

**3GPP2 C.S0063-0 v1.0**

**Version 1.0**

**Date: March 2005**



**3RD GENERATION  
PARTNERSHIP  
PROJECT 2  
"3GPP2"**

---

## ***cdma2000 High Rate Packet Data Supplemental Services***

### ***COPYRIGHT***

3GPP2 and its Organizational Partners claim copyright in this document and individual Organizational Partners may copyright and issue documents or standards publications in individual Organizational Partner's name based on this document. Requests for reproduction of this document should be directed to the 3GPP2 Secretariat at [secretariat@3gpp2.org](mailto:secretariat@3gpp2.org). Requests to reproduce individual Organizational Partner's documents should be directed to that Organizational Partner. See [www.3gpp2.org](http://www.3gpp2.org) for more information.



**CONTENTS**

1	FOREWORD.....	i
2	NOTES.....	i
3	REFERENCES.....	i
4	<b>1 Overview.....</b>	<b>1-1</b>
5	1.1 Scope of This Document .....	1-1
6	1.2 Requirements Language.....	1-1
7	1.3 Architecture Reference Model.....	1-1
8	1.4 Protocols .....	1-1
9	1.4.1 Interfaces .....	1-1
10	1.4.2 States.....	1-1
11	1.5 Terms.....	1-1
12	1.6 Notation .....	1-1
13	<b>2 Enhanced Multi-Flow Packet Application .....</b>	<b>2-1</b>
14	2.1 Introduction .....	2-1
15	2.1.1 General Overview.....	2-1
16	2.1.2 Public Data.....	2-1
17	2.1.3 Data Encapsulation for the InUse Instance of the Application .....	2-1
18	2.2 Protocol Initialization.....	2-1
19	2.2.1 Protocol Initialization for the InConfiguration Application Instance.....	2-1
20	2.2.2 Protocol Initialization for the InUse Application Instance.....	2-1
21	2.3 Procedures and Messages for the InConfiguration Instance of the Packet	
22	Application .....	2-1
23	2.3.1 Procedures.....	2-1
24	2.3.2 Commit Procedures.....	2-1
25	2.3.3 Message Formats .....	2-1
26	2.3.3.1 ConfigurationRequest.....	2-1
27	2.3.3.2 ConfigurationResponse.....	2-1
28	2.4 Route Selection Protocol .....	2-1
29	2.4.1 Overview.....	2-1
30	2.4.2 Primitives and Public Data .....	2-1
31	2.4.2.1 Commands.....	2-1
32	2.4.2.2 Return Indications .....	2-1

**CONTENTS**

1        2.4.3 Protocol Data Unit ..... 2-1

2        2.4.4 Procedures and Messages for the InUse Instance of the Protocol ..... 2-1

