TS 101 658 V1.1.1 (1999-04)

Technical Specification

Terrestrial Trunked Radio (TETRA); Digital Advanced Wireless Service (DAWS); Logical Link Control (LLC) service description

Reference

DTS/TETRA-04033 (fco00icr.PDF)

Keywords

data, TETRA

ETSI

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

Office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet

secretariat@etsi.fr
Individual copies of this ETSI deliverable
can be downloaded from
http://www.etsi.org
If you find errors in the present document, send your
comment to: editor@etsi.fr

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Project Terrestrial Trunked Radio (TETRA).

An overview of the requirements for DAWS can be found in TR 101 156.

1 Scope

The present document specifies the service requirements for the Digital Advanced Wireless Service (DAWS) Logical Link Control (LLC) layer. The present document provides a conceptual architecture useful for specifying requirements but is not intended to imply a particular implementation. The present document contains preliminary LLC protocol requirements which will be moved into the formal LLC protocol specification document (Part 6) when it is drafted.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, subsequent revisions do apply.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1] TR 101 156: "Terrestrial Trunked Radio (TETRA); Technical requirements specification for Digital Advanced Wireless Service (DAWS)".

[2] TS 101 659: "Terrestrial Trunked Radio (TETRA); Digital Advanced Wireless Service (DAWS); Medium Access Control (MAC); Requirements Specification".

[3] Void.

[4] Void.

[5] IETF RFC 2211: "Specification of the Controlled-Load Network Element Service".

[6] IETF RFC 2205: "Resource Reservation Protocol (RSVP) – Version 1 Functional Specification".

[7] IETF RFC 2215: "General Characterization Parameters for Integrated Service Network Elements".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

base station: piece of equipment providing simultaneous, bi-directional network access to mobile stations.

downlink: general term meaning "from the base station to the mobile station".

flow: sequence of data packets originating from a single source and addressed to the same destination for which special handling by intervening routers is desired.

mobile station: piece of equipment able to create and consume data but only having network access via a base station.

protocol data unit: set of parameters and/or data passed from peer to peer by a protocol primitive.

protocol instance: two protocol processes which exchange messages in order to transfer data from one protocol process to the other.

protocol primitive: request, response, or informative message sent from peer to peer.

protocol process: entity created to manage one end of a peer-to-peer protocol. For unidirectional data flows, a protocol process can be further described as either a sender process or a receiver process.

service data unit: set of parameters and/or data passed between adjacent layers by a service primitive.

service primitive: request, response, or informative message sent between adjacent layers.

uplink: general term meaning "from the mobile station to the base station".

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACK	Acknowledged
BEP	Best-Effort Plus
BS	Base Station
DAWS	Digital Advanced Wireless Service
DL	Downlink
GW	Gateway
IP	Internet Protocol
LLC	Logical Link Control
LLC_ADM	LLC Admission Control Service
LLC_REG	LLC Registration Service
LLC_TPT	LLC Transport Service
LPDU	LLC Protocol Data Unit
MAC	Medium Access Control
MS	Mobile Station
MSH	Mobile Station Handle
MSI	Mobile Station Identifier
NWK	Network
PDU	Protocol Data Unit
QOS	Quality Of Service
RSVP	Resource Reservation Protocol
SAP	Service Access Point
SDU	Service Data Unit
SW	Switch
UL	Uplink

4 Introduction

The DAWS protocol architecture is provided in TR 101 156 [1]. The Logical Link Control (LLC) provides services to the network layer (NWK) and requests services from the Medium Access Control (MAC) TS 101 659 [2]. The present document provides the requirements the LLC service has to satisfy to operate successfully within a Digital Advanced Wireless Service (DAWS) network. As described in TR 101 156 [1], LLC functionality may be distributed across several DAWS nodes. The following prefixes will be used to specify the scope of a requirement:

- LLC - the requirement applies to the LLC in general;
- GW_LLC - the requirement applies to Gateway functionality;
- SW_LLC - the requirement applies to Switch functionality;
- BS_LLC - the requirement applies to Base Station functionality;
- MS_LLC - the requirement applies to Mobile Station functionality.

Figure 1 shows the architecture of the LLC for the minimum complexity DAWS network described in TR 101 156 [1]. The network layer (NWK) accesses LLC services via service access points (SAPs) A and B. LLC_SAP_A is for data transfer service primitives and LLC_SAP_B is for local control and status service primitives, including RSVP IETF RFC 2205 [6] operations.

