initial Set up Operation (accept port list exchange)

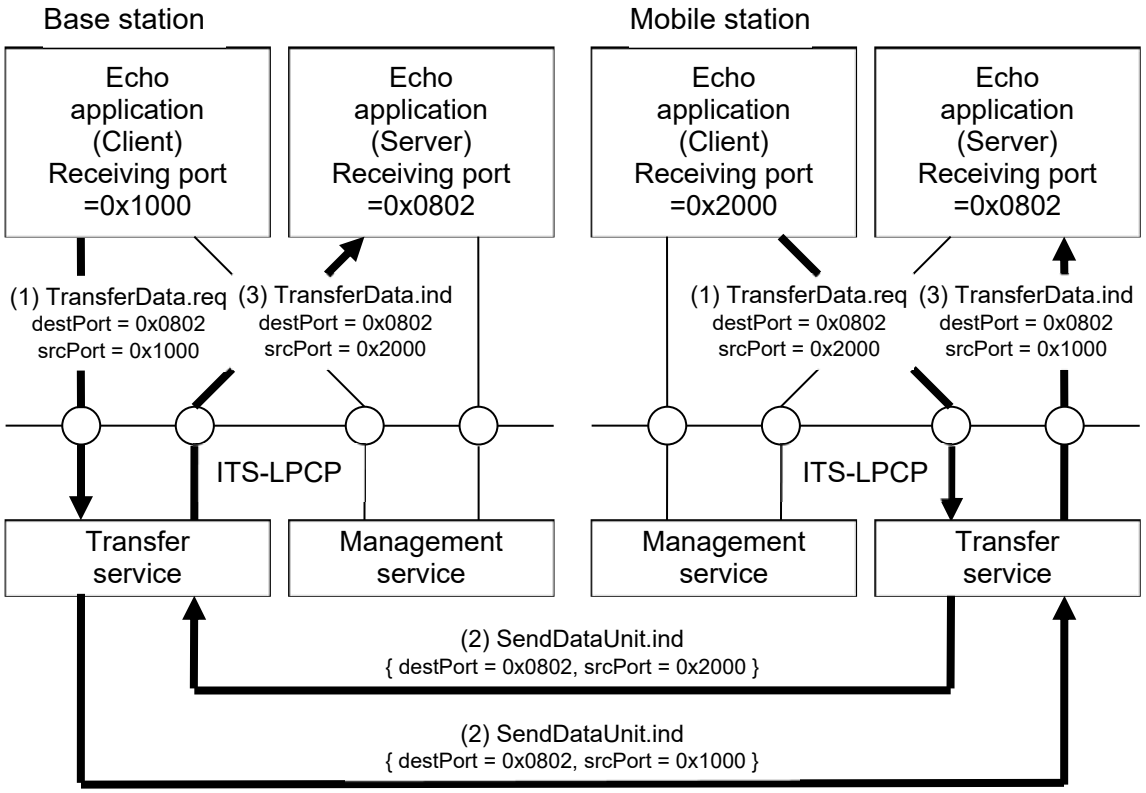


Figure 3.2-12 Example of Data Transfer Operation (echo request)

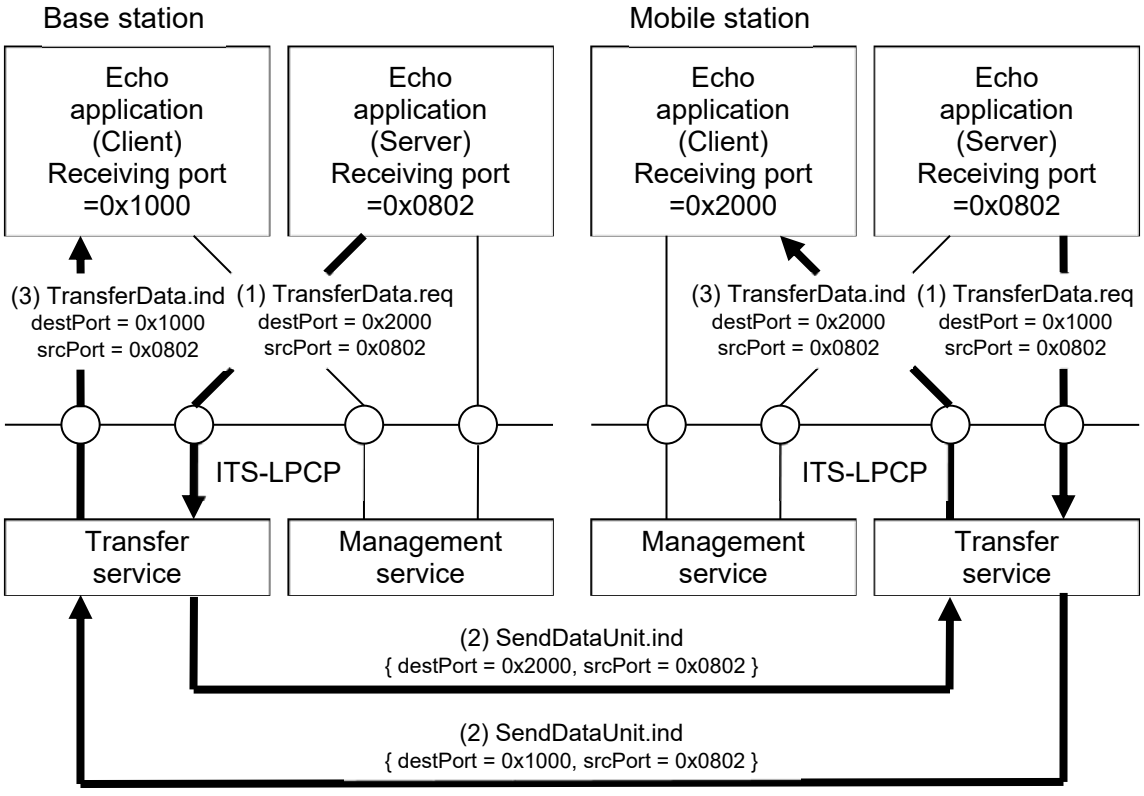


Figure 3.2-13 Example of Data Transfer Operation (echo response)

3.2.6.4.2 **Peer to Peer Communication Type Service (Peer-to-Peer Model) Process Procedure**

Figure 3.2-8 to Figure 3.2-13 show as examples of processing procedure in the peer-to-Peer model.

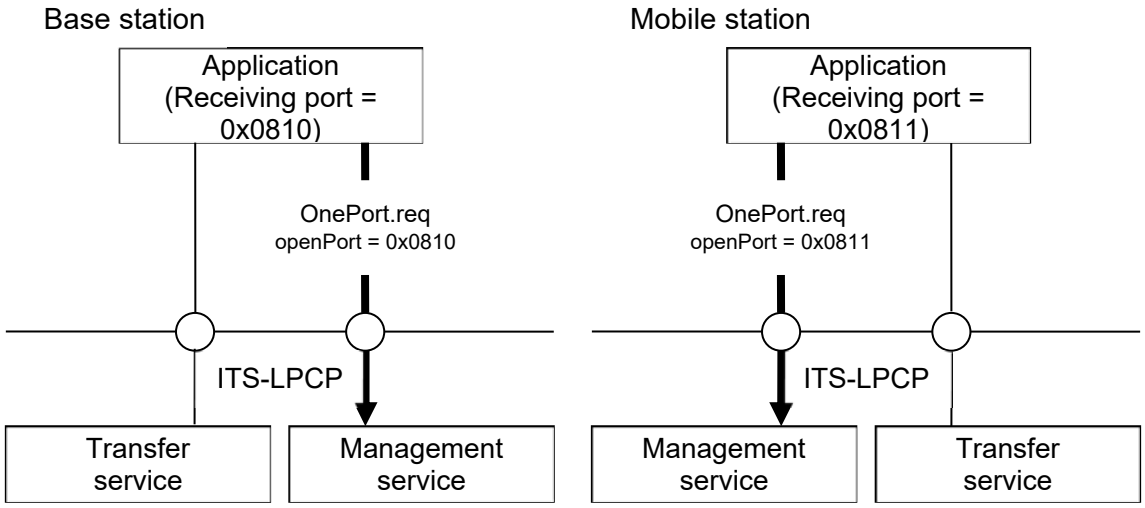


Figure 3.2-14 Example of Application Registration Operation (1)

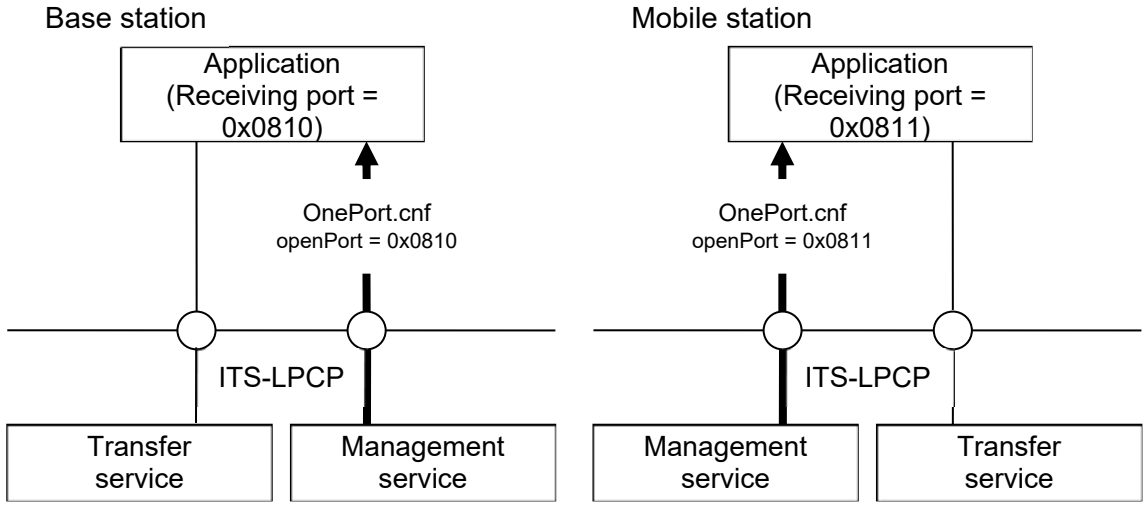


Figure 3.2-15 Example of Application Registration Operation (2)

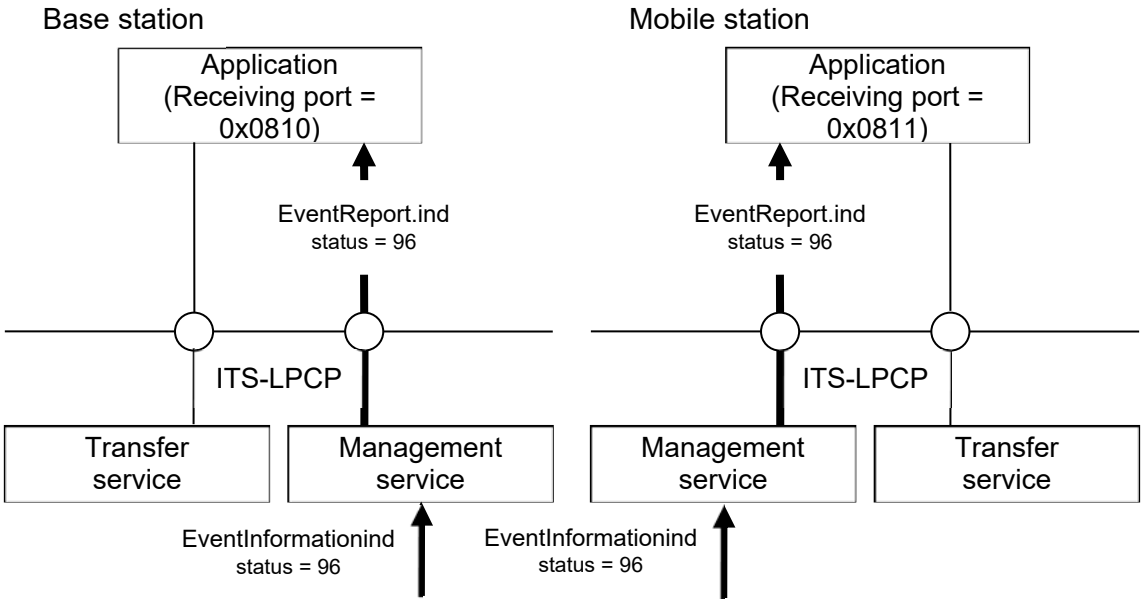


Figure 3.2-16 Initial Set up Operation (connection notice)

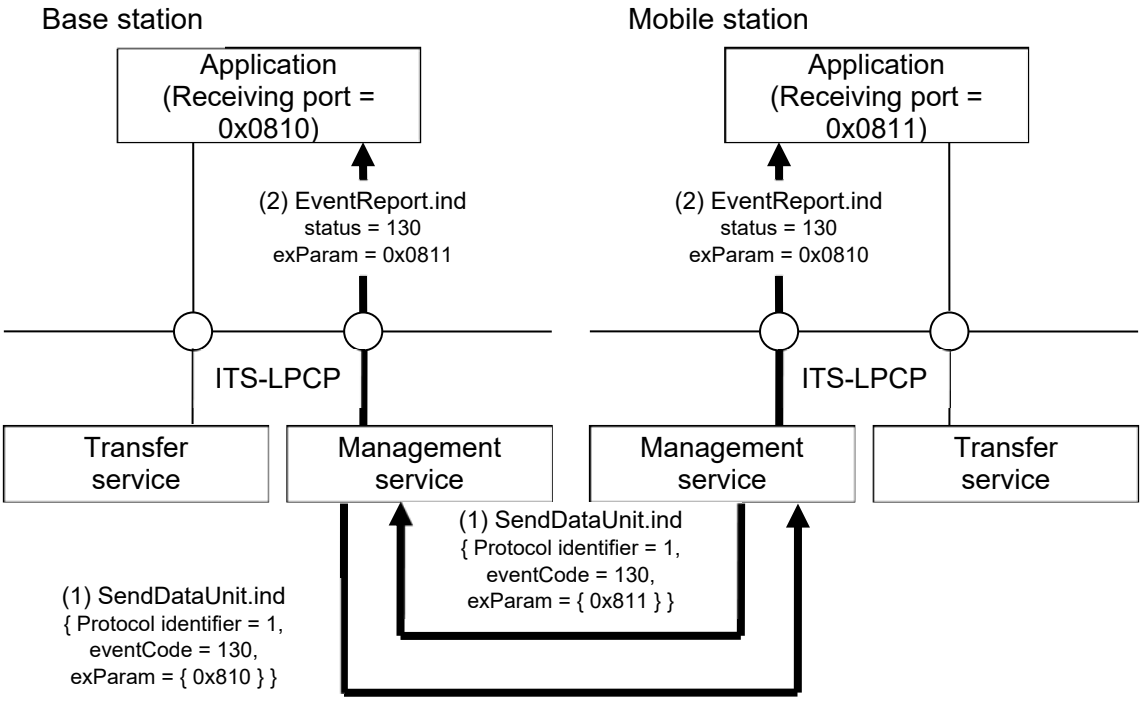


Figure 3.2-17 Initial Set up Operation (accept port list exchange)

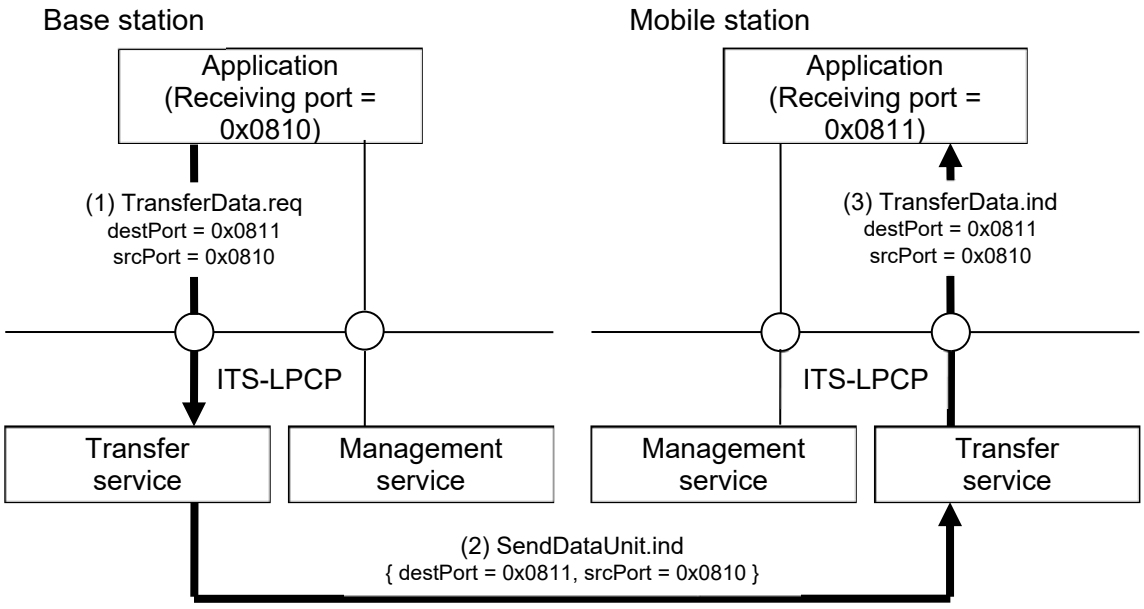


Figure 3.2-18 Example of Data Transfer Operation (base station to mobile station)

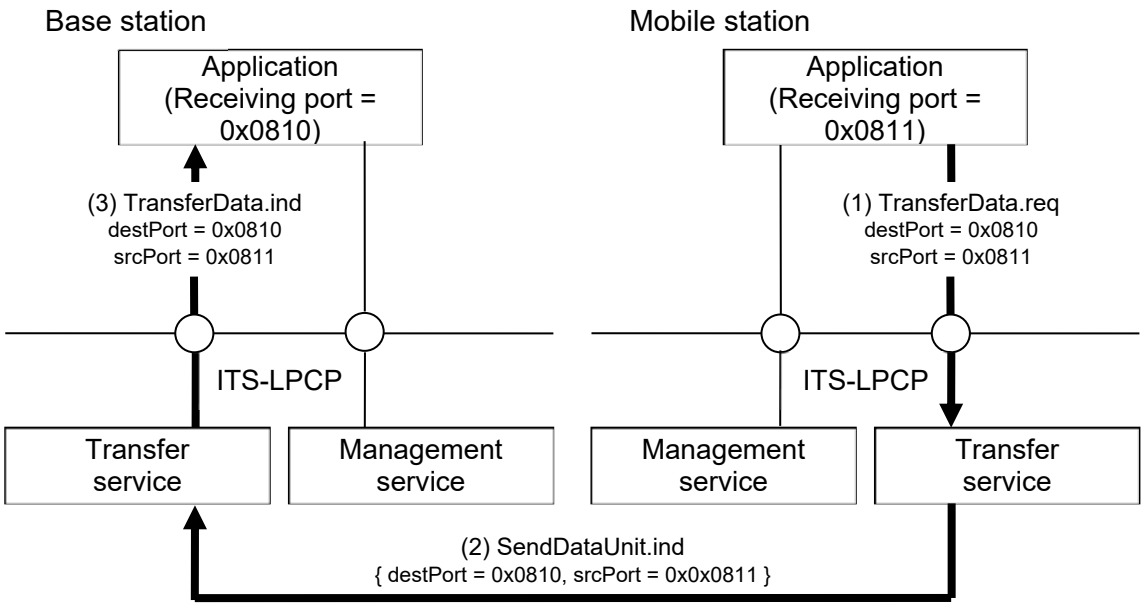


Figure 3.2-19 Example of Data Transfer Operation (mobile station to base station)

3.2.6.4.3 Broadcast Type Service Process Procedure

Figure 3.2-8 to Figure 3.2-8 show as examples of processing procedure in broadcast type service.

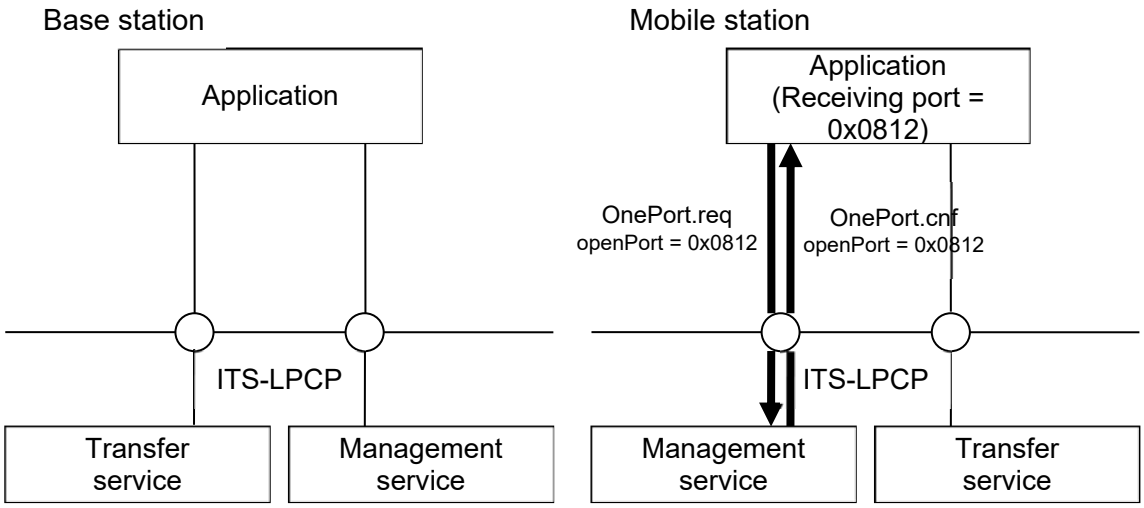


Figure 3.2-20 Application Registration Operation

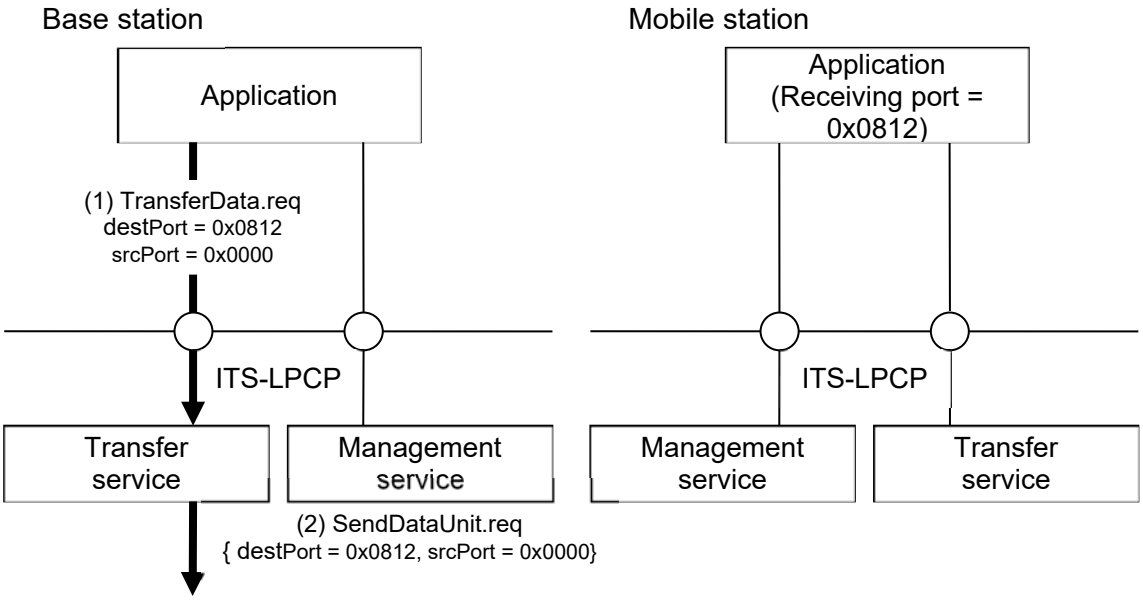


Figure 3.2-21 Startup of Application on Base Station Side

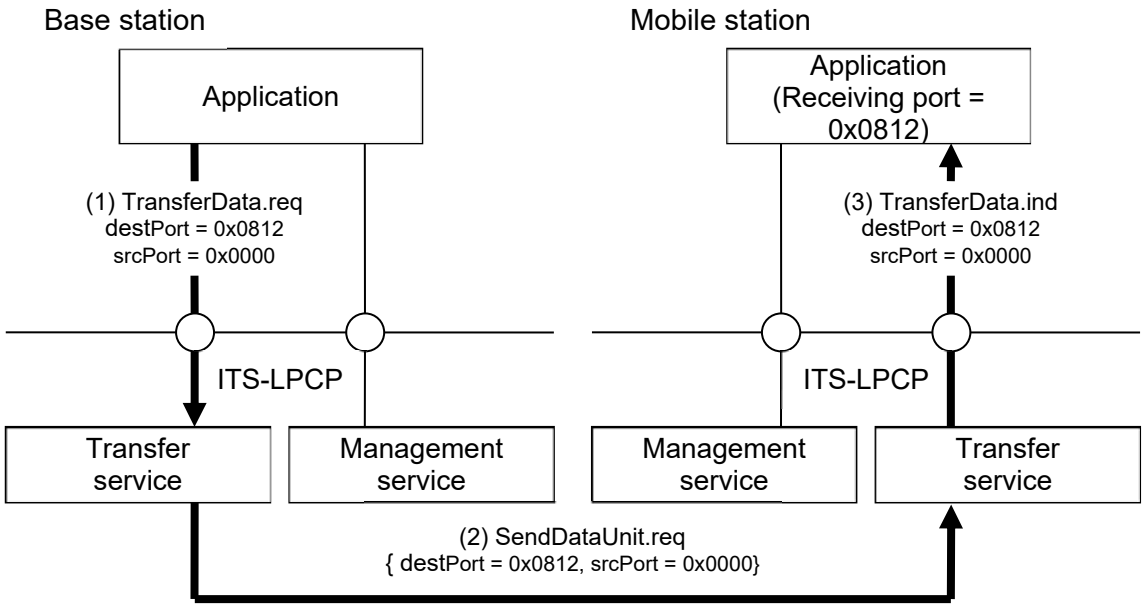


Figure 3.2-22 Entry into Communication Process

3.3 **Local Port Protocol (ITS-LPP)**

3.3.1 **Overview**

The local port protocol (LPP) is a transaction-oriented protocol. It is located between the LPCP and non-network type applications, extends the functions of the LPCP, and provides the transaction service and connection management services described below to non-network type applications in the DSRC mobile stations/base station to improve the efficiency of the application building as shown in Figure 3.3-1 . The LPP consists of the transaction service entity, which extends the communication functions of the LPCP, and the connection management service entity, which manages the communication status such as initial set up and disconnection. Each service has the following functions:

(1) Transaction service entity

(a) Data exchange functions per transaction

(b) Unidirectional data-sending transaction service

(c) Request-response type transaction service

(d) Data resend function (option)

(e) Message segmentation/re-assembly function (option)

(f) Transaction abortion function

(2) Connection management service entity

(a) Communication connection query service

(b) Communication disconnection notice service

(c) Accept port query service

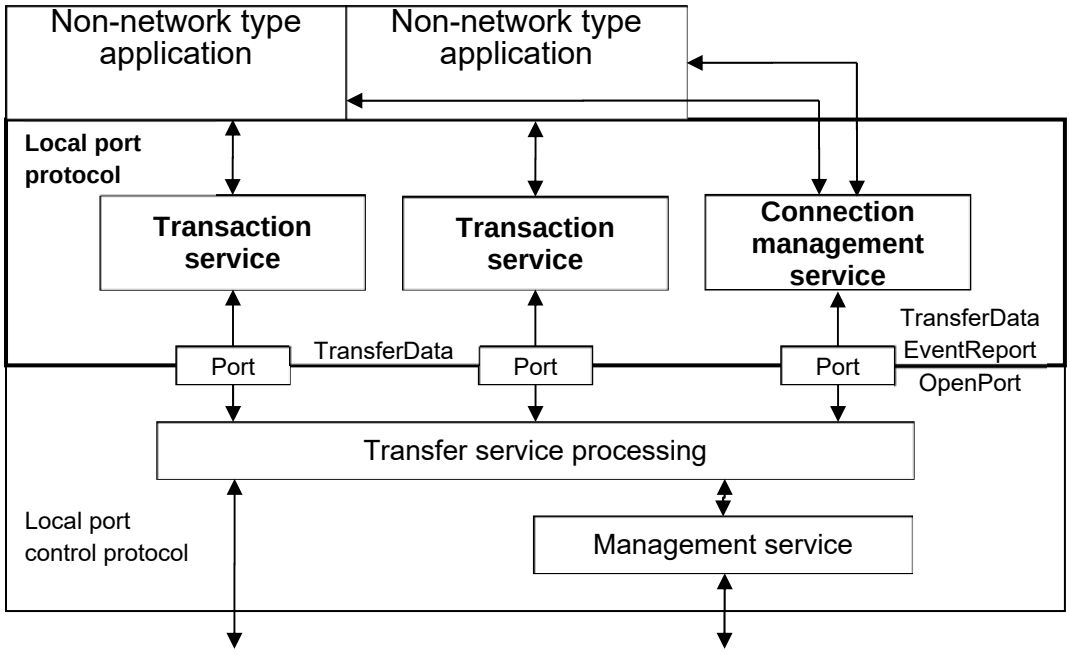


Figure 3.3-1 Overview of Local Port Protocol (LPP)

3.3.1.1 **Transaction Services**

3.3.1.1.1 **Data Exchange per Transaction**

The LPP exchanges the application data for each single transaction.

Each transaction is identified by the transaction ID as shown in Figure 3.3-2. By this function, the system can handle situations in which two or more transactions exist simultaneously in the same application.

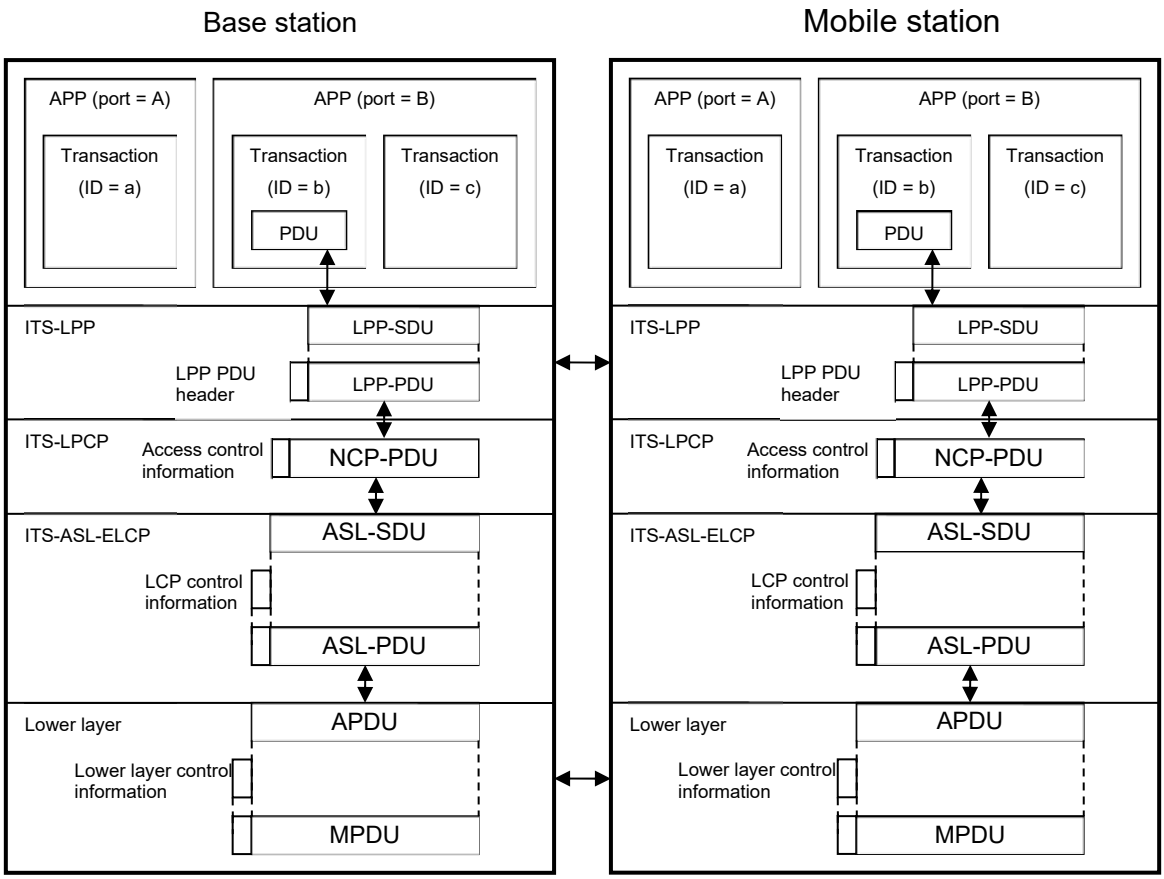


Figure 3.3-2 Example of Data Exchange between Transactions in ITS-LPP

The transaction ID is assigned by the following method:

(1) The ID consists of 16 bits.

(2) The first bit indicates the transaction starting side. (“0” indicates a mobile station, and “1” indicates the base station.)

(3) Every time a new transaction is created, the ID is incremented by “1”.

3.3.1.1.2 Two Types of Transaction Services

The LPP provides the following two types of transaction services:

(1) Unidirectional data-sending transaction service

(2) Request-response type transaction service

Each transaction service is selected according to the communication requirements of each application. Accordingly, the optimum communication service for each application is available.

(1) Unidirectional data-sending transaction service

The LPP provides data-sending service to non-network type applications on both the base and mobile stations as shown in Figure 3.3-3.

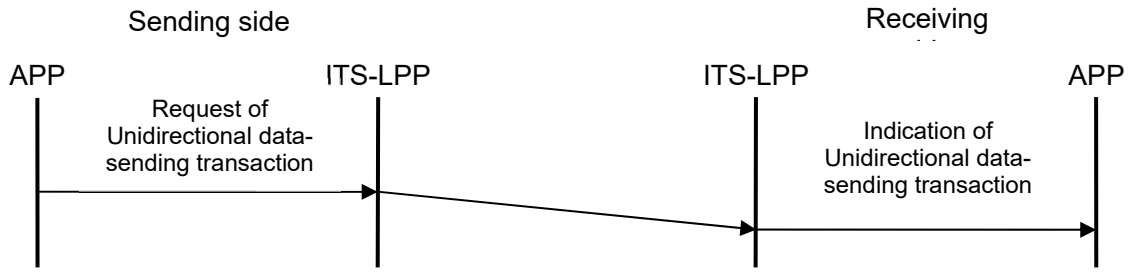


Figure 3.3-3 Example of Unidirectional Data-sending Service

(2) Request-response type transaction service

The LPP notifies the counterpart of a message, and acquires the returned value for the message. This service can be used such as to call a method on a remote station. (Refer to Figure 3.3-4.)

This transaction is available only for point to point communication.

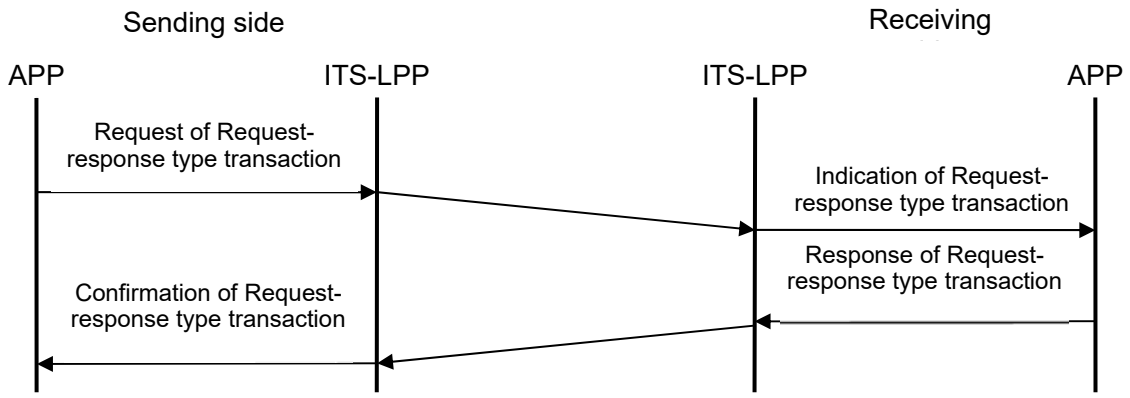


Figure 3.3-4 Example of Request-response Type Transaction Service

3.3.1.1.3 Data Resend Function (Optional)

This function is provided to ensure the reliability of communication, and controls resending of data using the resend timer and resend counter. When the resend timer expires, the LPP resends the data (up to the maximum number of times for resending) to ensure communication reliability as shown in Figure 3.3-5. This function can be applied to request and response data, and an application specifies whether or not to enable this function. The sequence is as follows:

When sending a packet, the LPP starts the resend timer and sets the resend counter to “0”. If data acknowledgement is not received before expiration of the resend timer, the LPP increments the resend counter and sends the packet again.

If the resend counter exceeds the maximum number of times for resending, the LPP concludes the transaction and notifies the application of transaction conclusion.

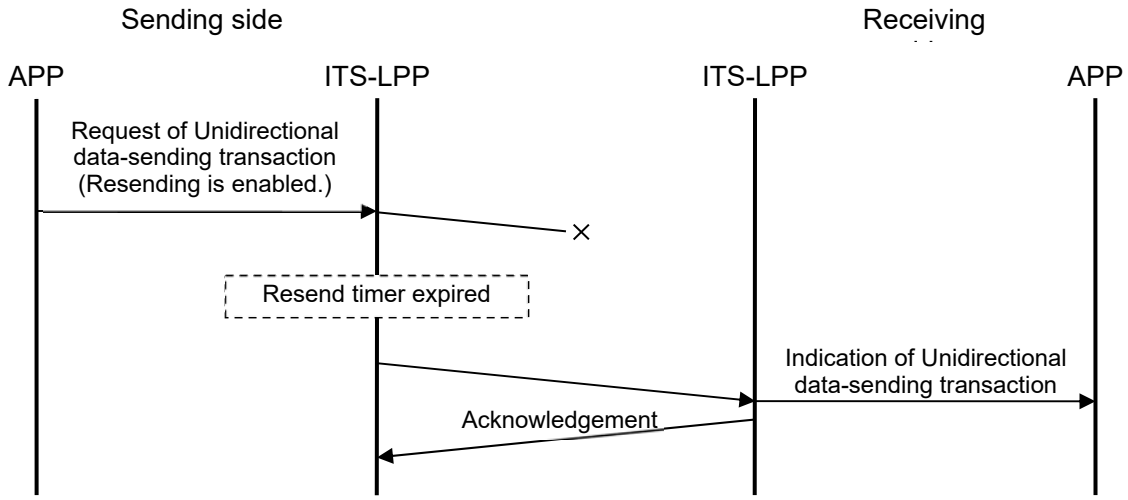


Figure 3.3-5 Example of Data Resending

In a transaction using the data resend function, the PDU received previously may be received again due to acknowledgement not being received, etc. The LPP checks for such duplicate receiving using the transaction ID as shown in Figure 3.3-6. The duplicate receiving check method is implementation issue, and is not specified in this specification.