3            2.4.4.1 Procedures ..... 2-1

4                2.4.4.1.1 General Requirements ..... 2-1

5                2.4.4.1.2 Access Terminal Requirements ..... 2-1

6                    2.4.4.1.2.1 A Open B Setting State ..... 2-1

7                        2.4.4.1.2.1.1 State Transitions ..... 2-1

8                        2.4.4.1.2.1.2 Transmitter Requirements ..... 2-1

9                        2.4.4.1.2.1.3 Receiver Requirements ..... 2-1

10                2.4.4.1.2.2 A Open B Rising State ..... 2-1

11                    2.4.4.1.2.2.1 State Transitions ..... 2-1

12                    2.4.4.1.2.2.2 Transmitter Requirements ..... 2-1

13                    2.4.4.1.2.2.3 Receiver Requirements ..... 2-1

14                2.4.4.1.2.3 A Setting B Open State ..... 2-1

15                    2.4.4.1.2.3.1 State Transitions ..... 2-1

16                    2.4.4.1.2.3.2 Transmitter Requirements ..... 2-1

17                    2.4.4.1.2.3.3 Receiver Requirements ..... 2-1

18                2.4.4.1.2.4 A Rising B Open State ..... 2-1

19                    2.4.4.1.2.4.1 State Transitions ..... 2-1

20                    2.4.4.1.2.4.2 Transmitter Requirements ..... 2-1

21                    2.4.4.1.2.4.3 Receiver Requirements ..... 2-1

22                2.4.4.1.3 Access Network Requirements ..... 2-1

23        2.4.4.2 Message Formats ..... 2-1

24                2.4.4.2.1 RouteSelect ..... 2-1

25                2.4.4.2.2 RouteSelectAck ..... 2-1

26                2.4.4.2.3 ActivateRoute ..... 2-1

27                2.4.4.2.4 ActivateRouteAck ..... 2-1

28        2.4.4.3 Interface to Other Protocols ..... 2-1

29                2.4.4.3.1 Commands ..... 2-1

30                2.4.4.3.2 Indications ..... 2-1

31        2.4.5 Protocol Numeric Constants ..... 2-1

## CONTENTS

1	2.5 Radio Link Protocol.....	2-1
2	2.5.1 Overview.....	2-1
3	2.5.2 Primitives and Public Data .....	2-1
4	2.5.2.1 Commands.....	2-1
5	2.5.2.2 Return Indications .....	2-1
6	2.5.3 Protocol Data Unit .....	2-1
7	2.5.4 Procedures and Messages for the InUse Instance of the Protocol.....	2-1
8	2.5.4.1 Procedures .....	2-1
9	2.5.4.1.1 Initialization and Reset .....	2-1
10	2.5.4.1.1.1 Initialization Procedure .....	2-1
11	2.5.4.1.1.1.1 Initialization Procedure for the RLP Transmitter .....	2-1
12	2.5.4.1.1.1.2 Initialization Procedure for the RLP Receiver .....	2-1
13	2.5.4.1.1.2 Reset Procedure .....	2-1
14	2.5.4.1.1.2.1 Reset Procedure for the Initiating Side when it is an RLP	
15	Transmitter .....	2-1
16	2.5.4.1.1.2.2 Reset Procedure for Initiating Side when it is an RLP	
17	Receiver .....	2-1
18	2.5.4.1.1.2.3 Reset Procedure for the Responding Side when it is an RLP	
19	Receiver .....	2-1
20	2.5.4.1.1.2.4 Reset Procedure for the Responding Side when it is an RLP	
21	transmitter .....	2-1
22	2.5.4.1.1.2.5 RLP Reset Message Flows .....	2-1
23	2.5.4.2 Data Transfer.....	2-1
24	2.5.4.2.1 RLP Transmit Procedures.....	2-1
25	2.5.4.2.1.1 Close State .....	2-1
26	2.5.4.2.1.1.1 Access Terminal Requirements .....	2-1
27	2.5.4.2.1.1.2 Access Network Requirements .....	2-1
28	2.5.4.2.1.2 Open State .....	2-1
29	2.5.4.2.1.2.1 Access Terminal Requirements .....	2-1
30	2.5.4.2.1.2.2 Access Network Requirements .....	2-1
31	2.5.4.2.2 RLP Receive Procedures .....	2-1
32	2.5.4.3 RLP Packet Header .....	2-1
33	2.5.4.4 Message Formats .....	2-1

**CONTENTS**

1	2.5.4.4.1 ResetTxIndication.....	2-1
2	2.5.4.4.2 ResetRxIndication .....	2-1
3	2.5.4.4.3 ResetTxIndicationAck .....	2-1
4	2.5.4.4.4 ResetTxComplete.....	2-1
5	2.5.4.4.5 ResetRxComplete .....	2-1
6	2.5.4.4.6 ReceiverStatus .....	2-1
7	2.5.4.4.7 ReservationOnRequest .....	2-1
8	2.5.4.4.8 ReservationOffRequest .....	2-1
9	2.5.4.4.9 ReservationAccept .....	2-1
10	2.5.4.4.10 ReservationReject.....	2-1
11	2.5.4.4.11 ReservationOn .....	2-1
12	2.5.4.4.12 ReservationOff .....	2-1
13	2.5.4.4.13 FwdReservationOff .....	2-1
14	2.5.4.4.14 AttributeUpdateRequest .....	2-1
15	2.5.4.4.15 AttributeUpdateAccept .....	2-1
16	2.5.4.4.16 AttributeUpdateReject .....	2-1
17	2.5.4.5 Interface to Other Protocols.....	2-1
18	2.5.4.5.1 Commands .....	2-1
19	2.5.4.5.2 Indications.....	2-1
20	2.5.4.6 RLP Packet Priorities.....	2-1
21	2.5.5 Protocol Numeric Constants.....	2-1
22	2.6 Data Over Signaling Protocol .....	2-1
23	2.6.1 Overview .....	2-1
24	2.6.2 Primitives and Public Data.....	2-1
25	2.6.2.1 Commands .....	2-1
26	2.6.2.2 Return Indications.....	2-1
27	2.6.3 Protocol Data Unit.....	2-1
28	2.6.4 Procedures and Messages for the InUse Instance of the Protocol .....	2-1
29	2.6.4.1 Procedures .....	2-1
30	2.6.4.2 Message Formats .....	2-1
31	2.6.4.2.1 DataOverSignaling .....	2-1

**CONTENTS**

1	2.6.4.2.2 DataOverSignalingAck .....	2-1
2	2.6.4.3 Interface to Other Protocols .....	2-1
3	2.6.4.3.1 Commands .....	2-1
4	2.6.4.3.2 Indications .....	2-1
5	2.7 Location Update Protocol .....	2-1
6	2.7.1 Overview .....	2-1
7	2.7.2 Primitives and Public Data .....	2-1
8	2.7.2.1 Commands .....	2-1
9	2.7.2.2 Return Indications .....	2-1
10	2.7.3 Protocol Data Unit .....	2-1
11	2.7.4 Procedures and Messages for the InUse Instance of the Protocol .....	2-1
12	2.7.4.1 Procedures .....	2-1
13	2.7.4.1.1 Access Network Requirements .....	2-1
14	2.7.4.1.2 Access Terminal Requirements .....	2-1
15	2.7.4.2 Message Formats .....	2-1
16	2.7.4.2.1 LocationRequest .....	2-1
17	2.7.4.2.2 LocationNotification .....	2-1
18	2.7.4.2.3 LocationAssignment .....	2-1
19	2.7.4.2.4 LocationComplete .....	2-1
20	2.7.4.2.5 ServiceNetworkIDRequest .....	2-1
21	2.7.4.2.6 ServiceNetworkIDNotification .....	2-1
22	2.7.4.2.7 ServiceNetworkIDAssignment .....	2-1
23	2.7.4.2.8 ServiceNetworkIDComplete .....	2-1
24	2.7.4.3 Interface to Other Protocols .....	2-1
25	2.7.4.3.1 Commands .....	2-1
26	2.7.4.3.2 Indications .....	2-1
27	2.8 Flow Control Protocol .....	2-1
28	2.8.1 Overview .....	2-1
29	2.8.2 Primitives and Public Data .....	2-1
30	2.8.2.1 Commands .....	2-1
31	2.8.2.2 Return Indications .....	2-1
32	2.8.3 Protocol Data Unit .....	2-1