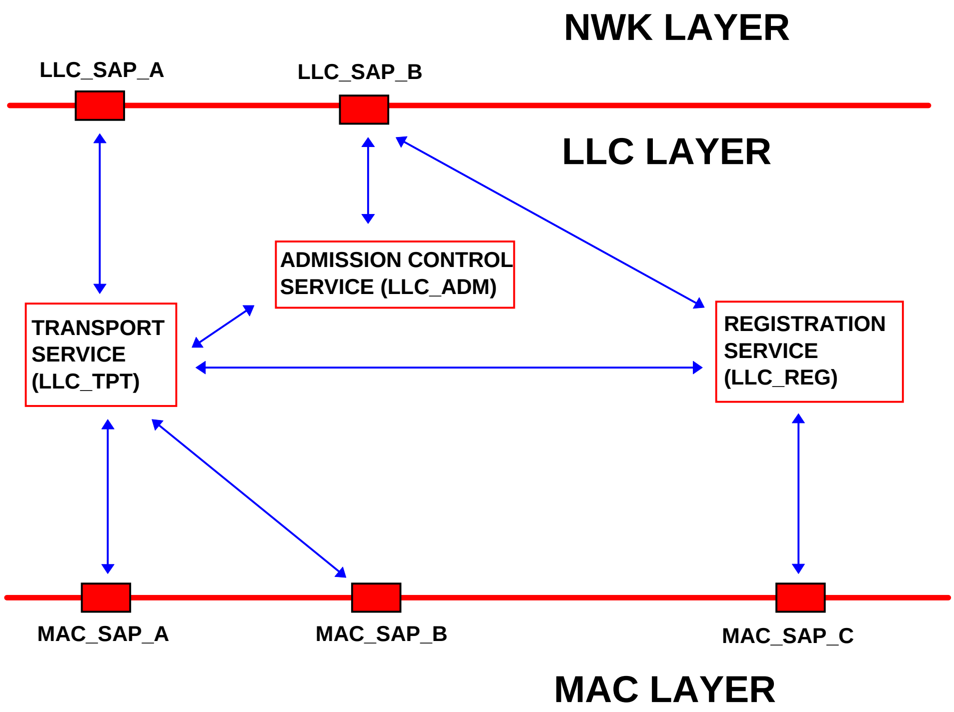

Figure 1: DAWS LLC Architecture

The LLC accesses MAC services via service access points A, B, and C. MAC_SAP_A is for service primitives relating to PDU transfers using an unacknowledged protocol; MAC_SAP_B is for service primitives relating to PDU transfers using an acknowledged protocol; and MAC_SAP_C is for local control and status service primitives.

Requirements for the registration, admission control, and transport services are provided in clauses 5, 6, and 7. Service primitives and associated service data units are provided in clause 8. Annex A discusses the IPv4 to IPv6 transition.

5 Registration Services

The LLC registration service (LLC_REG) supports BS registration, MS registration, and de-registration operations.

5.1 Base station registration

Every DAWS GW is assigned a globally unique identifier called the GWI. The GWI shall be assigned when the GW is manufactured and shall not be dynamically alterable. The GW address space shall be administered by an industry body to prevent GWI address duplication among manufacturers.

Every DAWS BS is assigned a globally unique identifier called the BSI. The BSI shall be assigned when the BS is manufactured and shall not be dynamically alterable. The BSI address space shall be administered by an industry body to prevent BSI address duplication among manufacturers.

Before a DAWS BS can provide wireless access service to a MS, the BS must register with a DAWS GW. BS registration usually occurs immediately after BS power-on, and is composed of the following steps:

1) BS_LLC sends a BS registration request message toward the GW via the next upstream switch;

2) SW_LLC adds the binding (BSI, output_interface) to its routing table, and forwards the BS registration request toward the GW. For simplicity, this example will assume only one upstream switch, so the next node will be the GW;

3) GW_LLC adds the binding (BSI, output_interface) to its routing table, and sends a BS registration response toward the BS. The BS registration response contains the GWI of the GW;

4) SW_LLC forwards the BS registration response towards the BS;

5) BS_LLC begins providing wireless access to MS within its cell. The BS regularly broadcasts the GWI of its serving GW and its own BSI in a system information message to all MS within the serving cell. Mobile Stations use GWI information during the hand-over procedure to differentiate between intra-network and inter-network hand-overs.