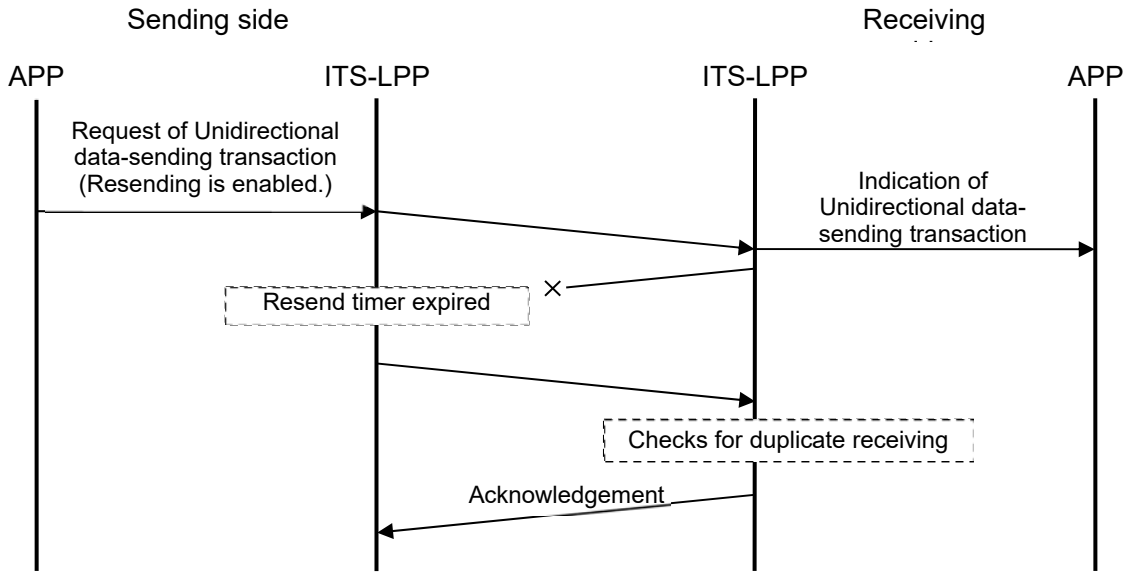


Figure 3.3-6 Example of Duplicate Receiving Check

Although this resend function is optional, the response of acknowledgement (ACK) and duplicate receiving check function is mandatory. The resend function is available only for point to point communication.

3.3.1.1.4 **Segmentation/re-assembly of Message**

This function segments and assembles a message so that the sending interface for a message, which exceeds the MTU of the LPCP can be provided to an application.

Figure 3.3-7 shows the message communication procedure using the segmentation/re-assembly function. When the LPP receives a message which exceeds the MTU of the LPCP from an application, the LPP segments the LPP-SDU into segments (packets) of the size less than the MTU of the LPCP (SUL: segment unit for LPP, 1386 octets), then transfers these packets in turn to the LPCP. In this process, segmented packets are added in the sending queue of the ITS-ASL, and then transferred in turn to the lower layer. At this time, because it is postulated that the sending queue of the DSRC-ASL will overflow, the LPP sends again packets whose sending has failed and controls the flow to assure that all packets can be sent.

The receiving side acquires in turn the segmented packets passed from the LPCP, and adds them in the receiving queue prepared by the application on the receiving side. At this time, it is not certain that each packet is stored in the receiving queue in the sending order due to the resend processing in lower layer and the other. Accordingly, the receiving side judges the assembly order based on

the ordering number assigned to each packet, and then assembles each packet into the PDU. After receiving all packets, the receiving side returns acknowledgement to the sending side.

Some packets may be missing due to overflow in the receiving queue in the DSRC-ASL or data loss in the DSRC, so it is not certain that all sent packets reach the LPP in the receiving side. In this case, loss of one packet means loss of the entire message data. Therefore the following method ensures the arrival of the entire message. If a packet has not been received when the final packet is received, the receiving side notifies those packets by negative acknowledgement, and the sending side sends missing packets included in negative acknowledgement again (selective resend process). With regard to loss of the final packet, its arrival is ensured by the normal resend process. The same control is applied for packets sent by selective resending.

Figure 3.3-8 shows an example of selective resend processing.

Because it is postulated that the size of the receiving queue required in each application is considerably different, each application prepares its own receiving queue for this function. Accordingly, with regard to transactions requiring disassembly and assembly, it is be possible to perform only one such transaction at a time for each sending destination (which is identified by the link address and destination port number).

In the case of sending data to a broadcast address, return of arrival acknowledgement, selective resend processing and final segment resend control are not performed. Instead, the required reliability of communication is assured by the transaction re-execution request.

Figure 3.3-8 shows an example of last Packet Resend Processing.

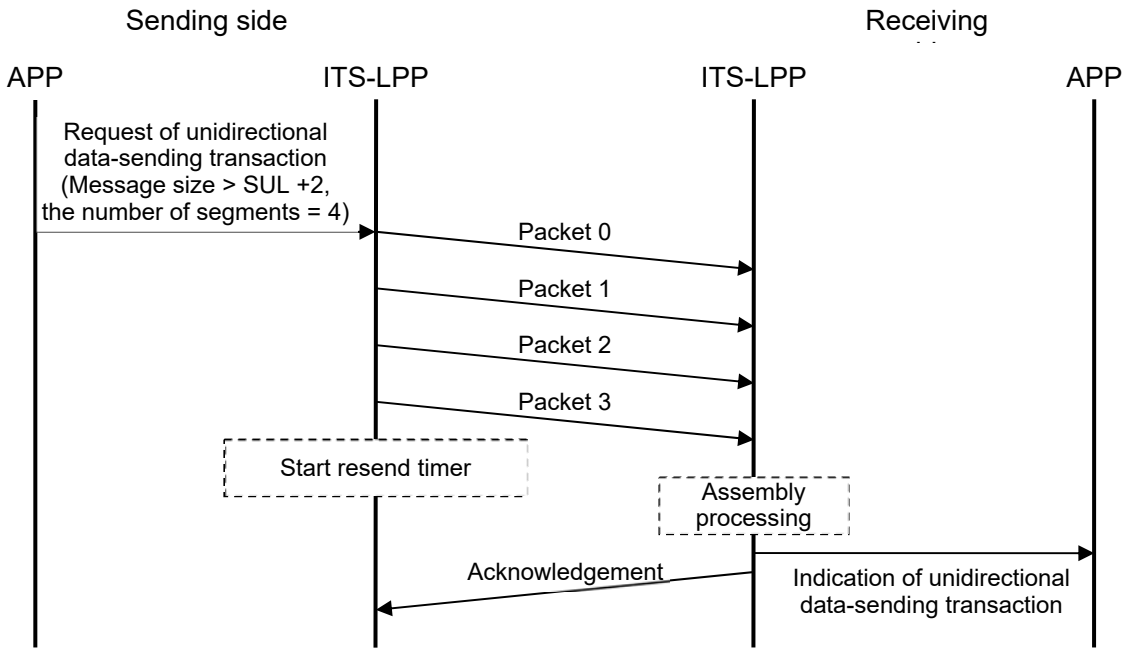


Figure 3.3-7 Example of Segmentation/re-assembly of Message

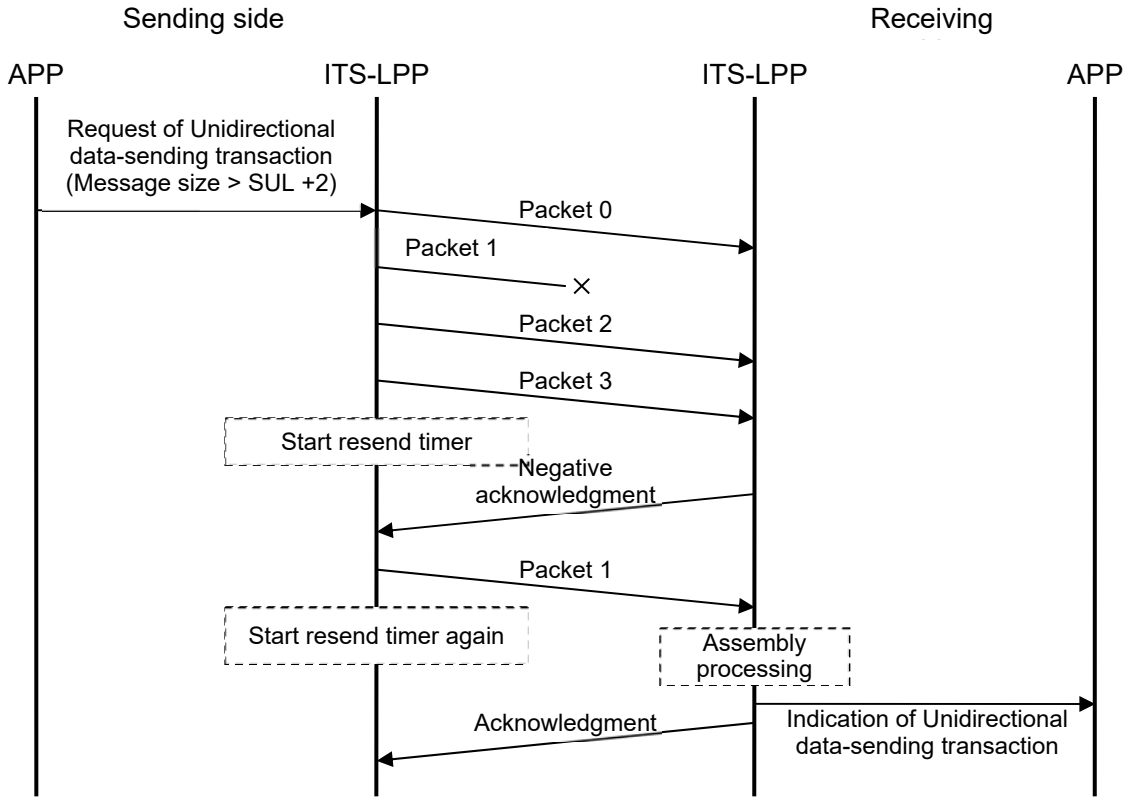


Figure 3.3-8 Example of Selective Resend Process

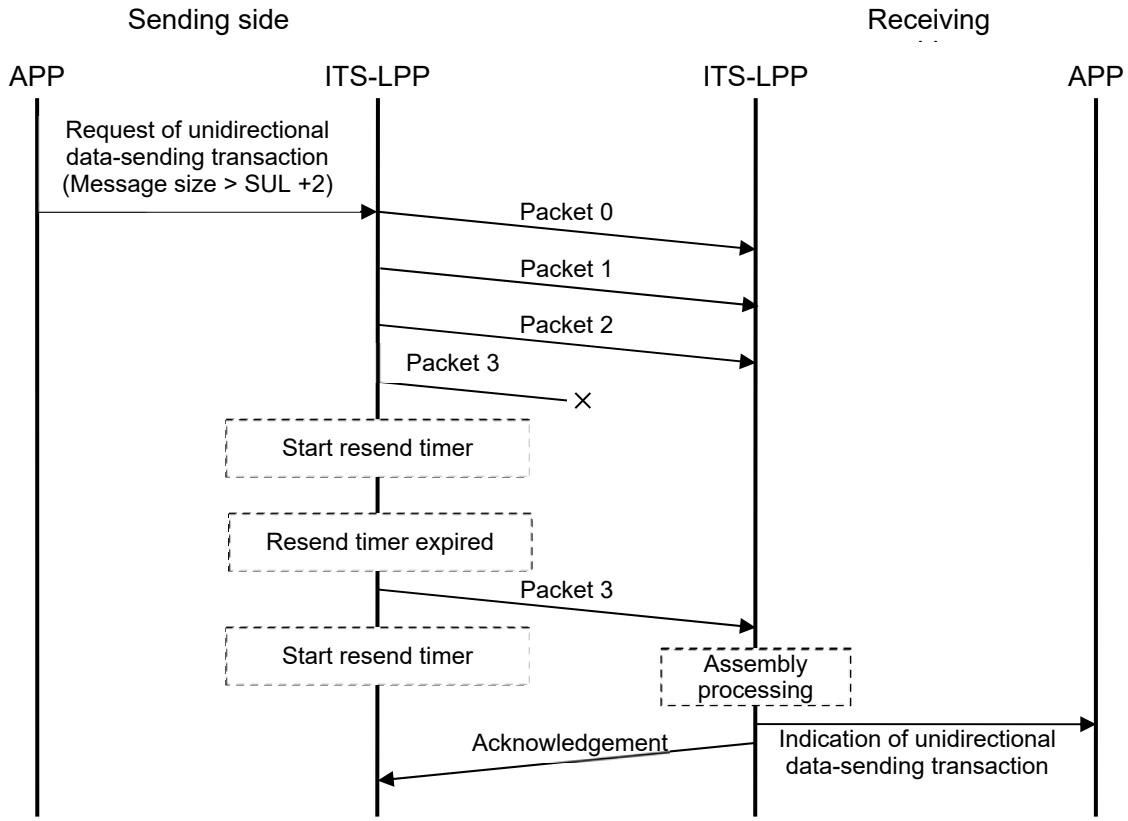


Figure 3.3-9 Example of Last Packet Resend Process

3.3.1.1.5 **Transaction Abortion Function**

Abortion of a transaction can be requested from an application or the system as shown in Figure 3.3-10. The LPP performs one of the following processes according to the transaction status at the time of request.

(1) When a message has not been sent, the LPP abandons the message.
(2) When a message has been already sent or is being sent, the LPP abandons all data related to the transaction, and notifies own station and remote station that the transaction has been aborted.
(3) When the transaction abortion request is received from the remote station, the LPP notifies the application that the transaction will be aborted, and then abandon all data related to the transaction.

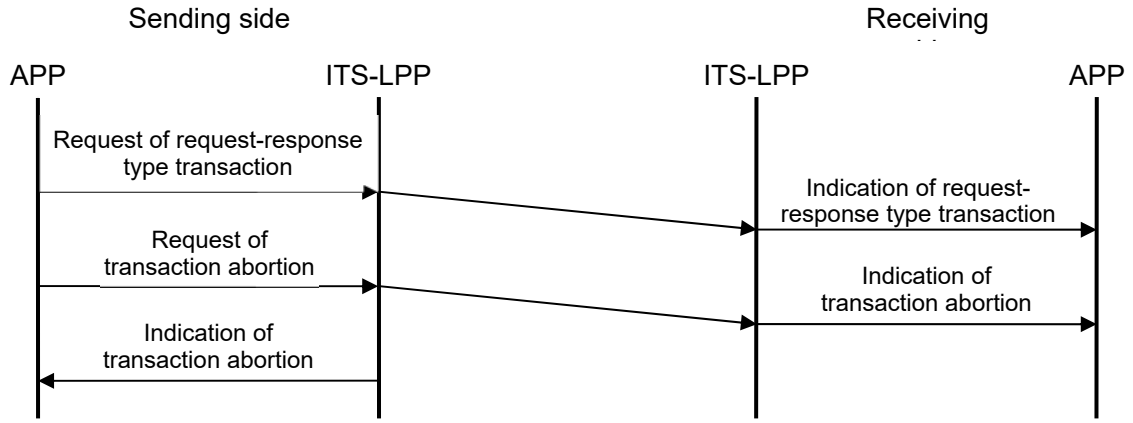


Figure 3.3-10 Example of Transaction Abortion

And in the following cases:

(1) when the DSRC is disconnected or
(2) when the destination port is not an accept port,

The LPP does not start the transaction, but notifies the application that the request has failed so that unnecessary communication is omitted.

3.3.1.2 **Connection Management Service**

In the connection management service, the LPP provides the start/end trigger of a communication to the application by the following services:

(1) The LPP manages and monitors the DSRC connection status, then reports the connection status and notifies new connections and disconnections by requests from an application.
(2) The service, which reports the situation of port by the request from an application or notifies to an application that the attribute of port is changed to acceptable. To provide this service, the LPP receives the accept port list event from the LPCP when the DSRC is connected and the LPP manages accept port numbers in the remote station by notifying accept port numbers and reject port numbers between the connection management service on the base station and the mobile station when accept ports are changed.

The connection management service is regarded as similar to an application on the local port control protocol. For sending and receiving events between the connection management service on the base station and the mobile station, the data transfer service of the local port control protocol is used. The port number used by the connection management service is 0x0FFF.

3.3.1.2.1 **Communication Connection Query Service**

The LPP has a function to query whether or not communication is connected.

Two types of services, reference service and notification service, are specified. In the reference service, the LPP immediately returns the communication connection status at the time of query. In the notification service, the LPP waits for connection if the communication is not connected, then provides notification of connection at the time of connection.

3.3.1.2.2 **Communication Disconnection Notify Service**

This service notifies the disconnection of communication to an application, which requires notification.

3.3.1.2.3 **Accept Port Query Service**

This service queries whether or not a port is present as receiving port in the remote station. There are three types of port statuses as follows:

(1) Accept port

This port is opened by the remote station as a data-receiving port.

(2) Reject port

This port is not opened by the remote station as a data-receiving port.

(3) Unknown port

It is not known whether or not this port is opened by the remote station as a data-receiving port. This is the initial status.

Two types of accept port query services, reference service and notification service, are specified. In the reference service, the LPP immediately returns the port status at the time of query. In the notification service, the LPP waits until the queried port changes to accept status, and then notifies the status when the accept port notification is received from the remote station (If it is already known that the queried port is a accept port, the service replies immediately.).

In order to enable the two types of services above, the management service in the LPP on the base station and a mobile station receives the accept port list event from the LPCP when the DSRC is connected, and then notify accept/reject port numbers in own station to remote station when accept ports are changed.

3.3.2 **Implementation Scope of LPP**

Table 3.3-1 shows the implementation range of each function of the LPP on the bidirectional type service and broadcast type service specified in 3.2.2.

Table 3.3-1 Implementation Scope of Local Port Protocol

		Broadcast type service		Bidirectional type service		
		Road traffic information (See NOTE below.)	Traffic / area guidance / sightseeing information	Collection of uplink information	Selective information provisioning	Charge / reservation
Functions of LPP	Unidirectional data-sending transaction service	M	M	M	M	M
	Request-response type transaction service	-	-	M	M	M
	Resend function	-	-	-	-	M
	Segmentation/ re-assembly function	-	M	M	M	-
	Transaction abortion	-	-	-	M	M
	Communication connection notification/ accept port query service	-	-	M	M	M
	Communication disconnection notification service	-	-	M	M	M
	Receiving port registration	M (mobile)	M (mobile)	M	M	M
	Receiving port deregistration	M (mobile)	M (mobile)	M	M	M

NOTE: When the sending data is disassembled into segments whose size is the MTU or less.

Symbol Description

M	Mandatory
M (mobile/base)	Mandatory only in mobile/base station
-	Not applicable

3.3.3 **Service Specification**

3.3.3.1 **Notation**

Table 3.3-2 shows the list of primitive types specified in this document.

Table 3.3-2 Primitive Types

Primitive type	Abbreviation	Description
Request	req	Used when an upper layer requests a service from a lower layer.
Indication	ind	Used when a lower layer notifies an upper layer of a service from the counterpart.
Response	res	Used when an upper layer gives a response to a lower layer for a service to the counterpart.
Confirm	cnf	Used when a lower layer notifies an upper layer that the requested service is completed.

Table 3.3-3 shows the list of parameter types used in the primitive definition table in this document.

Table 3.3-3 Parameter Types

Symbol	Description
M (mandatory)	Mandatory parameter
C (conditional)	Parameter specified when specified in the immediately preceding primitive (“req” in the case of “ind”, and “res” in the case of “cnf”)
O (optional)	Optional parameter
(=)	Indicates that the parameter value is equivalent to the value of the immediately preceding primitive (“req” in the case of “ind”, and “res” in the case of “cnf”).

3.3.3.2 **Service Primitives for Transaction Service**

The LPP prepares the following two types of primitives as the transaction service for applications.

(1) Invoke (transaction start primitive)

(2) Abort (transaction abortion primitive)

3.3.3.2.1 **Invoke**

(1) Function

The invocation of the “Invoke” service results in the generation of a new transaction.

Every transaction is started by calling this primitive.

(2) Parameters of service primitive

Table 3.3-4 Parameter of InvokePrimitive

Primitive parameter	Invoke			
	req	ind	Res	cnf
LinkAddress	M	M (=)	-	-
SourcePort	M	M (=)	-	-
DestinationPort	M	M (=)	-	-
UserDataSize	M	M (=)	M	M (=)
UserData	O	C (=)	O	C (=)
TransactionType	M	M (=)	-	-
RequireAck	O	-	O	-
ResultTimeout	O	-	-	-
Handle	M	M	M	M

The parameter “Link Address” is specific to the link address used in the ITS-ASL-ELCP.

The parameter “Source Port” is specific to the local port number of the sending source application.

The parameter “Destination Port” is specific to the local port number of the sending destination application.

The parameter “User Data Size” is specific to the size of the User Data (units: octets).

The parameter “User Data” s specific to the sending data.

The parameter “Transaction Type” is specific to the transaction service type. The value is shown in Table 3.3-5.

The parameter “Require Ack” is specific to the flag indicating whether resend processing is enabled. “0” indicates that resend processing is not necessary, and “1” indicates that the resend processing is necessary.

The parameter “Result Timeout” is specific to the timeout time concern with the receiving of PDU “Result” or PDU “ResultSegment” in the request-response type transaction service. If PDU “Result” or PDU “ResultSegment” does not arrive within this time after “Invoke.req” is executed, the transaction is aborted.

The parameter “Handle” is specific to the ID assigned to identify the transaction in the local station. The “Handle” specified here is required to satisfy the following conditions:

(a) On the sending side, the transaction ID is determined uniquely from the “Handle” and“Source Port”.

(b) On the receiving side, “Link Address”, “Source Port” and transaction ID is determined uniquely respectively from “Handle”. The parameter “Link Address” is specific to the link address used in the DSRC.

Table 3.3-5 Transaction Types

Transaction Type	Description	Remarks
0	Data-sending transaction service	
1	Request-response type transaction service	

3.3.3.2.2 **Abort**

(1) Function

The invocation of “Abort” service results in the abortion of an active transaction.

(2) Parameters of service primitive

Table 3.3-6 Parameter of AbortPrimitive

Primitive parameter	Abort	
	req	ind
AbortType	O	C (=)
AbortCode	O	C (=)
Handle	M	M

The parameter “Abort Type” is specific to the type for aborting: system error (0) or user request (1).

The parameter “Abort Code” is specific to the reason for aborting the transaction. The details of system errors are shown in Table 3.3-7.

The parameter “Handle” is the ID assigned to identify the transaction in the local station.

Table 3.3-7 Abort Code List

AbortCode	Code	Description
Unknown	0x00	-
Protocol error	0x01	The received PDU structure is abnormal.
TID is invalid.	0x02	The TID is invalid.
Transaction service is not supported.	0x03	The receiving side does not support the request-response type transaction service.
LPP version is different.	0x04	The LPP version is different between the sending side and the receiving side.
Receive buffer overflow	0x05	The receive buffer has overflowed.
MTU error	0x06	Because the send data exceeded the MTU in the LPCP, the transaction could not be started (when segmentation/re-assembly processing is not supported).
Resend timer timeout	0x07	The resend timer expired, and the transaction was aborted.
Result timer timeout	0x08	The result timer expired, and the transaction was aborted.
Link Address error	0x09	(Point to point) The vehicle is not present inside the zone. (Broadcast) The broadcast address is illegal.
Destination port error	0x0A	The destination port number is not present in the counterpart.
LPP is not supported.	0x0B	The DSRC-ASL does not support this protocol.
Aborted by DSRC-ASL.	0x0C	Because there was no space in the send queue in the DSRC-ASL, the requested service was aborted.
Transaction was not started.	0x0D	Because the number of transactions exceeded the number which can be executed at the same time, the transaction could not be started.
Under segmentation/assembly processing	0x0E	A transaction used segmentation/re-assembly processing is being executed.
Reserved for future use	0x0F to 0xFF	Reserved for future use

3.3.3.3 **Connection Management Service Primitive**

As the connection management service, the LPP provides the following four types of primitives for applications.

(1) Connect (transaction start query/notification primitive)

(2) Disconnect (Communication disconnection notification primitive)

(3) RegisterPort (Port registration primitive)

(4) DeregisterPort (Port deregistration primitive)

3.3.3.3.1 **Connect**

(1) Function

The invocation of “Connect” service results in the query whether or not a transaction may be started. The “Connect.cnf” primitive notifies the result of query by “Connect.req”. The results are

the Communication connection status, link address and accept port number in the remote station, which is indicated by the link address.

(2) Parameters of service primitive

Table 3.3-8 Parameter of ConnectPrimitive

Primitive parameter	Connect			
	req	ind	res	cnf
QueristPort	M			-
QueryLID	O			-
QueryPort	O			-
TimeOut	O			-
ConnectedLID	-			M
AcceptPort	-			M

The parameter “QueristPort” is specific to the local port number of the querist application.

The parameter “Query LID” is specific to the link address to be queried. When this parameter is specified, this primitive results in the query for a connected link. When this parameter is not specified, this primitive results in the waiting for new connection. When both “Query LID” and “Query Port” are omitted, “Connect.cnf” is issued immediately after the DSRC connects (fast connection mode). When “Query Port” is specified, “Connect.cnf” is issued after the accept port notification is received (normal connection mode).

The parameter “Query Port” is specific to the destination local port number to be queried.

The parameter “TimeOut” is specific to the timer value of “Connect” service. When the communication is connected during this timer is operating, “Connect.cnf” is issued immediately. If not this parameter, the timer value is infinity.

The parameter “Connected LID” is the same link address as “Query LID” when “Query LID” is specified and its link address is connected. When “Query LID” is specified and its link address is not connected or when “Query LID” is not specified and a new connection is not achieved within the time specified by “TimeOut”, “Connected LID” indicates “-1”.

The parameter “Accept Port” is the accept port number held by the remote station indicated by “Connected LID”. When the port number is specified by “Query Port”, “Accept Port” indicates only

the specified port number. When the specified local port number is a reject port number, “Accept Port” indicates “-1”. When “Query Port” is omitted, “Accept Port” indicates “0”.

3.3.3.3.2 **Disconnect**

(1) Function

The primitive “Disconnect” is used to notify an application that the DSRC is disconnected.

(2) Parameters of service primitive

Table 3.3-9 Parameter of DisconnectPrimitive

Primitive parameter	Disconnect	
	req	ind
LinkAddress	-	M

The “Link Address” parameter is the link address used in the ITS-ASL-ELCP.

3.3.3.3.3 **RegisterPort**

(1) Function

The primitive “RegisterPort” is used to register a receiving port for the LPP.

(2) Parameters of service primitive

Table 3.3-10 Parameter of RegisterPortPrimitive

Primitive parameter	RegisterPort
	req
PortNo	M
BulkArea	O
BulkAreaSize	O

The parameter “PortNo” is a receiving local port number.

The parameter “BulkArea” is the area to assemble disassembled segments of a message. This parameter is optional.

The parameter “BulkAreaSize” is the size of “BulkArea”. This parameter is optional.

3.3.3.3.4 **DeregisterPort**

(1) Function

The primitive “DeregisterPort” is used to deregister a receiving port for the LPP.

(2) Parameters of service primitive

Table 3.3-11 Parameter of DeregisterPortPrimitive

Primitive parameter	DeregisterPort
	Req
PortNo	M

The parameter “PortNo” is a receiving local port number to be deregistered.

3.3.4 **Protocol Data Unit (PDU)**

3.3.4.1 **PDUs in Transaction Service**

PDUs used in the transaction service are classified into 7 types as shown in Table 3.3-12. The PDU used in the transaction service consists of the header area defined for each PDU type and the data area, which indicates the application data. Figure 3.3-11 shows the basic structure of the LPP-PDU.

Table 3.3-12 PDU Type List

PDU type	Usage scenario
Invoke	Used in the primitive “Invoke.req”
Result	Used in the primitive “Invoke.res”
Acknowledgement	Used in arrival acknowledgement.
Abort	Used when a transaction is aborted (due to the primitive “Abort” or system error).
InvokeSegment	Used when the message size exceeds the MTU in the LPCP in the primitive “Invoke.req”.
ResultSegment	Used when the message size exceeds the MTU in the LPCP in the primitive “Invoke.res”.
Nack	Used in selective resend processing for segmentation/assembly processing.

Header area	Data area

Figure 3.3-11 Basic Structure of LPP-PDU

3.3.4.1.1 **Invoke PDU**

Table 3.3-13 Invoke PDU Header Information

Bit/Octet	7	6	5	4	3	2	1	0
1	PDU Type = Invoke (0x01)			Version		TT	RA	RD
2	TID							
3								

(1) PDU Type

This field indicates the PDU type. In the Invoke PDU, this field always indicates Invoke (1).

(2) Version

This field indicates the LPP version. The current version is 0x00.

(3) TT

This field indicates for Transaction Type, which is the flag indicating the transaction type. (1: in the case of request-response type transaction service 0: in the case of data-sending type transaction service)

(4) RA

This field indicates for Require Ack, which is the flag indicating whether or not resend processing is enabled. (1: enable 0: disable)

(5) RD

This field indicates for Retransmitted Data, which is the flag indicating whether or not the data are resent data. (1: resent 0: not resent)

(6) TID

This field indicates the transaction ID.

3.3.4.1.2 **Result PDU**

Table 3.3-14 Result PDU Header Information

Bit/Octet	7	6	5	4	3	2	1	0
1	PDU Type = Result (0x02)			RES	RES	RES	RA	RD
2	TID							
3								

(1) PDU Type

This field indicates the PDU type. In the Result PDU, this field always indicates Result (2).

(2) RA

This field indicates for Require Ack, which is the flag indicating whether or not resend processing is enabled. (1: enable 0: disable)

(3) RD

This field indicates for Retransmitted Data, which is the flag indicating whether or not the data are resent data. (1: resent 0: not resent)

(4) TID

This field indicates the transaction ID.

(5) RES

This field is reserved.

3.3.4.1.3 **Acknowledgement PDU**

Table 3.3-15 Acknowledgement PDU Header Information

Bit/Octet	7	6	5	4	3	2	1	0
1	PDU Type = Ack (0x03)			RES	RES	RES	RES	RD
2	TID							
3								

(1) PDU Type

This field indicates the PDU type. In the PDU Acknowledgement, this field always indicates Ack (3).

(2) RD

This field indicates for Retransmitted Data, which is the flag indicating whether or not the data are resent data. (1: resent 0: not resent)

(3) TID

This field indicates the transaction ID.

(4) RES

This field is reserved.

3.3.4.1.4 **Abort PDU**

Table 3.3-16 Abort PDU Header Information

Bit/Octet	7	6	5	4	3	2	1	0
1	PDU Type = Abort (0x04)			RES	RES	RES	RES	AT
2	TID							
3								
4	Abort Code							

(1) PDU Type

This field indicates the PDU type. In the PDU “Abort”, this field always indicates Abort (4).

(2) AT

This field indicates for Abort Type which is the flag indicating the reason for aborting. (1: request from the user 0: system error)

(3) TID

This field indicates the transaction ID.

(4) Abort Code

This field indicates the reason for aborting the transaction Abort code for system error is shown in Table 3.3-7.

(5) RES

This field is reserved.

NOTE: The destination port number and sending source port number should be judged based on

the TID.

3.3.4.1.5 **InvokeSegment PDU**

Table 3.3-17 InvokeSegment PDU Header Information

Bit/Octet	7	6	5	4	3	2	1	0
1	PDU Type = InvokeSegment (0x05)			Version		TT	FIN	RD
2	TID							
3								
4	Segment No							
5								

(1) PDU Type

This field indicates the PDU type. In the InvokeSegment PDU, this field always indicates InvokeSegment (5).

(2) Version

This field indicates the LPP version. The current version is 0x00.

(3) TT

This field indicates for Transaction Type which is the flag indicating the transaction type. (1: in the case of request-response type transaction service 0: in the case of data-sending type transaction service)

(4) FIN

This field indicates whether or not the segment is final. (1: when the segment is final 0: in any other case)

(5) RD

This field indicates for Retransmitted Data, which is the flag indicating whether or not the data are resent data. (1: resent 0: not resent)

(6) TID

This field indicates the transaction ID.

(7) Segment No

This field indicates the sequence number of the PDU.

3.3.4.1.6 **ResultSegment PDU**

Table 3.3-18 ResultSegment PDU Header Information

Bit/Octet	7	6	5	4	3	2	1	0
1	PDU Type = ResultSegment (0x06)			RES	RES	RES	FIN	RD
2	TID							
3								
4	Segment No							
5								

(1) PDU Type

This field indicates the PDU type. In the ResultSegment PDU, this field always indicates ResultSegment (6).

(2) FIN

This field indicates whether or not the segment is final. (1: when the segment is final 0: in any other case)

(3) RD

This field indicates for Retransmitted Data, which is the flag indicating whether or not the data are resent data. (1: resent 0: not resent)

(4) TID

This field indicates the transaction ID.

(5) RES

This field is reserved.

(6) Segment No

This field indicates the serial number of the PDU.

3.3.4.1.7 **Nack PDU**

Table 3.3-19 Nack PDU Header Information

Bit/Octet	7	6	5	4	3	2	1	0
1	PDU Type = Nack (0x07)			RES	RES	RES	RES	RD
2	TID							
3								
4	NumSeg = *n*							
5								
6 : 6+2n-1	Segment Number List							

(1) PDU Type

This field indicates the PDU type. In the Nack PDU, this field always indicates Nack (7).

(2) RD

This field indicates for Retransmitted Data, which is the flag indicating whether or not the data are resent data. (1: resent 0: not resent)

(3) TID

This field indicates the transaction ID.

(4) RES

This field is reserved.

(5) NumSeg

This field indicates the count of sequence numbers of PDUs not received yet.

(6) Segment Number List

This field indicates the list of sequence numbers of PDUs not received yet.

3.3.4.2 **PDUs in Connection Management Service**

The connection management service of the LPP notifies the accept port list and reject port list to the remote station by using the transfer service of the LPCP when the communication is newly connected or when the number of accept ports has increased or decreased. PDUs used in these notifications are shown below and they are stored in the user data area in the LPCP PDU.

3.3.4.2.1 **Accept Port PDU**

Table 3.3-20 Accept Port PDU

Bit/Octet	7	6	5	4	3	2	1	0
1	Status = acceptPort (1)							
2	AcceptPort							
3								

(1) Status

This field indicates the event type. This field always indicates “acceptPort (1)” in case of accept port list notification.

(2) AcceptPort

This field indicates accept port numbers.

3.3.4.2.2 **Reject Port PDU**

Table 3.3-21 Reject Port PDU

Bit/Octet	7	6	5	4	3	2	1	0
1	Status = rejectPort (2)							
2	RejectPort							
3								

(1) Status

This field indicates the event type. This field always indicates rejectPort (2) in case of reject port list notification.

(2) RejectPort

This field indicates reject port numbers.

3.3.5 **Procedure**

3.3.5.1 **Initial Set up Procedure**

This paragraph describes the initial set up procedure for normal connection applications and fast connection applications using point to point communication. An application with broadcast communication can be executed without the initial set up procedure.

(1) Initial Set up Procedure for Normal Connection Applications

(a) Each application in a mobile station and the base station registers accept port numbers in the LPP by the port registration primitive (RegisterPort).

(b) The LPP updates the connection management table, and registers the accept port numbers specified in (a) and the connection management service port as the data receiving ports and the event receiving ports in the LPCP.

(c) Each application with “QueryLID” and “Query Port”, issues the “Connect.req” primitive, and waits for connection of the DSRC (blocking call).

(d) The connection management service of the LPP receives the event “Connection notice (96)” from the LPCP through the “EventReport.indication”.

(e) The connection management service of the LPP prepares the connection management table for the link address received through the primitive above. For an application requiring fast connection, the LPP accepts thereafter transaction start requests to all ports at this link address and broadcast addresses until the accept port list of the remote station is received from the LPCP.

(f) When the connection management service of the LPP receives the accept port list from the LPCP through the “EventReport.indication”, the LPP registers the accept ports in the connection management table at the link address specified by the primitive above. Thereafter, the LPP accepts transaction start requests to this link address only for these accept ports.

(g) The LPP notifies an application which issued the “Connect.req” primitive to a port number included in the accept port list received in (f) the link address and sending accept port numbers by the “Connect.cnf” primitive.

(h) When the application issues “Invoke.req” primitive with the link address specified by the“Connect.cnf” primitive or broadcast addresses and the destination port number, the transaction is started.

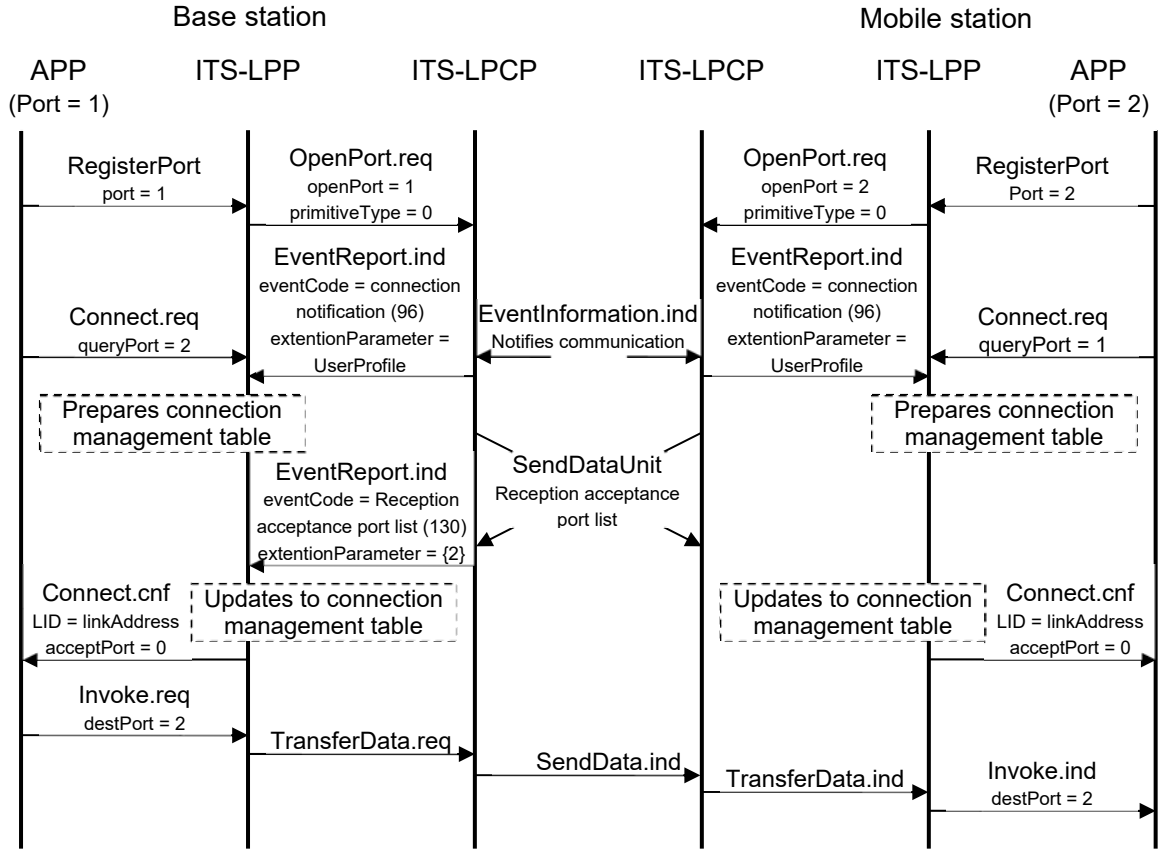


Figure 3.3-12 Example of Initial Set up Procedure of LPP

(2) Initial Setup Procedure for Fast Connection Applications

(a) Each application in a mobile station and the base station registers accept port numbers to the LPP by the port registration primitive (RegisterPort).