**CONTENTS**

1        2.8.4 Procedures and Messages for the InUse Instance of the Protocol ..... 2-1

2        2.8.4.1 Procedures ..... 2-1

3            2.8.4.1.1 Transmission and Processing of DataReady Message..... 2-1

4            2.8.4.1.2 Transmission and Processing of RestartNetworkInterface Message..... 2-1

5            2.8.4.1.3 Close State..... 2-1

6                2.8.4.1.3.1 Access Terminal Requirements ..... 2-1

7                2.8.4.1.3.2 Access Network Requirements ..... 2-1

8            2.8.4.1.4 Open State..... 2-1

9                2.8.4.1.4.1 Access Terminal Requirements ..... 2-1

10              2.8.4.1.4.2 Access Network Requirements ..... 2-1

11        2.8.4.2 Message Formats ..... 2-1

12            2.8.4.2.1 XonRequest ..... 2-1

13            2.8.4.2.2 XonResponse ..... 2-1

14            2.8.4.2.3 XoffRequest ..... 2-1

15            2.8.4.2.4 XoffResponse ..... 2-1

16            2.8.4.2.5 DataReady ..... 2-1

17            2.8.4.2.6 DataReadyAck..... 2-1

18            2.8.4.2.7 RestartNetworkInterface ..... 2-1

19            2.8.4.2.8 RestartNetworkInterfaceAck ..... 2-1

20        2.8.5 Protocol Numeric Constants..... 2-1

21        2.9 Configuration Attributes for the Enhanced Multi-Flow Packet Application..... 2-1

22        2.9.1 Simple Attributes ..... 2-1

23        2.9.2 Complex Attributes..... 2-1

24            2.9.2.1 FlowNNTimersFwd Attribute..... 2-1

25            2.9.2.2 FlowNNTimersRev Attribute ..... 2-1

26            2.9.2.3 FlowNNReservationFwd Attribute ..... 2-1

27            2.9.2.4 FlowNNReservationRev Attribute ..... 2-1

28            2.9.2.5 SupportedFlowProtocolIDs Attribute..... 2-1

29            2.9.2.6 SupportedRouteProtocolIDs Attribute ..... 2-1

30            2.9.2.7 ATSupportedQoSProfiles Attribute..... 2-1

31            2.9.2.8 ANSupportedQoSProfiles Attribute ..... 2-1

**CONTENTS**

1	2.9.2.9 ReservationKKQoSRequestFwd Attribute.....	2-1
2	2.9.2.10 ReservationKKQoSRequestRev Attribute .....	2-1
3	2.9.2.11 ReservationKKQoSResponseFwd Attribute .....	2-1
4	2.9.2.12 ReservationKKQoSResponseRev Attribute .....	2-1
5	2.9.2.13 FlowNNFlowProtocolFwd Attribute .....	2-1
6	2.9.2.13.1 Definition of ProtocolParameters record when the Flow Protocol is	
7	ROHC .....	2-1
8	2.9.2.14 FlowNNFlowProtocolRev Attribute .....	2-1
9	2.9.2.15 FlowNNRouteProtocolFwd Attribute.....	2-1
10	2.9.2.16 FlowNNRouteProtocolRev Attribute .....	2-1
11	2.10 Session State Information .....	2-1
12	2.10.1 Location Parameter .....	2-1
13	2.10.2 FlowControlState Parameter .....	2-1
14	2.10.3 DataOverSignalingMessageSequence Parameter.....	2-1
15	2.10.4 ServiceNetworkID Parameter .....	2-1
16	2.10.5 ReservationState Parameter .....	2-1
17	2.10.6 RouteState Parameter .....	2-1

**FIGURES**

1 Figure 1.3-1. Architecture Reference Model ..... 1-1

2 Figure 2.1.1-1. Reference Architecture for a Forward Link Flow ..... 2-1

3 Figure 2.1.1-2. Reference Architecture for a Reverse Link Flow ..... 2-1

4 Figure 2.1.1-3. Enhanced Multi-Flow Packet Application Protocols..... 2-1

5 Figure 2.1.3-1. Enhanced Multi-Flow Packet Application Encapsulation when the  
6 Link Flow carries an Octet Stream..... 2-1

7 Figure 2.1.3-2. Enhanced Multi-Flow Packet Application Encapsulation when the  
8 Link Flow carries a Packet Stream..... 2-1