The size of the routing table within a DAWS switch is independent of the number of MS served by the DAWS network. The routing table size is proportional to the number of BS in the network.

A DAWS network shall automatically correct its routing table entries if the network topology changes. Manual intervention by the system administrator shall not be required.

5.2 Mobile station registration

5.2.1 Cell selection

When an MS is powered on, MS_LLC_REG shall issue a request to MS_MAC_REG to do a scan of available cells and report the results. MS_LLC_REG shall select the best cell and instruct MS_MAC_REG to camp on the cell. MS_LLC_TPT is then able to exchange PDUs with BS_LLC_TPT using the unacknowledged protocols.

5.2.2 Registration

After cell selection, MS_LLC automatically registers with BS_LLC. MS_LLC registration with a BS involves the following steps:

1) MS_LLC obtains the BSI of the BS and GWI of the GW serving the BS from a system information message;

2) MS_LLC sends a registration request to BS_LLC, providing its MSI;

3) BS_LLC generates a MSH for the MS and adds the binding (MSI, MSH) to its registration table;

4) BS_LLC generates two protocol instance identifiers for the ACK_BE_UL and a ACK_BE_DL protocol instances;

5) BS_LLC creates ACK_BE_UL and ACK_BE_DL protocol processes in BS_MAC;

6) BS_LLC sends a registration response message containing the MSH and protocol instance identifiers to MS_LLC;

7) MS_LLC stores its assigned MSH for future uplink and downlink signalling with the BS;

8) MS_LLC creates ACK_BE_UL and ACK_BE_DL protocol processes in MS_MAC;

9) MS_LLC sends a service indication message to MS_NWK indicating that LLC registration is complete, providing the registration triplet (GWI, BSI, MSI).

The MS is now able to exchange PDUs with the BS using the two best-effort acknowledged protocols ACK_BE_UL and ACK_BE_DL. MS_NWK can create additional protocol instances by the procedures described in clause 7.

5.2.3 Service interruption

When MS_LLC receives a service interruption indication from MS_MAC, MS_LLC sends a service indication interruption to MS_NWK. MS_NWK will report the service interruption to higher layers in the protocol stack and will discard uplink traffic until service is restored.

When MS_MAC indicates that service is restored, MS_LLC will re-register with BS_LLC and send a service indication message to MS_NWK indicating that LLC service is available. The triplet (GWI, BSI, MSI) reported during the prior registration will be reported again to MS_NWK, indicating that the serving BS and GW have not changed. If the duration of the LLC service interruption is sufficiently short, MS_NWK will not need to re-register with GW_NWK.

5.2.4 Intra-network hand-over

For an intra-network hand-over, the MS loses services with a BS and re-establishes service with another BS within the same DAWS network (i.e., both the prior and new BS have registered the same GW).

When MS_LLC receives a service interruption indication from MS_MAC, MS_LLC sends a service indication interruption to MS_NWK. MS_NWK will report the service interruption to higher layers in the protocol stack and will discard uplink traffic until service is restored.

When MS_MAC indicates that service is restored, MS_LLC will register with the new BS and send a service indication message to MS_NWK indicating that LLC service is available. The BSI reported to MS_NWK will be different from the BSI reported during the prior registration, reflecting service by a new BS. The GWI will not change. If the duration of the LLC service interruption is sufficiently short, MS_NWK will not need to re-register with GW_NWK.

5.2.5 Inter-network hand-over

For an inter-network hand-over, the MS loses services with a BS and re-establishes service with another BS within a different DAWS network (i.e., the prior and new BS have registered with different GWs).

When MS_LLC receives a service interruption indication from MS_MAC, MS_LLC sends a service indication interruption to MS_NWK. MS_NWK will discard uplink traffic until service is restored and report the service interruption to higher layers in the protocol stack.

When MS_MAC indicates that service is restored, MS_LLC will register with the new BS and send a service indication message to MS_NWK indicating that LLC service is available. The BSI and GWI reported to MS_NWK will be different than the BSI and GWI reported during the prior registration, reflecting service by a new BS and GW. MS_NWK will be required to perform network-level registration with GW_NWK regardless of the duration of the service interruption.