(b) The LPP updates the connection management table, and registers the accept port numbers in the LPCP.

(c) Each application issues the “Connect.req” primitive, without “QueryLID” and “Query Port”, and waits for connection of the communication.

(d) The LPP receives the event “connection notice (96)” from the LPCP through the “EventReport.indication”.

(e) The LPP prepares the connection management table for the link address received through the primitive above. For an application requiring fast connections, the LPP accepts thereafter transaction start requests for all ports for this link address and broadcast addresses until the accept port list of the remote station is received from the LPCP.

(f) The LPP notifies an application issuing the “Connect.req” primitive of (the link address) through the “Connect.cnf” primitive.

(g) Each application issues the “Invoke.req” primitive with the link address notified through the “Connect.cnf” primitive or broadcast addresses to start the transaction.

(h) If the port number specified in (g) is present in the remote station, this transaction succeeds. If the port number specified in (g) is not present in the remote station, the LPCP in the remote station notifies the event “The destination local port is invalid. (129)” through the event notification primitive, and the LPP updates the connection management table for this link address. In the case of Transaction Type = “1”, the LPP notifies the corresponding application that the transaction has failed through the “Abort.ind” primitive. If an “Invoke.req” primitive is given with this combination of link address and port number after that, the LPP notifies that the transaction is aborted through the “Abort.ind” primitive.

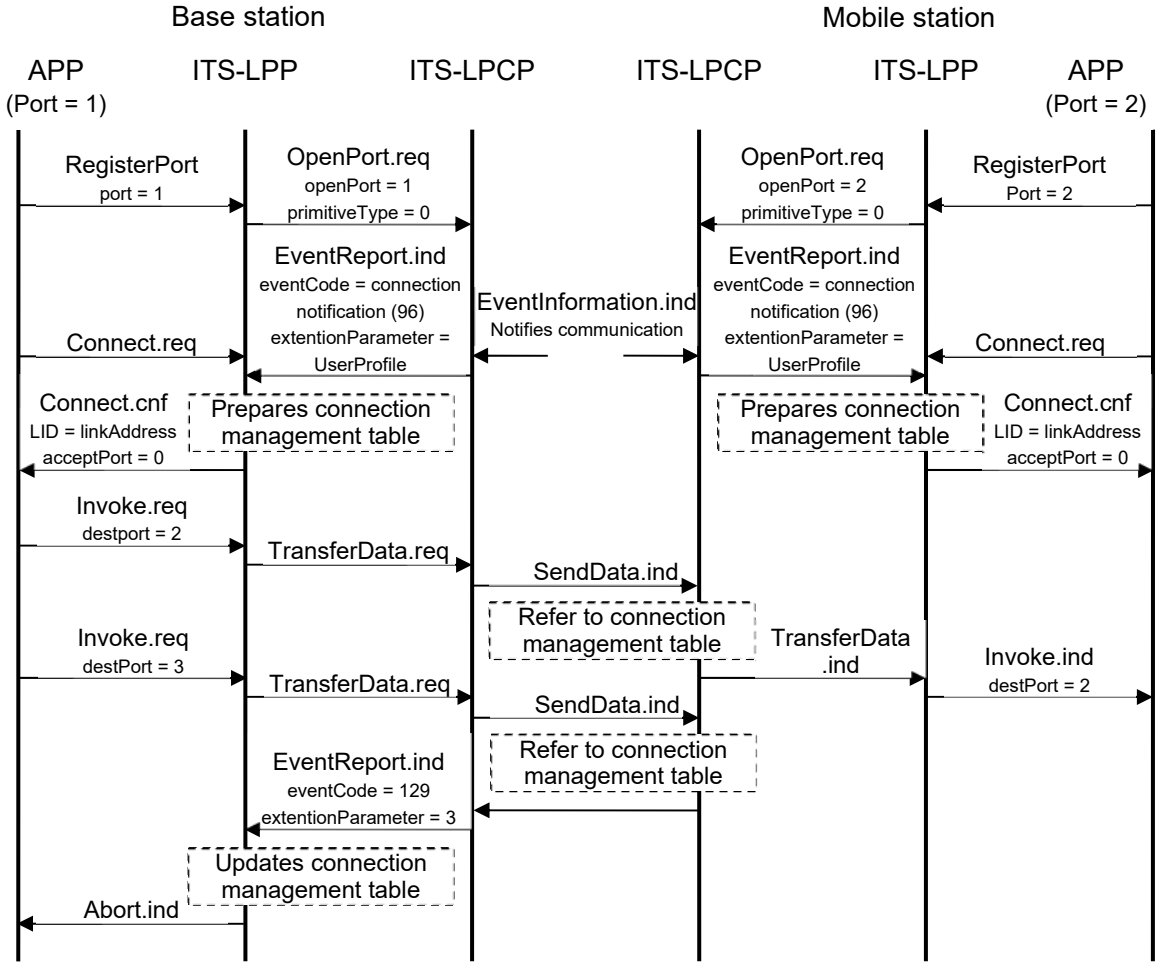


Figure 3.3-13 Example of Initial Set up Sequence in a Fast Connection Application

3.3.5.2 **Unidirectional Data-sending Transaction Service**

(1) Send Processing

(a) When an application issues the “Invoke.req” primitive with Transaction Type = “0”, a data-sending transaction service is started.

(b) When the combination of specified link address and destination port number indicates a reject port, the LPP notifies the application of “destination port error” through “Abort.ind”, and the transaction is ended.

(c) When the specified message exceeds the MTU in the LPCP and segmentation/assembly processing is not supported, the LPP notifies the application of “MTU error” through the “Abort.ind” primitive, and the transaction is ended. The processing to be performed when segmentation/assembly processing is supported is described in 3.3.5.5.

(d) In any case other than (b) and (c), the LPP prepares the Invoke PDU with TT set to 0, and then sends it to the remote station through the “TransferData.request” in the LPCP. The processing to be performed when resend processing is valid is described in 0.

(2) Receive processing

(a) When the Invoke PDU sent in (1)-(d) through the “TransferData.indication” in the LPCP is received, the LPP notifies the application of the received data through the “Invoke.ind” primitive.

Figure 3.3-14 shows an example of the sequence of the data transfer procedure in a data-sending transaction service.

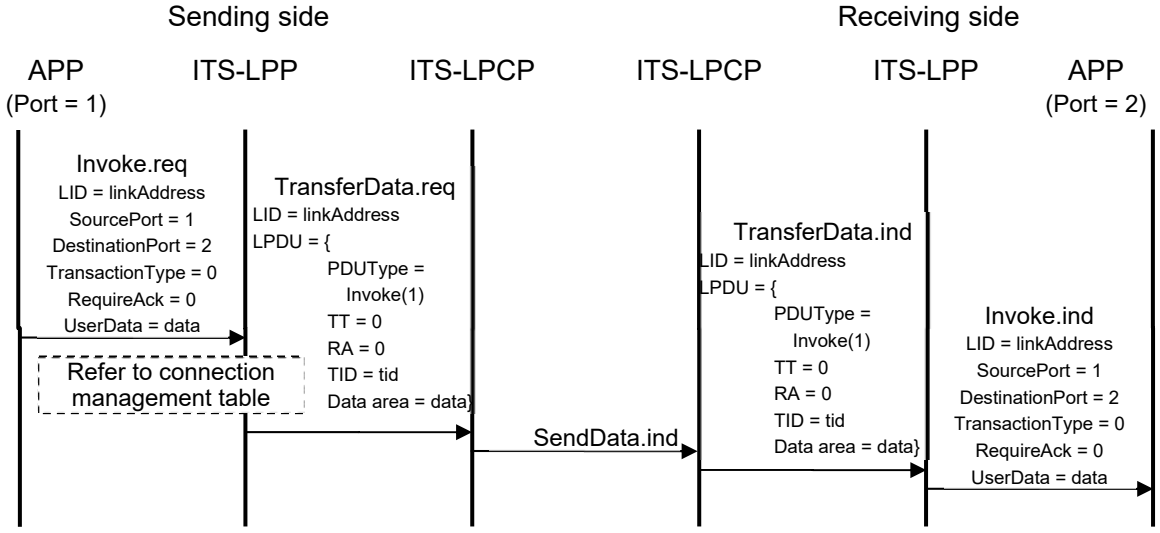


Figure 3.3-14 Example of Unidirectional Data-sending Transaction Service

3.3.5.3 **Request-response Type Transaction Service**

(1) Send Processing

(a) When an application issues the “Invoke.req” primitive with Transaction Type = 1, a request-response type transaction service is started.

(b) When the combination of specified link address and destination port number indicates a reject port, the LPP notifies the application of “destination port error” through “Abort.ind”, and the transaction is ended.

(c) When the number of transactions exceeded executable at a time, the LPP notifies the application of “Transaction could not be started.” through “Abort.ind”, and the transaction is ended.

(d) In any case other than (b) and (c), the LPP prepares the Invoke PDU with TT set to 1, sends it to the remote station through the “TransferData.request” in the LPCP, starts up the Result timer (with the timeout value specified by “Invoke.req”), and waits to receive the Result PDU from the remote station.

(e) When the Result timer started up in (d) expires, the LPP prepares the Abort PDU with AT set to 0 and Abort Code set to 0x08, notifies the remote station of “Result timer timeout”, and then notifies the application that the transaction has failed through the Abort.ind primitive.

(f) When the Result PDU sent by the remote station through the “TransferData.indication” in the LPCP is received before the Result timer expires, the LPP stops the Result timer started up in (d), and then notifies the application of the response data through the “Invoke.cnf” primitive.

(2) Receive processing

(a) When the Invoke PDU sent by the remote station through the ”TransferData.indication” in the LPCP is received, the LPP notifies the application of the receiving data through the “Invoke.ind” primitive, and waits to receive the “Invoke.res” primitive from the application.

(b) When the Abort PDU sent by the remote station through the “TransferData.indication” in the LPCP is received, the LPP issues the “Abort.ind” primitive, and notifies the application that the transaction has failed. The transaction is then ended.

(c) The application issues the “Invoke.res” primitive, and requests the LPP to send a response.

(d) The LPP prepares the Result PDU, and sends it to the remote station through the “TransferData.request” in the LPCP.

NOTE: When segmentation/assembly processing is used in “Invoke.req” and “Invoke.res”, the send processing and receive processing is performed according to the procedure described in 0.

Figure 3.3-15 shows an example of the basic sequence in a request-response type transaction service. Figure 3.3-16 shows an example of the sequence when the Result timer expires.

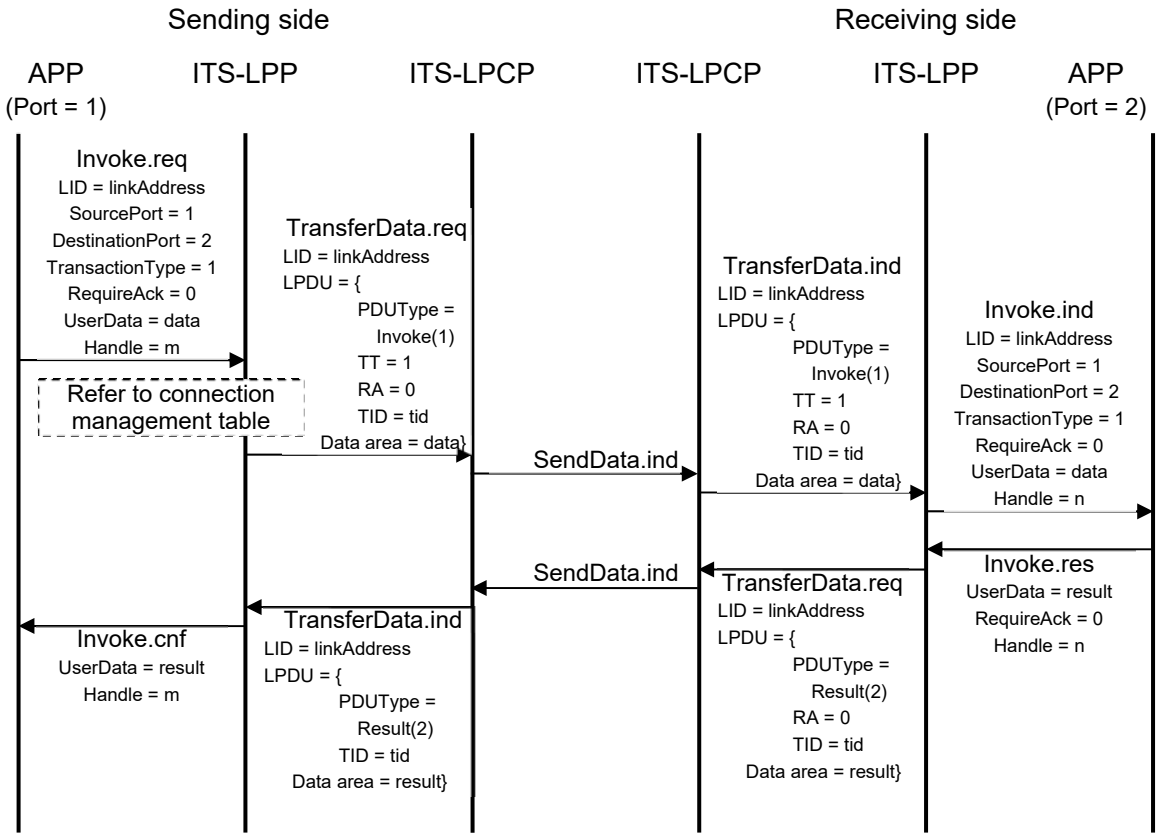


Figure 3.3-15 Example of Request-response Type Transaction Service

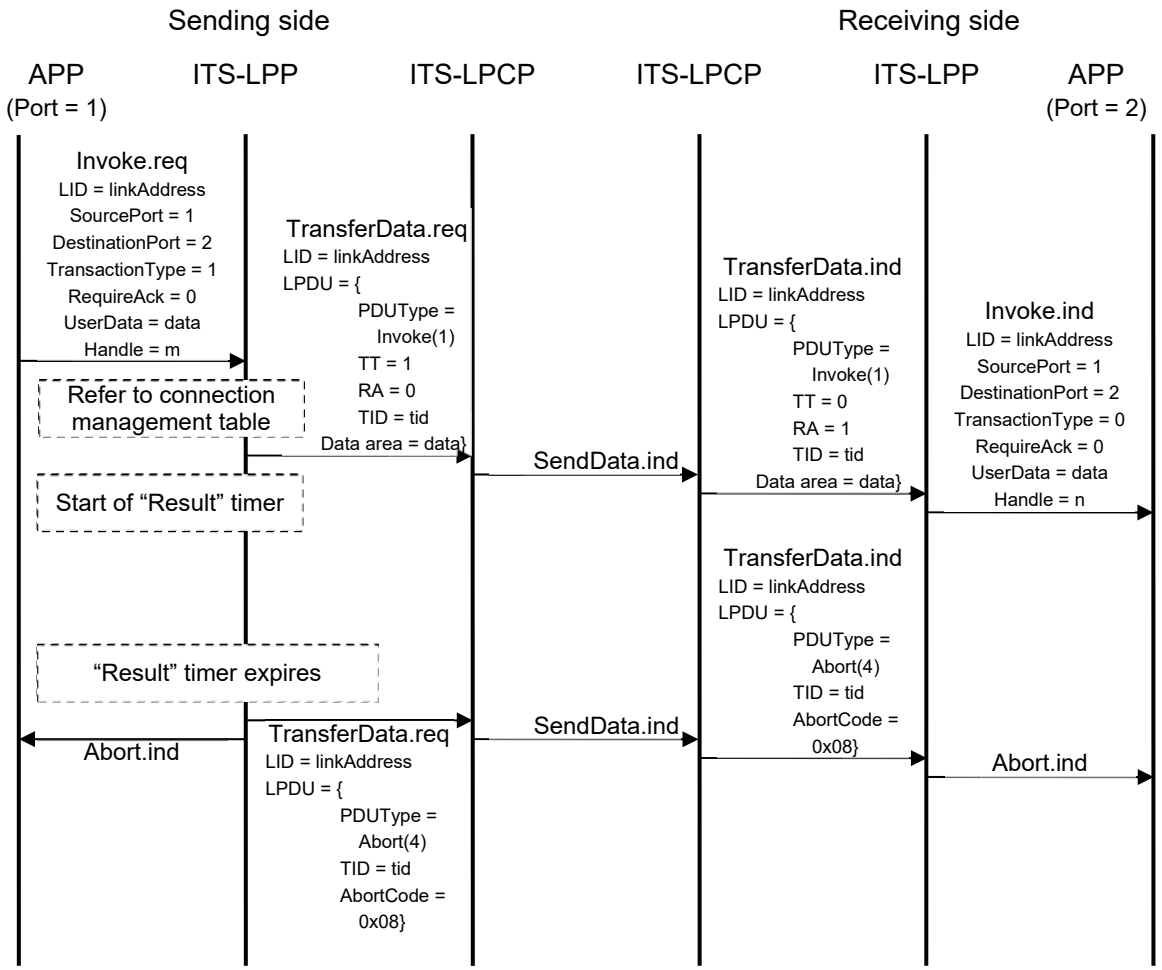


Figure 3.3-16 Example of Request-response Type Transaction Service (Result timer expires)

3.3.5.4 **Resend Process**

Resend processing is used when “Require Ack” is set to “1” in “Invoke.req” and “Invoke.res”. This paragraph describes the sequence when resend processing is used in “Invoke.req” in a data-sending transaction. In the request-response type transaction service, the same processing is also available in “Invoke.res”.

(1) Send processing

(a) When an application issues the “Invoke.req” primitive with Require Ack = “1”, data transfer service in which resend processing is enabled is started.

(b) The LPP prepares the Invoke PDU with RA set to 1, sends it to the remote station using the “TransferData.request” in the LPCP, starts the resend timer, and waits to receive the PDU Acknowledgement from the remote station.

(c) When the resend timer started in (b) expires before the LPP receives the PDU Acknowledgement for some reason (such that the Invoke PDU sent in (b) does not arrive), the LPP sets the RD flag to 1 in the Invoke PDU sent in (b), sends the Invoke PDU again to the remote station, restarts the resend timer, and increments the resend counter.

(d) When the resend counter exceeds the maximum number of resend times after resend in(c), the LPP prepares the Abort PDU with AT set to “0” and Abort Code set to “0x07”, notifies the remote station of “Resend timer timeout”, notifies the application that the transaction has failed through the “Abort.ind” primitive, and transaction is ended.

(e) When the PDU Acknowledgement sent by the remote station through the “TransferData.indication” in the LPCP is received before the resend timer expires, the LPP stops the resend timer started up in (b) or (c), and completes this transaction.

(2) Receive processing

(a) When the Invoke PDU is received through the “TransferData.indicaton” in the LPCP, the LPP notifies the application of the received data through the “Invoke.ind” primitive.

(b) When the RA flag is valid in the PDU received in (a), the LPP prepares the PDU Acknowledgement, sends it to the remote station through the “TransferData.request” in the LPCP, and starts up the wait timer.

(c) When the Invoke PDU received in (a) is received again for some reason (such as that the PDU Acknowledgement sent in (b) does not arrive), the LPP aborts this PDU, prepares the PDU Acknowledgement again, sends it to the remote station through “TransferData.request” in the LPCP, and starts up the wait timer again.

(d) When the wait timer started up in (b) or (c) expires, the LPP ends this transaction is ended.

Figure 3.3-17 shows an example of the sequence when resend processing is enabled.

Figure 3.3-18 shows an example of the sequence when resend processing has succeeded.

Figure 3.3-19 shows an example of the sequence when resend processing has failed.

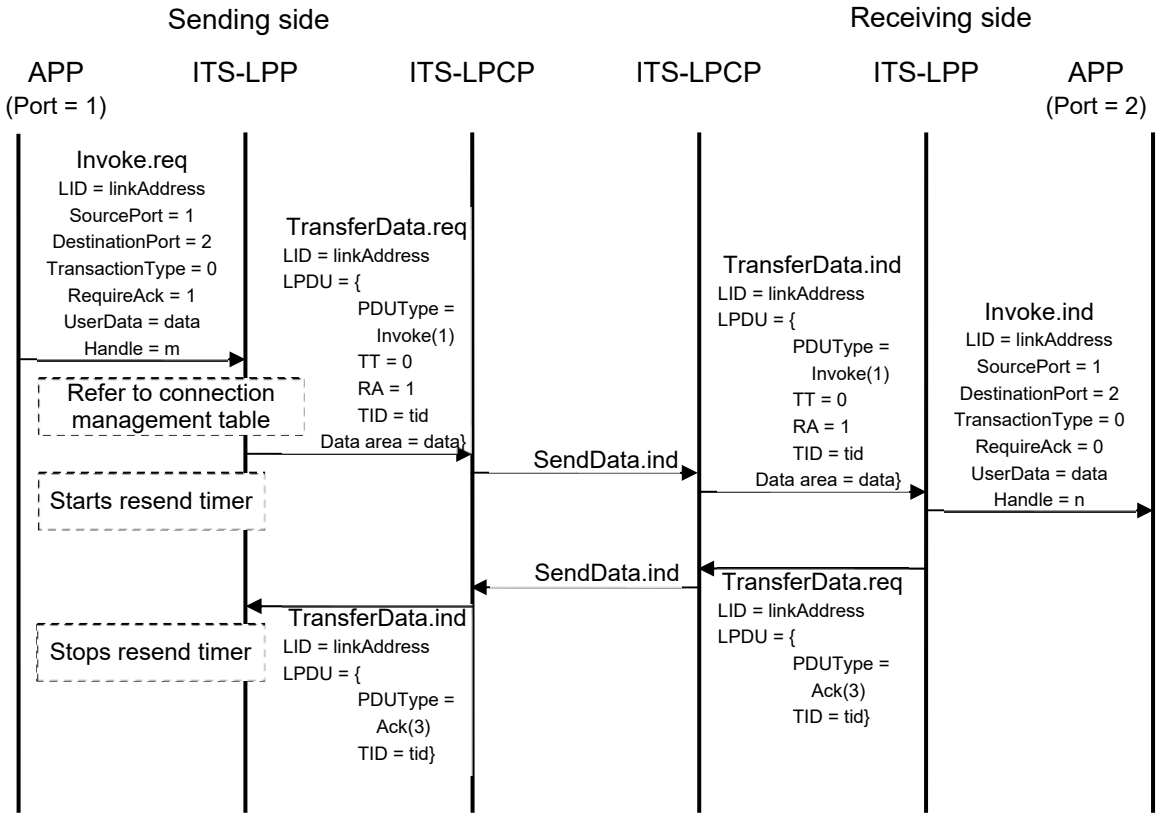


Figure 3.3-17 Example of Resend Processing (Basic sequence)

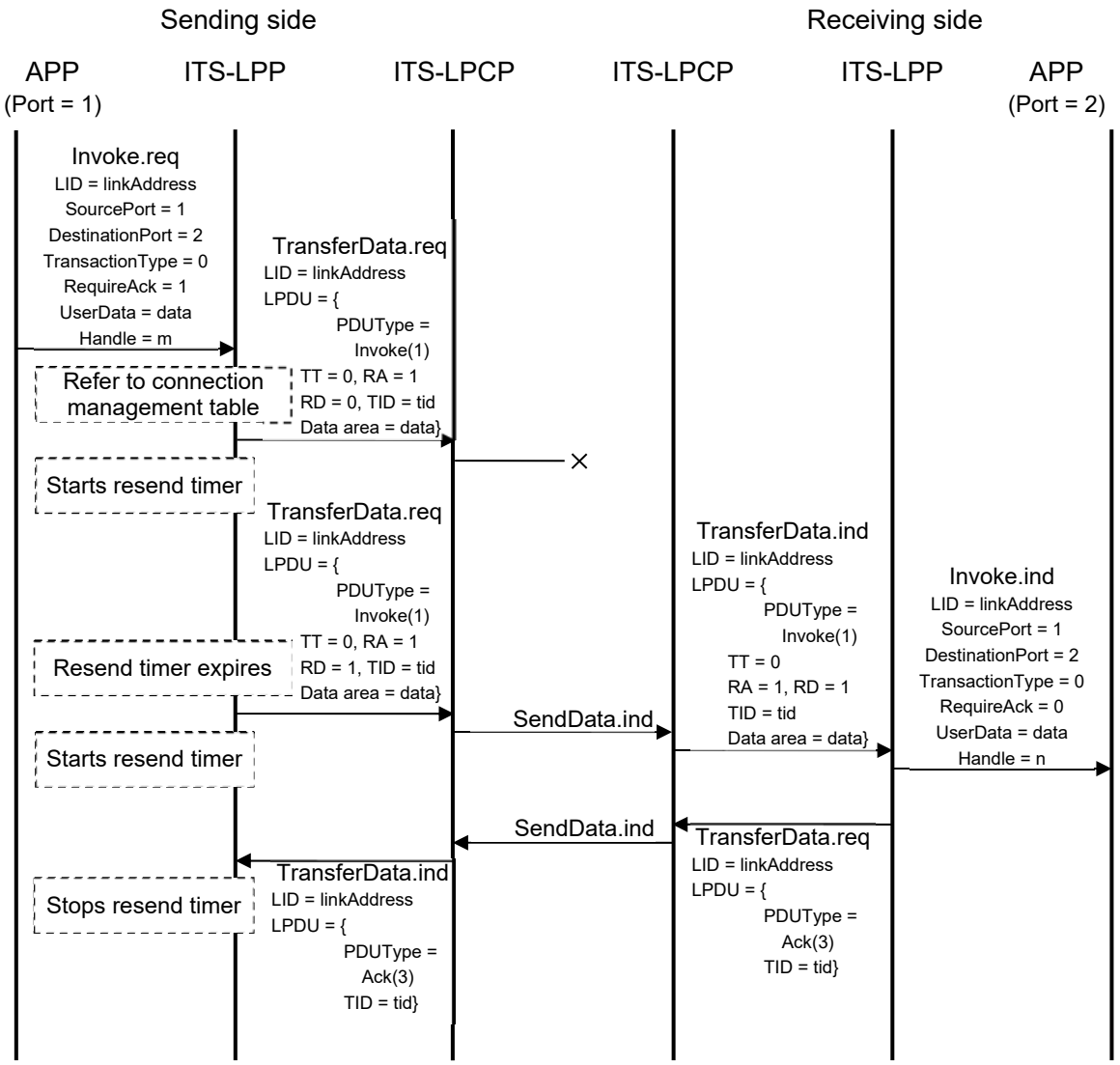


Figure 3.3-18 Example of Resend Processing (When resend has succeeded)

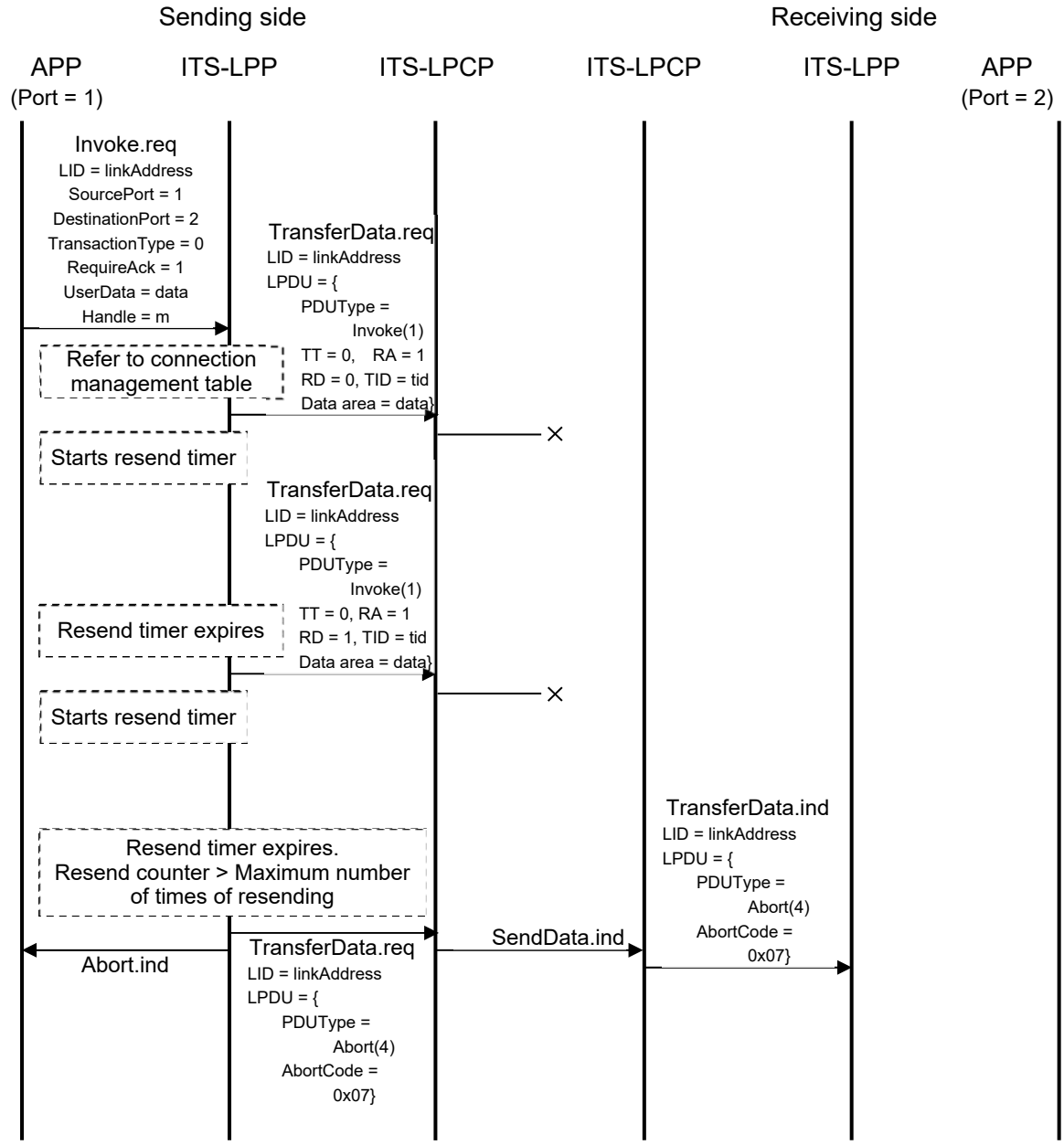


Figure 3.3-19 Example of Resend Processing (When resend has failed)

3.3.5.5 **Segmentation/re-assembly Process**

Segmentation/re-assembly processing is used when a message exceeding the MTU in the LPCP is specified in “Invoke.req” and “Invoke.res”. This paragraph describes the sequence when segmentation/re-assembly processing is used in “Invoke.req”.

(1) Send processing

(a) When an application specifies a message whose size exceeds the MTU in the LPCP and issues the “Invoke.req” primitive, a transaction is started in the data sending service in which the segmentation/assembly processing is enabled.

(b) When the combination of specified link address and destination port number indicates a reject port, the LPP notifies the application of “Destination port error” through the “Abort.ind” primitive.

(c) When a transaction with the segmentation/assembly processing is already being executed for the combination of specified link address and destination port number, he LPP notifies the application of “Under segmentation/assembly processing” through the “Abort.ind” primitive.

(d) In any case other than (b) and (c), the LPP disassembles the message from the top sequentially in the SUL, adds the header of the “InvokeSegment” PDU (described in 3.3.4.1.5) to each disassembled segment, and then send each segment in turn through the “TransferData.request” in the LPCP. At this time, the LPP assigns segment number (Segment No) “0” to the first segment, and then assigns an incremented value as the segment number in turn to each of the following segments. If the link address is a broadcast one and the Handle is same as that in the transaction with broadcast communication executed just before preceding the current transaction, the LPP sets the RD flag into each segment, and uses the same transaction ID in the transaction executed just before with the broadcast communication.

(e) When the sending queue in the ASL has overflowed and the LPCP notifies the LPP of the “The sending service is aborted because of sending queue overflow.” through the “EventReport.indication”, the LPP waits for a certain period of time, then starts sending again, including data whose sending failed previously.

(f) In the point to point communication, the LPP starts up the resend timer after sending the last segment data, waits to receive the PDU Acknowledgement or Nack from the remote station, and then performs the following processing. In the broadcast communication, the LPP completes the processing after sending the last segment data without performing the following processing.

(g) When the PDU Nack sent by the remote station through the “TransferData.indication” in the LPCP is received, the LPP resends the segments specified in Segment Number List in the

PDU Nack. At this time, the LPP sets the RD flag to 1 in all segments to be sent again, and sets the FIN flag to 1 in the segment sent again at the end. After resending all segments, the LPP starts up the resend timer, and waits to receive the PDU Acknowledgement or Nack from the remote station.

(h) When the resend timer started up in (f) of (g) expires, the LPP sends the final segment again, and starts up the resend timer again.

(i) When the PDU Acknowledgement sent by the remote station through the “TransferData.indication” in the LPCP is received, the LPP stops the resend timer started up in (f), (g) or (h), and then completes this transaction.

(2) Receive processing

(a) An application specifies the buffer area for assembly of receiving data through the port registration primitive (RegisterPort).

(b) When the “InvokeSegment” PDU is received through the “TransferData.indication” in the LPCP, the LPP sets it based on sequence number in the buffer area.

(c) When the final segment is received, the LPP checks whether there are any un-received segments. If there are un-received segments, the LPP prepares the PDU Nack, sends it to the remote station through the “TransferData.request” in the LPCP, and then memorizes the final segment number.

(d) When data whose RD flag is not set for some reason (such as a change of the arrival order after sending the PDU Nack in (c)) are received, the LPP aborts the data.

(e) In point to point communication, if the LPP has received all segment data when the final segment is received, the LPP notifies the application of the received data using the “Invoke.ind” primitive, prepares the PDU Acknowledgement, and sends it to the remote station using the “TransferData.request” in the LPCP.

(f) In the broadcast communication, when the LPP has received all segment data, the LPP notifies the application of the received data using the “Invoke.ind” primitive. If segment data included in this transaction are received after issuing the “Invoke.ind” primitive, the LPP aborts it.

Figure 3.3-20 shows an example of the basic sequence with the segmentation/assembly processing. Figure 3.3-21 shows an example of the sequence when some segment data are missing and selective resend processing is performed. Figure 3.3-22 shows an example of the sequence when the final segment data are missing and resend processing is performed. Figure 3.3-23 shows an example of the sequence for broadcast communication.