9 Figure 2.5.4.1.1.2.5-1. RLP Reset Procedure Initiated by RLP Transmitter ..... 2-1

10 Figure 2.5.4.1.1.2.5-2. RLP Reset Procedure Initiated by RLP Receiver ..... 2-1

11 Figure 2.5.4.2.1-1. RLP Transmit Sequence Number Variable ..... 2-1

12 Figure 2.5.4.2.1-2. Reverse Link Reservation State Diagram (Access Terminal) ..... 2-1

13 Figure 2.5.4.2.1-3. Forward Link Reservation State Diagram (Access Network) ..... 2-1

14 Figure 2.5.4.2.2-1. RLP Receive Sequence Number Variables ..... 2-1

15 Figure 2.8.1-1. Flow Control Protocol State Diagram (Access Terminal) ..... 2-1

16 Figure 2.8.1-2. Flow Control Protocol State Diagram (Access Network) ..... 2-1

**TABLES**

1	Table 2.7.4.2-1. LocationType Encoding.....	2-1
2	Table 2.7.4.2-2. Subfields of LocationValue when LocationType = 0x01.....	2-1
3	Table 2.9-1. Configurable Values .....	2-1
4	Table 2.9.2.5-1. Default Values of ProtocolIdentifier in SupportedFlowProtocolIDs	
5	Attribute .....	2-1
6	Table 2.9.2.13-1. ProtocolID for Flow Protocol.....	2-1
7	Table 2.9.2.15-1. ProtocolID for Route Protocol .....	2-1
8	Table 2.10.1-1. The Format of the Parameter Record for the Location Parameter.....	2-1
9	Table 2.10.2-1. The Format of the Parameter Record for the FlowControlState	
10	Parameter .....	2-1
11	Table 2.10.3-1. The Format of the Parameter Record for the	
12	DataOverSignalingMessageSequence Parameter .....	2-1
13	Table 2.10.4-1. The Format of the Parameter Record for the ServiceNetworkID	
14	Parameter .....	2-1
15	Table 2.10.5-1. The Format of the Parameter Record for the ReservationState	
16	Parameter .....	2-1
17	Table 2.10.6-1. The Format of the Parameter Record for the RouteState Parameter .....	2-1
18	Table 2.10.6-2. RouteSelectionProtocolState Encoding .....	2-1

## FOREWORD

1 **(This foreword is not part of this Standard)**

2 This standard was prepared by Technical Specification Group C of the Third Generation  
3 Partnership Project 2 (3GPP2). This standard is evolved from and is a companion to the  
4 cdma2000<sup>®1</sup> standards. This air interface standard provides high rate packet data  
5 supplemental services.

6

---

<sup>1</sup> “cdma2000<sup>®</sup> is the trademark for the technical nomenclature for certain specifications and standards of the Organizational Partners (OPs) of 3GPP2. Geographically (and as of the date of publication), cdma2000<sup>®</sup> is a registered trademark of the Telecommunications Industry Association (TIA-USA) in the United States.”

**REFERENCES**

1 The following documents contain provisions, which, through reference in this text,  
2 constitute provisions of this document. References are either specific (identified by date of  
3 publication, edition number, version number, etc.) or non-specific. For a specific reference,  
4 subsequent revisions do not apply. For a non-specific reference, the latest version applies.  
5 In the case of a reference to a 3GPP2 document, a non-specific reference implicitly refers to  
6 the latest version of that document in the same Release as the present document.

- 7  
8 [1] X.S0011-001, cdma2000 Wireless IP Network Standard: Introduction.
- 9 [2] C.S0005-D, Upper Layer (Layer 3) Signaling Specification for cdma2000 Spread  
10 Spectrum Systems.
- 11 [3] C.R1001, Administration of Parameter Value Assignments for cdma2000 Spread  
12 Spectrum Standards.
- 13 [4] IETF RFC 1662, PPP in HDLC-like Framing.
- 14 [5] IETF RFC 791, Internet Protocol.
- 15 [6] IETF RFC 3095, Robust Header Compression (ROHC): Framework and four  
16 profiles: RTP, UDP, ESP, and uncompressed.
- 17 [7] IETF draft-ietf-pana-pana-04, Protocol for Carrying Authentication for Network  
18 Access (PANA).
- 19 [8] C.S0024-A, cdma2000 High Rate Packet Data Air Interface Specification.
- 20 [9] C.S0002-D, Physical Layer standard for cdma2000 Spread Spectrum Systems.
- 21 [10] IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification.
- 22 [11] A.S0007, Inter-Operability Specification (IOS) for High Rate Packet Data (HRPD)  
23 Access Network Interfaces.



## 1 OVERVIEW

### 1.1 Scope of This Document

These technical requirements form a compatibility standard for supplemental services on cdma2000 high rate packet data systems. These requirements ensure that a compliant access terminal can obtain service through any access network conforming to this standard. These requirements do not address the quality or reliability of that service, nor do they cover equipment performance or measurement procedures.

This specification is primarily oriented toward requirements necessary for the design and implementation of access terminals. As a result, detailed procedures are specified for access terminals to ensure a uniform response to all access networks. Access network procedures, however, are specified only to the extent necessary for compatibility with those specified for the access terminal.

This specification includes provisions for future service additions and expansion of system capabilities. The architecture defined by this specification permits such expansion without the loss of backward compatibility to older access terminals.

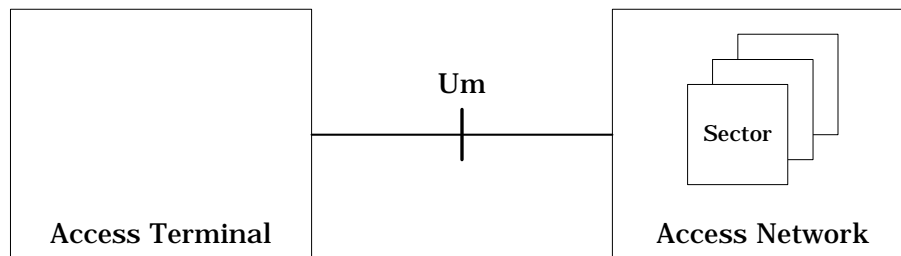
### 1.2 Requirements Language

Compatibility, as used in connection with this standard, is understood to mean: Any access terminal can obtain service through any access network conforming to this standard. Conversely, all access networks conforming to this standard can service access terminals.

“Shall” and “shall not” identify requirements to be followed strictly to conform to the standard and from which no deviation is permitted. “Should” and “should not” indicate that one of several possibilities is recommended as particularly suitable, without mentioning or excluding others, that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. “May” and “need not” indicate a course of action permissible within the limits of the standard. “Can” and “cannot” are used for statements of possibility and capability, whether material, physical, or causal.