5.3 De-registration

A registration-related state created in a DAWS node has an associated de-registration timer. If the de-registration timer expires, the registration state is deleted. The following registration states have de-registration timers:

1) GW_LLC: the (BSI, output_interface) binding;
2) SW_LLC: the (BSI, output_interface) binding;
3) BS_LLC: the (MSI, MSH) binding;
4) MS_LLC: the MSH.

A de-registration timer is automatically restarted whenever traffic to or from the MS passes through the node.

A timer can also be restarted upon receipt by the managing entity of a "timer restart" message. BS_LLC will issue a "timer restart" message when necessary to maintain binding states between itself and the GW. MS_LLC will issue a "timer restart" message when necessary to maintain binding states between itself and a serving BS.

When BS_LLC deletes a (MSI,MSH) binding, it also deletes all protocol processes in the BS_MAC layer dedicated to flows between the BS and MS. When MS_LLC deletes its MSH, it also deletes all protocol processes in the MS_MAC layer.

6 Transport Services

The LLC transport service (LLC_TPT) provides link layer address resolution, packet classification, and traffic measurement, policing, and shaping.

6.1 Link layer address resolution

Link layer address resolution is the process of translating a network level address into a link layer address (the MSH), and involves the following steps for downlink PDUs:

1) GW_NWK translates the network level address (the MS care-of address) into a BSI and MSI. GW_NWK sends the PDU with attached BSI and MSI to GW_LLC for routing through intermediate DAWS switches (if any) to a particular BS;

2) BS_LLC translates the MSI into the MSH for downlink packet transfer of the PDU over the wireless link.

For an uplink packet, MS_LLC uses the MSH assigned during MS_LLC registration for the uplink transfer of the PDU over the wireless link.

6.2 Packet classification

At any given time, there will be multiple protocol instances in place between a BS and a particular MS. Packet classification is the process of selecting a particular protocol process to handle the transfer of a PDU to a peer entity. BS_LLC_TPT shall use the MSH, PDU destination address, and PDU destination port to obtain the protocol instance identifier for downlink IPv4 packets. BS_LLC_TPT shall use the MSH and PDU flow label to obtain the protocol instance identifier for downlink IPv6 packets. For uplink packets, MS_LLC_TPT shall use the same packet classification procedure as BS_LLC_TPT, except that MS_LLC_TPT does not need to index based on the MSH because at any time the MS has at most one MSH.

The best-effort downlink or uplink protocol instance is the default method of transferring PDUs if no controlled-load instance has been created for a particular flow.

LLC_TPT may also support best-effort plus (BEP) service. BEP analyses the flow of best-effort PDUs and detects when there is a sufficient flow directed to a single MS to justify a persistent bandwidth reservation. LLC_TPT then establishes a persistent bandwidth reservation to handle the best-effort traffic. In contrast to RSVP controlled-load QOS, which has network-level scope, BEP has only link-level scope and is intended to optimize bandwidth utilization. The definition of requirements for BEP is for further study.

6.3 Traffic measurement, policing, and shaping

LLC_TPT shall monitor incoming controlled-load flows for compliance with the associated IETF RFC 2215 [7]. As defined in IETF RFC 2211 [5], the following requirements apply to MAC_TPT handling of non conformant controlled-load flows:

1) MAC_TPT shall continue to provide the contracted QOS to conformant controlled-load flows;

2) MAC_TPT should prevent non conformant controlled-load flows from unfairly impacting the handling of best-effort traffic;

3) MAC_TPT shall attempt to forward the excess traffic of non conformant controlled-load flows on a best-effort basis if resources are available and requirements 1) and 2) are satisfied.

7 Admission Control Services

The LLC Admission Control Service (LLC_ADM) processes resources reservation requests. For the purposes of this clause, the RSVP entity will be assumed to exist within the network layer, although in reality it may be more appropriately placed in the transport layer. The MS implementations of RSVP will be specified by MS_NWK_RSVP. A resource reservation is always initiated by MS_NWK_RSVP regardless of the source of the RSVP RESV message (the MS or a correspondent host).