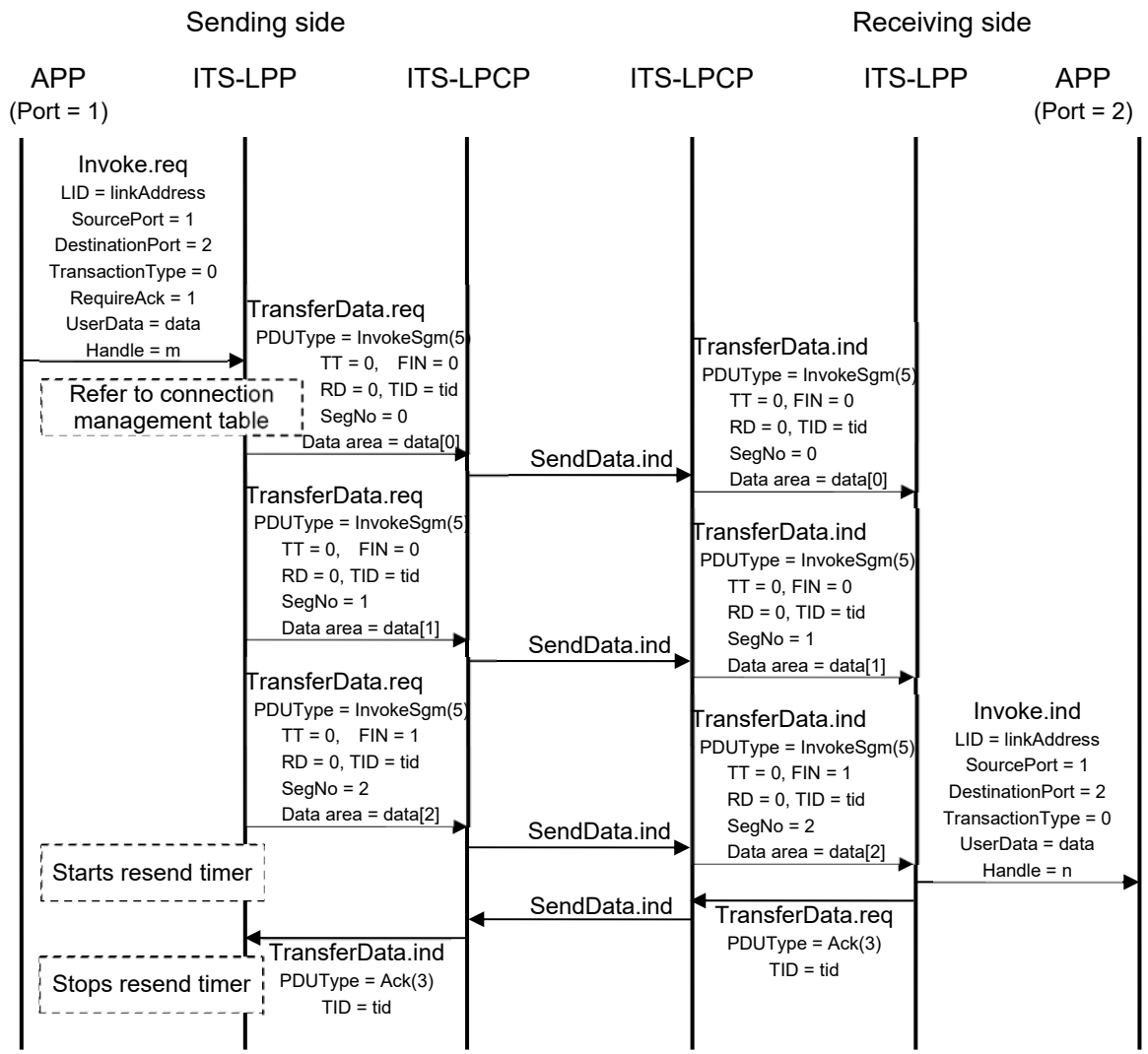


Figure 3.3-20 Example of Segmentation/re-assembly processing

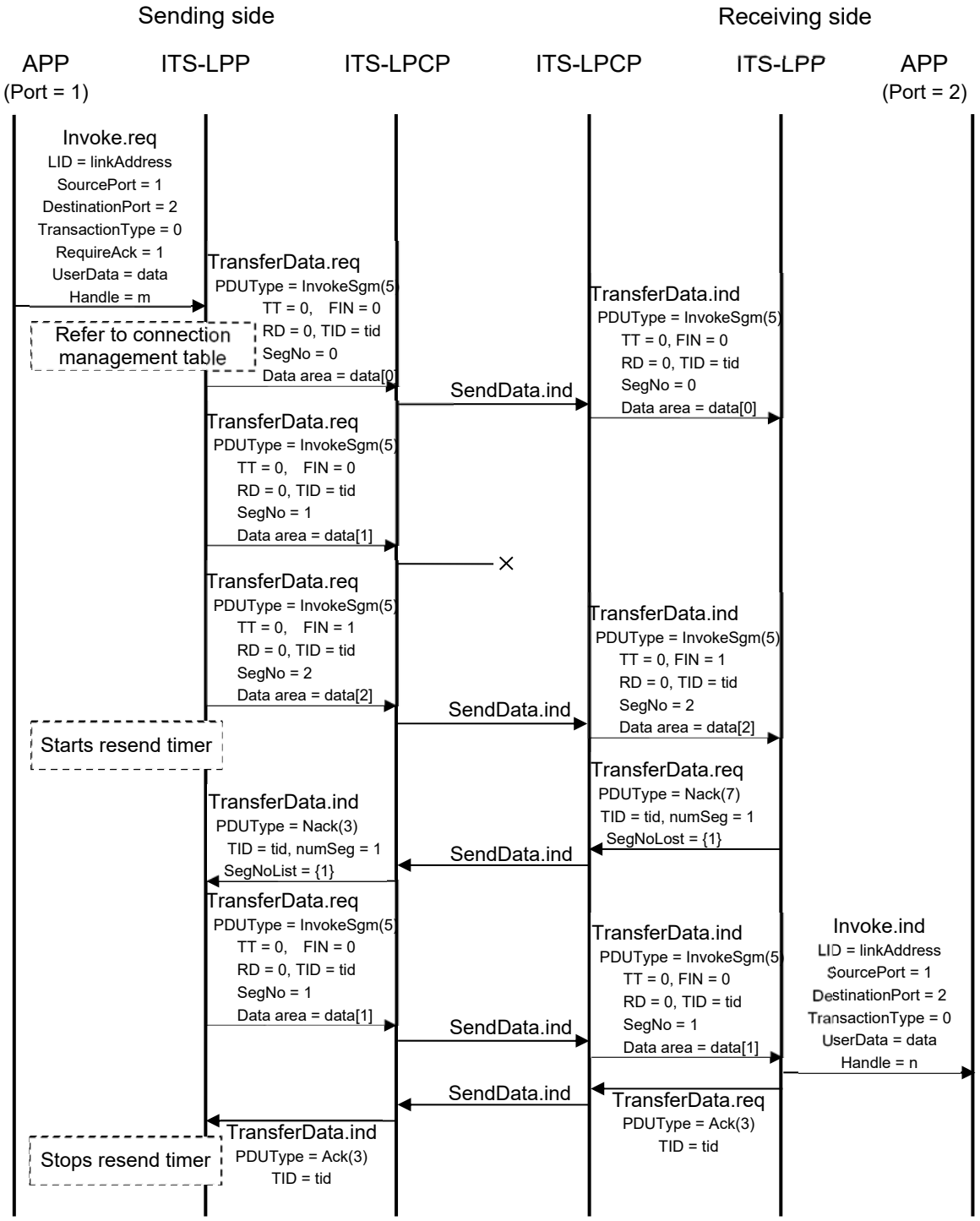


Figure 3.3-21 Example of Segmentation/re-assembly Process (Selective Resend process)

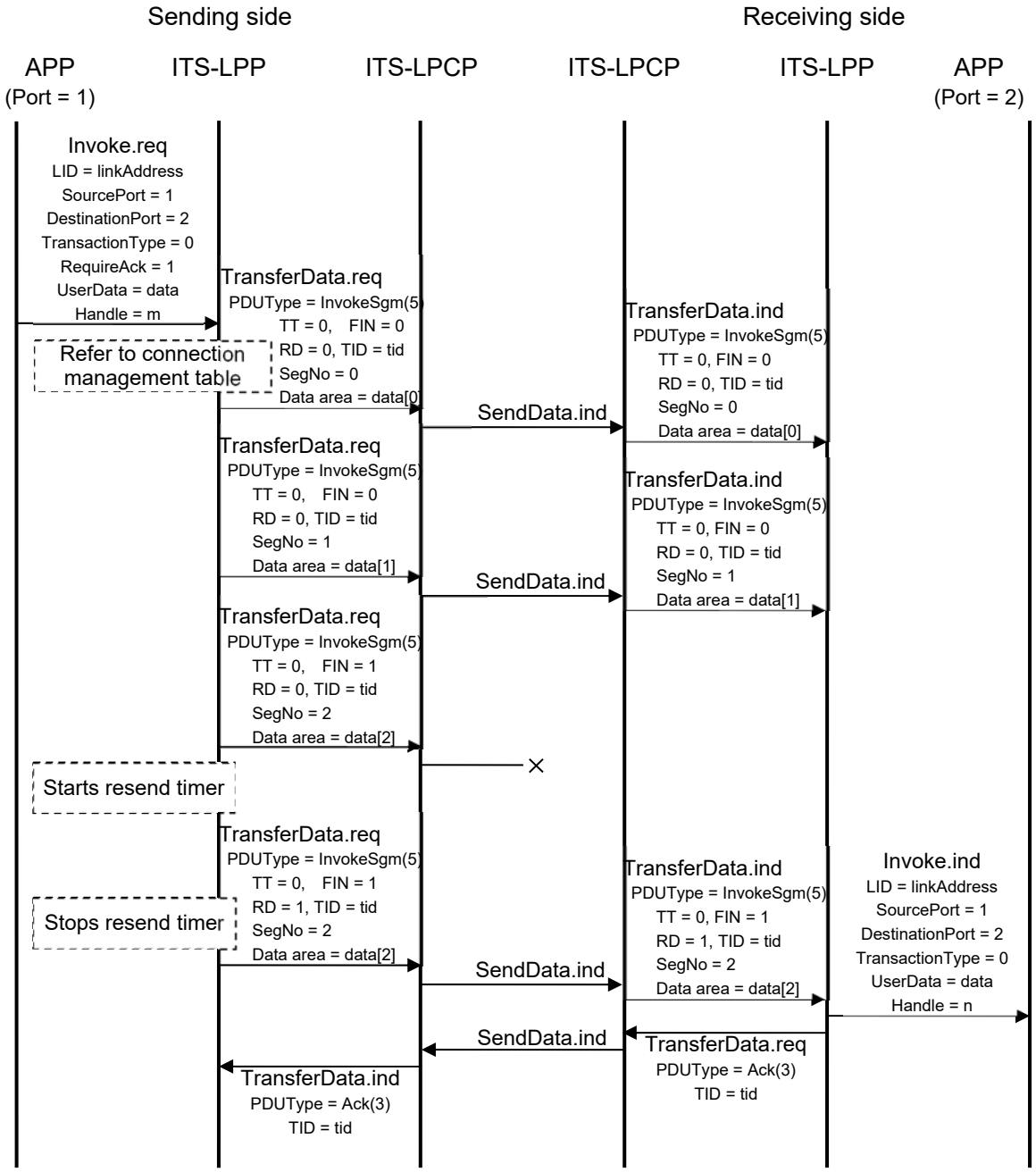


Figure 3.3-22 Example of Segmentation/re-assembly Process

(When final segment data have not been received)

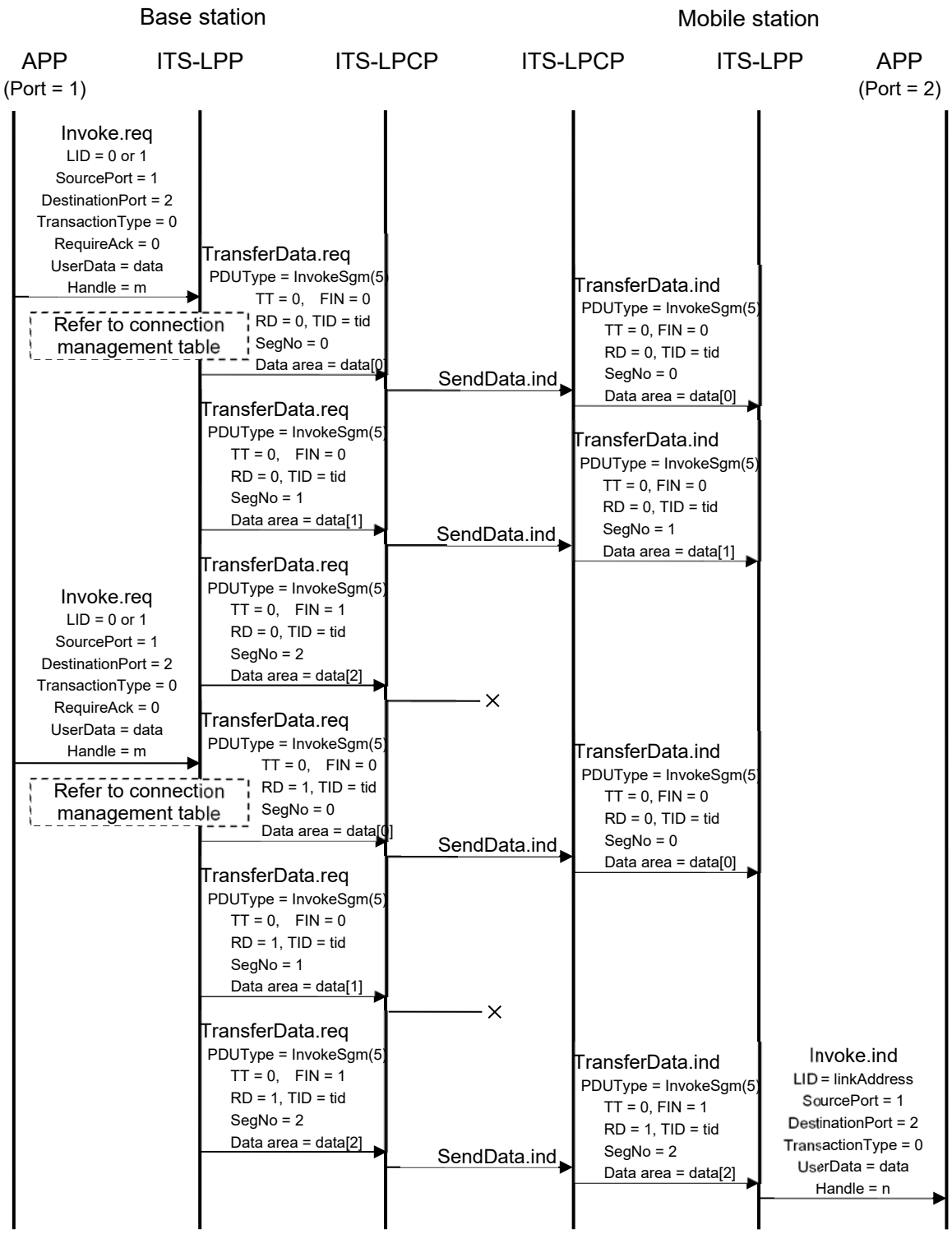


Figure 3.3-23 Example of Segmentation/re-assembly process in Broadcast Communication

3.3.5.6 **Communication End Procedure**

(1) Communication End Procedure

(a) The LPP receives the event “disconnection notice (97)” from the LPCP through the “EventReport.indication”.

(b) The LPP issues the “Disconnect.ind” primitive to the application, which is using the corresponding link address.

(c) The LPP deletes the connection management table for the link address received in the primitive above. Thereafter, the LPP will not accept any transaction start each segment for this link address.

Figure 3.3-24 shows an example of the sequence in the communication end procedure.

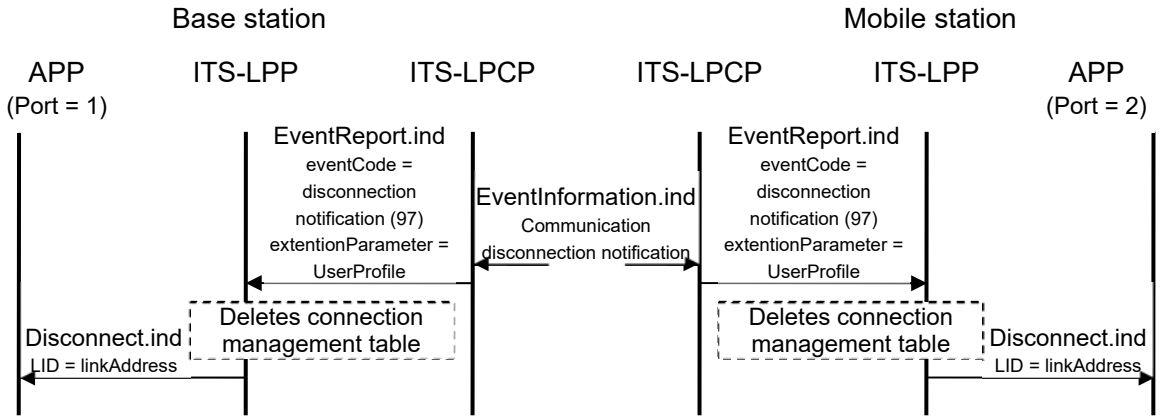


Figure 3.3-24 Procedure when Communication is Disconnected

3.3.5.7 **Transaction Abort Procedure**

The LPP accepts a request from an application to abort a transaction when the transaction is in the following status:

(1) Sending side

From accepting “Invoke.req” to issuing “Invoke.cnf” in a request-response type transaction

(2) Receiving side

From issuing “Invoke.ind” to sending the Result PDU in a request-response type transaction

The sequence of transaction about procedure is as follows:

(a) When the LPP receives the “Abort.req” primitive from an application, this sequence is

started.

(b) The LPP prepares the Abort PDU for the transaction specified in the primitive above, and then sends it to the remote station through the “TransferData.request” in the LPCP.

(c) The LPP issues the “Abort.ind” primitive to the requesting application to notify that aborting of the transaction has been completed.

(d) When the Abort PDU is received through the “TransferData.indication” in the LPCP, the LPP aborts all resources related to the transaction specified in the PDU above if it is being executed in the local station, and then issues the “Abort.ind” primitive to the application to notify that the transaction has been aborted.

Figure 3.3-25 shows an example of the sequence in the transaction abort procedure.

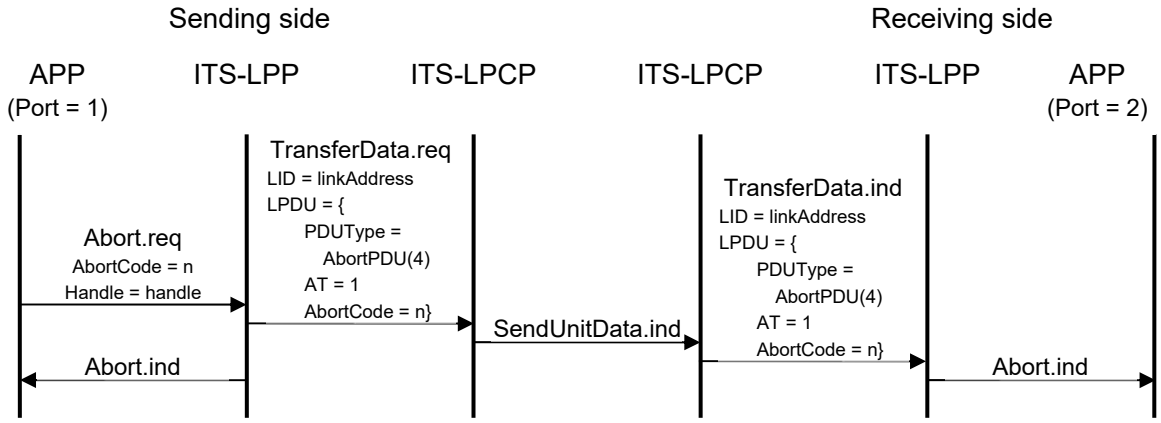


Figure 3.3-25 Transaction Abort Procedure

3.3.5.8 **Accept Port Change Procedure**

The LPP may dynamically change accept ports while the DSRC is connecting according to the registration and deregistration of accept ports by applications.

(1) Accept Port Addition Procedure

(a) While the communication is connecting, each application in a mobile station and the base station registers accept port numbers to the LPP using the port registration primitive (RegisterPort).

(b) The LPP updates the connection management table, and registers the accept port numbers specified in (a) as data receiving ports in the LPCP.

(c) The management service of the LPP prepares the PDU for notification of accept ports using the registered accept port numbers, and sends the PDU to the management service in the remote station using the “TransferData.request” in the LPCP.

(d) When the accept port notification sent in (c) is received, the management service of the LPP registers the received ports in the connection management table for the link address of this message. Thereafter, the LPP may accept transaction start requests to these accept ports.

Figure 3.3-26 shows an example of the sequence for the accept port addition procedure.

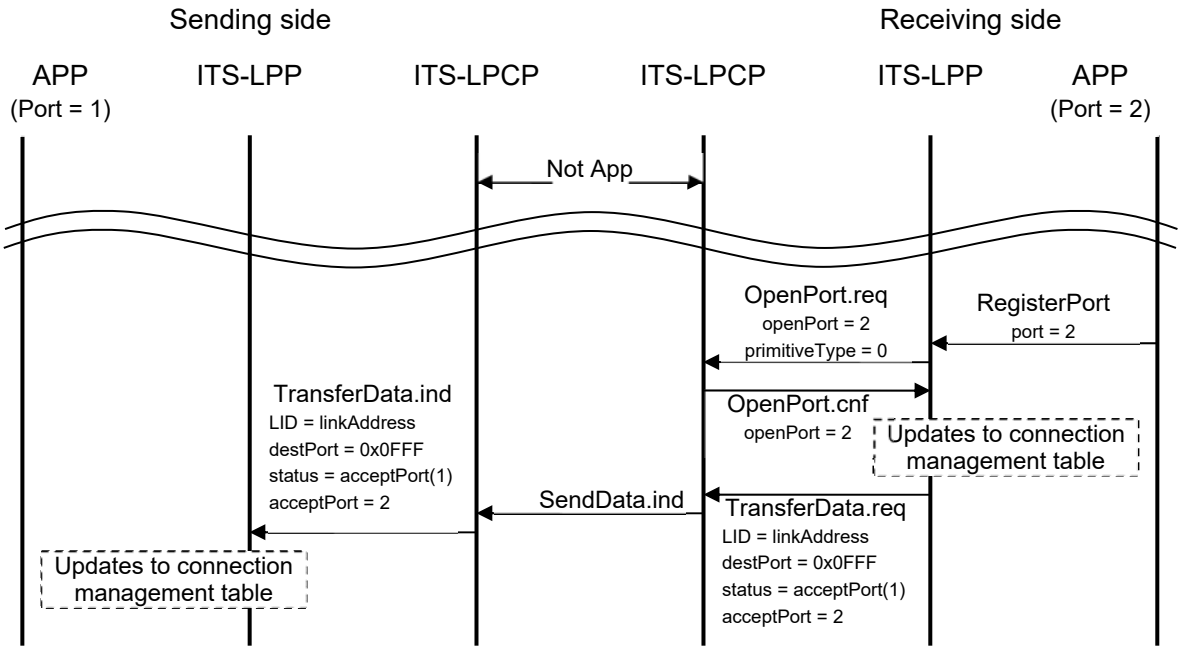


Figure 3.3-26 Example of Accept Port Addition Procedure

(2) Accept port deletion procedure

(a) While the communication is connecting, each application in a mobile station and the base station deregisters accept port numbers from the LPP using the port deregistration primitive (DeregisterPort).

(b) The LPP updates the connection management table, and deregisters the accept port numbers deregistered in (a) from the accept port list in the LPCP using the “ClosePort.request”.

(c) The management service of the LPP prepares the PDU for notification of reject ports using the deregistered port numbers, and sends the PDU to the management service in the remote station using the “TransferData.request” in the LPCP.

(d) When the reject port notification sent in (c) is received, the management service in the LPP deregisters the received ports from the connection management table for the link address of this message. Thereafter, the LPP will not accept transaction start requests to these reject ports.

Figure 3.3-27 shows an example of the sequence for the accept port deletion procedure.

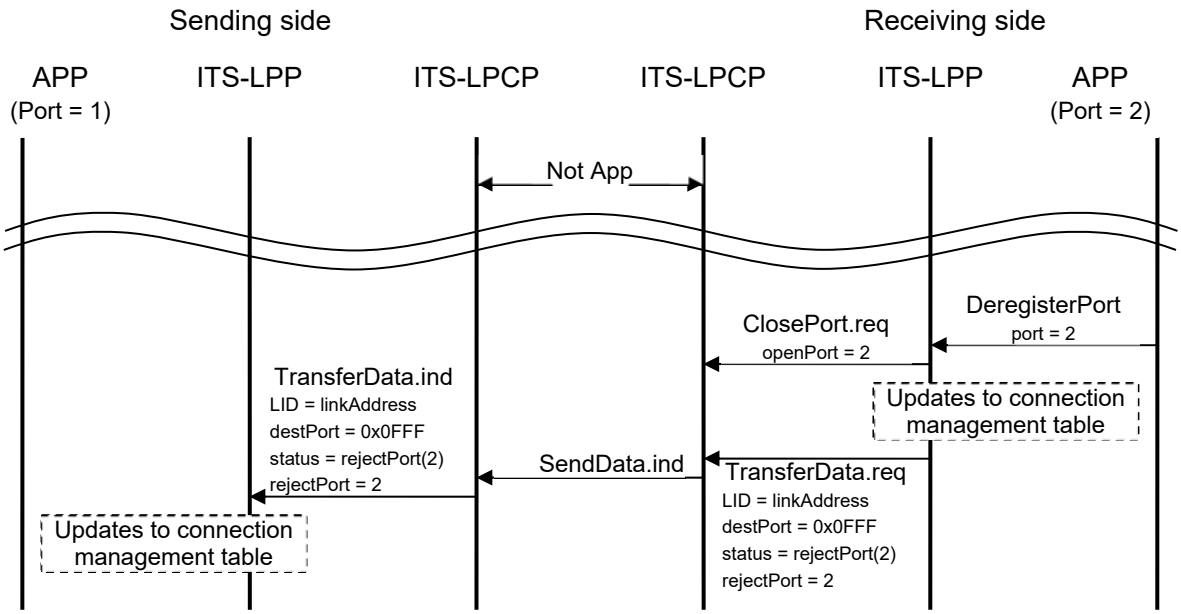


Figure 3.3-27 Example of Accept Port Deletion Procedure

3.3.6 **Extension for Separated Type Configuration in Base Station**

3.3.6.1 **Outline**

The LPPoUDP is an extension protocol allowing the LPP in the external terminal to use the interface of the LPCP in the base station by using the datagram transfer service of UDP.

This extension protocol consists of the transfer service in the base station, the interface-providing entity in the external terminal and the event handling mechanism in the external terminal, and provides the following three types of interfaces which are provided to LPP by the LPCP in the base station to the LPP external terminal:

(1) TransferData.req
(2) TransferData.ind
(3) EventReport.ind

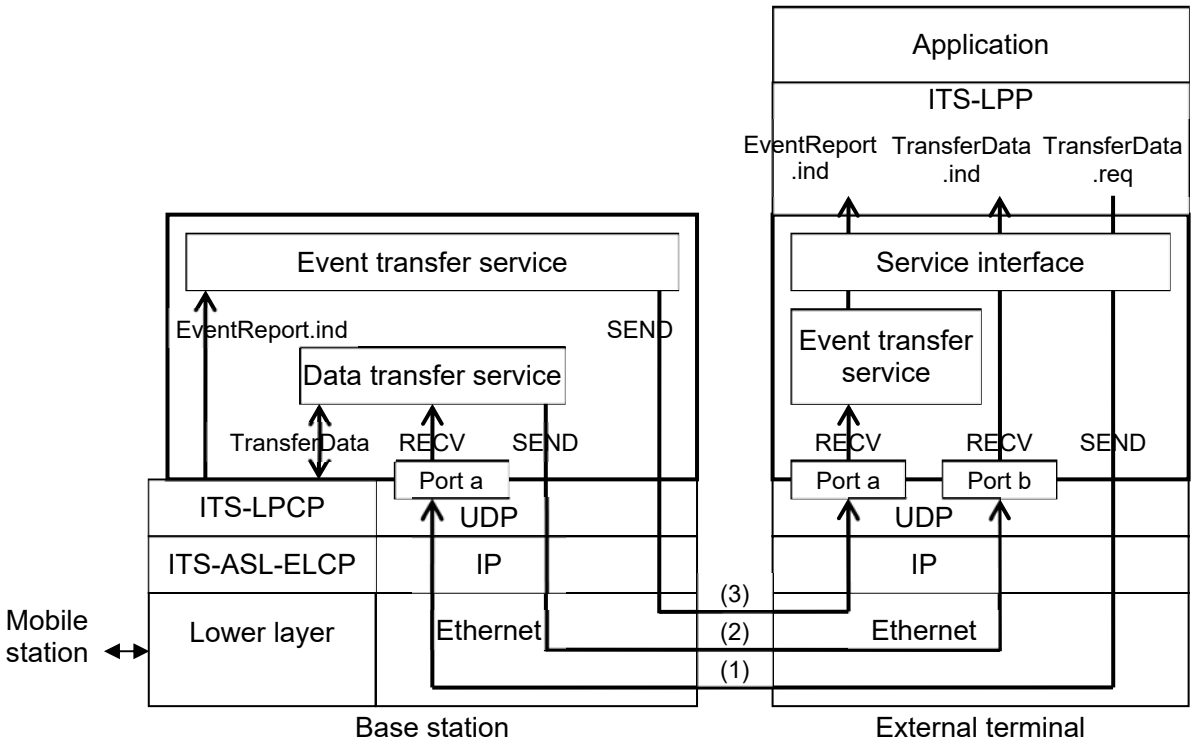


Figure 3.3-28 Outline of LPP over UDP

3.3.6.2 **Definition of UDP and LPCP**

Figure 3.3-29 and Figure 3.3-30 show the protocol data unit of the UDP and LPCP.

Source port number (16)	Destination port number (16)	Length (16)	Check sum (16)	Data area

Figure 3.3-29 UDP Datagram Format

Source port number (16)	Destination port number (16)	Length (8 or 16)	Data area

Figure 3.3-30 PDU Format in LPCP

The notations for primitives of the LPCP and interfaces of the UDP are shown below:

(1) Primitives of the LPCP

TransferData.req (link address, source port number, destination port number, data sent)

TransferData.ind (link address, source port number, destination port number, data received)

EventReport.ind (status identifier, event additional information)

(2) Interfaces of the UDP

SEND (source IP address, destination IP address, source port No, destination port number, data

sent)

RECV (source IP address, source port number, data received)

3.3.6.3 **Details of Mapping Method**

3.3.6.3.1 **Mapping in Data Transfer Function (TransferData.req)**

When the request “TransferData.req” is received from the LPP (implemented in the external terminal), the transfer service in the external terminal sends the request to the receiving port (Wellknown port) of the data transfer service in the base station by the SEND interface of the UDP. The data area in the UDP indicates the source port, destination port, link address and data area in the LPCP are stored in the format shown in Figure 3.3-31, and it is transferred to the ASL-NCP shown in Figure 3.3-32.

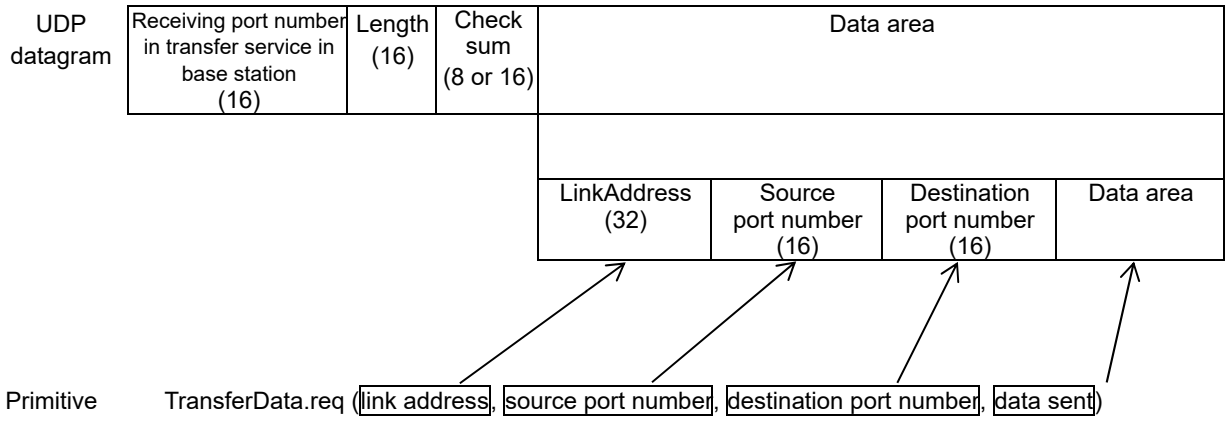


Figure 3.3-31 PDU Mapping in TransferData.req

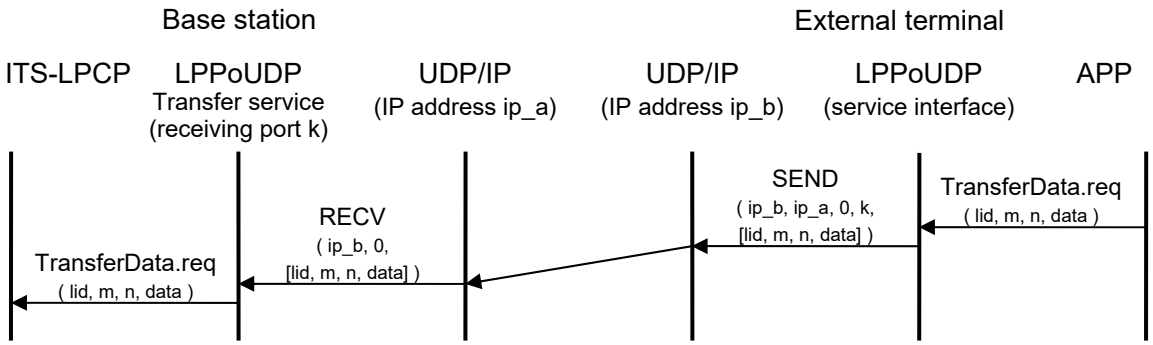


Figure 3.3-32 TransferData.req Sequence

3.3.6.3.2 **Mapping in Data Transfer Function (TransferData.ind)**

When “TransferData.ind” is received, the data transfer service in the base station sends it received message, whose destination source/port is set to the received LPCP source/port destination and to an external terminal by the SEND interface of the UDP source port. The destination port number

in the LPCP is used to determine the sending destination external terminal. Accordingly, if there are two or more external terminals, port numbers in applications is unique in every external terminal. In the data area of the UDP indicates the link address and data area in the LPCP in the format shown in Figure 3.3-33, and it is notified of the LPP in the external terminal is stored shown in Figure 3.3-34.

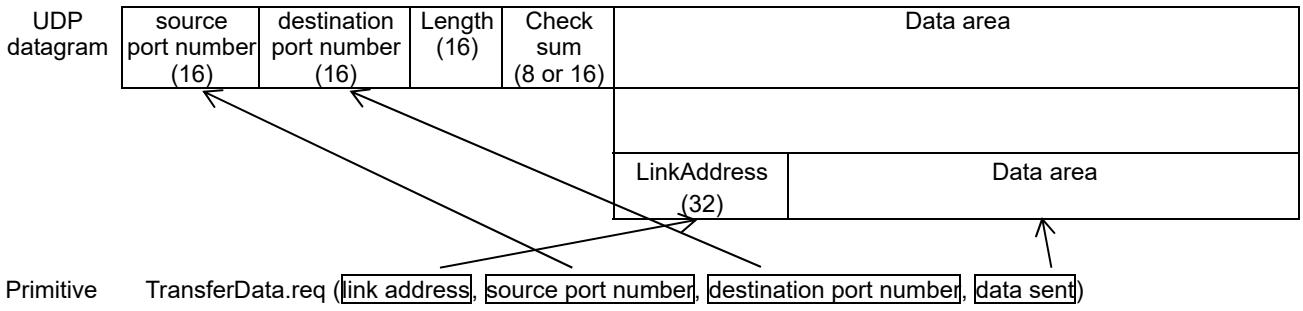


Figure 3.3-33 PDU Mapping in TransferData.ind

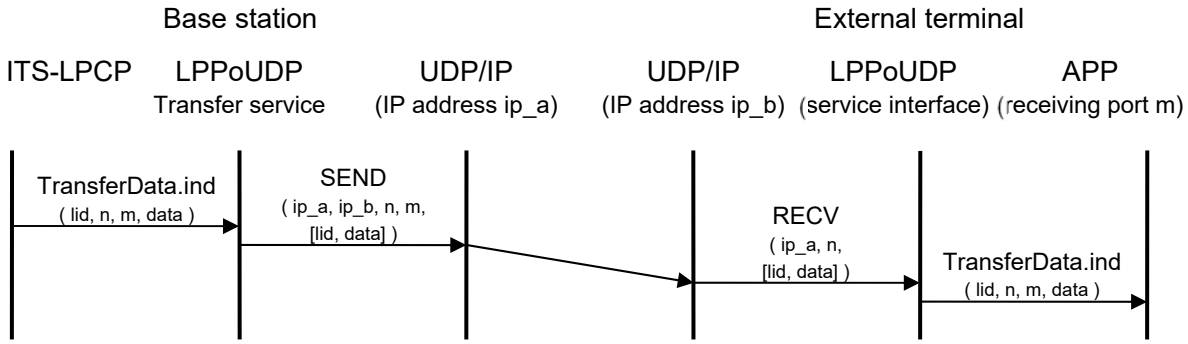


Figure 3.3-34 TransferData.ind Sequence

3.3.6.3.3 **Mapping in Event Notification Function**

The event transfer service works as an application with LPCP on the base station, and as an application with UDP on the external terminal. The information “EventReport” received by the event transfer service of the base station is sent to the event transfer service of the external terminal using the SEND interface of UDP. The receiving port number of the event notification service is 0x0ffe in the external terminal. In the data area of UDP, PDU indicate the status identifier and additional event information are stored in the format shown in Figure 3.3-35, and it is notified of the LPP of the external terminal shown in Figure 3.3-36.