### 1.3 Architecture Reference Model

The architecture reference model is presented in Figure 1.3-1. The reference model consists of the following functional units:



**Figure 1.3-1. Architecture Reference Model**

The access terminal, the access network, and the sector are formally defined in Section 1.5.

The reference model includes the air interface between the access terminal and the access network. The protocols used over the air interface are defined in this document.

## 1.4 Protocols

### 1.4.1 Interfaces

This standard defines a set of interfaces for communications between protocols in the same entity and between a protocol executing in one entity and the same protocol executing in the other entity.

In the following the generic term “entity” is used to refer to the access terminal and the access network.

Protocols in this specification have four types of interfaces:

- Headers and messages are used for communications between a protocol executing in one entity and the same protocol executing in the other entity.
- Commands are used by a protocol to obtain a service from another protocol within the same access network or access terminal.
- Indications are used by a protocol to convey information regarding the occurrence of an event to another protocol within the same access network or access terminal. Any protocol can register to receive these indications.
- Public Data is used to share information in a controlled way between protocols/applications. Public data is shared between protocols/applications in the same layer, as well as between protocols/applications in different layers. The public data of the InUse protocol/application is created when an InUse instance of a protocol/application is created. All configurable attributes of the InConfiguration instance of a protocol or application are also public data of that protocol or application.

Commands and indications are written in the form of *Protocol.Command* and *Protocol.Indication*. When the context is clear, the *Protocol* part is dropped.

Commands are always written in the imperative form, since they direct an action. Indications are always written in the past tense since they notify of events that happened.

1 Headers and messages are binding on all implementations. Commands, indications, and  
2 public data are used as a device for a clear and precise specification. Access terminals and  
3 access networks can be compliant with this specification while choosing a different  
4 implementation that exhibits identical behavior.

#### 5 1.4.2 States

6 When protocols exhibit different behavior as a function of the environment, this behavior is  
7 captured in a set of states and the events leading to a transition between states.

8 Unless otherwise specifically mentioned, the state of the access network refers to the state  
9 of a protocol engine in the access network as it applies to a particular access terminal.  
10 Since the access network communicates with multiple access terminals, multiple  
11 independent instantiations of a protocol will exist in the access network, each with its own  
12 independent state machine.

13 Unless otherwise specifically shown, the state transitions due to failure are not shown in  
14 the figures.

15 Typical events leading to a transition from one state to another are the receipt of a message,  
16 a command from a higher layer protocol, an indication from a lower layer protocol, or the  
17 expiration of a timer.

18 When a protocol is not functional at a particular time the protocol is placed in a state called  
19 the Inactive state. This state is common for most protocols.

20 Other common states are Open, indicating that the session or connection (as applicable to  
21 the protocol) is open and Close, indicating that the session or connection is closed.

22 If a protocol has a single state other than the Inactive state, that state is always called the  
23 Active state. If a protocol has more than one state other than the Inactive state, all of these  
24 states are considered active, and are given individual names.

#### 25 1.5 Terms

26 **Access Network (AN).** The network equipment providing data connectivity between a  
27 packet switched data network (typically the Internet) and the access terminals. An access  
28 network is equivalent to a base station in [9].

29 **Access Terminal (AT).** A device providing data connectivity to a user. An access terminal  
30 may be connected to a computing device such as a laptop personal computer or it may be a  
31 self-contained data device such as a personal digital assistant. An access terminal is  
32 equivalent to a mobile station in [9].

33 **Channel.** The set of channels transmitted between the access network and the access  
34 terminals within a given frequency assignment. A Channel consists of a Forward Link and a  
35 Reverse Link.

36 **Forward Channel.** The portion of the Channel consisting of those Physical Layer Channels  
37 transmitted from the access network to the access terminal.

38 **Forward Control Channel.** The channel that carries data to be received by all access  
39 terminals monitoring the Forward Channel.

1 **Forward Traffic Channel.** The portion of the Forward Channel that carries information for  
 2 a specific access terminal. The Forward Traffic Channel can be used as either a Dedicated  
 3 Resource or a non-Dedicated Resource. Prior to successful access terminal authentication,  
 4 the Forward Traffic Channel serves as a non-Dedicated Resource. Only after successful  
 5 access terminal authentication can the Forward Traffic Channel be used as a Dedicated  
 6 Resource for the specific access terminal.

7 **FCS.** Frame Check Sequence.

8 **NULL.** A value which is not in the specified range of the field.

9 **Reservation.** Air interface resources set up by the access network to carry a higher layer  
 10 flow. A Reservation is identified by its ReservationLabel. ReservationLabels are bound to  
 11 Link Flows that carry higher layer flows. A Reservation can be either in the Open or Close  
 12 state.

13 **Reverse Access Channel.** The portion of the Reverse Channel that is used by access  
 14 terminals to communicate with the access network when they do not have a traffic channel  
 15 assigned. There is a separate Reverse Access Channel for each sector of the access network.

16 **Reverse Channel.** The portion of the Channel consisting of those Physical Layer Channels  
 17 transmitted from the access terminal to the access network.

18 **Reverse Traffic Channel.** The portion of the Reverse Channel that carries information from  
 19 a specific access terminal to the access network. The Reverse Traffic Channel can be used  
 20 as either a Dedicated Resource or a non-Dedicated Resource. Prior to successful access  
 21 terminal authentication, the Reverse Traffic Channel serves as a non-Dedicated Resource.  
 22 Only after successful access terminal authentication can the Reverse Traffic Channel be  
 23 used as a Dedicated Resource for the specific access terminal.

24 **RLP.** Radio Link Protocol provides reliable delivery if needed, in-order delivery if needed,  
 25 and duplicate detection for a higher layer data stream.