The MS LLC Admission Control Service (MS_LLC_ADM) interacts with MS_NWK_RSVP to establish and release resource reservations. MS_LLC_ADM will not reject a resource reservation request due to limited bandwidth but may reject a request for another reason, for example, lack of buffer space. If the reservation request is accepted by MS_LLC_ADM, then MS_LLC_ADM will perform signalling with BS_LLC_ADM to establish the reservation. BS_LLC_ADM shall consider available free bandwidth (and other factors to be specified) when arriving at an admission control decision. BS_LLC_ADM shall base admission control decisions on statistics calculated from traffic measurements (available from BS_LLC_TPT) as well as on calculations done with TSPEC parameters. Both MS_LLC_ADM and BS_LLC_ADM must accept a reservation request before it will be established. After a reservation request has been accepted, LLC_ADM exchanges service primitives with MAC_BWM in order to create and delete protocol processes which satisfy the reservation request.

DAWS does not specify requirements for the handling of reservation requests by the DAWS GW and intermediate switches.

The following clauses provide an overview of the procedures for reservation creation, deletion, and modification. All service requests are assumed to complete successfully.

7.1 Reservation creation procedure

The final result of the reservation creation procedure is a new protocol instance supplying the requested QOS to the specified flow.

1) MS_NWK_RSVP sends the **LLC_create_protocol_request** primitive to MS_LLC_ADM.
2) MS_LLC_ADM makes an admission control decision.
3) MS_LLC_ADM sends a protocol primitive to BS_LLC_ADM requesting resource reservation.
4) BS_LLC_ADM makes an admission control decision.
5) BS_LLC_ADM creates a protocol process in the MAC layer (identified by protocol_instance_ID) to handle the reservation. BS_LLC_ADM configures BS_MAC_BWM to allocate bandwidth to the MS for the flow.
6) BS_LLC_ADM sends a resource reservation response primitive to MS_LLC_ADM. The response primitive contains the new protocol_instance_ID.
7) MS_LLC_ADM creates a protocol process in the MAC layer (identified by protocol_instance_ID) to handle the reservation. MS_LLC_ADM configures MS_MAC_BWM to allocate bandwidth assigned to the MS for the flow.
8) MS_LLC_ADM sends the **LLC_create_protocol_confirm** primitive to MS_NWK_RSVP.

The BS and MS now each have a new protocol process in the MAC layer to support the new protocol instance. The packet classifier in the BS (for a downlink flow) or MS (for an uplink flow) is routing PDUs to the new instance and the bandwidth manager in the BS is allocating bandwidth to the MS for the flow.

7.2 Reservation deletion procedure

The reservation deletion procedure assumes that a reservation was made as specified in subclause 7.1. The reservation deletion procedure releases all resources associated with an existing reservation.

1) MS_NWK_RSVP sends the **LLC_delete_protocol_request** primitive to MS_LLC_ADM.
2) MS_LLC_ADM sends a protocol primitive to BS_LLC_ADM requesting reservation deletion.
3) BS_LLC_ADM tells BS_MAC_BSM to stop allocating bandwidth to the MS for the flow. BS_LLC_ADM deletes the protocol process from the MAC layer which handled the flow.
4) BS_LLC_ADM sends a reservation delete response primitive to MS_LLC_ADM.
5) MS_LLC_ADM deletes the protocol process in the MAC layer which handled the flow.
6) MS_LLC_ADM sends the **LLC_delete_protocol_confirm** primitive to MS_NWK_RSVP.

7.3 Reservation modification procedure, resource increase

The reservation modification procedure provided in this subclause assumes that a reservation was made as specified in subclause 7.1. This procedure increases the resources dedicated to a particular flow.

1) MS_NWK_RSVP sends the **LLC_modify_protocol_request** primitive to MS_LLC_ADM.
2) MS_LLC_ADM makes an admission control decision.
3) MS_LLC_ADM tentatively modifies the protocol process in the MAC layer which handles the flow.
4) MS_LLC_ADM sends a protocol primitive to BS_LLC_ADM requesting reservation modification.
5) BS_LLC_ADM makes an admission control decision.
6) BS_LLC_ADM modifies the protocol process in the MAC layer which handles the flow and/or BS_MAC_BWM.
7) BS_LLC_ADM sends a resource reservation response primitive to MS_LLC_ADM.
8) MS_LLC_ADM sends the **LLC_modify_protocol_confirm** primitive to MS_NWK_RSVP.