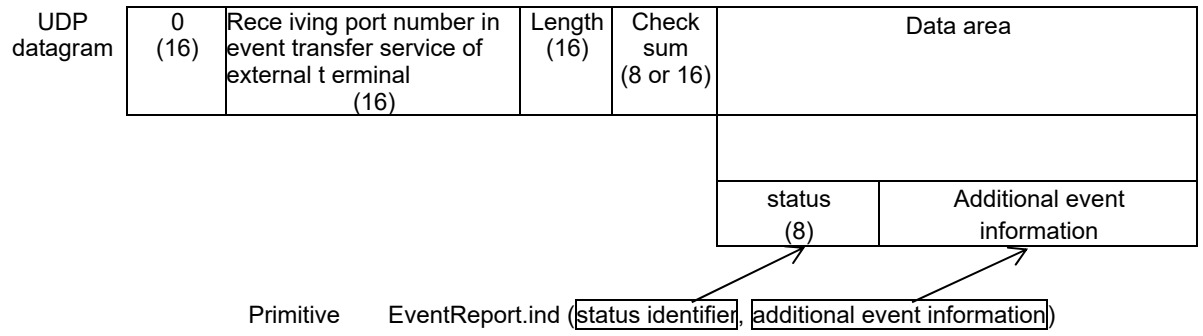


Figure 3.3-35 PDU Mapping in EventReport.ind

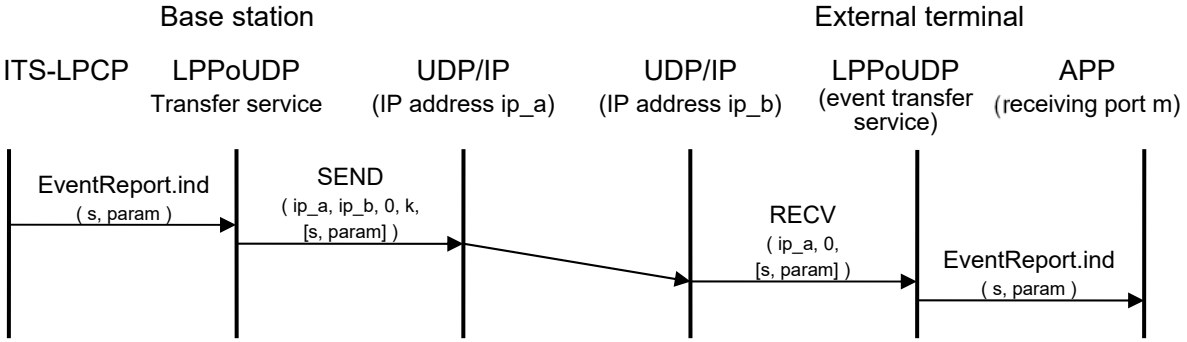


Figure 3.3-36 EventReport.ind Sequence

3.3.7 **Definition of Parameter Types**

```
LppParameter DEFINITIONS ::=
BEGIN

IMPORTS
  PortNo          FROM   LocalControlParameter;

        -- Transfer service processing
        LppTransferDataPDU ::= CHOICE {
                        notUse                          [0]     NULL,
                        invokePdu                       [1]     InvokePDU,
                        resultPdu                       [2]     ResultPDU,
                        acknowledgementPdu              [3]     AcknowledgementPDU,
                        abortPdu                        [4]     AbortPDU,
                        invokeSegmentPdu                [5]     InvokeSegmentPDU,
                        resultSegmentPdu                [6]     ResultSegmentPDU,
                        nackPdu                 [7]     NackPDU
        }

        InvokePDU ::= SEQUENCE {
                        version                 INTEGER(0..3),
                        transactionType         INTEGER(0..1),
                        requireAck              BOOLEAN,
                        retransmitData          BOOLEAN,
                        tid                     INTEGER(0..65535),
                        userData                OCTET STRING
```

```
}

ResultPDU ::= SEQUENCE {
        fill                    BIT STRING(SIZE(3)),
        requireAck              BOOLEAN,
        retransmitData          BOOLEAN,
        tid                     INTEGER(0..65535),
        userData                OCTET STRING
}

AcknowledgementPDU ::= SEQUENCE {
        fill                    BIT STRING(SIZE(4)),
        retransmitData          BOOLEAN,
        tid                     INTEGER(0..65535),
}

AbortPDU ::= SEQUENCE {
        fill                    BIT STRING(SIZE(4)),
        abortType               BOOLEAN,
        tid                     INTEGER(0..65535),
        abortCode               AbortCode
}

InvokeSegmentPDU ::= SEQUENCE {
        version                 INTEGER(0..3),
        transactionType         INTEGER(0..1),
        fin                     BOOLEAN,
        retransmitData          BOOLEAN,
        tid                     INTEGER(0..65535),
        segmentNo               INTEGER(0..65535),
        userData                OCTET STRING
}

ResultSegmentPDU ::= SEQUENCE {
        fill                    BIT STRING(SIZE(3)),
        fin                     BOOLEAN,
        retransmitData          BOOLEAN,
        tid                     INTEGER(0..65535),
        segmentNo               INTEGER(0..65535),
        userData                OCTET STRING
}

NackPDU::= SEQUENCE {
        fill                    BIT STRING(SIZE(4)),
        retransmitData          BOOLEAN,
        tid                     INTEGER(0..65535),
        segmentNumberList       SEQUENCE SIZE(0..65535)
                                        OF INTEGER(0..65535 )
}

AbortCode ::= INTEGER {
        Unknown error                                   (0),
        Protocol error                                  (1),
        TID is invalid.                                 (2),
        Transaction service is not supported.           (3),
        LPP version is different.                       (4),
        Receiving buffer has overflowed.                (5),
```

```
                    MTU error                                   (6),
                    Resend timer timeout                        (7),
                    Result time timeout                         (8),
                    Link Address error                          (9),
                    Destination port error                      (10),
                    LPP is not supported.                       (11),
                    Aborted by DSRC-ASL.                        (12),
                    Transaction could not be started.           (13),
                    Under segmentation/assembly processing      (14)
                    -- In AbortCode, values 15 to 255 are reserved.
} (0..255)

-- Connection management service
LppStatusPDU ::= CHOICE {
        notUse          [0]         NULL,
        acceptPort      [1]         AcceptPort,     -- Accept port
        rejectPort      [2]         RejectPort,     -- Reject port
        dummy3 [3] NULL, dummy4 [4] NULL, dummy5 [5] NULL,
        dummy6 [6] NULL, dummy7 [7] NULL, dummy8 [8] NULL,
        dummy9 [9] NULL, dummy10 [10] NULL, dummy11 [11] NULL,
        dummy12 [12] NULL, dummy13 [13] NULL, dummy14 [14] NULL,
        dummy15 [15] NULL, dummy16 [16] NULL, dummy17 [17] NULL,
        dummy18 [18] NULL, dummy19 [19] NULL, dummy20 [20] NULL,
        dummy21 [21] NULL, dummy22 [22] NULL, dummy23 [23] NULL,
        dummy24 [24] NULL, dummy25 [25] NULL, dummy26 [26] NULL,
        dummy27 [27] NULL, dummy28 [28] NULL, dummy29 [29] NULL,
        dummy30 [30] NULL, dummy31 [31] NULL, dummy32 [32] NULL,
        dummy33 [33] NULL, dummy34 [34] NULL, dummy35 [35] NULL,
        dummy36 [36] NULL, dummy37 [37] NULL, dummy38 [38] NULL,
        dummy39 [39] NULL, dummy40 [40] NULL, dummy41 [41] NULL,
        dummy42 [42] NULL, dummy43 [43] NULL, dummy44 [44] NULL,
        dummy45 [45] NULL, dummy46 [46] NULL, dummy47 [47] NULL,
        dummy48 [48] NULL, dummy49 [49] NULL, dummy50 [50] NULL,
        dummy51 [51] NULL, dummy52 [52] NULL, dummy53 [53] NULL,
        dummy54 [54] NULL, dummy55 [55] NULL, dummy56 [56] NULL,
        dummy57 [57] NULL, dummy58 [58] NULL, dummy59 [59] NULL,
        dummy60 [60] NULL, dummy61 [61] NULL, dummy62 [62] NULL,
        dummy63 [63] NULL, dummy64 [64] NULL, dummy65 [65] NULL,
        dummy66 [66] NULL, dummy67 [67] NULL, dummy68 [68] NULL,
        dummy69 [69] NULL, dummy70 [70] NULL, dummy71 [71] NULL,
        dummy72 [72] NULL, dummy73 [73] NULL, dummy74 [74] NULL,
        dummy75 [75] NULL, dummy76 [76] NULL, dummy77 [77] NULL,
        dummy78 [78] NULL, dummy79 [79] NULL, dummy80 [80] NULL,
        dummy81 [81] NULL, dummy82 [82] NULL, dummy83 [83] NULL,
        dummy84 [84] NULL, dummy85 [85] NULL, dummy86 [86] NULL,
        dummy87 [87] NULL, dummy88 [88] NULL, dummy89 [89] NULL,
        dummy90 [90] NULL, dummy91 [91] NULL, dummy92 [92] NULL,
        dummy93 [93] NULL, dummy94 [94] NULL, dummy95 [95] NULL,
        dummy96 [96] NULL, dummy97 [97] NULL, dummy98 [98] NULL,
        dummy99 [99] NULL, dummy100 [100] NULL, dummy101 [101] NULL,
        dummy102 [102] NULL, dummy103 [103] NULL, dummy104 [104] NULL,
        dummy105 [105] NULL, dummy106 [106] NULL, dummy107 [107] NULL,
        dummy108 [108] NULL, dummy109 [109] NULL, dummy110 [110] NULL,
        dummy111 [111] NULL, dummy112 [112] NULL, dummy113 [113] NULL,
        dummy114 [114] NULL, dummy115 [115] NULL, dummy116 [116] NULL,
        dummy117 [117] NULL, dummy118 [118] NULL, dummy119 [119] NULL,
        dummy120 [120] NULL, dummy121 [121] NULL, dummy122 [122] NULL,
```

```
                dummy123 [123] NULL, dummy124 [124] NULL, dummy125 [125] NULL,
                dummy126 [126] NULL, dummy127 [127] NULL, dummy128 [128] NULL,
                dummy129 [129] NULL, dummy130 [130] NULL, dummy131 [131] NULL,
                dummy132 [132] NULL, dummy133 [133] NULL, dummy134 [134] NULL,
                dummy135 [135] NULL, dummy136 [136] NULL, dummy137 [137] NULL,
                dummy138 [138] NULL, dummy139 [139] NULL, dummy140 [140] NULL,
                dummy141 [141] NULL, dummy142 [142] NULL, dummy143 [143] NULL,
                dummy144 [144] NULL, dummy145 [145] NULL, dummy146 [146] NULL,
                dummy147 [147] NULL, dummy148 [148] NULL, dummy149 [149] NULL,
                dummy150 [150] NULL, dummy151 [151] NULL, dummy152 [152] NULL,
                dummy153 [153] NULL, dummy154 [154] NULL, dummy155 [155] NULL,
                dummy156 [156] NULL, dummy157 [157] NULL, dummy158 [158] NULL,
                dummy159 [159] NULL, dummy160 [160] NULL, dummy161 [161] NULL,
                dummy162 [162] NULL, dummy163 [163] NULL, dummy164 [164] NULL,
                dummy165 [165] NULL, dummy166 [166] NULL, dummy167 [167] NULL,
                dummy168 [168] NULL, dummy169 [169] NULL, dummy170 [170] NULL,
                dummy171 [171] NULL, dummy172 [172] NULL, dummy173 [173] NULL,
                dummy174 [174] NULL, dummy175 [175] NULL, dummy176 [176] NULL,
                dummy177 [177] NULL, dummy178 [178] NULL, dummy179 [179] NULL,
                dummy180 [180] NULL, dummy181 [181] NULL, dummy182 [182] NULL,
                dummy183 [183] NULL, dummy184 [184] NULL, dummy185 [185] NULL,
                dummy186 [186] NULL, dummy187 [187] NULL, dummy188 [188] NULL,
                dummy189 [189] NULL, dummy190 [190] NULL, dummy191 [191] NULL,
                dummy192 [192] NULL, dummy193 [193] NULL, dummy194 [194] NULL,
                dummy195 [195] NULL, dummy196 [196] NULL, dummy197 [197] NULL,
                dummy198 [198] NULL, dummy199 [199] NULL, dummy200 [200] NULL,
                dummy201 [201] NULL, dummy202 [202] NULL, dummy203 [203] NULL,
                dummy204 [204] NULL, dummy205 [205] NULL, dummy206 [206] NULL,
                dummy207 [207] NULL, dummy208 [208] NULL, dummy209 [209] NULL,
                dummy210 [210] NULL, dummy211 [211] NULL, dummy212 [212] NULL,
                dummy213 [213] NULL, dummy214 [214] NULL, dummy215 [215] NULL,
                dummy216 [216] NULL, dummy217 [217] NULL, dummy218 [218] NULL,
                dummy219 [219] NULL, dummy220 [220] NULL, dummy221 [221] NULL,
                dummy222 [222] NULL, dummy223 [223] NULL, dummy224 [224] NULL,
                dummy225 [225] NULL, dummy226 [226] NULL, dummy227 [227] NULL,
                dummy228 [228] NULL, dummy229 [229] NULL, dummy230 [230] NULL,
                dummy231 [231] NULL, dummy232 [232] NULL, dummy233 [233] NULL,
                dummy234 [234] NULL, dummy235 [235] NULL, dummy236 [236] NULL,
                dummy237 [237] NULL, dummy238 [238] NULL, dummy239 [239] NULL,
                dummy240 [240] NULL, dummy241 [241] NULL, dummy242 [242] NULL,
                dummy243 [243] NULL, dummy244 [244] NULL, dummy245 [245] NULL,
                dummy246 [246] NULL, dummy247 [247] NULL, dummy248 [248] NULL,
                dummy249 [249] NULL, dummy250 [250] NULL, dummy251 [251] NULL,
                dummy252 [252] NULL, dummy253 [253] NULL, dummy254 [254] NULL,
                dummy255 [255] NULL
            -- In PDU LppStatus, identifier and type definition of tag numbers 3 to 255 are
            -- reserved.
        }

        AcceptPort ::= PortNo

        RejectPort ::= PortNo

END
```

Chapter 4 Compatibility Confirmation Test

4.1 **Objective of the Test**

The testing based on the testing system is intended to confirm that mobile stations (OBEs) manufactured by a mobile station manufacturer meet the functional requirements for mobile stations in accordance with the standard specification for ITS-ASL specified in this technical report. It is assumed that for these tests, ITS FORUM RC-005, IEEE 802.11, ARIB STD-T104, ARIB STD-T109, or ARIB STD-T120 are used as the communication lower layer.
Hereafter in this document, a mobile station will be expressed as OBE (On Board Equipment) and a testing system as TS (Test System).

4.2 **Test Items and Test Conditions**

4.2.1 **Test Items**

Test items are described as to the operation test and performance test for each function.

As for the performance test, however, only a method of measurement to obtain criteria for numerical value evaluations should be described and conformance of numerical values obtained in this method of measurement is not defined.

In a case where there are separate requirements requested by the service, performance should be evaluated in accordance with this method of measurement.

4.2.1.1 **Test Items for Extended Link Control Protocol (ITS-ASL-ELCP)**

The test items for ITS-ASL-ELCP by TS are as shown below. When test items differ depending on the lower layer, "-a" is attached to test numbers for ITS FORUM RC-005, IEEE 802.11, ARIB STD-T109, and ARIB STD-T120 (PC5), and "-b" is attached to test numbers for ARIB STD-T104 and ARIB STD-T120 (Uu). Furthermore, “-c” is attached for standards other than ARIB STD-T109, and “-d” is attached for ARIB STD-T109. These relationships are shown in Table 4.2-1.

Test number	Test item
1-1	Operation test for communication control management
1-1-1-a	Initial connection
1-1-1-b	Initial connection
1-1-2-a	Maintaining connection
1-1-2-b	Maintaining connection
1-1-3-a	Disconnection
1-1-3-b	Disconnection
1-1-4-a	Event notification service
1-1-4-b	Event notification service
1-2	Extended link control operation test
1-2-1-a	Transmission service
1-2-1-b	Transmission service
1-2-2-c	Bulk transmission control
1-2-2-d	Bulk transmission control
1-2-3	Broadcast mode control
1-2-4	Conbination of bulk transmission control and broadcast mode control
1-3	Performance test
1-3-1-a	Measurement of initial connection time

Table 4.2-1 Test numbers and corresponding standards

Test number	ITS FORUM RC-005	IEEE 802.11	ARIB STD-T109	ARIB STD-T104	ARIB STD-T120(Uu)	ARIB STD-T120(PC5)
1-1-1-a	○	○	○			○
1-1-1-b				○	○	
1-1-2-a	○	○	○			○
1-1-2-b				○	○	
1-1-3-a	○	○	○			○
1-1-3-b				○	○	
1-1-4-a	○	○	○			○

1-1-4-b				○	○	
1-2-1-a	○	○	○			○
1-2-1-b				○	○	
1-2-2-c	○	○		○	○	○
1-2-2-d			○			
1-2-3	○	○	○	○	○	○
1-2-4	○	○	○	○	○	○
1-3-1-a	○	○	○			○

4.2.1.2 **Test Items for Local Port Control Protocol (ITS-LPCP)**

The test items for ITS-LPCP by TS are as shown below. For information, this test is based on the existence of test application for operation test.

Test number	Test item
2-1	ITS-LPCP operation test
2-1-1	Initial connection
2-1-2	Disconnection
2-1-3	Management service
2-1-4	Data transfer service
2-1-5	Data transfer service in broadcast mode
2-2	ITS-LPCP performance test
2-2-1	Measurement of initial connection time
2-2-2	Transmission performance

4.2.1.3 **Test items for Local Port Protocol (ITS-LPP)**

The test items for ITS-LPP by TS are as shown below. For information, this test is based on the existence of test application for operation test.

Test number	Test item
3-1	ITS-LPP operation test
3-1-1	Initial set up operation
3-1-2	Disconnection operation
3-1-3	Unilateral data-send transaction service
3-1-4	Unilateral data-send transaction service in broadcast mode
3-1-5	Request-response type transaction service
3-1-6	Data retransmission operation
3-1-7	Segmentation/assembly operation

4.2.2 **Test Parameters**

The parameters are set or registered when performing testing are as follows. These parameters are applied unless otherwise specified.

The meanings of <TS>, <OBE>, and <Common> below are as indicated.

<TS>: This is a parameter that is set or registered by TS.

<OBE>: This is a parameter that is set or registered by OBE.

<Common>: This is a parameter that is set or registered by both TS and OBE.

4.2.2.1 **ITS FORUM RC-005, IEEE 1609.3, and ARIB STD-T120 (PC5) Related Parameters**

・PSID：any value from 0x28 to 0x7E <Common>

4.2.2.2 **ITS Application Sub Layer (ITS-ASL) Related Parameters**

・Version information：0 <Common>

・Connection management timer：to be set by TS <TS>

・Link address：to be set by OBE <OBE>

4.2.3 **Details of Test**

4.2.3.1 **Extended Link Control Protocol (ITS-ASL-ELCP) Test**

Details of the test as to ITS-ASL-ELCP are as shown below.

4.2.3.1.1 **Operation Test for Communication Control Management**

Test number	1-1-1-a	Items	Communication control management operation Initial connection
Test overview ・ Verify that TS and OBE enter a communication-connected state.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Lower layer is available.			
Test procedure 1. Put TS into a transmission state from the transmission-stopped state. 2. In TS, verify the details of the ASL mobile station profile.			
Confirmation items ・ In TS, verify that the details of the ASL mobile station profile received by TS are identical with the details of the declared ASL mobile station profile.			

Test number	1-1-1-b	Items	Communication control management operation Initial connection
Test overview ・ Verify that TS and OBE enter a communication-connected state.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Lower layer is available.			
Test procedure 1. Add "Connection standby" service provision area information to the TS area management table and place the OBE in the service provision area. 2. In TS, confirm the details of the mobile station profile registered in the link management table.			
Confirmation items ・ In TS, verify that the link address linked to the service provision area in the TS link management table is newly registered.			

Test number	1-1-2-a	Items	Communication control management operation Maintaining connection
Test overview ・ Verify that the communication-connected state with TS and OBE can be maintained.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Verification has been performed for the operation test in the test number 1-1-1			
Test procedure 1. TS transmits the connection keep request message repeatedly at an interval of within a CTO value of T1max milli-second.			
Confirmation items ・ In TS, verify that OBE responds to the transmission of the connection keep request message from TS within a CTR value of T2max milli-second.			

Test number	1-1-2-b	Items	Communication control management operation Maintaining connection
Test overview ・ Verify that the communication-connected state with TS and OBE can be maintained.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Verification has been performed for the operation test in the test number 1-1-1.			
Test procedure 1. With numbered link addresses linked to a service provision area in TS and OBE, maintain the OBE within the service provision area.			
Confirmation items ・ In TS, verify that the OBE regularly transmits service request messages and that the TS sends a service response message indicating "Maintaining connection."			

Test number	1-1-3-a	Items	Communication control management operation Disconnection
Test overview ・ Verify that TS and OBE enter a communication-disconnected state.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Verification has been performed for the operation test in the test numbers 1-1-1 and 1-1-2.			
Test procedure 1. Put TS into a transmission-stopped state from the transmission state. 2. After an elapse of a CTO value of T1max milli-second in OBE, put TS into a transmission state from the transmission-stopped state.			
Confirmation items ・ In the test procedure step 2, put TS into a transmission-stopped state. Then, verify in TS that after an elapse of a CTO value of T1max milli-second in OBE plus the value in the release timer, OBE goes into the initial connection operation starting from the transmission of connection keep request message.			

Test number	1-1-3-b	Items	Communication control management operation Disconnection
Test overview ・ Verify that TS and OBE enter a communication-disconnected state.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Verification has been performed for the operation tests in test numbers 1-1-1 and 1-1-2.			
Test procedure 1. With numbered link addresses linked to a service provision area in TS and OBE, move the OBE outside the service provision area.			
Confirmation items ・ In TS, verify that the OBE regularly transmits service request messages and that the TS sends a service response message indicating "Disconnection."			

Test number	1-1-4-1-a	Items	Communication control management operation Event notification service 1
Test overview ・ Verify the operation procedure of the event notification service provided by Communication control management. ・ TS issues an abnormality in which the status identifier status corresponds to 1 (no access point exists) to OBE. Then, verify the occurrence of an event, and afterwards, verify that OBE does not enter abnormality.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established.			
Test procedure 1. TS produces NCP-PDU, which stores an access point identifier not installed in OBE, and transmits it to OBE. 2. TS receives the event notification message from OBE. 3. TS transmits the connection keep request message and maintain the communication-connected state.			
Confirmation items ・ Verify in TS that the value of the status identifier status of the event notification message received in the test procedure step 2 is 1. ・ Verify in TS that in response to the transmission of the connection keep request message from TS, OBC responds within a CTR value of T2max milli-second in the base station.			

Test number	1-1-4-1-b	Items	Communication control management operation Event notification service 1
Test overview ・ Verify the operation procedure of the event notification service provided by Communication control management. ・ TS issues an abnormality in which the status identifier status corresponds to 1 (no access point exists) to OBE. Verify the occurrence of an event, and afterwards, verify that OBE does not enter an abnormal state.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established.			
Test procedure 1. TS produces an NCP-PDU, which stores an access point identifier not installed in OBE, and transmits it to OBE. 2. TS receives the event notification message from OBE.			
Confirmation items ・ Verify in TS that the value of the status identifier status of the event notification message received in the test procedure step 2 is 1.			

Test number	1-1-4-2-a	Items	Communication control management operation Event notification service 2
Test overview ・ Verify the operation procedure of the event notification service provided by Communication control management. ・ TS causes an event in which the status identifier status corresponds to 1 (no access point exists). Then, verify that OBE receives the event, and afterwards, OBE does not go into an abnormality.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established. ・ Test data to be used for the operation test does not specifically indicated.			
Test procedure 1. OBE transmits data to TS through ITS-LPCP. 2. TS confirms the data reception. Then, TS transmits an event notification message in which the status identifier status is 1. 3. TS transmits the connection keep request message and maintains the communication-connected state.			
Confirmation items ・ Verify in TS that in response to the transmission of the connection keep request message from TS, OBE responds within a CTR value of T2max milli-second in the base station. ・ Verify in OBE that the value of the status identifier status of the event notification message received in the test procedure step 2 is 1.			

Test number	1-1-4-2-b	Items	Communication control management operation Event notification service 2
Test overview ・ Verify the operation procedure of the event notification service provided by Communication control management. ・ TS causes an event in which the status identifier status corresponds to 1 (no access point exists). Verify that OBE receives the event, and afterwards, verify that OBE does not enter an abnormal state.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established.			
Test procedure 1. OBE transmits data to TS through ITS-LPCP. 2. TS confirms data reception and then transmits an event notification message in which the status identifier status is 1.			
Confirmation items ・ Verify in OBE that the value of the status identifier status of the event notification message received in the test procedure step 2 is 1.			

4.2.3.1.2 **Operation Test for Extended Link Control**

Test number	1-2-1-1-a	Items	Extended link control operation Transmission service 1 (TS to OBE)
Test overview ・ Verify the operation procedure for data transfer (TS to OBE).			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established. ・ The test data to be used as NCP-SDU in the operation test is the following data: Test data 1: [32] octet arbitrary data row			
Test procedure 1. OBE declares ITS-LPCP as ASL-NCP to which TS transmits data. 2. TS transmits the connection keep request message repeatedly at an interval of within a CTO value of T1max milli-second in OBE. 3. TS transmits test data 1 to ITS-LPCP by using the data transmission primitive.			
Confirmation items ・ OBE verifies in ITS-LPCP declared that the test data 1 transmitted by TS to the data transmission primitive is stored in NCP-SDU.			

Test number	1-2-1-1-b	Items	Extended link control operation Transmission service 1 (TS to OBE)
Test overview ・ Verify the operation procedure for data transfer (TS to OBE).			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established. ・ The test data to be used as NCP-SDU in the operation test is the following data: Test data 1: [32] octet arbitrary data row			
Test procedure 1. OBE declares ITS-LPCP as ASL-NCP to which TS transmits data. 2. TS transmits test data 1 to ITS-LPCP by using the data transmission primitive.			
Confirmation items ・ OBE verifies in ITS-LPCP that the test data 1 transmitted by TS to the data transmission primitive is stored in NCP-SDU.			

Test number	1-2-1-2-a	Items	Extended link control operation Transmission service 2 (OBE to TS)
Test overview ・ Verify the operation procedure for data transfer (OBE to TS).			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established. ・ The test data to be used as NCP-SDU in the operation test is the following data: Test data 1: [32] octet arbitrary data row			
Test procedure 1. OBE declares ITS-LPCP as ASL-NCP to which TS transmits data. 2. TS transmits the connection keep request message repeatedly at an interval of within a CTO value of T1max milli-second in OBE. 3. OBE transmits test data 1 to TS ITS-LPCP by using the data transmission primitive.			
Confirmation items ・ TS verifies in ITS-LPCP declared that the test data 1 transmitted by OBE to the data transmission primitive is stored in NCP-SDU.			

Test number	1-2-1-2-b	Items	Extended link control operation Transmission service 2 (OBE to TS)
Test overview ・ Verify the operation procedure for data transfer (OBE to TS).			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established. ・ The test data to be used as NCP-SDU in the operation test is the following data: Test data 1: [32] octet arbitrary data row			
Test procedure 1. OBE declares ITS-LPCP as ASL-NCP to which OBE transmits data. 2. OBE transmits test data 1 to the TS ITS-LPCP by using the data transmission primitive.			
Confirmation items ・ TS verifies in ITS-LPCP that the test data 1 transmitted by OBE to the data transmission primitive is stored in NCP-SDU.			

Test number	1-2-2-1-c	Items	Extended link control operation Bulk transmission control 1 (TS to OBE)
Test overview ・ Verify the operation procedure of the bulk transmission control (TS to OBE).			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Verification has been performed for the operation test in the test number 1-2-1-1. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established. ・ The test data to be used as NCP-SDU in the operation test is the following data: * The size of the test data should include size information (1 to 2 octets) and error check code (4 octets). Test data 1: [SUU - 5] octet arbitrary data row Test data 2: [SUU - 4] octet arbitrary data row Test data 3: [MRU - 5 of ITS-ASL-ELCP on the OBE side] octet arbitrary data row			
Test procedure 1. OBE declares ITS-LPCP as ASL-NCP to which TS transmits data. 2. TS transmits test data 1 to ITS-LPCP by using the data transmission primitive. 3. TS transmits test data 2 to ITS-LPCP by using the data transmission primitive. 4. TS transmits test data 3 to ITS-LPCP by using the data transmission primitive.			
Confirmation items ・ Verify in OBE that it was possible to receive test data 1. ・ Verify in OBE that it was possible to receive test data 2. ・ Verify in OBE that it was possible to receive test data 3.			

Test number	1-2-2-2-c	Items	Extended link control operation Bulk transmission control 2 (OBE to TS)
Test overview ・ Verify the operation procedure of the bulk transmission control (OBE to TS).			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Verification has been performed for the operation test in the test number 1-2-1-2. ・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established. ・ The test data to be used as NCP-SDU in the operation test is the following data: * The size of the test data should include size information (1 to 2 octets) and error check code (4 octets). Test data 1: [SUU - 5] octet arbitrary data row Test data 2: [SUU - 4] octet arbitrary data row Test data 3: [MRU - 5 of ITS-ASL-ELCP on the TS side] octet arbitrary data row			
Test procedure 1. OBE declares ITS-LPCP as ASL-NCP to which OBE transmits data. 2. TS transmits test data 1 to ITS-LPCP by using the data transmission primitive. 3. TS transmits test data 2 to ITS-LPCP by using the data transmission primitive. 4. TS transmits test data 3 to ITS-LPCP by using the data transmission primitive.			
Confirmation items ・ Verify in OBE that it was possible to receive test data 1. ・ Verify in OBE that it was possible to receive test data 2. ・ Verify in OBE that it was possible to receive test data 3.			

Test number	1-2-2-1-d	Items	Extended link control operation Bulk transmission control 1 (TS to OBE)

Test overview

・ Verify the operation procedure of the bulk transmission control (TS to OBE).

Test conditions

・ The basic parameters are those specified in subclause 4.2.2

・ Verification has been performed for the operation test in the test number 1-2-1-1.

・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established.

・ The test data to be used as NCP-SDU in the operation test is the following data:

* The size of the test data should include size information (1 to 2 octets) and error check code (4 octets).

Test data 1: [SUU - 5] octet arbitrary data row

Test data 2: [SUU - 4] octet arbitrary data row

Test data 3: [MRU - 5 of ITS-ASL-ELCP on the OBE side] octet arbitrary data row

Test data 4: [SUU - 5] octet arbitrary data row

Test data 5: [SUU - 4] octet arbitrary data row

Test data 6: [MRU - 5 of ITS-ASL-ELCP on the OBE side] octet arbitrary data row

Test procedure

1. OBE declares ITS-LPCP as ASL-NCP to which TS transmits data
2. TS sets the OBE link address as the communication control information destination link address and transmits test data 1 to ITS-LPCP by using the data transmission primitive.
3. TS sets the OBE link address as the communication control information destination link address and transmits test data 2 to ITS-LPCP by using the data transmission primitive.
4. TS sets the OBE link address as the communication control information destination link address and transmits test data 3 to ITS-LPCP by using the data transmission primitive.
5. TS sets a link address other than the OBE link address as the communication control information destination link address and transmits test data 4 to ITS-LPCP by using the data transmission primitive.
6. TS sets a link address other than the OBE link address as the communication control information destination link address and transmits test data 5 to ITS-LPCP by using the data transmission primitive.
7. TS sets a link address other than the OBE link address as the communication control information destination link address and transmits test data 6 to ITS-LPCP by using the data transmission primitive.

・ Confirmation itemsVerify in OBE that it was possible to receive test data 1.

・ Verify in OBE that it was possible to receive test data 2.

・ Verify in OBE that it was possible to receive test data 3.

・ Verify in OBE that it was not possible to receive test data 4.

・ Verify in OBE that it was not possible to receive test data 5.

・ Verify in OBE that it was not possible to receive test data 6.

Test number	1-2-2-2-d	Items	Extended link control operation Bulk transmission control 2 (OBE to TS)

Test overview

・ Verify the operation procedure of the bulk transmission control (OBE to TS).

Test conditions

・ The basic parameters are those specified in subclause 4.2.2.

・ Verification has been performed for the operation test in test number 1-2-1-2.

・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established.

・ The test data to be used as NCP-SDU in the operation test is the following data:

* The size of the test data should include size information (1 to 2 octets) and error check code (4 octets).

Test data 1: [SUU - 5] octet arbitrary row data
Test data 2: [SUU - 4] octet arbitrary row data
Test data 3: [MRU - 5 of ITS-ASL-ELCP on the TS side] octet arbitrary data row
Test data 4: [SUU - 5] octet arbitrary row data
Test data 5: [SUU - 4] octet arbitrary row data
Test data 6: [MRU - 5 of ITS-ASL-ELCP on the TS side] octet arbitrary data row

Test procedure

1. OBE declares ITS-LPCP as ASL-NCP to which OBE transmits data.
2. OBE sets the OBE link address as the communication control information destination link address and transmits test data 1 to the TS ITS-LPCP by using the data transmission primitive.
3. OBE sets the OBE link address as the communication control information destination link address and transmits test data 2 to the TS ITS-LPCP by using the data transmission primitive.
4. OBE sets the OBE link address as the communication control information destination link address and transmits test data 3 to the TS ITS-LPCP by using the data transmission primitive.
5. OBE sets a link address other than the OBE link address as the communication control information destination link address and transmits test data 4 to the TS ITS-LPCP by using the data transmission primitive.
6. OBE sets a link address other than the OBE link address as the communication control information destination link address and transmits test data 5 to the TS ITS-LPCP by using the data transmission primitive.
7. OBE sets a link address other than the OBE link address as the communication control information destination link address and transmits test data 6 to the TS ITS-LPCP by using the data transmission primitive.

Confirmation items

・ Verify in OBE that it was possible to receive test data 1.

・ Verify in OBE that it was possible to receive test data 2.

・ Verify in OBE that it was possible to receive test data 3.

・ Verify in OBE that it was not possible to receive test data 4.

- Verify in OBE that it was not possible to receive test data 5.
- Verify in OBE that it was not possible to receive test data 6.

Test number	1-2-3-1	Items	Extended link control operation Broadcast mode control 1 (TS to OBE)
Test overview ・ Verify the operation procedure of the broadcast mode control (TS to OBE).			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ TS use the bulk transmission function. ・ The number of consecutive transmissions, k, is [3]. ・ The test data to be used as NCP-SDU in the operation test are the following data: * The size of the test data should include size information (1 to 2 octets) and error check code (4 octets). Test data 1: [SUM - 5] octet arbitrary data row Test data 2: [SUM - 4] octet arbitrary data row Test data 3: [MRU - 5 of ITS-ASL-ELCP on the OBE side] octet arbitrary data row			
Test procedure 1. OBE declares ITS-LPCP as ASL-NCP to which TS transmits broadcast data. 2. TS transmits test data 1 to ITS-LPCP on OBE at the group broadcast link address by using the data transmission primitive 3. TS transmits test data 2 to ITS-LPCP on OBE at the group broadcast link address by using the data transmission primitive 4. TS transmits test data 3 to ITS-LPCP on OBE at the group broadcast link address by using the data transmission primitive			
Confirmation items ・ Verify in OBE that it was possible to receive test data 1. ・ Verify in OBE that it was possible to receive test data 2. ・ Verify in OBE that it was possible to receive test data 3.			

Test number	1-2-3-2	Items	Extended link control operation Broadcast mode control 2 (OBE to TS)

Test overview

・ Verify the operation procedure of the broadcast mode control (OBE to TS).

Test conditions

・ The basic parameters are those specified in subclause 4.2.2.

・ OBE use the bulk transmission function.

・ The number of consecutive transmissions, k, is [3].

・ The test data to be used as NCP-SDU in the operation test are the following data:

* The size of the test data should include size information (1 to 2 octets) and error check code (4 octets).

Test data 1: [SUM - 5] octet arbitrary data row

Test data 2: [SUM - 4] octet arbitrary data row

Test data 3: [MRU - 5 of ITS-ASL-ELCP on the OBE side] octet arbitrary data row

Test procedure

1. OBE declares ITS-LPCP as ASL-NCP to which OBE transmits broadcast data.
2. OBE transmits test data 1 to ITS-LPCP on TS at the group broadcast link address by using the data transmission primitive
3. OBE transmits test data 2 to ITS-LPCP on TS at the group broadcast link address by using the data transmission primitive
4. OBE transmits test data 3 to ITS-LPCP on TS at the group broadcast link address by using the data transmission primitive

Confirmation items

・ Verify in OBE that it was possible to receive test data 1.

・ Verify in OBE that it was possible to receive test data 2.

・ Verify in OBE that it was possible to receive test data 3.

Test number	1-2-4-1	Items	Extended link control operation Combination of bulk transmission control and broadcast mode control 1 (TS to OBE)

Test overview

- Verify the operation sequential procedure of the unicast and broadcast (TS to OBE).

Test conditions

- The basic parameters are those specified in subclause 4.2.2.
- TS use the bulk transmission function.
- The number of consecutive transmissions, k, is [3].
- Initial connection processing between TS and OBE has been completed and the communication-connected state has been established.
- The test data to be used as NCP-SDU in the operation test is the following data:
  * The size of the test data should include size information (1 to 2 octets) and error check code (4 octets).
  - Test data 1: [SUM - 5] octet arbitrary data row
  - Test data 2: [SUM - 4] octet arbitrary data row
  - Test data 3: [MRU - 5 of ITS-ASL-ELCP on the OBE side] octet arbitrary data row
  - Test data 4: [SUU - 5] octet arbitrary data row
  - Test data 5: [SUU - 4] octet arbitrary data row
  - Test data 6: [MRU - 5 of ITS-ASL-ELCP on the OBE side] octet arbitrary data row

Test procedure

1. OBE declares ITS-LPCP as ASL-NCP to which TS transmits broadcast data.
2. TS transmits data in the order of the test data 1, test data 2 and test data 3 repeatedly to ITS-LPCP on OBE by using the data transmission primitive. The link address in this case is the group broadcast link address.
3. TS transmits data in the order of test data 1, test data 2 and test data 3 repeatedly to ITS-LPCP on OBE by using the data transmission primitive. The link address in this case is the private link address of OBE.

Confirmation items

- Verify in OBE that it was possible to receive test data 1.
- Verify in OBE that it was possible to receive test data 2.
- Verify in OBE that it was possible to receive test data 3.
- Verify in OBE that it was possible to receive test data 4.
- Verify in OBE that it was possible to receive test data 5.
- Verify in OBE that it was possible to receive test data 6.

Test number	1-2-4-2	Items	Extended link control operation Combination of bulk transmission control and broadcast mode control 2 (OBE to TS)

Test overview

・ Verify the operation sequential procedure of the unicast and broadcast (OBE to TS).

Test conditions

・ The basic parameters are those specified in subclause 4.2.2.

・ OBE uses the bulk transmission function.

・ The number of consecutive transmissions, k, is [3].

・ Initial connection processing between TS and OBE has been completed and the communication-connected state has been established.

・ The test data to be used as NCP-SDU in the operation test is the following data:

* The size of the test data should include size information (1 to 2 octets) and error check code (4 octets).

Test data 1: [SUM - 5] octet arbitrary data row

Test data 2: [SUM - 4] octet arbitrary data row

Test data 3: [MRU - 5 of ITS-ASL-ELCP on the TS side] octet arbitrary data row

Test data 4: [SUU - 5] octet arbitrary data row

Test data 5: [SUU - 4] octet arbitrary data row

Test data 6: [MRU - 5 of ITS-ASL-ELCP on the TS side] octet arbitrary data row

Test procedure

1. OBE declares ITS-LPCP as ASL-NCP to which OBE transmits broadcast data.
2. OBE transmits data in the order of test data 1, test data 2 and test data 3 repeatedly to ITS-LPCP on TS by using the data transmission primitive. The link address in this case is the group broadcast link address.
3. OBE transmits data in the order of test data 1, test data 2 and test data 3 repeatedly to ITS-LPCP on TS by using the data transmission primitive. The link address in this case is the private link address of OBE.

Confirmation items

・ Verify in OBE that it was possible to receive test data 1.

・ Verify in OBE that it was possible to receive test data 2.

・ Verify in OBE that it was possible to receive test data 3.

・ Verify in OBE that it was possible to receive test data 4.

・ Verify in OBE that it was possible to receive test data 5.

・ Verify in OBE that it was possible to receive test data 6.

4.2.3.1.3 **Performance Test**

Test number	1-3-1-a	Items	Extended link control operation Measurement of initial connection time
Test overview ・ Measure the time from the moment OBE gets into the communication zone to the moment it is possible to communicate as seen from ITS-LPCP.			
Test conditions ・ The basic parameters are those specified in subclause 4.2.2. ・ Verification has been done as to the operation test in the test number 1-1-1.			
Test procedure 1. Turn TS into a transmission state from the transmission-stopped state. Then, in TS, measure the time, t1, when TS is turned into the transmission state. 2. In TS, measure the time, t2, when ITS-ASL-ELCP generates the event notification primitive (communication connection notification) to ITS-LPCP.			
Confirmation items ・ The time, ti, determined from the following equation should be considered as the initial connection time from the channel selection to ITS-LPCP activation. Initial connection time ti = t2 - t1 ・ The number of samples should be [10] and based on the average value obtained from them, evaluations should be performed.			

4.2.3.2 **Local Port Control Protocol (ITS-LPCP) Test**

Test method and procedures for the ITS-LPCP are described below.

The tests described here are specified on condition that confirmation has been completed for the operation tests required on the corresponding OBE in the test items shown 4.2.1.1.

4.2.3.2.1 **Operation Test**

Test number	2-1-1	Items	ITS-LPCP operation Initial connection
Test overview ・ Verify that initial set up process between the TS (test system) and the OBE is completed.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ For the operation test, local port number [0x0FF0] is stored in the acceptable local port list (common).			
Test procedure 1. Connect the communication. 2. The “test application <OBE>” receives an event notification message from the ITS-LPCP in the OBE via the eventReport.indication. 3. The TS receives the event notification message from the OBE.			
Confirmation items ・ Verify that the event type received by the “test application <OBE>” in test procedure step 2 was “connection notice”. ・ Verify that the event code received by the TS in test procedure step 3 was the “accept port list”, and that the received additional event information is the same as the accept port list in the OBE.			

Test number	2-1-2	Items	ITS-LPCP operation Disconnection
Test overview ・ Verify that the communication between the TS and the OBE changes to disconnected status.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ The communication between the TS and the OBE is in connected status. ・ For the operation test, local port number [0x0FF0] is stored in the acceptable local port list in the OBE.			
Test procedure 1. Disconnect the communication. 2. The “test application <OBE>” receives an event notification message from the ITS-LPCP in the OBE via the EventReport.indication.			
Confirmation items ・ Verify that the event type received by the “test application <OBE>” in test procedure step 2 is “disconnection notice”.			

Test number	2-1-3	Items	ITS-LPCP operation Management service

Test overview

- Verify the operation procedure of the management service provided by the ITS-LPCP.
- Verify the local port open operation.

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-ASL-ELCP operates normally.
- For the operation test, local port number [0x0FF0] is stored in the acceptable local port list in the OBE, and local port number [0x0FF1] is not stored. The test data stored as the send data in the OpenPort.request primitive are as follows:

  Test data 1: The “openPort” number [0x0FF1] not registered in the acceptable local port list.

  Test data 2: The “openPort” number [0x0FF0] registered in the acceptable local port list.

  Test data 3: “openPort” is omitted.

Test procedure

1. Send the test data 1 from the “test application <OBE>” to the ITS-LPCP in the OBE via the OpenPort.request primitive.
2. The “test application <OBE>” receives the OpenPort.confirm primitive from the ITS-LPCP in the OBE.
3. Set the test data 2 as the message in test procedure step 1, and repeat steps 1 and 2.
4. Set the test data 3 as the message in test procedure step 1, and repeat steps 1 and 2.

Confirmation items

- Verify that the OpenPort.confirm primitive in which the specified local port number is stored in the “openPort” parameter is received by the “test application <OBE>” in test procedure step 2.
- Verify that the OpenPort.confirm primitive in whom the “openPort” parameter is omitted is received by the “test application <OBE>” in test procedure step 3.
- Verify that the OpenPort.confirm primitive in which the local port number opened by the ITS-LPCP is stored in the “openPort” parameter is received by the “test application <OBE>” in test procedure step 4.