26 **Rx.** Receive.

27 **Sector.** The part of the access network that provides one CDMA channel.

28 **SNP.** Signaling Network Protocol provides message transmission services for signaling  
 29 messages. The protocols that control each layer use SNP to deliver their messages to their  
 30 peer protocols. SNP is defined in [8].

31 **Stream Layer.** The Stream Layer provides multiplexing of distinct streams. Stream 0 is  
 32 dedicated to signaling and defaults to the default signaling stream (SNP / SLP). Stream 1,  
 33 Stream 2 and Stream 3 are not used by default. The Stream Layer is defined in [8].

34 **Tx.** Transmit.

### 35 1.6 Notation

36 **A[i]** The  $i^{\text{th}}$  element of array A. The zeroeth element of the array is A[0].

37 **<e<sub>1</sub>, e<sub>2</sub>, ..., e<sub>n</sub>>** A *structure* with elements 'e<sub>1</sub>', 'e<sub>2</sub>', ..., 'e<sub>n</sub>'.  
 38 Two structures E = <e<sub>1</sub>, e<sub>2</sub>, ..., e<sub>n</sub>> and F = <f<sub>1</sub>, f<sub>2</sub>, ..., f<sub>m</sub>> are equal if

1		and only if 'm' is equal to 'n' and $e_i$ is equal to $f_i$ for $i=1, \dots, n$ .
2		Given $E = \langle e_1, e_2, \dots, e_n \rangle$ and $F = \langle f_1, f_2, \dots, f_m \rangle$ , the assignment "E =
3		F" denotes the following set of assignments: $e_i = f_i$ , for $i=1, \dots, n$ .
4	<b>S.e</b>	The member of the structure 'S' that is identified by 'e'.
5	<b>M[i:j]</b>	Bits $i^{\text{th}}$ through $j^{\text{th}}$ inclusive ( $i \geq j$ ) of the binary representation of
6		variable M. M[0:0] denotes the least significant bit of M.
7		Concatenation operator. (A   B) denotes variable A concatenated with
8		variable B.
9	×	Indicates multiplication.
10	$\lfloor x \rfloor$	Indicates the largest integer less than or equal to x: $\lfloor 1.1 \rfloor = 1$ , $\lfloor 1.0 \rfloor =$
11		1.
12	$\lceil x \rceil$	Indicates the smallest integer greater or equal to x: $\lceil 1.1 \rceil = 2$ , $\lceil 2.0 \rceil =$
13		2.
14	$ x $	Indicates the absolute value of x: $ -17  = 17$ , $ 17  = 17$ .
15	$\oplus$	Indicates exclusive OR (modulo-2 addition).
16	$\min(x, y)$	Indicates the minimum of x and y.
17	$\max(x, y)$	Indicates the maximum of x and y.
18	$x \bmod y$	Indicates the remainder after dividing x by y: $x \bmod y = x - (y \times \lfloor x/y \rfloor)$ .
19	Unless otherwise specified, the format of field values is unsigned binary.	
20	Unless indicated otherwise, this standard presents numbers in decimal form. Binary	
21	numbers are distinguished in the text by the use of single quotation marks. Hexadecimal	
22	numbers are distinguished by the prefix '0x'.	
23	Unless specified otherwise, each field of a packet shall be transmitted in sequence such	
24	that the most significant bit (MSB) is transmitted first and the least significant bit (LSB) is	
25	transmitted last. The MSB is the left-most bit in the figures in this document. If there are	
26	multiple rows in a table, the top-most row is transmitted first. If a table is used to show the	
27	sub-fields of a particular field or variable, the top-most row consists of the MSBs of the	
28	field. Within a row in a table, the left-most bit is transmitted first. Notations of the form	
29	"repetition factor of N" or "repeated N times" mean that a total of N versions of the item are	
30	used.	