7.4 Reservation modification procedure, resource decrease

The reservation modification procedure provided in this subclause assumes that a reservation was made as specified in subclause 7.1. This procedure decreases the resources dedicated to a particular flow.

1) MS_NWK_RSVP sends the **LLC_modify_protocol_request** primitive to MS_LLC_ADM.
2) MS_LLC_ADM makes an admission control decision.
3) MS_LLC_ADM tentatively modifies the protocol process in the MAC layer which handles an uplink flow. For a downlink flow, go to the next step.
4) MS_LLC_ADM sends a protocol primitive to BS_LLC_ADM requesting reservation modification.
5) BS_LLC_ADM makes an admission control decision.
6) BS_LLC_ADM modifies the protocol process in the MAC layer which handles the flow and/or BS_MAC_BWM.
7) BS_LLC_ADM sends a resource reservation response primitive to MS_LLC_ADM.

8) MS_LLC_ADM tentatively modifies the protocol process in the MAC layer which handles a downlink flow. For an uplink flow, go to the next step.

9) MS_LLC_ADM sends the **LLC_modify_protocol_confirm** primitive to MS_NWK_RSVP.

8 Service Primitives

8.1 Primitive Definitions

8.1.1 LLC_transfer_request

Table 1

LLC_transfer_request	
Usage	MS
Source	NWK Layer
Destination	LLC Layer
Service Access Point	A
Multiple Outstanding	No
SDU Parameters	*NPDU*

This primitive is used by the MS NWK layer for the uplink transfer of a *NPDU* to the BS NWK layer.

8.1.2 LLC_transfer_confirm

Table 2

LLC_transfer_confirm	
Usage	MS
Source	LLC Layer
Destination	NWK Layer
Service Access Point	A
SDU Parameters	*transfer_receipt_ack*

This primitive acknowledges the receipt of the *NPDU* associated with a **LLC_transfer_request**. It does not indicate that the *NPDU* has been transferred to a peer SAP. It does indicate that the LLC is ready to receive another *NPDU*.

8.1.3 LLC_transfer_indication

Table 3

LLC_transfer_indication	
Usage	MS
Source	LLC Layer
Destination	NWK Layer
Service Access Point	A
SDU Parameters	*NPDU*

This primitive passes a received *NPDU* to the NWK layer.

8.1.4 LLC_create_protocol_request

Table 4

LLC_create_protocol_request	
Usage	MS
Source	NWK Layer
Destination	LLC Layer
Service Access Point	B
Multiple Outstanding	No
SDU Parameters	*protocol_type*
	protocol_parameters
	packet_classification_params

This primitive starts the procedure which instantiates an acknowledged protocol between the BS and MS. After instantiation, BS_LLC_TPT (for a downlink protocol) or MS_LLC_TPT (for an uplink protocol) routes LPDUs to the new protocol instance, and BS_MAC_BWM allocates bandwidth for the protocol.

8.1.5 LLC_create_protocol_confirm

Table 5

LLC_create_protocol_confirm	
Usage	MS
Source	LLC Layer
Destination	NWK Layer
Service Access Point	B
SDU Parameters	*create_protocol_result*
	protocol_instance_ID

This primitive confirms the creation of a protocol instance. All subsequent primitives referencing the protocol instance shall use the returned protocol instance ID.

8.1.6 LLC_create_protocol_indication

Table 6

LLC_create_protocol_indication	
Usage	MS
Source	LLC Layer
Destination	NWK Layer
Service Access Point	B
SDU Parameters	*create_protocol_result*
	protocol_instance_ID

This primitive indicates the creation of a protocol instance not specifically requested by the NWK layer. All subsequent primitives referencing the protocol instance shall use the returned protocol instance ID.

8.1.7 LLC_delete_protocol_request

Table 7

LLC_delete_protocol_request	
Usage	MS
Source	NWK Layer
Destination	LLC Layer
Service Access Point	B
Multiple Outstanding	No
SDU Parameters	*protocol_instance_ID*

This primitive deletes a protocol instance. Any PDUs awaiting transfer or in the process being transferred are discarded.

8.1.8 LLC_delete_protocol_confirm

Table 8

LLC_delete_protocol_confirm	
Usage	MS
Source	LLC Layer
Destination	NWK Layer
Service Access Point	B
SDU Parameters	*delete_protocol_result*
	protocol_instance_ID

This primitive confirms the deletion of a protocol instance.