Test number	2-1-4-1	Items	ITS-LPCP operation Data transfer service 1
Test overview ・ Verify the operation procedure of the data transfer service provided by the ITS-LPCP. ・ Send a data transfer message including a valid link address from the OBE to the TS via the TransferData.request primitive, and verify that the TS receives the data transfer message. (OBE to TS)			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in the connected status. ・ For the operation test, the test data stored in the NCP-SDU is as follows: Test data 1: Arbitrary data whose size doesn't exceed the MTU			
Test procedure 1. Send a data transfer message from the “test application <OBE>” to the TS via the TransferData.request primitive, including a valid link address to achieve communication connection, a valid destination local port number [0x0FF0] registered in the accept local port list in the TS, and test data 1. 2. The TS receives the data transfer message from the OBE.			
Confirmation items ・ Verify that the data received by the TS in test procedure step 2 matches test data 1.			

Test number	2-1-4-2	Items	ITS-LPCP operation Data transfer service 2
Test overview ・ Verify the operation procedure of the data transfer service provided by the ITS-LPCP. ・ Send a data transfer message from the TS to a valid link address in the OBE via the TransferData.request primitive, and then verify that the “test application <OBE>” receives the data transfer message. (TS to OBE)			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in the connected status ・ For the operation test, the test data stored in the NCP-SDU is as follows: Test data 1: Arbitrary data whose size doesn't exceed the MTU			
Test procedure 1. Send a data transfer message from the TS to the OBE via the TransferData.request primitive, with a valid link address, a valid destination local port number [0x0FF0] registered in the accept local port list in the OBE, and test data 1. 2. The “test application <OBE>” receives the data transfer message.			
Confirmation items ・ Verify that the data received by the “test application <OBE>” in test procedure step 2 matches test data 1.			

Test number	2-1-4-3	Items	ITS-LPCP operation Data transfer service 3
Test overview ・ Verify the operation procedure of the data transfer service provided by the ITS-LPCP. ・ Send a data transfer message from the OBE to an unconnected link address in the TS via the TransferData.request primitive, and then verify that the “test application <OBE>” receives the notice “The communication is not connected.” via the EventReport.indication.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in the connected status. ・ For the operation test, the test data stored in the NCP-SDU is as follows: Test data 1: Arbitrary data whose size doesn't exceed the MTU			
Test procedure 1. Send a data transfer message from the “test application <OBE>” to the TS via the TransferData.request primitive, including an unconnected link address and test data 1. 2. The “test application <OBE>” receives an EventReport.indication primitive from the ITS-LPCP in the OBE.			
Confirmation items ・ Verify that the event type received by the “test application <OBE>” in test procedure step 2 is “The communication is not connected.”.			

Test number	2-1-4-4	Items	ITS-LPCP operation Data transfer service 4
Test overview ・ Verify the operation procedure of the data transfer service provided by the ITS-LPCP. ・ Send a data transfer message from the TS to an unregistered local port number in the OBE via the TransferData.request primitive. The OBE sends an event notification message “The destination port is invalid.” to the TS. Verify that the TS receives this message.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in the connected status. ・ The port number [0x0FF2] is not registered in the accept port list in the OBE. ・ In the operation test, the test data stored in the NCP-SDU is as follows: Test data 1: Arbitrary data whose size doesn't exceed the MTU			
Test procedure 1. Send a data transfer message from the TS to the OBE via the TransferData.request primitive, with a valid link address, a sending destination local port number [0x0FF2] not registered in the accept local port list in the OBE, and test data 1. 2. The TS receives an event notification message from the OBE.			
Confirmation items ・ Verify that the event code in the event notification message received by the TS in test procedure step 2 is “The destination port is invalid.”.			

Test number	2-1-5	Items	ITS-LPCP operation Data transfer service in broadcast mode

Test overview

- Verify the operation procedure of the data transfer service provided by the ITS-LPCP.
- Send a data transfer message whose link address is a group broadcast address from the TS to the OBE via the TransferData.request primitive, and then verify that the “test application <OBE>” receives the data transfer message. (TS to OBE)

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-ASL-ELCP operates normally.
- Connection processing between the TS and the OBE has been completed, and they are in the connected status.
- For the operation test, the test data stored in the NCP-SDU is as follows:

  Test data 1: Arbitrary data whose size doesn't exceed the MTU

Test procedure

1. Send a data transfer message from the TS to the OBE via the TransferData.request primitive, with a group broadcast address specified as the link address, a valid destination local port number [0x0FF0] registered in the accept local port list in the OBE, and test data 1.

   At this time, 2 is used as group broadcast address.
2. The “test application <OBE>” receives the data transfer message from the TS.

Confirmation items

- Verify that the data received by the “test application <OBE>” in test procedure step 2 is the same as test data 1.

4.2.3.2.2 **Performance test**

Test number	2-2-1	Items	ITS-LPCP operation Measurement of initial connection time
Test overview ・ Measure the time between connection of the communication and receipt of an event notification message via the EventReport.indication primitive.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ For the operation test, the test data stored in the accept local port list are as follows: Test data 1: Port number: [0x0FF0], “primitiveType”: Omitted, “recvEventCode”: Omitted			
Test procedure 1. On the condition that test data 1 stored in the acceptable local port list, connect the communication. 2. The TS receives an event notification message via the EventReport.indication. 3. Record the time between the point at which the communication achieves connected status and the point at which the TS receives the event notification message.			
Confirmation items ・ Repeat test procedure steps 1 to 3 eleven times, and measure the time. ・ Exclude only the longest time among the 11 results, determine the average of the remaining 10 results, and use this time as the connection time. NOTE: The longest time is excluded so that the time required starting up the program in the test system server, etc. is excluded. ・ Verify the connection time is reasonable.			

Test number	2-2-2	Items	ITS-LPCP operation Transmission performance
Test overview ・ Transfer data from the TS using the echo service, and then calculate the data transfer speed from the time required for data transfer.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ For the operation test, the test data to be transferred is as follows: Test data 1: Arbitrary data whose size doesn't exceed the MTU			
Test procedure 1. Send a data transfer message from the TS to the OBE via the TransferData.request primitive, with test data 1. 2. The OBE sends back the data from the TS to the TS using the echo service. 3. The TS receives the data transfer message from the OBE.			
Confirmation items ・ Repeat test procedure steps 1 to 3 ten times. ・ In the TS, measure the time required for the test procedure steps 1 to 3. The time measurement method is not specified because it depends on the equipment specifications. Moreover, the unit of measured values may be “seconds”, and visual measurement is allowed. ・ Obtain the transfer speed from the measured time and the transferred data size.			

4.2.3.3 **Local Port Protocol (ITS-LPP) Test**

Test method and procedures for the ITS-LPP are as follows.

The tests described here are described on condition that confirmation has been completed for the operation tests required for the corresponding mobile station regarding the test items shown in 4.2.1.1 and 4.2.1.2.

4.2.3.3.1 **Operation Test**

Test number	3-1-1	Items	ITS-LPP operation Connection operation test
Test overview ・ Verify that Initial set up process between the TS and the OBE is completed.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-LPCP operates normally. ・ In the operation test, a local port number [0x0FF3] is stored in the accept local port list (common).			
Test procedure 1. Send the Connect.req from the “test application <OBE>” to the ITS-LPP in the OBE, with the “QueryLid” omitted and “QueryPort” set to [0x0FF3]. 2. Connect the communication. 3. The “test application <OBE>” receives the Connect.cnf primitive from the ITS-LPP in the OBE.			
Confirmation items ・ Verify that the link address and port number corresponding to the acceptable local port list registered in the TS are received by the “test application <OBE>”.			

Test number	3-1-2	Items	ITS-LPP operation Disconnection operation
Test overview ・ Verify that disconnection operation between the TS and the OBE is completed.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-LPCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in connected status. ・ For the operation test, “openPort” [0x0FF3] is stored and “primitiveType” and “recvEventCode” are omitted in the connection management table in the OBE.			
Test procedure 1. Disconnect the communication. 2. The “test application <OBE>” receives the Disconnect.ind primitive from the ITS-LPP in the OBE.			
Confirmation items ・ Verify that the Disconnect.ind primitive is received by the “test application <OBE>” in test procedure step 2.			

Test number	3-1-3	Items	ITS-LPP operation Unilateral data-send transaction service
Test overview ・ Verify the operation procedure of the unilateral data-send transaction service provided by the ITS-LPP. ・ Send the Invoke.req primitive from the OBE in the unilateral data-send transaction service, and then verify that the TS receives the Invoke.ind primitive.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-LPCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in connected status. ・ For the operation test, the test data in the Invoke.req primitive is as follows: Test data 1: Arbitrary [500] octets data			
Test procedure 1. Send the Invoke.req primitive from the “test application <OBE>” to the TS, with a valid port number [0x0FF3], a link address, “Transaction Type = 0” and test data 1. 2. The TS receives the Invoke.ind with the data from the OBE.			
Confirmation items ・ Verify that the data received by the TS in test procedure step 2 is the same as test data 1.			

Test number	3-1-4	Items	ITS-LPP operation Unilateral transaction service in broadcast mode
Test overview ・ Verify the operation procedure of the unilateral data-send transaction service provided by the ITS-LPP. ・ Send the Invoke.req primitive from the TS in the unilateral data-send transaction service in broadcast mode, and then verify that the OBE receives the Invoke.ind primitive.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-LPCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in connected status. ・ For the operation test, the test data the Invoke.req primitive is as follows: Test data 1: Arbitrary data whose size doesn't exceed the MTU			
Test procedure 1. Send the Invoke.req primitive from the TS to the OBE, with a port number [0x0FF3], a group broadcast address specified as the link address, “Transaction Type = 0” and test data 1. At this time, 2 is used as group broadcast address. 2. The “test application <OBE>” receives the Invole.ind with the data from the TS.			
Confirmation items ・ Verify that the data received by the “test application <OBE>” in test procedure step 2 is the same as test data 1.			

Test number	3-1-5	Items	ITS-LPP operation Request-response type transaction service
Test overview ・ Verify the operation procedure of the request-response type transaction service provided by the ITS-LPP. ・ Verify the operation of the request-response type transaction service.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-LPCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in connected status. ・ For the operation test, the test data in the Invoke.req primitive is as follows: Test data 1: Arbitrary data whose size doesn't exceed the MTU			
Test procedure 1. Send the Invoke.req primitive from the “test application <OBE>” to the TS, with a valid port number [0x0FF3], a link address, “Transaction Type = 1” and test data 1. 2. The TS receives the Invoke.ind primitive with the data from the OBE. 3. The TS sends the Invoke.res primitive whose send data is test data 1 to the OBE. 4. The “test application <OBE>” receives the Invoke.cnf primitive with the data from the TS.			
Confirmation items ・ Verify that the data received by the TS in test procedure step 2 is the same as test data 1. ・ Verify that the data received by the “test application <OBE>” in test procedure step 4 is the same as test data 1.			

Test number	3-1-6-1	Items	ITS-LPP operation Data retransmission operation 1
Test overview ・ Verify the operation procedure of the data retransmission operation provided by the ITS-LPP. ・ Send the transaction Invoke.req primitive from the OBE, with “Require Ack = 1” in the unilateral data-send transaction service, and then verify that the TS receives the Invoke.ind primitive.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-LPCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in connected status. ・ For the operation test, the test data in the Invoke.req primitive is as follows: Test data 1: Arbitrary data whose size doesn't exceed the MTU			
Test procedure 1. Send the Invoke.req primitive from the “test application <OBE>” to the TS, with a valid port number [0x0FF3], a link address, “Transaction Type = 0”, “Require Ack = 1” and test data 1. 2. The TS receives the Invoke.ind primitive from the OBE. 3. The TS sends the “Acknowledgement PDU” to the OBE via the TransferData.req primitive.			
Confirmation items ・ Verify that the data received by the TS in test procedure step 2 is the same as test data 1.			

Test number	3-1-6-2	Items	ITS-LPP operation Data retransmission operation 2

Test overview

- Verify the operation procedure of the data retransmission operation provided by the ITS-LPP.
- Send the Invoke.req primitive from the OBE, with “Require Ack = 1” in the unilateral data-send transaction service. The TS aborts receiving of the Invoke PDU once (that is, the TS does not send the “Acknowledgement PDU” to the OBE), and the OBE sends the Invoke PDU again. Verify that the TS receives the Invoke.ind.

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-LPCP operates normally.
- Connection processing between the TS and the OBE has been completed, and they are in connected status.
- For the operation test, the test data in the Invoke.req primitive is as follows:

  Test data 1: Arbitrary data whose size doesn't exceed the MTU

Test procedure

1. Send the Invoke.req primitive from the “test application <OBE>” to the TS, with a valid port number [0x0FF4], a link address, “Transaction Type = 0”, “Require Ack = 1” and the test data 1.
2. The TS receives the Invoke PDU from the OBE.
3. The TS does not send the “Acknowledgement PDU” to the OBE.
4. The TS receives the Invoke PDU resent from the OBE.
5. The TS sends the “Acknowledgement PDU” to the OBE.

Confirmation items

- Verify that the data received by the TS in test procedure step 4 is the same as test data 1.

Test number	3-1-7-1	Items	ITS-LPP operation Segmentation/assembly operation 1
Test overview ・ Verify the operation procedure of the segmentation/assembly operation provided by the ITS-LPP. ・ Send the Invoke.req primitive from the OBE, with send data that exceeds the maximum transmission unit (MTU) in the unilateral data-send transaction service, and then verify that the TS receives the Invoke.ind.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-LPCP operates normally. ・ Connection processing between the TS and the OBE has been completed, and they are in connected status. ・ For the operation test, the test data in the Invoke.req primitive is as follows: Test data 1: Arbitrary data whose size exceeds the MTU			
Test procedure 1. Send the Invoke.req primitive from the “test application <OBE>” to the TS, with a valid port number [0x0FF3], a link address, “Transaction Type = 0” and test data 1. 2. The TS receives the Invoke.ind with the data sending from the OBE. 3. The TS sends the “Acknowledgement PDU” to the OBE.			
Confirmation items ・ Verify that the data received by the TS in test procedure step 2 is the same as test data 1.			

Test number	3-1-7-2	Items	ITS-LPP operation Segmentation/assembly operation 2

Test overview

- Verify the operation procedure of the segmentation/assembly operation provided by the ITS-LPP.
- Send the Invoke.req primitive from the OBE, with send data that exceeds the MTU in the unilateral data-send transaction service. The TS aborts once receiving of the InvokeSegment PDUs for all segments except the final segment, the TS sends the “Nack PDU” for the aborted segments and the OBE sends again InvokeSegment PDUs. Verify that the TS receive the Invoke.ind primitive.

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-LPCP operates normally.
- Connection processing between the TS and the OBE has been completed, and they are in connected status.
- For the operation test, the test data in the Invoke.req primitive is as follows:
  Test data 1: Arbitrary data which size exceeds the MTU

Test procedure

1. Send the Invoke.req primitive from the “test application <OBE>” to the TS, with a valid ort number [0x0FF5], a link address, “Transaction Type = 0” and test data 1.
2. The TS aborts receiving of InvokeSegment PDUs for all segments except the final egment from the OBE.
3. The TS sends the “Nack PDU” to the OBE for aborted segments.
4. The TS receives resent InvokeSegment PDUs and the Invoke.ind primitive with the data rom the OBE
5. The TS sends the “Acknowledgement PDU” to the OBE.

Confirmation items

- Verify that the data received by the TS in test procedure step 4 is the same as test data 1.

Test number	3-1-7-3	Items	ITS-LPP operation Segmentation/assembly operation 3

Test overview

- Verify the operation procedure of the segmentation/assembly operation provided by the ITS-LPP.
- Send the Invoke.req primitive from the OBE, with send data which size exceeds the MTU in the unilateral data-send transaction service. The TS aborts once receiving of an InvokeSegment PDU for the final segment, and the TS retransmits the final segment.Verify that the TS receives the Invoke.ind.

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-LPCP operates normally.
- Connection processing between the TS and the OBE has been completed, and they are in connected status.
- For the operation test, the test data in the Invoke.req primitive is as follows:

  Test data 1: Arbitrary data which size exceeds the MTU

Test procedure

1. Send the Invoke.req primitive from the “test application <OBE>” to the TS, with a valid port number [0x0FF6], a link address, “Transaction Type = 0” and test data 1.
2. The TS aborts receiving of an InvokeSegment PDU for the final segment from the OBE.
3. The TS receives the resent final segment and Invoke.ind primitive with the data sending from the OBE.
4. The TS sends the “Acknowledgement PDU” to the OBE.

Confirmation items

- Verify that the data received by the TS in test procedure step 3 is the same as test data 1.

Test number	3-1-8	Items	ITS-LPP operation Transaction abort operation

Test overview

- Verify the operation procedure of the transaction abort operation provided by the ITS-LPP.
- Send the Abort.req primitive from the OBE to the TS, and then verify the transaction abort operation in both the OBE and the TS.

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-LPCP operates normally.
- Connection processing between the TS and the OBE has been completed, and they are in connected status.
- Confirmation has been completed for operation test number 3-1-5.
- For the operation test, the test data in the Invoke.req primitive is as follows:

Test procedure

1. Send the Invoke.req primitive from the “test application <OBE>” to the TS, with a valid port number [0x0FF7], a link address, “Transaction Type = 1” and test data 1.
2. The TS receives the Invoke.ind with the data sending from the OBE.
3. The “test application <OBE>” sends the Abort.req primitive to the TS.
4. The TS receives the Abort PDU from the OBE.
5. The “test application <OBE>” receives the Abort.ind primitive from the ITS-LPP in the OBE.

Confirmation items

- Verify that the Abort.ind primitive is received by the TS in test procedure step 4.
- Verify that the Abort.ind primitive is received by the “test application <OBE>”in test procedure step 5.

Test number	3-1-9-1	Items	ITS-LPP operation Connection management service 1

Test overview

- Verify the operation procedure of the connection management service provided by the ITS-LPP.
- Verify the operation of the transaction start possibility query service (query the counterpart port number).

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-LPCP operates normally.
- Connection processing between the TS and the OBE has been completed, and they are in connected status.
- Confirmation has been completed for operation test number 3-1-1.
- The number [0x0FF3] is stored, and [0x0FF8] is not stored in the acceptable local port list of the TS.
- For the operation test, the test data in the Connect.req primitive are as follows:

  Test data 1: “QueryLid” is specified, and a number [0x0FF3] registered in the TS is specified for “QueryPort”.

  Test data 2: “QueryLid” is specified, and a number [0x0FF8] not registered in the TS is specified for “QueryPort”.

  Test data 3: “QueryLid” is specified, and “QueryPort” is not specified.

Test procedure

1. Send the Connect.req primitive from the “test application <OBE>” to the ITS-LPP in the OBE, with test data 1.
2. The “test application <OBE>” receives the Connect.cnf primitive from the ITS-LPP in the OBE.
3. Set test data 2 as the send data for test procedure step 1, and repeat test procedure steps 1 and 2.
4. Set test data 3 as the send data for test procedure step 1, and repeat test procedure steps 1 and 2

Confirmation items

- Verify that the link address and port number corresponding to the acceptable local port list registered in the TS is received by the “test application <OBE>” in test procedure steps 2, 3 and 4.

Test number	3-1-9-2	Items	ITS-LPP operation Connection management service 2

Test overview

- Verify the operation procedure of the receiving port registration and receiving port deregistration provided by the ITS-LPP.
- Register and deregister receiving ports in the OBE, and verify receiving by the TS.

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-LPCP operates normally.
- Connection processing between the TS and the OBE has been completed, and they are in connected status.
- The port number [0x0FF8] is not stored in the acceptable local port list in the TS.

Test procedure

1. Register an accept port number [0x0FF8] to the ITS-LPP in the OBE via the RegisterPort.req primitive from the “test application <OBE>”.
2. The TS receives the accept port list from the OBE.
3. Deregister the accept port number registered to the ITS-LPP in the OBE in test procedure step 1 via the DeregisterPort.req primitive from the “test application <OBE>”.
4. The TS receives the reject port notification from the OBE.

Confirmation items

- Verify that the accept port list received by the TS in test procedure step 2 is the same as the port number registered in test procedure step 1.
- Verify that the reject port received by the TS in test procedure step 4 is the same as the port number deregistered in test procedure step 3.

4.2.3.3.2 **Performance Test**

Test number	3-2-1	Items	ITS-LPP operation Initial set up time
Test overview ・ Measure the time from connection of the communication to the time when the Connect.cnf is sent.			
Test conditions ・ The basic parameters are those in subclause 4.2.2. ・ The ITS-ASL-ELCP operates normally. ・ For the operation test, a local port number [0x0FF3] is stored in the acceptable local port list (common). The test data in the Connect.req primitive is as follows: Test data 1: “QueryLid” is not specified, and “QueryPort” is not specified.			
Test procedure 1. Send the Connect.req primitive from the “test application <OBE>” to the ITS-LPP in the OBE, with test data 1. 2. Connect the connection. 3. The TS receives the Connect.cnf primitive from the OBE. 4. Record the time from the point the communication achieves connected status to the point when the TS receive the Connect.cnf primitive.			
Confirmation items ・ Repeat test procedure steps 1 to 3 ten times, and measure the time. ・ Obtain the average of the 10 results, and use it as the connection time. ・ Verify the connection time is reasonable.			

Test number	3-2-2	Items	ITS-LPP operation Data transfer performance

Test overview

- Transfer data from the TS using the echo service, and then calculate the data transfer speed from the time required for data transfer.

Test conditions

- The basic parameters are those in subclause 4.2.2.
- The ITS-ASL-ELCP operates normally.
- For the operation test, the test data to be transferred is as follows:

  Test data 1: Arbitrary data whose size doesn't exceed the MTU

Test procedure

1. Send a data transfer message from the TS to the OBE via the Invoke.req primitive, with test data 1.
2. The OBE sends back the data from the TS to the TS using the echo service.
3. The TS receives the data transfer message from the OBE.

Confirmation items

- Repeat test procedure steps 1 to 3 ten times.
- In the TS, measure the time required for test procedure steps 1 to 3. The time measurement method is not specified because it depends on the equipment specifications. Moreover, the unit of measured values may be “seconds”, and visual measurement is allowed.
- Obtain the transfer speed from the measured time and the transferred data size.

[BLANK]

Annex A: Protocol Stack for ITS-ASL

A.1 Outline

This guideline defines the ITS application sublayer (hereinafter referred to as the ITS-ASL: ITS Application Sub-layer) that extends the functions of the DSRC platform, which is defined by the standard ARIB-STD-T88, in order to support applications such as automated driving support apps, satellite billing apps, and GNSS positioning data usage apps, and to support both existing wireless communications devices and new wireless communications devices that use V2X technology, by using the basic application interface (hereinafter referred to as basic API) in narrow area communications (DSRC) defined by the standard ARIB STD-T75.

ITS-ASL was designed to be applied to narrow area communications such as the lower protocol stacks of ITS FORUM RC-005, IEEE 802.11, ETSI EN 302 636, and ARIB STD-T109, and wide area communications such as LTE, 4G, and 5G. It supplements these wireless communication protocol functions to make it possible to execute applications that use basic API, which are upper layer API as seen from ITS-ASL, without being bound by the restrictions placed by lower protocol stacks.

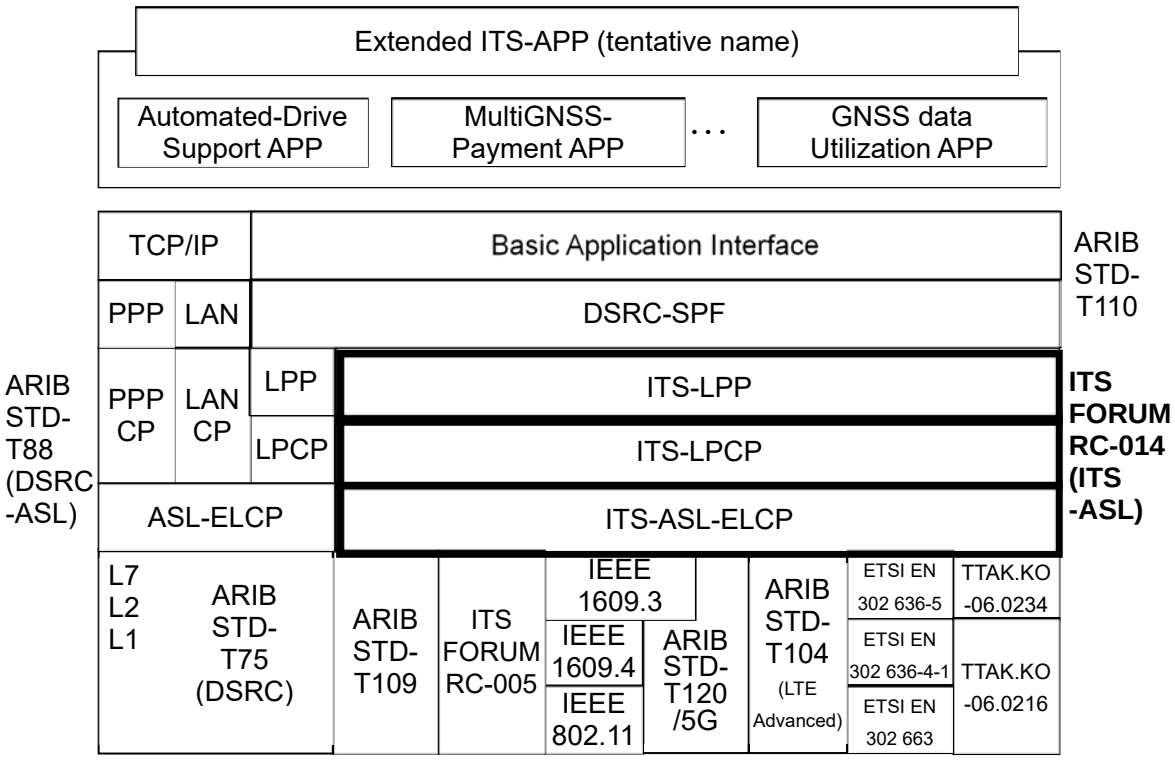
This guideline adds lower protocol stacks with each revision, as shown in Table A-1, and will continue to support new wireless communications produced by advances in V2X technologies.

Table A-1 ITS-ASL guideline versions and supported wireless communication standards

Version	ITS FORUM RC-005	IEEE 802.11	ARIB STD-T104 （LTE Advance）	ARIB STD-T109	ARIB STD-T120/5G	ETSI EN 302 663	TTAL.KO-06.0216	・・
1.0	○	○						
2.0	○	○	○					
3.0	○	○	○	○	○			
:								

A.2 Location of ITS-ASL in the Protocol Stack

Figure A.2-1 shows the location of ITS-ASL in the protocol stack.



FigureA.2-1 Protocol configuration of the extended ITS platform

Annex B: Differences from ARIB STD-T88

Table B-1 shows the differences between this guideline and ARIB STD-T88.

Table B-1 Differences between this guideline and ARIB STD-T88.

Item		DSRC-ASL	ITS-ASL	Description
General				
	Protocol construction	It consist of below protocol - ASL-NCP : LPCP, PPPCP, LANCP - ASL-ELCP	It consist of below protocol - ITS-LPCP (same function as LPCP of DSRC-ASL) - ITS-ELCP (same function as ELCP of DSRC-ASL)	1.2 2.1
	Communication sublayer	DSRC (ARIB STD-T75)	RC-005, IEEE 802.11, TTA, ETSIEN 302 636, ARIB STD-T109, LTE	1.2 2.2 3.1.2
	Link Address	DSRC (ARIB STD-T75) generates link address. Link address is distinguished simultitaneous broadcast, group broadcast, and private	ITS-ASL-ELCP generates link address. Link address is distinguished group broadcast and private.	2.3
Function				
	Multiprotocol	Compatible	Compatible (It is necessary for distinguish LPCP and communication control management.)	3.1.1
	Client / Server type communication control	Compatible	Non-compatible (It is not necessary, because mobile station is able to transmit voluntarily.)	3.1.1
	Bulk transmission control	Compatible	Compatible (It is necessary if sendable data size of lower layer is the smaller than MTU size. It is applied to the transaction from mobile station to base station.)	3.1.1 3.1.2.4

Item		DSRC-ASL	ITS-ASL	Description
	Broadcast mode control	Compatible (It is not applied to the transaction from mobile station to base station.)	Compatible (It is necessary for executing broadcast communication. It is applied to the transaction from mobile station to base station.)	3.1.1 3.1.2.4
	Access control	Compatible	Non-compatible (It is not necessary, because current DSRC basic application doesn't use this function.)	3.1.1
	Communication link management	Compatible	Compatible (It is necessary for the link management of individual communication.)	3.1.1
Extended Link Protocol				
	Service type	communication service	Communication service (It is almost the same function as DSRC-ASL- ELCP)	3.1.2.1
	Value of Group broadcast link address	- 2 (Case: Base station support bulk transmission control) - 3 (Case: Base station doesn't support bulk transmission control)	- 0 (It is support bulk transmission control in any case.)	
Link Control Management				
	Service type	- EventInformation, Echo/ EchoReply, MIB access	- EventInformation - Protocol ID and Access control information and procedure is regulated. (They are equal function as DSRC-ASL-ELCP)	3.1.3.1 3.1.3.2.2 3.1.3.4.1 3.1.3.4.2

Item		DSRC-ASL	ITS-ASL	Description
	Link control management	- Initial connection: using the function of DSRC layer - Continuation transmitting: using Client/Server type communication control for managing the remote station's communication status.	- Initial connection: not using the function of the lower layer - Continuation transmitting: not using the Client/Server type communication control function. - Protocol ID and Access control information is additionally regulated.	3.1.3.2.2 3.1.3.3.1 3.1.3.3.2

[BLANK]

Annex C: Supplement in the Case of IEEE 802.11

C.1 Subdivision Layer

In the case of IEEE 802.11, service primitive is used for identification of connection. Service primitive is provided by WSMP of IEEE1609.3. (Refer to Figure C.1-1)

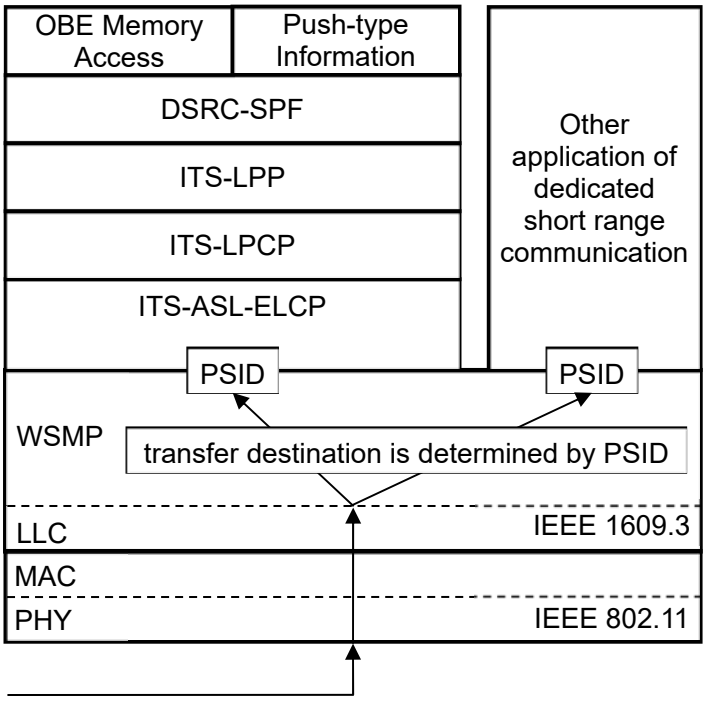


Figure C.1-1 Identification of connection for IEEE 802.11

C.2 Identification of Connecting

In the case of IEEE802.11, PSID is used for identification of the connection. Identification method is shown in the section 2.2.

C.3 Service Primitive and PDU

In the case of IEEE 802.11, when it receives SendDataUnit.request from ITS-LPCP, it constructs ASL-PDU and send to remote station by using WaveShortMessage.request of WSMP. (Refer to Figure C.3-1)

When it receives WaveShortMessage.indication from WSMP, it extracts ASL-SDU from ASL-PDU. And it send to ITS-LPCP by using SendDataUnit.indication. (Refer to Figure C.3-2)

Detail of service primitive is shown in section 3.1.2.3.3. PDU is shown in section 3.1.2.2.

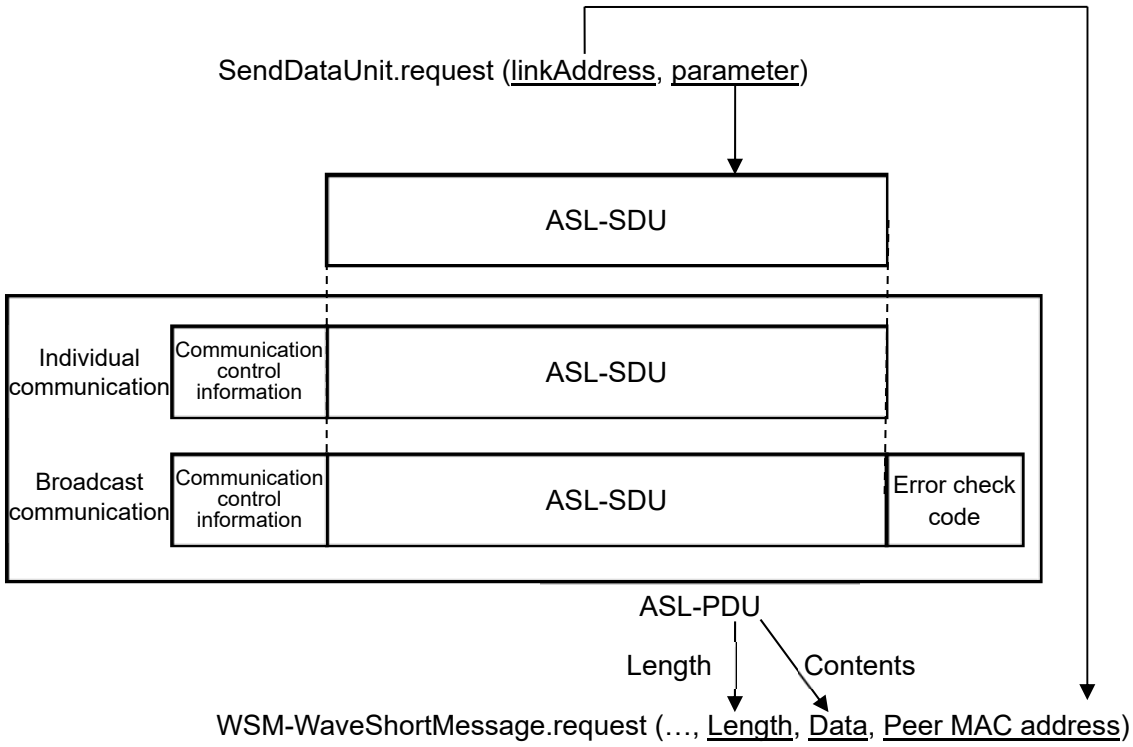


Figure C.3-1 Mapping of sublayer primitive for SendDataUnit.request

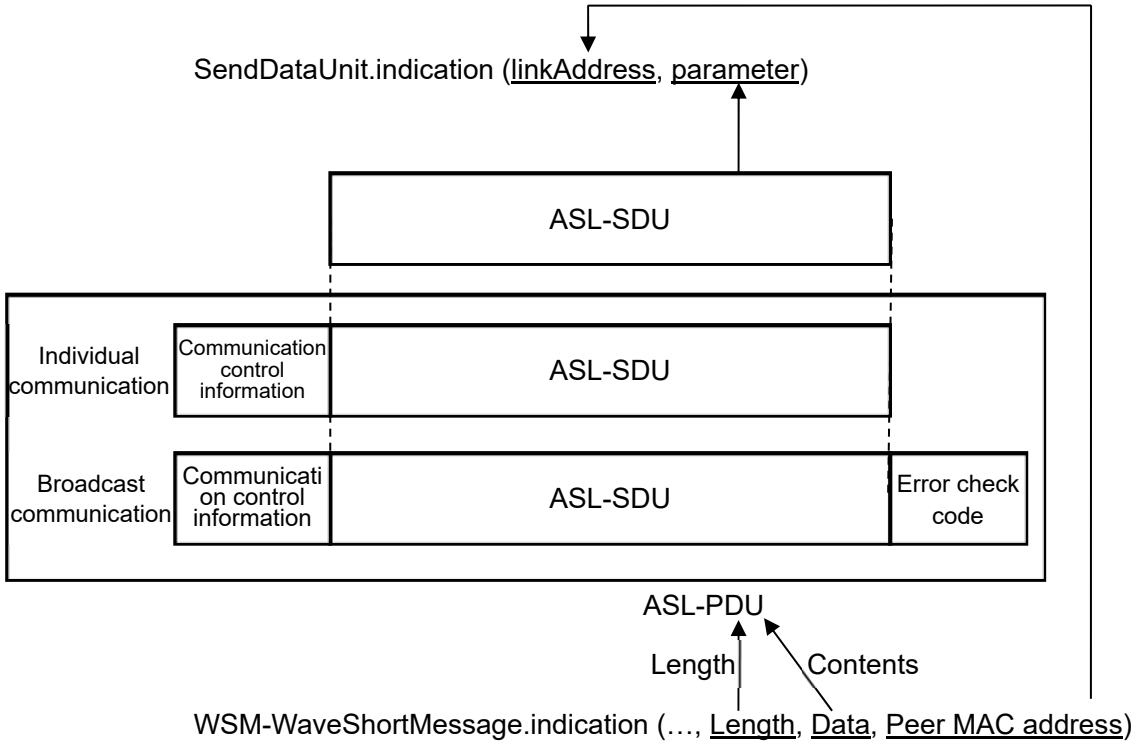


Figure C.3-2 Mapping of sublayer primitive for SendDataUnit.indication

Annex D: Study of use of ARIB STD-T104

D.1 Overview

Figure D.1-1 shows a conceptual image of the use of ITS-ASL with LTE-Advanced as its lowest layer. In this application example, virtual ITS spots are defined on top of wide area networks created using LTE-Advanced and are used to provide service. Other short-range communications and LTE-V2X are used to handle the delivery of information which must be provided in real time, such as control information. The delivery of this kind of information is not included within the scope of this study.

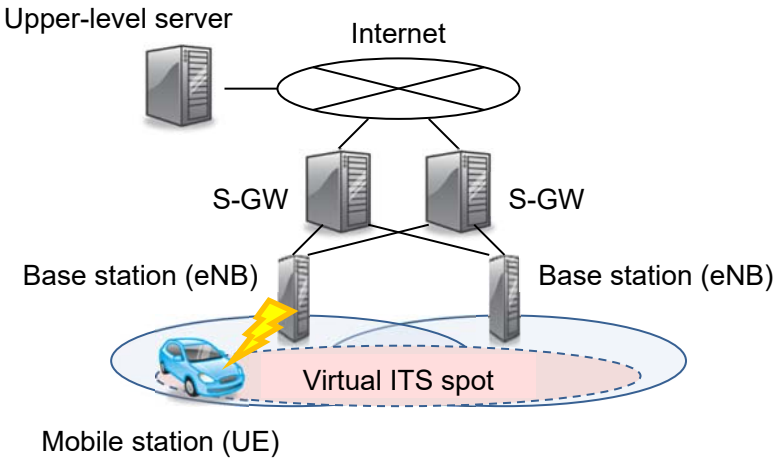


Figure D.1-1 Conceptual Image of Application of ITS-ASL Using LTE-Advanced

D.2 Conceptual Image of Service Provision

Figure D.2-1 shows a conceptual image of service provision. Information whose location does not change is provided in areas set in advance (e.g., travel time information is delivered in 1 kilometer intervals, information about areas with large numbers of accidents is distributed roughly 2 kilometers before the area). When dynamic traffic events occur, areas are defined based on actual conditions and service is provided (e.g., when an accident occurs, when restrictions are put in place due to an accident, and when traffic congestion occurs due to restrictions, information is delivered roughly 8 kilometers before each of the corresponding locations).