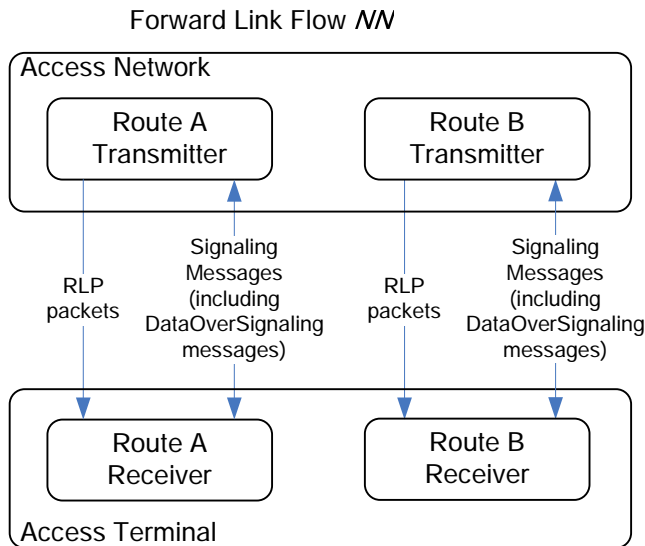


1 **2 ENHANCED MULTI-FLOW PACKET APPLICATION**

2 **2.1 Introduction**

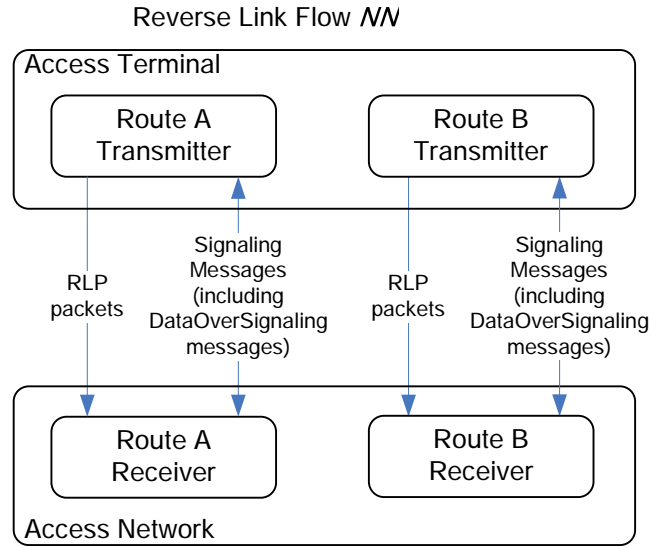
3 **2.1.1 General Overview**

4 The Enhanced Multi-Flow Packet Application provides multiple octet or packet streams that  
5 can be used to carry octets or packets between the access terminal and the access network.  
6 Each octet or packet stream is called a Link Flow. Each Link Flow provides two routes for  
7 transmission and reception of payloads from the higher layer. These routes are named  
8 Route A and Route B and can be carried using a single receiver-transmitter pair. Each  
9 route is associated with a transmitter-receiver pair. Figure 2.1.1-1 shows the association  
10 between a forward Link Flow and the transmitters and receivers for its two routes. Figure  
11 2.1.1-2 shows the reference architecture for a reverse Link Flow.



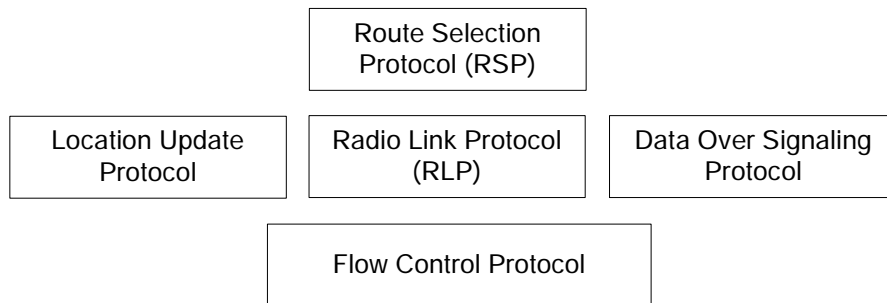
12 **Figure 2.1.1-1. Reference Architecture for a Forward Link Flow**

13



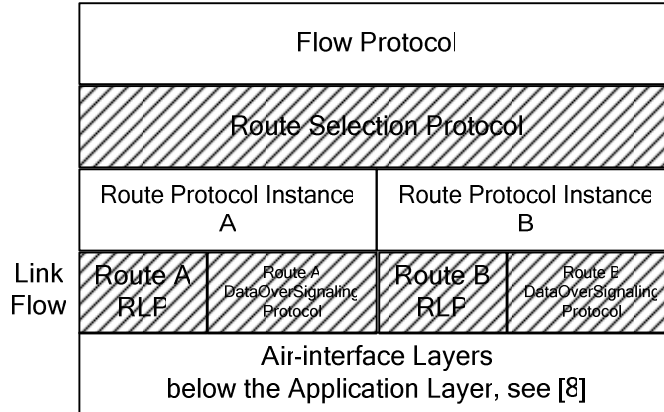
**Figure 2.1.1-2. Reference Architecture for a Reverse Link Flow**

The relationship between the Enhanced Multi-Flow Packet Application protocols is illustrated in Figure 2.1.1-3.



**Figure 2.1.1-3. Enhanced Multi-Flow Packet Application Protocols**

Figure 2.1.1-4 illustrates the relationship for each Link Flow between the Enhanced Multi-flow Packet Application and the higher layer protocols supported by the Enhanced Multi-flow Packet Application. The Flow Protocol and the Route Protocol are referred to as higher layer protocols. The protocols defined in the Enhanced Multi-flow Packet Application are shown shaded. The Route Selection Protocol routes Flow Protocol SDUs to either instance A or instance B of the Route Protocol. Instance A of the Route Protocol is bound to Route A of the Link Flow. Instance B of the Route Protocol is bound to Route B of the Link Flow.



1

2

3

**Figure 2.1.1-4. Relationship for each Link Flow between Enhanced Multi-Flow Packet Application and Higher Layer Protocols**

4

The Enhanced Multi-Flow Packet Application provides:

5

- the functionality defined in [1],

6

- the Route Selection Protocol, which routes Flow Protocol SDUs over either Route A or Route B of a Link Flow.

7

8

- the Radio Link Protocol (RLP), which provides retransmission (if needed) and duplicate detection of higher layer octets or packets transmitted on each route,

9

10

- the Data Over Signaling Protocol, which provides transmission and duplicate detection of higher layer data packets transmitted on each route using signaling messages,

11

12

- the Location Update Protocol, which defines location update procedures and messages in support of mobility management for the Packet Application,

13

14

- the Flow Control Protocol, which provides flow control for the Enhanced Multi-Flow Packet Application, and

15

16

- ability to negotiate Route Protocol and Flow Protocol parameters.

17

This application uses the Signaling Application to transmit and receive messages.