8.1.9 LLC_delete_protocol_indication

Table 9

LLC_delete_protocol_indication	
Usage	MS
Source	LLC Layer
Destination	NWK Layer
Service Access Point	B
SDU Parameters	*delete_protocol_result*
	protocol_instance_ID

This primitive indicates the deletion of a protocol instance not specifically requested by the NWK layer.

8.1.10 LLC_service_indication

Table 10

LLC_service_indication	
Usage	MS
Source	LLC Layer
Destination	NWK Layer
Service Access Point	B
SDU Parameters	*new_service_state*
	service_parameters

This primitive is used by the LLC to provide the NWK with the latest service status.

8.2 Parameter Definitions

8.2.1 *create_protocol_result*

Table 11

create_protocol_result	
0	success: requested protocol instantiated
1	failure: create protocol request already pending
2	failure: requested resources unavailable

8.2.2 *delete_protocol_result*

Table 12

delete_protocol_result	
0	success: requested protocol instance deleted
1	failure: delete protocol request already pending
2	failure: specified protocol instance does not exist

8.2.3 *new_service_state*

Table 13

new_service_state	
0	LLC service is available
1	LLC service is not available

8.2.4 *NPDU*

This parameter consists of an IPv4 or IPv6 packet.

8.2.5 *packet_classification_params*

This parameter has two different forms depending upon the IP protocol version. The LLC_TPT packet classification service associates the packet classification parameters with a protocol instance ID. Arriving packets which require enhanced QOS are identified by the packet classification service and are routed to the correct protocol instance ID for transfer over the air interface.

Table 14

packet_classification_params - IPv4	
0	IP Protocol ID = IPv4
1	Destination IP address
2	Destination port

Table 15

packet_classification_params - IPv6	
0	IP Protocol ID = IPv6
1	Flow label

8.2.6 *protocol_instance_ID*

This parameter uniquely identifies a protocol instance.

8.2.7 *protocol_parameters*

This parameter specifies the desired QOS to be delivered to the flow and any configuration options for the protocol instance to be created. More information on the reservation specification can be found in IETF RFC 2211 [5] and IETF RFC 2215 [7].

8.2.8 *protocol_type*

Table 16

protocol_type	
0	Reserved
1	Reserved
2	Controlled-load downlink (ACK_CL_DL)
3	Controlled-load uplink (ACK_CL_UL)

8.2.9 *service_parameters*

This parameter provides additional information regarding LLC service to the NWK when the *new_service_state* parameter in the **LLC_service_indication** primitive indicates that LLC service is available.

Table 17

service_parameters	
0	GWI: Gateway identifier
1	BSI: Base Station identifier
2	MSI: Mobile Station identifier

8.2.10 *transfer_receipt_ack*

Table 18

transfer_receipt_ack	
0	Success: receipt acknowledged
1	Failure: transfer request already pending
2	Failure: LLC service is not available

Annex A (informative): The IPv4 to IPv6 Transition

The terms used in this appendix can be defined as follows:

- **IPv6-only address:** an IPv6 address from which an IPv4 address cannot be derived;
- **IPv6 (IPv4-compatible) address:** an IPv6 address from which an IPv4 address can be derived;
- **IPv4-only address:** an address for routing via IPv4;
- **IPv6-capable BS:** a BS running IPv6 routing protocols;
- **IPv4-capable BS:** a BS running IPv4 routing protocols;
- **host-to-router tunnelling:** sending an IPv6 datagram encapsulated within an IPv4 datagram to an IPv6-capable router. The router decapsulates the datagram and forwards it using IPv6 routing protocols;
- **host-to-host tunnelling:** sending an IPv6 datagram encapsulated within an IPv4 datagram to a host. The host decapsulates the datagram and sends it to the transport layer.

During the IPv4 to IPv6 transition phase, a DAWS GW and a DAWS MS may have one of three configurations:

1) IPv4 only;
2) IPv4 and IPv6 simultaneously;
3) IPv6 only.

Table A.1 summarizes the possible interactions between a DAWS GW and MS. A "P" entry means that communication on the network level is possible between the GW and MS; a "NP" entry means that communication on the network level between the GW and MS is not possible. Table C.1 shows that dual-stack GW and MS configurations maximize compatibility.