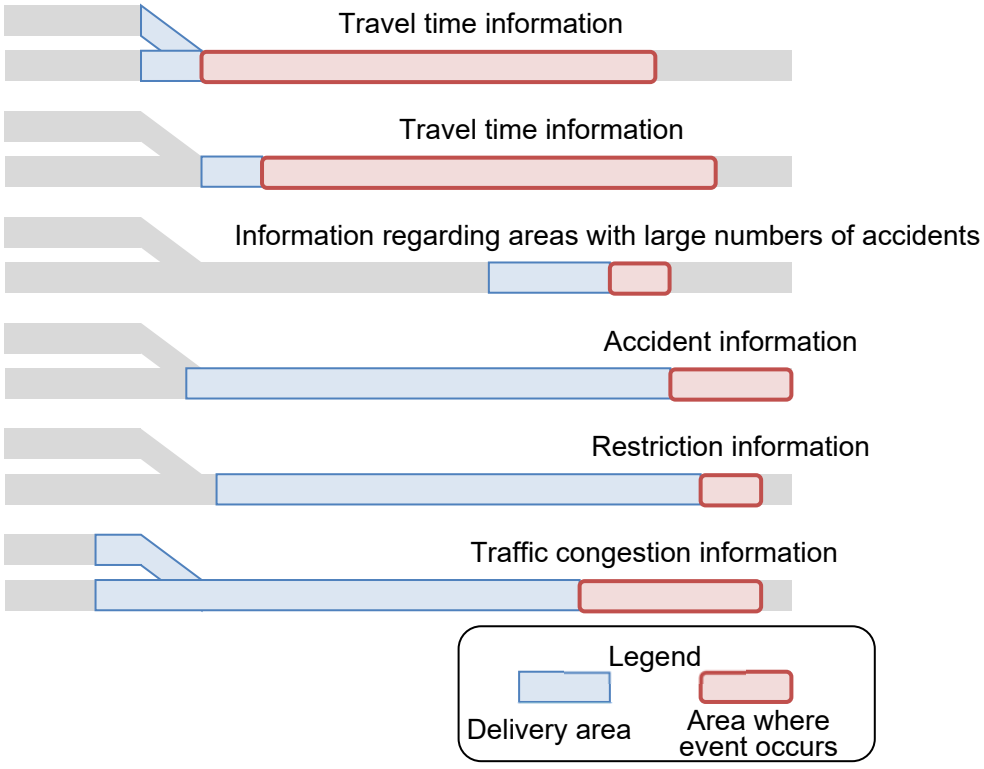


Figure D.2-1 Conceptual Image of Service Provision

D.3 Expression of service provision area location information

The service provision area scope is expressed based on the expression format, which in turn is based on the CRPs being studied by SIP-adus. Table D.3-1 shows, the location information expression type used varies depending on the type of service provision area.

Table D.3-1 Comparison of location information expression types

Type	Overview	Purpose
Location information expression type 1	Distance difference from CRP	Expresses service provision areas primarily for intersections.
Location information expression type 2	Road distance from CRP and offset from center line of road	Expresses service provision areas centered on roads.

(a) Location information expression type 1

Expresses service provision areas primarily for intersections. Figure D.3-1 shows a conceptual image, and Table D.3-2 shows the data format.

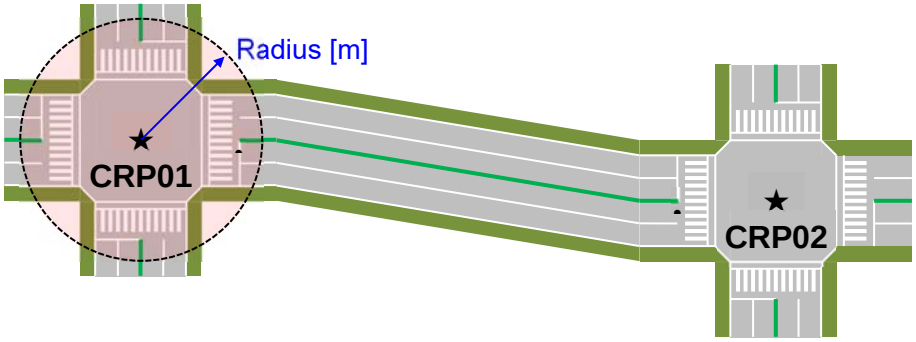


Figure D.3-1 Conceptual Image of Expression of Service Provision Area Location (Location Information Expression Type 1)

Table D.3-2 Data format of service provision area (location information expression type 1)

No.	Name	Contents
1	Area identifier	Number used to identify the service provision area
2	Location information expression type	"1"
3	CRP ID	Identifier of the CRP on which the service provision area is centered
4	Radius	Radius of service provision area around center CRP [m]

(b) Location information expression type 2

Expresses the scope of service provision areas centered on roads. Figure D.3-2 shows a conceptual image, and Table D.3-3 .

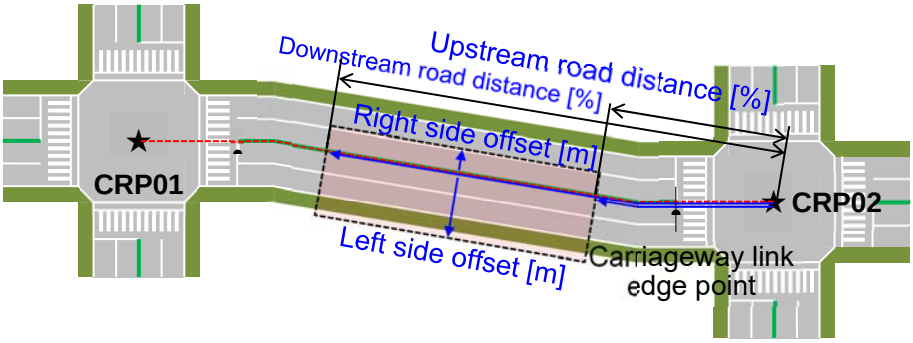


Figure D.3-2 Conceptual Image of Expression of Service Provision Area Location (Location Information Expression Type 2)

Table D.3-3 Data format of service provision area (location information expression type 2)

No.	Name	Contents
1	Area identifier	Number used to identify the service provision area
2	Location information expression type	"2"
3	Upstream CRP ID	Identifier of the CRP on the upstream side of the service provision area
4	Downstream CRP ID	Identifier of the CRP on the downstream side of the service provision area
5	Upstream road distance	Relative distance between the two CRPs indicating the upstream end point of the service provision area [%]
6	Downstream road distance	Relative distance between the two CRPs indicating the downstream end point of the service provision area [%]
7	Left side offset	Distance from the carriageway center line indicating the leftmost end point of the service provision area [m] (positive values indicate left side, negative values indicate right side)
8	Right side offset	Distance from the carriageway center line indicating the rightmost end point of the service provision area [m] (positive values indicate left side, negative values indicate right side)

D.4 Lower protocol

HTTP is believed to be easier to actually use than TCP due to the facility of IP address management, application handling, and the like. However, HTTP traffic takes slightly longer to process than TCP traffic, so this section explores whether or not this would present a problem. Assuming a maximum mobile station driving speed of 180km/h and a minimum service provision area length of 100m, it would take two or more seconds for a mobile station to traverse a service provision area. Assuming a processing latency on the order of 100ms, there would be no problem with information being missed. Information which must be delivered in real time is not included in this scope, so there would be no latency issues with regard to this information. Given this, the lower layer used by ITS-ASL will be switched from TCP to HTTP.

In this study, HTTP requests are sent from mobile stations to upper layer servers and HTTP responses are sent from upper layer servers to mobile stations.

D.5 Service provision area management and determination

Based on the principal that processing which is dependent on lower layers is performed in this layer, in ITS-ASL, this layer is used to compare mobile station location information and service provision area and perform area determination processing to decide what data is to be distributed. ITS-ASL shall also provide a primitive that allows applications to flexibly set service provision areas.

The main agents that could perform area determination are the upper layer servers and the mobile stations. Figure D.5-1 and Figure D.5-2 show the processing sequences when area determination is performed by an upper layer server and by a mobile station. In the figures, data to be delivered for service provision areas A, B, and C are indicated as Data A, Data B, and Data C, respectively.

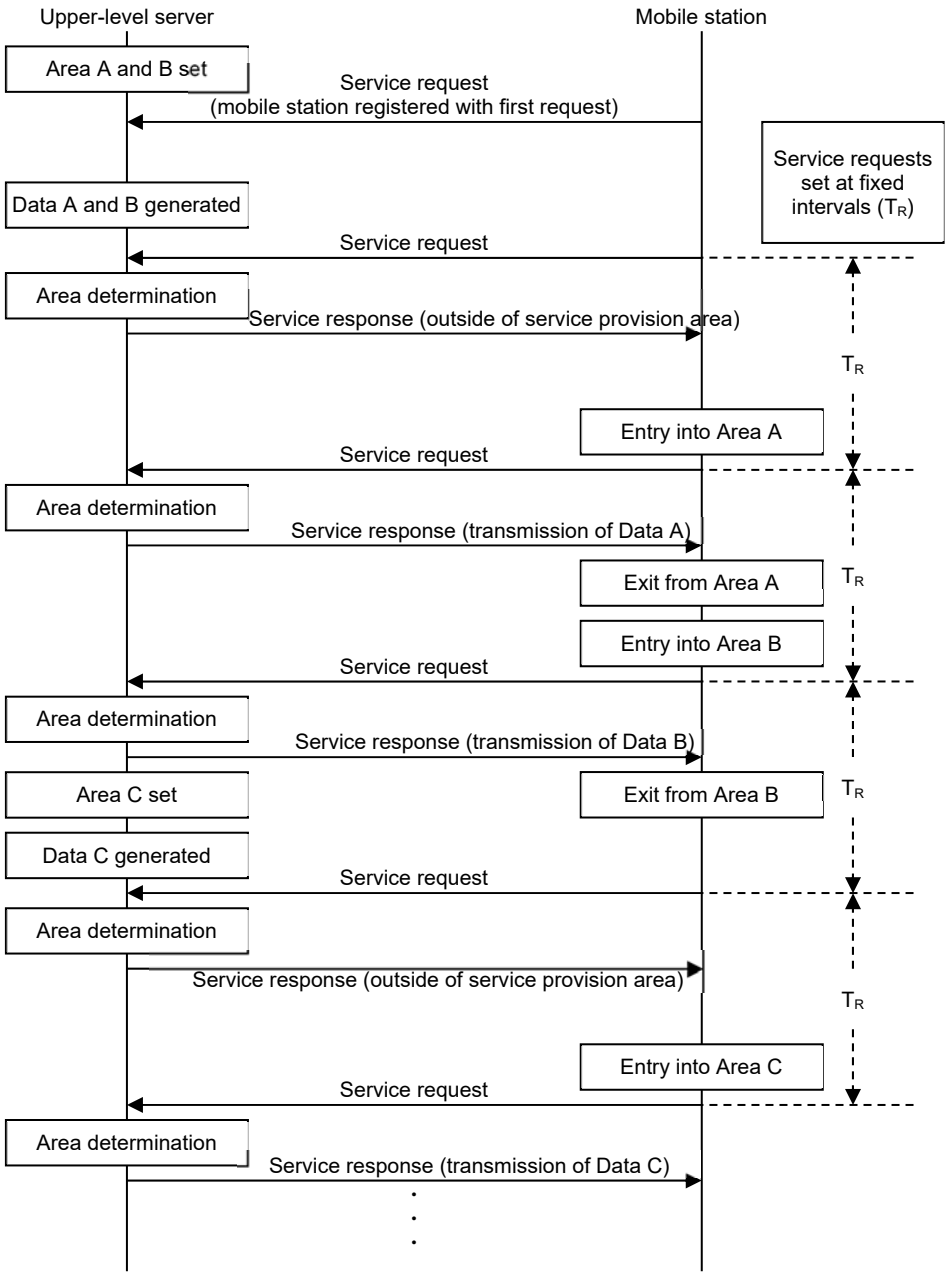


Figure D.5-1 Example of Processing Procedure for Data Delivery from Upper Layer Server (Area Determination Performed by Upper Layer Server)

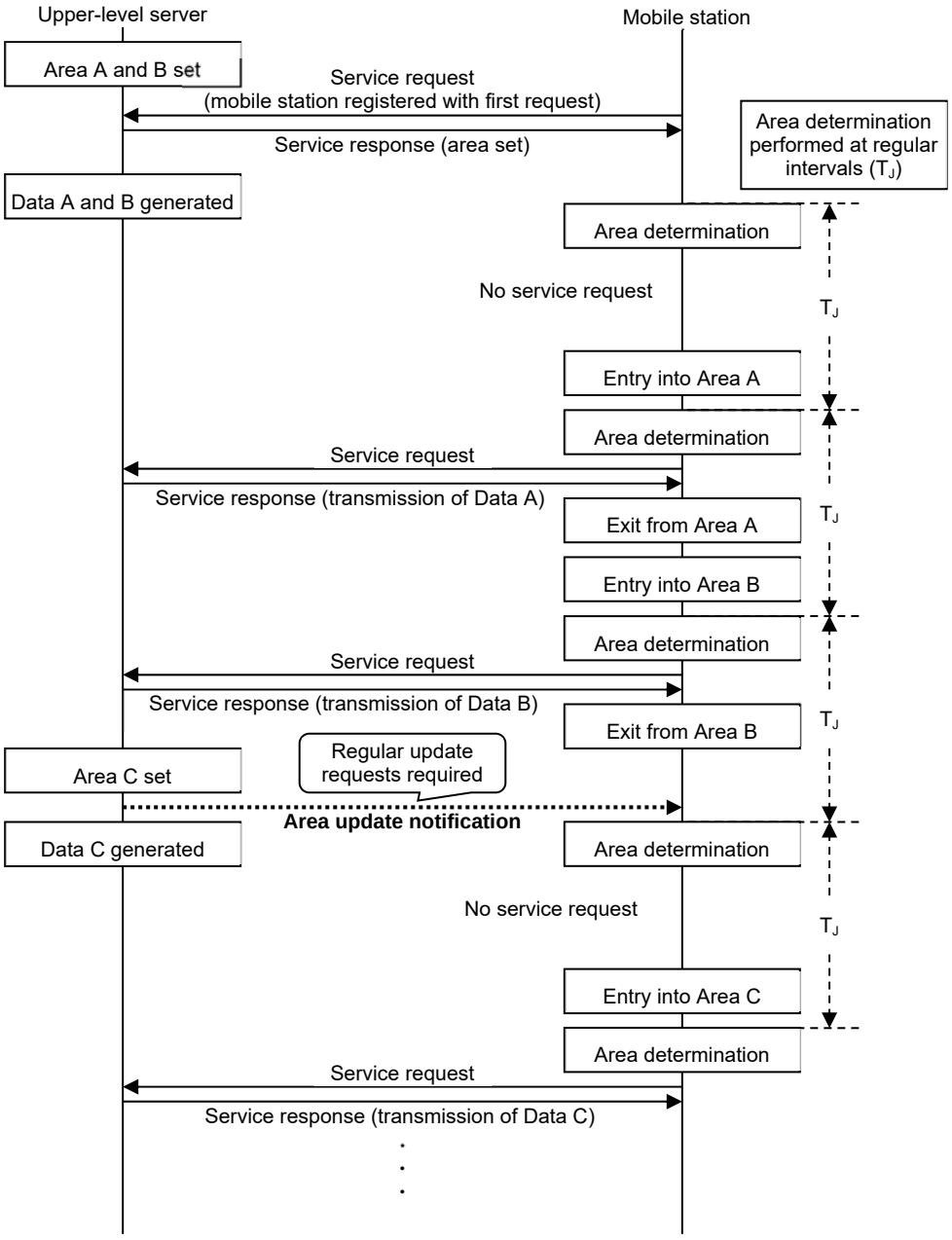


Figure D.5-2 Example of Processing Procedure for Data Delivery from Upper Layer Server (Area Determination Performed by Mobile Station)

Table D.5-1 shows a comparison of the agents used to perform area determination.

Table D.5-1 Comparison of implementing entities

Implementing entity	Upper-level server	Mobile station
Overview	Service provision area settings are retained by upper layer servers. Mobile stations regularly send data delivery requests with location information to upper layer servers. Upper layer servers use the location information they receive to perform area determination, and deliver information that matches the area conditions.	Upper layer servers regularly deliver service provision area settings to mobile stations. Mobile stations regularly perform area determination, and when there is information that matches the area conditions, they send data delivery requests to upper layer servers. Upper layer servers deliver data as requested.
Advantages	* There is no need to send area settings to mobile stations.	* Data delivery requests from mobile stations only need to be sent when inside the service provision area.
Disadvantages	* Data delivery requests must be regularly sent from mobile stations. When mobile stations are often outside of service provision areas, this is inefficient.	* Area settings must be sent to mobile stations. Mobile stations must regularly send requests.

When mobile stations perform area determination, it provides the advantage of not requiring data delivery requests to be sent when outside service provision areas. However, as indicated above, when area settings are dynamically changed, every time there are area setting changes, all mobile stations must be updated, making management difficult. Furthermore, from the standpoint of promoting adoption, it is important to minimize the amount of processing performed on the mobile station side. For these reasons, area determination is performed on the upper layer server side.

D.6 Conceptual image of service implementation

(a) Batch acquisition of on-board device memory

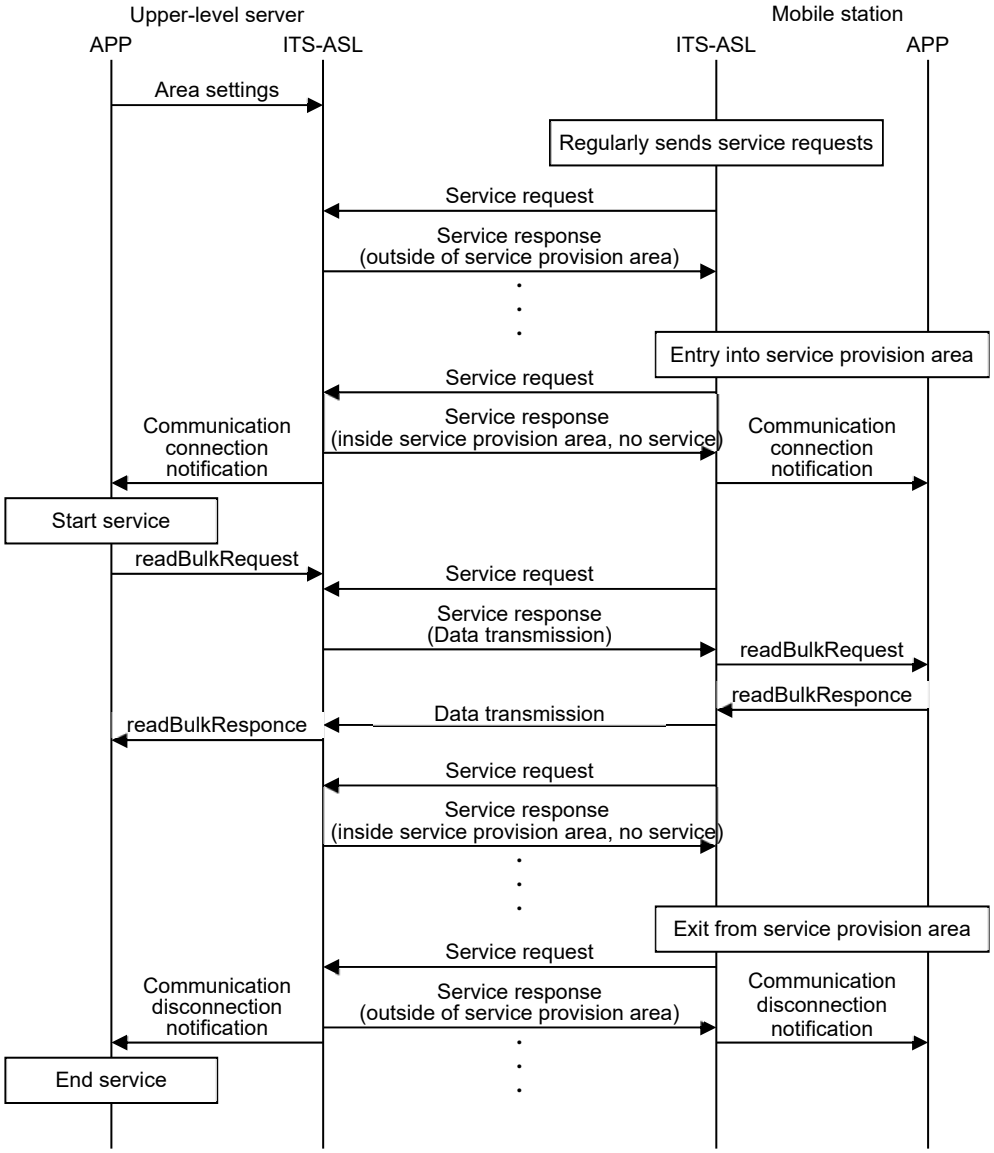


Figure D.6-1 Example of Processing Procedure for Batch Acquisition of On-board Device Memory Using ARIB STD-T104

(b) Downlink Continual Broadcast Transmission

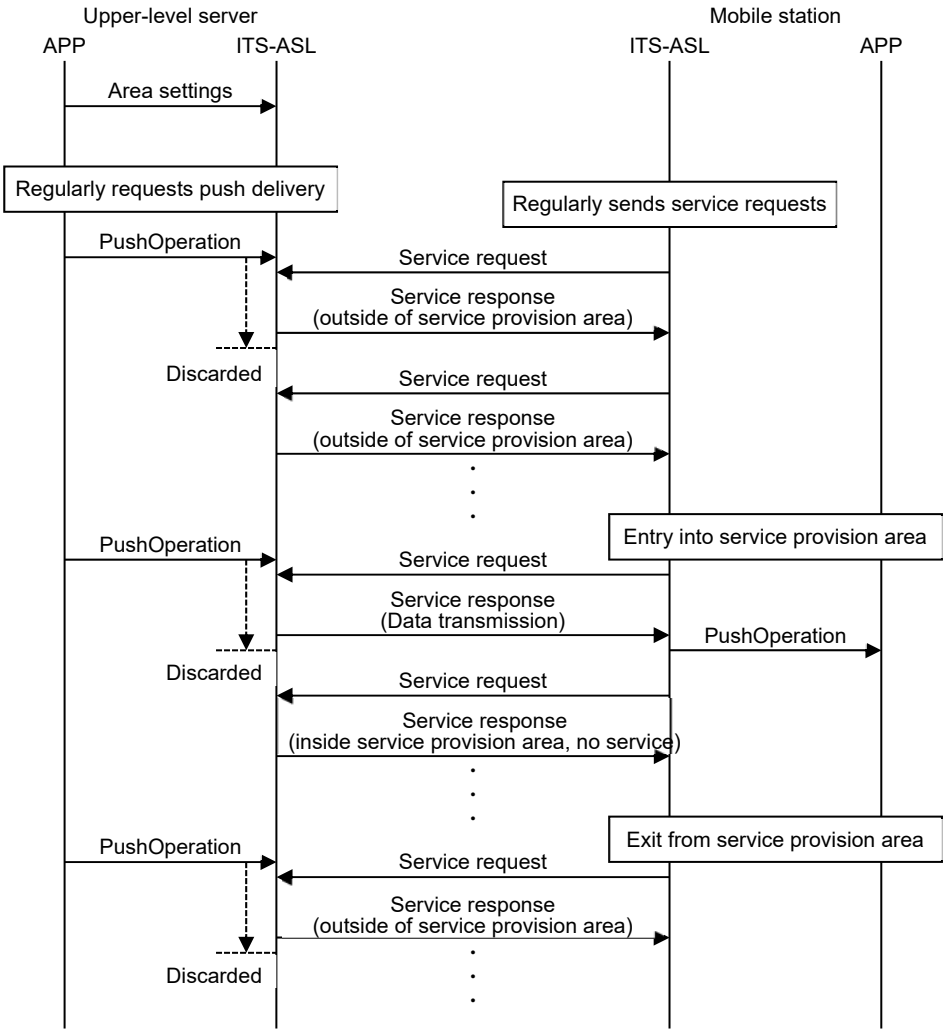


Figure D.6-2 Example of Processing Procedure for Downlink Continual Broadcast Transmission Using ARIB STD-T104

D.7 Overview of services and areas

(a) Service and service provision area management

Figure D.7-1 and Figure D.7-2 show the relationships between services and service provision areas and examples of management procedures. As the examples show, the relationships between services and service provision areas are expected to be many-to-many for upper layer servers and applications.

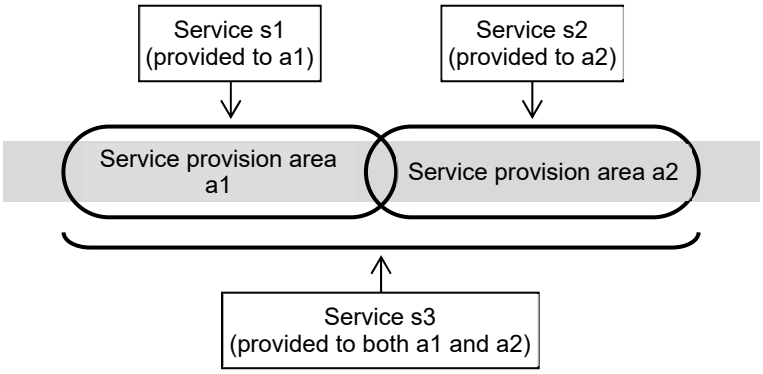


Figure D.7-1 Example of Relationships between Services and Service Provision Areas

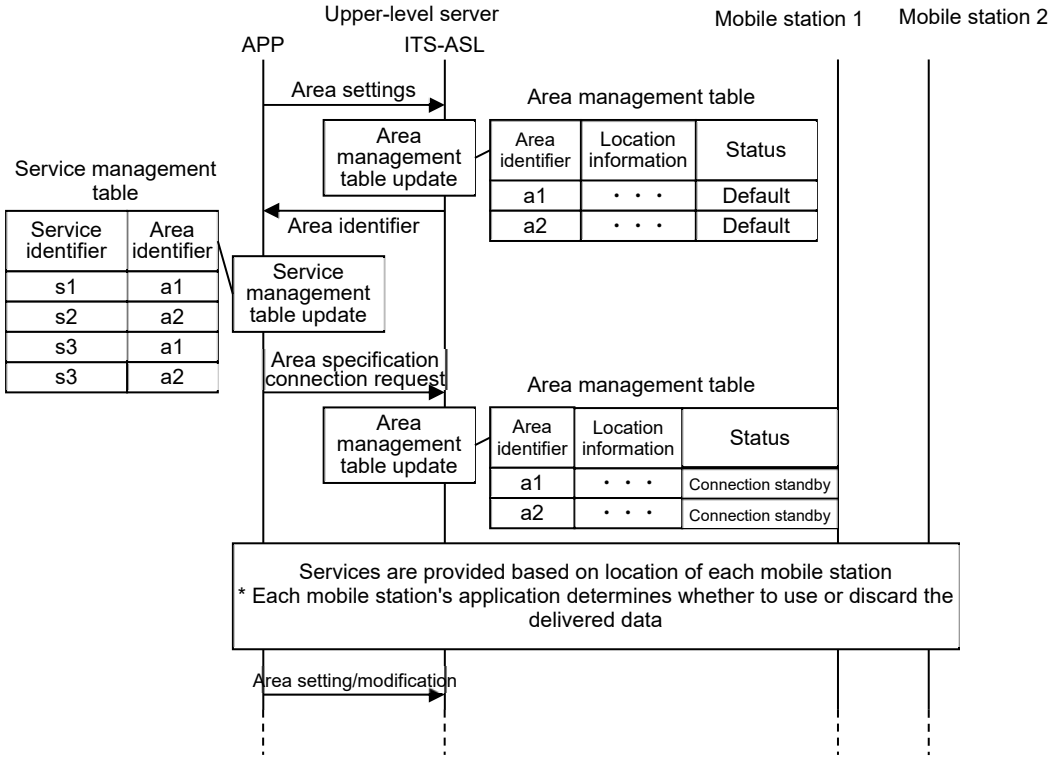


Figure D.7-2 Example of Management Procedure for Services and Service Provision Areas

(b) Transmission connection management

In transmission connection management, first a service request is sent from a mobile station to an upper layer server. In the first service request, the link address is set to the group's broadcast link address. The upper layer server then assigns link addresses for each mobile station and registers them in the link management table. The numbered link address is then sent back to the mobile station as a service response. In further service requests, the assigned link address is used.

When an upper layer server receives a service request, it determines what service provision area matches the location of the mobile station. Furthermore, it checks the linking between the link address registered in the link management table and the service provision area and determines entrance into and exit from the service provision area. The results of this determination are sent to the mobile station as a service response.

The ITS-ASL of the upper layer server and the mobile station send notifications to their respective transmitting station applications. When a mobile station starts a service request, a communication registration notification is sent, when a mobile station enters a service provision area a communication connection notification is sent, and when a mobile station exits a service provision area a communications disconnection notification is sent.

Figure D.7-3 to Figure D.7-9 show examples of transmission connection management procedures. First, Figure D.7-3 and Figure D.7-4 show examples of mobile stations starting service requests. These examples show the processes, in order, when requests are issued from outside a service provision area and when they are issued from inside a service provision area. Next, Figure D.7-5 to Figure D.7-8 show examples of the processes, in order, when mobile stations move from outside service provision areas into service provision areas, when they continue driving within service provision areas, and when they move from inside service provision areas to outside service provision areas. Figure D.7-9 then shows an example of multiple mobile stations entering the same service provision area.

Figure D.7-3 shows an example of a mobile station starting a service request while outside a service provision area. In this example, the upper layer server that receives the service request assigns and registers a link address. The registered link address is not linked to a service provision area. The upper layer server and mobile station generate communication registration notifications.

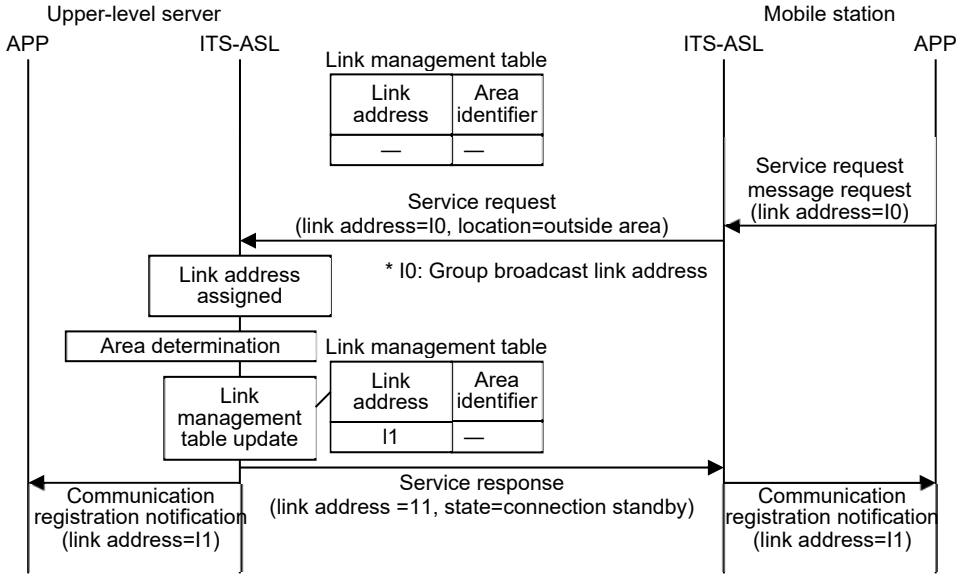


Figure D.7-3 Example of Transmission Connection Management Procedure
(Service Request Started Outside Service Provision Area)

Figure D.7-4 shows an example of a mobile station starting a service request while inside a service provision area. In this example, the upper layer server that receives the service request assigns and registers a link address. The registered link address is linked to the service provision area in which the mobile station is driving. The upper layer server and mobile station generate communication connection notifications for the service provision area in which the mobile station is driving.

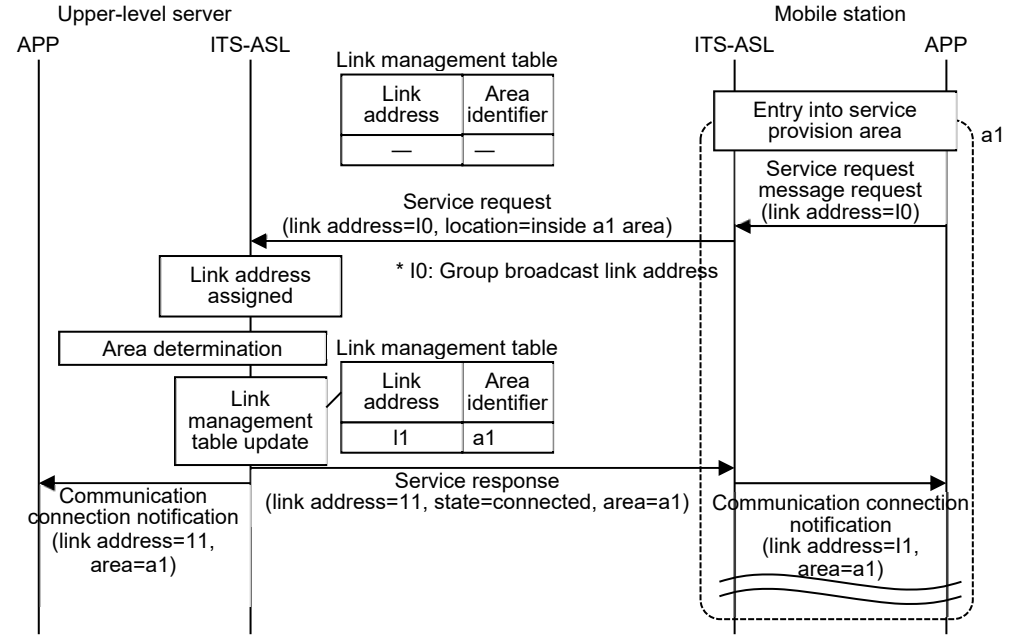


Figure D.7-4 Example of Transmission Connection Management Procedure
(Service Request Started Inside Service Provision Area)

Figure D.7-5 shows an example of a mobile station starting a service request while outside a service provision area and then entering the service provision area. In this example, the registered link address is linked to the service provision area into which the mobile station enters. The upper layer server and mobile station generate communication connection notifications for the service provision areas into which the mobile station enters.

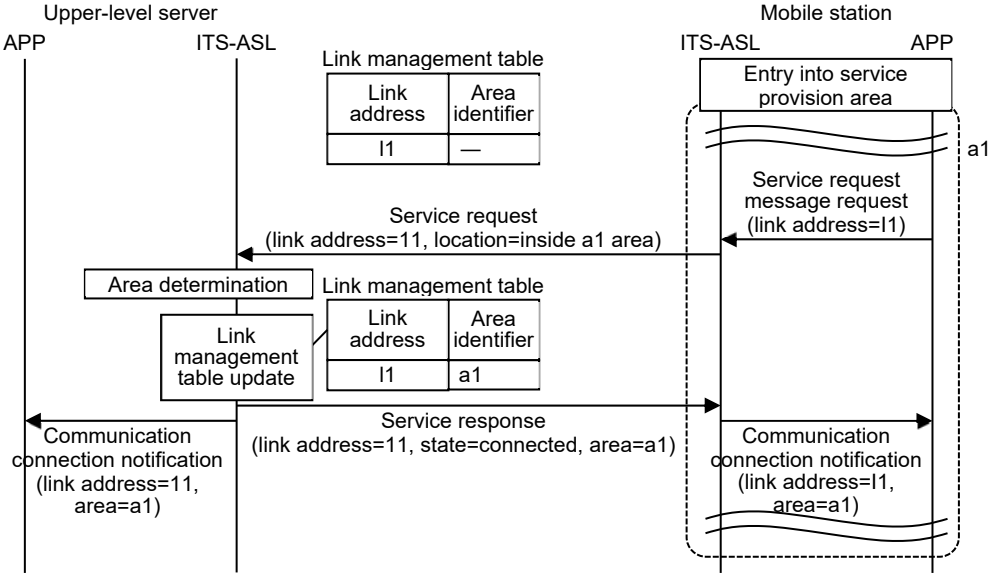


Figure D.7-5 Example of Transmission Connection Management Procedure
(Entry into a Single Service Provision Area)

Figure D.7-6 shows an example of a mobile station entering multiple overlapping service provision areas. In this example, the registered link address is additionally linked to the service provision areas into which the mobile station enters. The upper layer server and mobile station generate communication connection notifications for the service provision areas into which the mobile station enters.

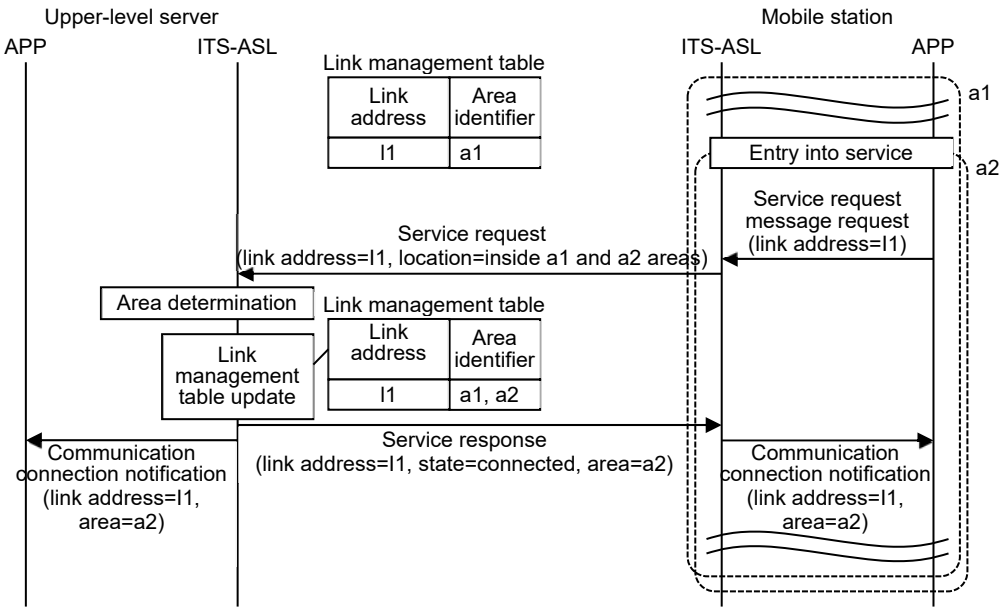


Figure D.7-6 Example of Transmission Connection Management Procedure
(Entry into Multiple Service Provision Areas)

Figure D.7-7 shows an example of a mobile station continuing to drive within a service provision area. In this example, the link between the registered link address and the service provision area is not updated. Neither the upper layer server nor the mobile station issue communication connection or communication disconnection notifications.