Table A.1: Possible GW-MS Combinations

		Mobile Station		
		IPv4	**IPv4/v6**	**IPv6**
Gateway	IPv4	P	P	NP
	IPv4/v6	P	P	P
	IPv6	NP	P	P

A DAWS GW should not be required to function as the source or termination of an IPv6-over-IPv4 tunnel. A DAWS GW will simply forward whatever type of datagram it receives without encapsulation or decapsulation. However, a dual-stack DAWS MS may be the source or termination of an IPv6-over-IPv4 tunnel.

For a dual-stack DAWS MS, there must be an entity in the network layer which decides which IP protocol stack to use when sending an IP datagram. In this annex, we will call this entity NWK_VSEL (for Version Select). The dual IP stack architecture for transmission can be visualized as shown in Figure A.1.

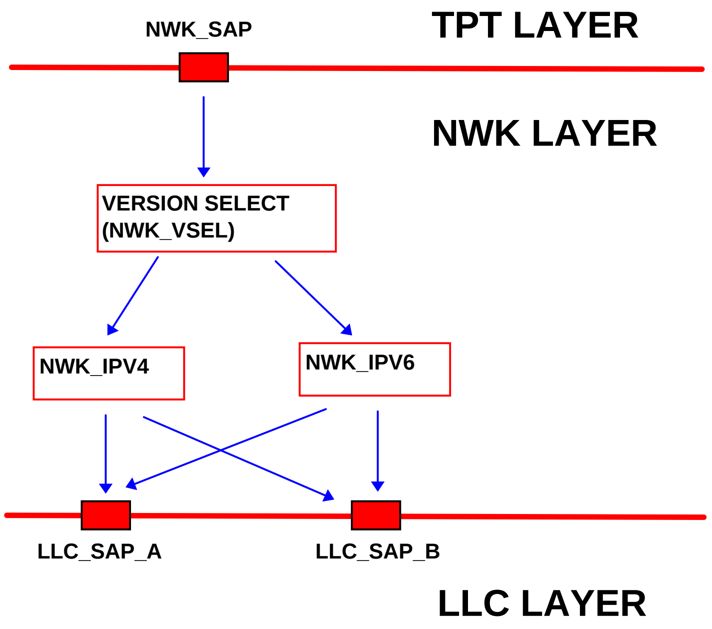

Figure A.1: Dual IP Protocol Stack Architecture - Transmission

The NWK_VSEL algorithm should favour the transmission of IPv6 packets. A possible NWK_VSEL algorithm is:

- if destination address is IPv6-only:
  - if GW is IPv6-capable:
    - send IPv6 native packet.
  - else if GW is IPv4-capable:
    - if tunnel endpoint address is available:
      - do host-to-router tunnelling.
    - else:
      - discard packet.
  - else:
    - discard packet.
- else if destination address is IPv6 (IPv4 compatible):
  - if GW is IPv6-capable:
    - send IPv6 native packet;
    - else if GW is IPv4-capable:
      - do host-to-host tunnelling.
  - else:
    - discard packet.
- else (destination address is IPv4-only):
  - if GW is IPv4-capable:
    - send IPv4 native packet.

- else:
    - discard packet.

This algorithm assumes that any host which advertises an IPv6 (IPv4-compatible) address is capable of functioning as the termination of a host-to-host tunnel.

The system administrator will configure the DAWS GW for IPv4-only, IPv6-only, or dual stack operation. The system administrator will also specify a tunnel endpoint address, if available. A BS will obtain GW configuration information when it registers with the Gateway (GW). A MS will obtain GW configuration information when it registers with the BS.

Bibliography

The following material, though not specifically referenced in the body of the present document (or not plublicly avaiblable), gives supporting information.

- Internet Draft: "IPv6 Stateless Address Auto configuration".
- IETF RFC 2210: "The use of RSVP with IETF Integrated Services".
- IETF RFC 2185: "Routing Aspects of IPv6 Transition".
- Internet Draft: "Transition Mechanisms for IPv6 Hosts and Routers".
- Internet Draft: "Dynamic Host Configuration Protocol".
- IETF RFC 1112: "Host Extensions for IP Multicasting".
- IETF RFC 791: "Internet Protocol".

History

Document history		
V1.1.1	April 1999	Publication

ISBN 2-7437-2969-4
Dépôt légal : Avril 1999