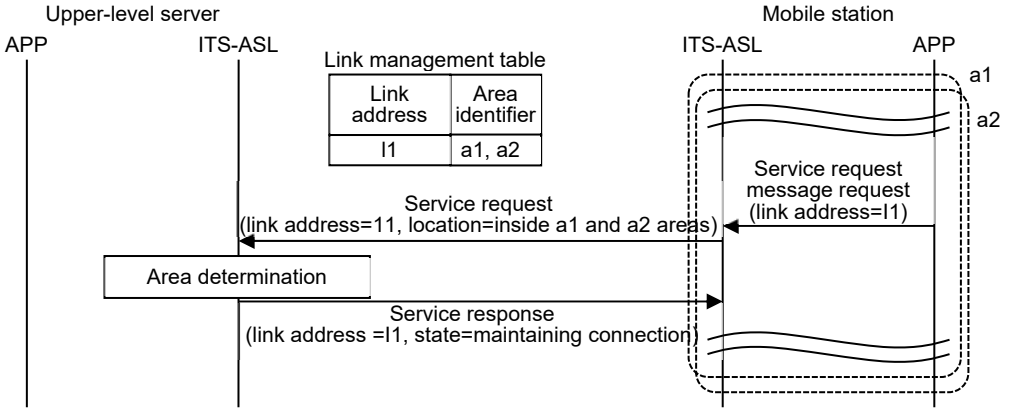


Figure D.7-7 Example of Transmission Connection Management Procedure
(Driving within a Service Provision Area)

Figure D.7-8 shows an example of a mobile station exiting from a service provision area. In this example, the link between the registered link address and the service provision area from which the mobile station exits is deleted. The upper layer server and mobile station generate communication disconnection notifications for the service provision area from which the mobile station exits.

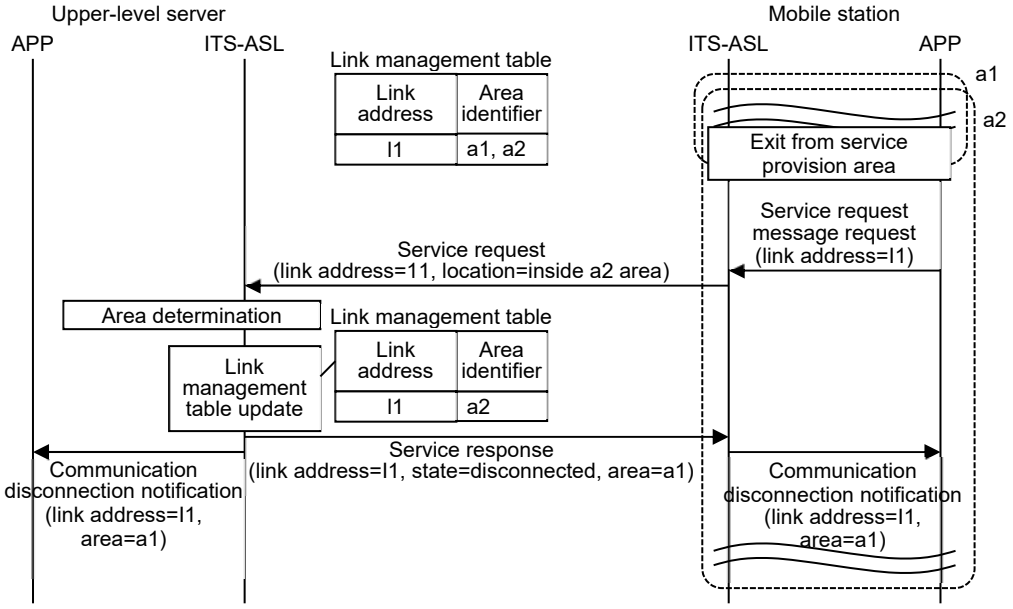


Figure D.7-8 Example of Transmission Connection Management Procedure (Exiting from a Service Provision Area)

Figure D.7-9 shows an example of multiple mobile stations entering the same service provision area. In this example, the same service provision area is linked to multiple link addresses.

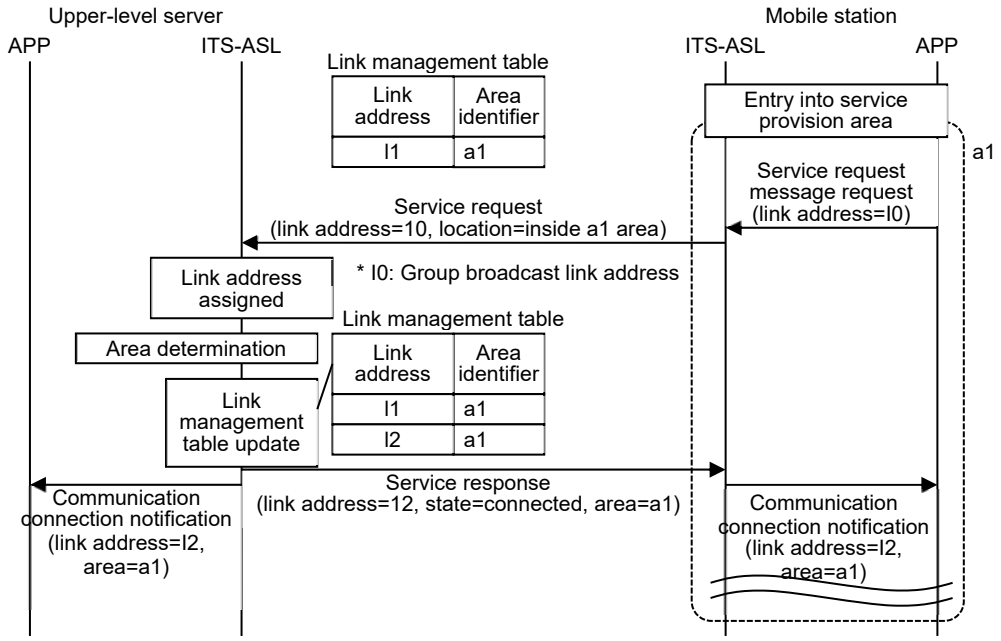


Figure D.7-9 Example of Transmission Connection Management Procedure
(Entry of Multiple Mobile Stations into Same Service Provision Area)

(c) Data transmission service process

In data transmission service processing from the upper layer server to the mobile station, the upper layer server's application first sends a data transmission request to the transmitting station's ITS-ASL. The ITS-ASL registers the content of the request in the transmission list. In this state, if the mobile station's ITS-ASL sends a service request to the upper layer server's ITS-ASL, the upper layer server's ITS-ASL decides on the data to send based on the specified link address and the mobile station's location. It then sends the data to the mobile station's ITS-ASL as a service response. The mobile station's ITS-ASL sends a data arrival notification containing the received data to the transmitting station's application.

Figure D.7-10 to Figure D.7-15 show examples of the procedures for data transmission service processing from the upper layer server to the mobile station.

First, Figure D.7-10 to Figure D.7-12 show examples of the processes when a mobile station drives without crossing a service provision area boundary. Respectively, they show the processes when a mobile station drives outside a service provision area, when it drives within a single service provision area, and when it drives within multiple overlapping service provision areas. Next, Figure D.7-13 and Figure D.7-14 show examples of the processes when a mobile station drives across a service provision area boundary. Respectively, they show the processes when a mobile station moves from outside a service provision area into the service provision area and when a mobile station moves from inside a service provision area to outside the service provision area. Furthermore, Figure D.7-15 shows an example of an upper layer server stopping data transmission.

Figure D.7-10 Figure D.7-10

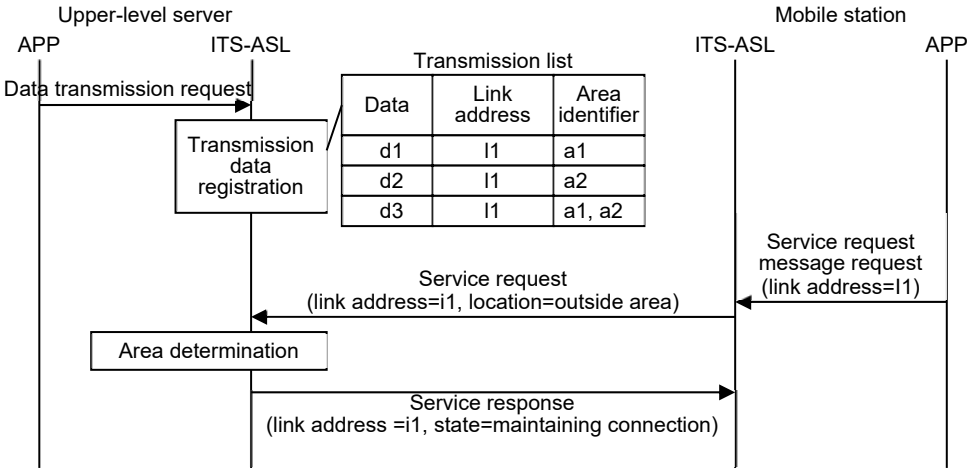


Figure D.7-10 Example of Transmission Service Process Procedure
(Driving Outside Service Provision Area)

Figure D.7-11 shows an example of a mobile station driving inside a single service provision area. In this example, there is data registered in the transmission list with a specified service provision area that matches the mobile station's location, so the upper layer server sends both maintaining connection information and the corresponding data.

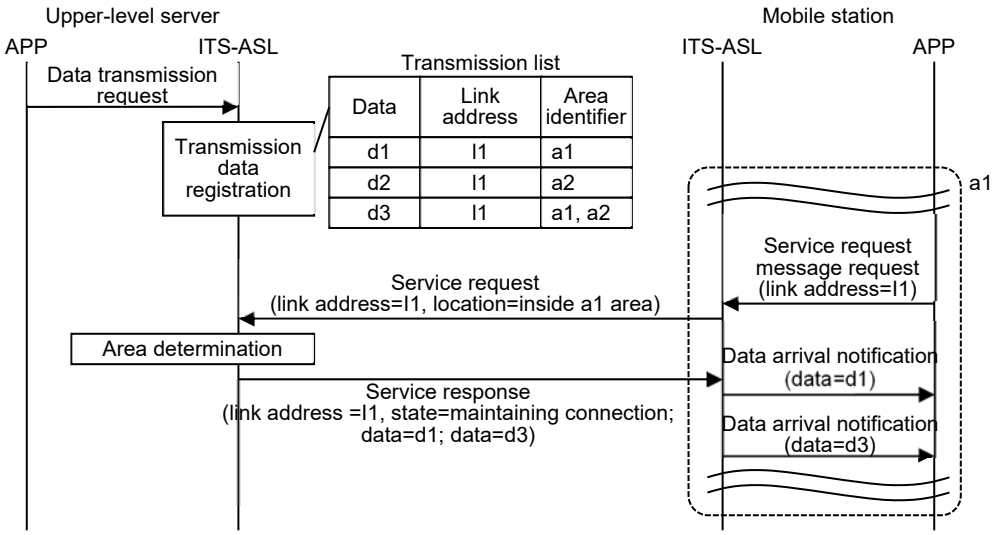


Figure D.7-11 Example of Transmission Service Process Procedure
(Driving Inside Single Service Provision Area)

Figure D.7-12 shows an example of a mobile station driving inside multiple overlapping service provision areas. In this example, the data to be transmitted is determined for all service provision areas that match the mobile station's location, and the upper layer server sends the corresponding data, without overlapping it.

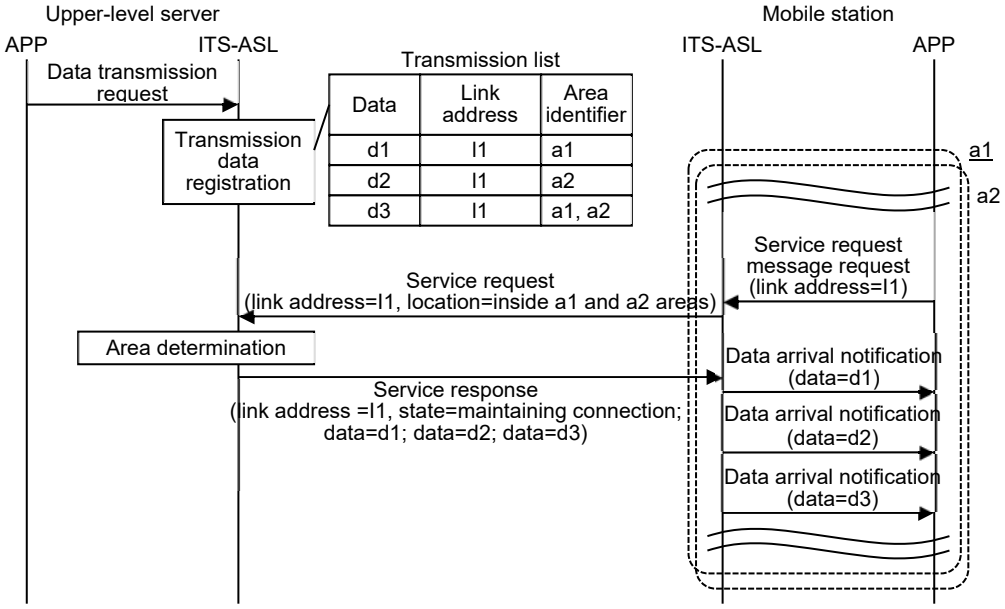


Figure D.7-12 Example of Transmission Service Process Procedure
(Driving Inside Overlapping Service Provision Areas)

Figure D.7-13 shows an example of a mobile station entering a service provision area. In this example, the upper layer server sends corresponding data together with connection information. The upper layer server's and mobile station's ITS-ASLs send a transmission connection notification to their transmitting stations' applications before providing data arrival notification.

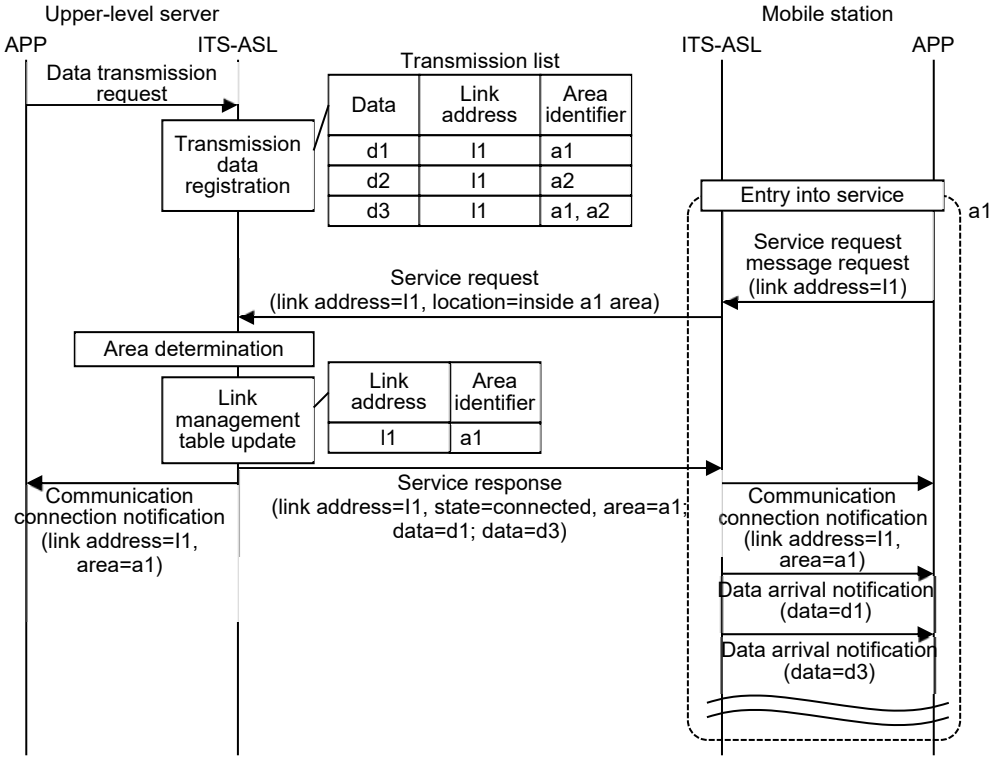


Figure D.7-13 Example of Transmission Service Process Procedure
(Entering Service Provision Area)

Figure D.7-14 shows an example of a mobile station exiting a service provision area. In this example, the upper layer server sends corresponding data together with disconnection information. The upper layer server's and mobile station's ITS-ASLs send a transmission disconnection notification to their transmitting stations' applications before providing data arrival notification.

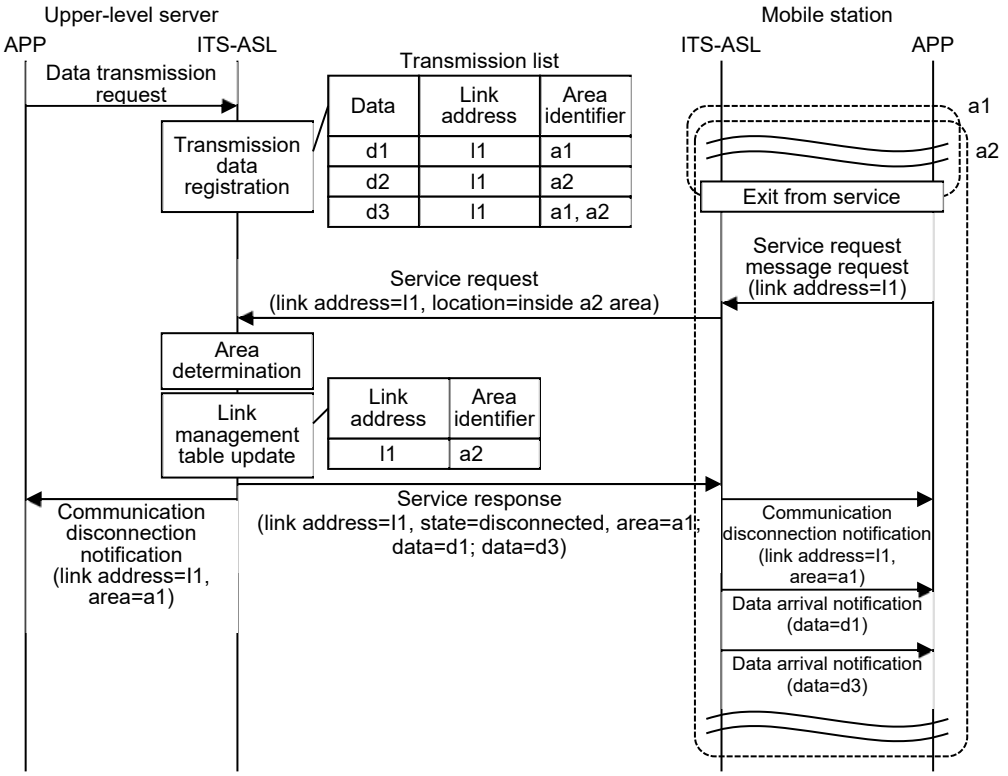


Figure D.7-14 Example of Transmission Service Process Procedure
(Exiting Service Provision Area)

Figure D.7-15 shows an example of an upper layer server stopping data transmission. In this example, the upper layer server's application sends a data stop request to ITS-ASL. After this, the data specified here is not sent to the mobile station.

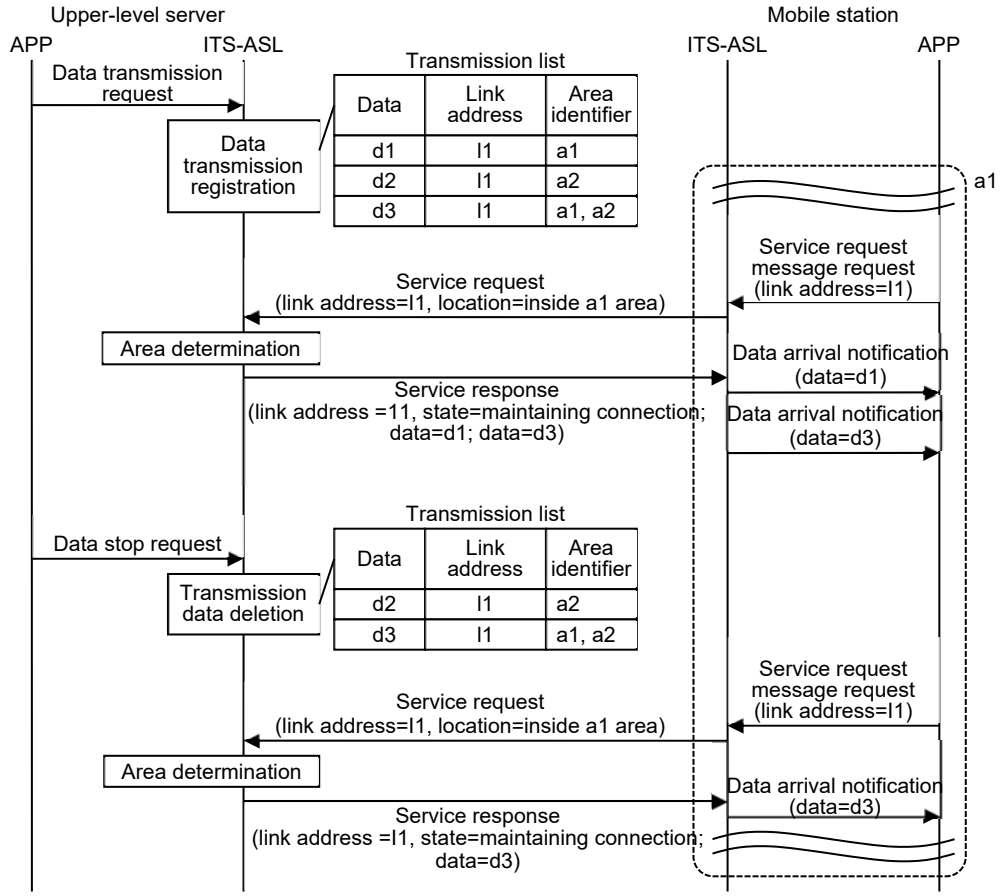


Figure D.7-15 Example of Transmission Service Process Procedure (Stopping Data Transmission)

[BLANK]

Annex E: Study of use of ARIB STD-T120

This section indicates the results of a study of the use of ITS-ASL with ARIB STD-T120 as its lower communication layer. ARIB STD-T120 includes both direct communications for narrow area communications (PC5) and communication via base stations for cellular networks (Uu). The communications protocols applied to upper layers differ between them.

For direct communications (PC5), V2X applications are situated on a layer above IP/Non-IP layers. IEEE 1609.3 (WSMP) and EN 302-636-4-1 (GeoNetwork) are applied to non-IP layers. Therefore, as Figure E-1 shows, ITS-ASL can be supported by performing the same processing as is used when applying ITS FORUM RC-005/IEEE 802.11 as the lowest layer,.

For communications via base stations (Uu), V2X applications are situated above IP layers. Therefore, as Figure E-2 shows, ITS-ASL can be supported by performing the same processing as is used when applying ARIB STD-T104 as the lowest layer.

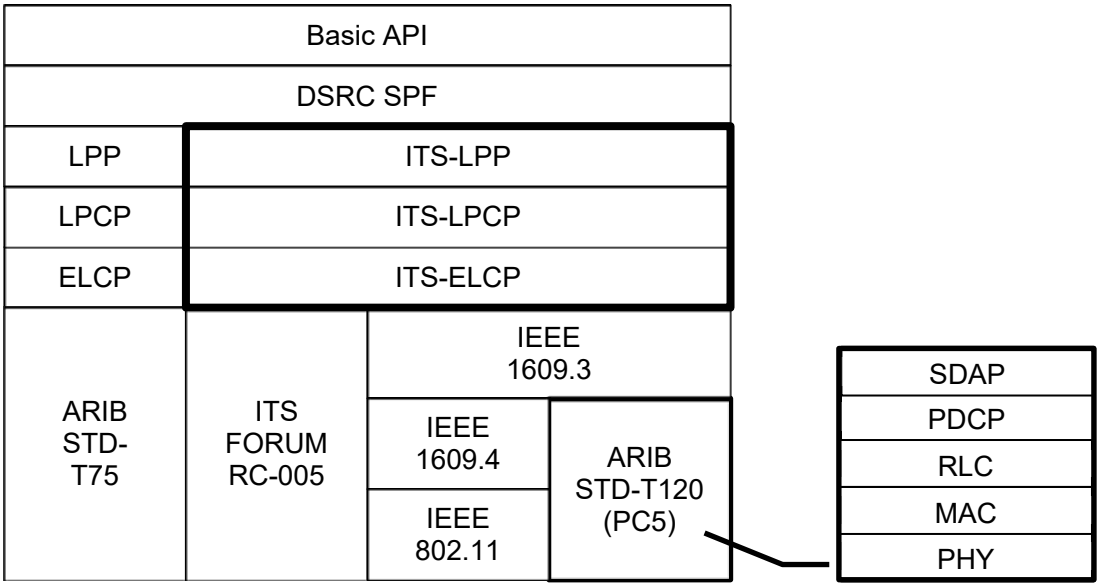


Figure E-1 Protocol configuration when applying ARIB STD-T120 (PC5)

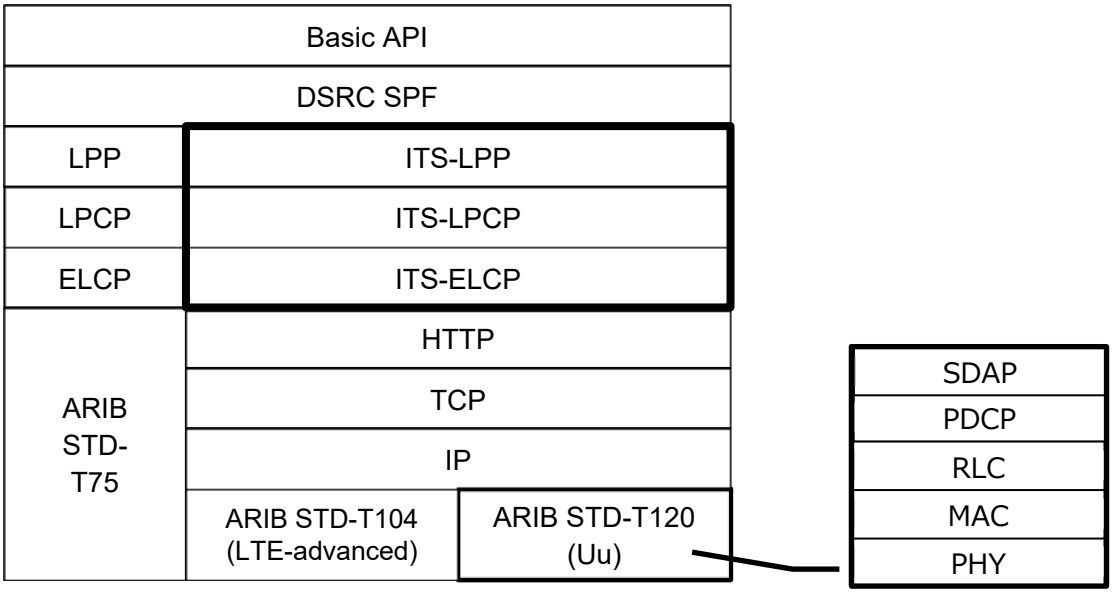


Figure E-2 Protocol configuration when applying ARIB STD-T120 (Uu)

Annex F: Study of use of ARIB STD-T109

ARIB STD-T109 uses Layer 7 BaseSationBroadcastData requests or MobileSationBroadcastData requests to send data to a remote station (see Figure F-1). ARIB STD-T109 specifies the LinkAddress parameter of the BaseSationBroadcastData request or the MobileSationBroadcastData request as the destination terminal's link address. Furthermore, the LinkAddress parameter of the Layer 7 BaseSationBroadcastData notification or the MobileSationBroadcastData notification is used as the sending terminal's link address. Therefore, when receiving data, the sender's link address can be identified, but the destination link address cannot be identified.

The destination link address can be set in the ITS-ASL-ELCP ASL-PDU communication control information and the receiving station can identify the data destination from the ITS-ASL, so it can determine if the destination link address is the same as the station's own link address.

When ITS-ASL-ELCP receives a SendDataUnit.request from ITS-LPCP, it creates an ASL-PDU based on the received ASL-SDU, stores the linkAdress parameter in the communication control information destinationLinkAddress, and uses the Layer 7 BaseSationBroadcastData requests or MobileSationBroadcastData request to transmit to the remote station (see Figure F-1).

When ITS-ASL-ELCP receives a Layer 7 BaseSationBroadcastData notification or a MobileSationBroadcastData notification, it extracts the ASL-SDU from the received ASL-PDU and determines if the communication control information in the ASL-PDU has the same destinationLinkAddress as the station's own link address. If they do not match, it discards the ASL-PDU. If the link addresses do match, it uses the SendDataUnit.indication to transmit to the ITS-LPCP (see Figure F-2).

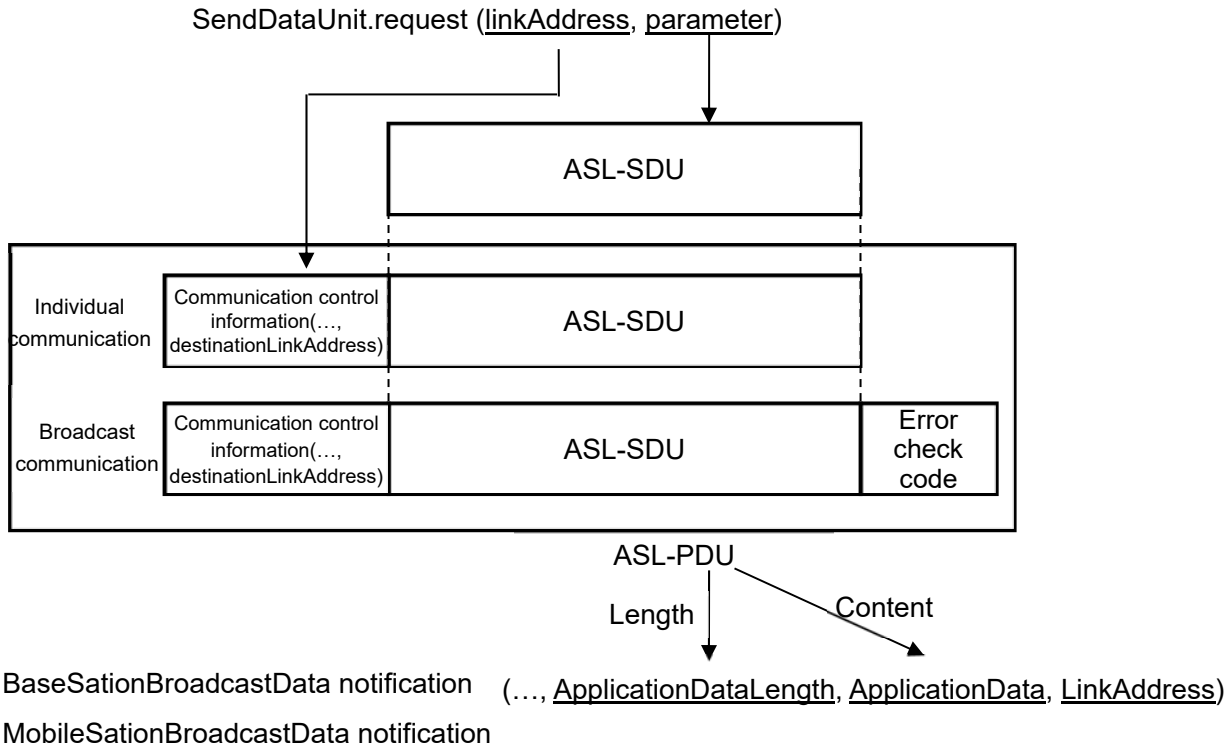


Figure F-1 Mapping of sublayer primitive for SendDataUnit.request

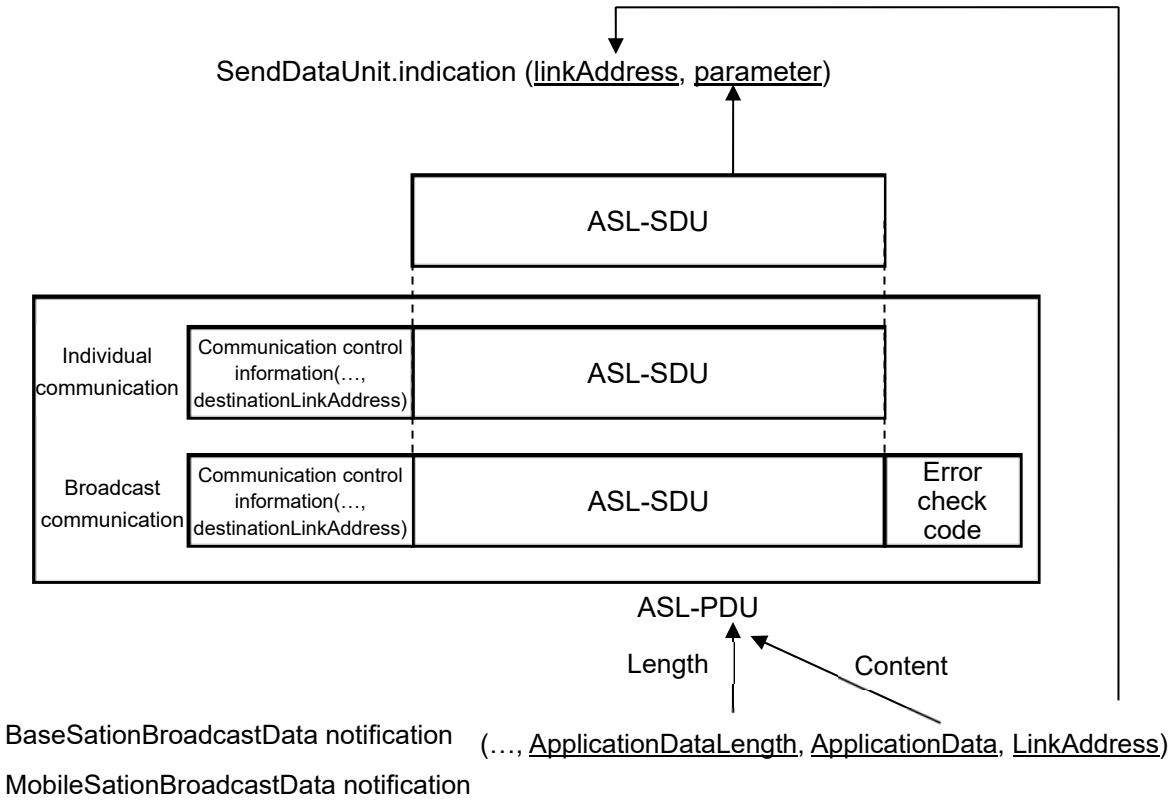


Figure F-2 Mapping of sublayer primitive for SendDataUnit.indication

Annex G: ITS-ASL-ELCP patterns for different lower communications layers

ITS-ASL-ELCP, which is situated at the adapter layer of ITS-ASL, supports diverse lower communication layers, so its functions are defined differently depending on the lower communication layer. This annex indicates the functions provided for each lower communication layer.

Table G-1 shows patterns of main ITS FORUM RC-014 functions. Also, as Figure G-1 shows, functions can be categorized into three patterns depending on the lower communication layer.

- Pattern #1: When using non-IP WSMP for the lower communication layer (RC-005/802.11/T120(PC5))
- Pattern #2: When using IP (HTTP) for the lower communication layer (T104/T120(Uu))
- Pattern #3: When using STD-T109 for the lower communication layer (T109)

This guideline is not intended for simultaneous use of patterns #1, #2, and #3.

Use only one of these patterns.

Table G-1 Classification of functions by standard

Function specification	Pattern type
2.2 Identification of destination application	Three patterns, different for: (1) RC-005/802.11/T120(PC5) (2) T104/T120(Uu) (3) T109
2.3 Link address	Same for all, regardless of lower communication layer
3.1.2 Extended communication control	—
3.1.2.1 Communication Service Interface	Two patterns, different for: (1) RC-005/802.11/T120(PC5)/T109 (2) T104/T120(Uu)
3.1.2.2 Protocol Data Unit (PDU)	Two patterns, different for: (1) RC-005/802.11/T120(PC5)/T104/T120(Uu) (2) T109
3.1.2.3 Procedure Elements of the Extended Communication Control Interface with lower communication layer	Three patterns, different for: (1) RC-005/802.11/T120(PC5) (2) T104/T120(Uu) (3) T109
3.1.2.4 ITS-ASL-ELCP Procedure	
3.1.3 Communication Control Management	—

3.1.3.1 Management Service Interface	Two patterns, different for: (1) RC-005/802.11/T120(PC5)/T109 (2) T104/T120(Uu)
3.1.3.2 Protocol Data Unit (PDU)	Same for all, regardless of lower communication layer
3.1.3.3 Communication Connection Management	
3.2 ITS-LPCP	
3.3 ITS-LPP	

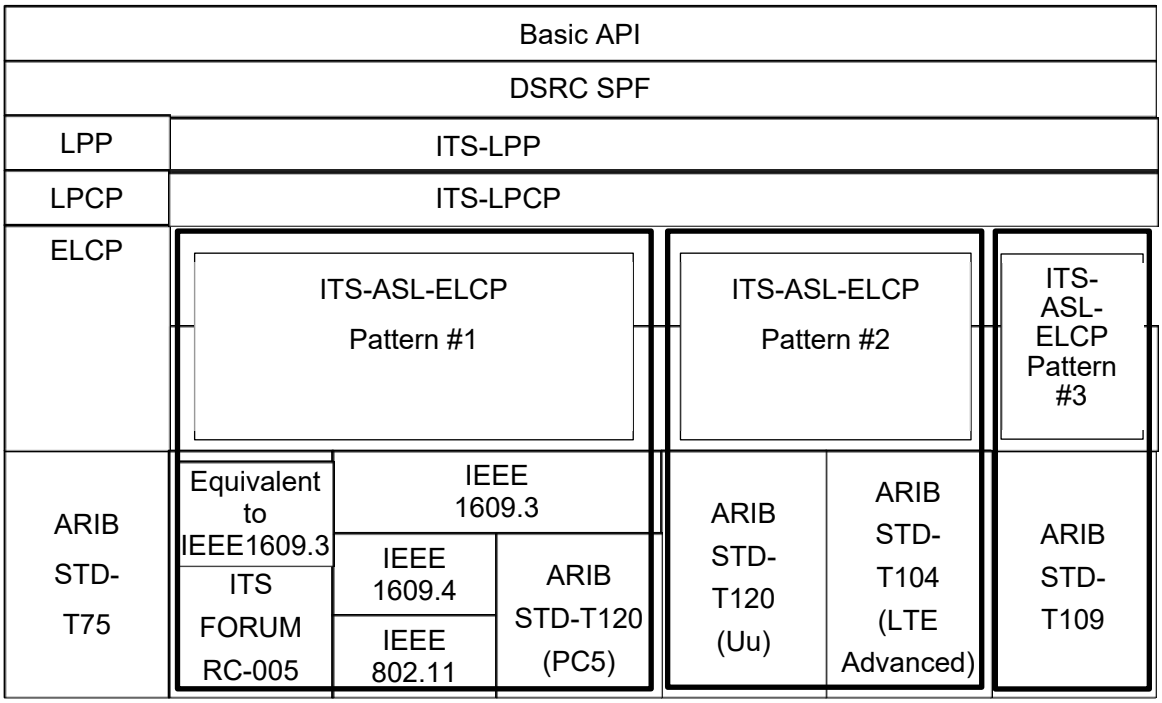


Figure G-1 Protocol stack patterns