18

**2.1.2 Public Data**

19

- Subtype for this application

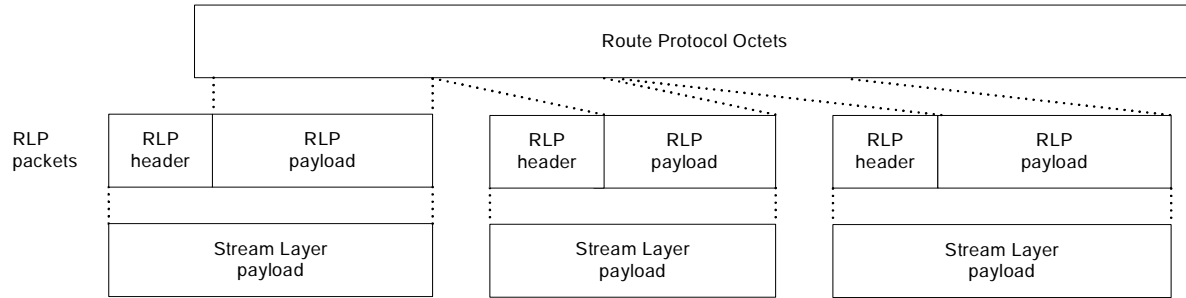
20

**2.1.3 Data Encapsulation for the InUse Instance of the Application**

21

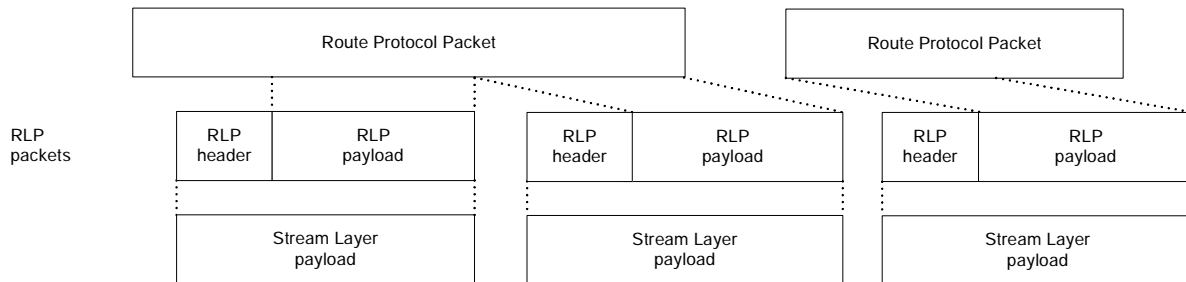
Figure 2.1.3-1 illustrates the relationship between octets from the Route Protocol, RLP packets, and Stream Layer payload for the case when the Link Flow carries an octet stream.

22



**Figure 2.1.3-1. Enhanced Multi-Flow Packet Application Encapsulation when the Link Flow carries an Octet Stream**

Figure 2.1.3-2 illustrates the relationship between packets from the Route Protocol, RLP packets, and Stream Layer payload for the case when the Link Flow carries a packet stream.



**Figure 2.1.3-2. Enhanced Multi-Flow Packet Application Encapsulation when the Link Flow carries a Packet Stream**

## 2.2 Protocol Initialization

### 2.2.1 Protocol Initialization for the InConfiguration Application Instance

Upon creation, the InConfiguration instance of this application in the access terminal and the access network shall perform the following in the order specified:

- The fall-back values of the attributes for this application instance shall be set to the default values specified for each attribute.
- If the InUse instance of this application (i.e., corresponding to the stream or virtual stream to which this application is bound) has the same application subtype as this InConfiguration application instance, then the fall-back values of the attributes defined by the InConfiguration application instance shall be set to the corresponding attribute values for the InUse application instance.
- The value for each attribute for this application instance shall be set to the fall-back value for that attribute.
- The value of the InConfiguration application instance public data shall be set to the value of the corresponding InUse application instance public data.

- 1 • The value of the application subtype associated with the InConfiguration application  
2 instance shall be set to the application subtype that identifies this application.

### 3 2.2.2 Protocol Initialization for the InUse Application Instance

4 Upon creation, the InUse instance of the Packet Application (i.e., corresponding to the  
5 stream or virtual stream to which this application is bound) in the access terminal and  
6 access network shall perform the following:

- 7 • The value of the attributes for this application instance shall be set to the default values  
8 specified for each attribute.

## 9 **2.3 Procedures and Messages for the InConfiguration Instance of the Packet** 10 **Application**

### 11 2.3.1 Procedures

12 This protocol uses the Generic Configuration Protocol (see [8]) to define the processing of  
13 the configuration messages.

### 14 2.3.2 Commit Procedures

15 The access terminal and the access network shall perform the procedures specified in this  
16 section, in the order specified, when directed by the InUse instance of the Session  
17 Configuration Protocol to execute the Commit procedures:

- 18 • All the public data that are defined by this application, but are not defined by the InUse  
19 application instance shall be added to the public data of the InUse application.
- 20 • If the InUse instance of this application (corresponding to the stream or virtual stream  
21 to which this application is bound) has the same subtype as this application instance,  
22 then
- 23 – The access terminal and the access network shall set the attribute values  
24 associated with the InUse instance of this application to the attribute values  
25 associated with the InConfiguration instance of this application, and
  - 26 – The access terminal and the access network shall purge the InConfiguration  
27 instance of the application.
- 28 • If the InUse instance of this application (corresponding to the stream or virtual stream  
29 to which this application is bound) does not have the same subtype as this application  
30 instance, then the access terminal and the access network shall perform the following  
31 in the order specified:
- 32 – The initial state of the Flow Control Protocol associated with the  
33 InConfiguration instance of the Packet Application at the access terminal and  
34 access network shall be set to the Close State.
  - 35 – The receive pointer for DataOverSignaling message validation on Route A,  
36  $V(R_A)$ , shall be initialized to 127.

- 1           – The receive pointer for DataOverSignaling message validation on Route B,  
2           V(R<sub>B</sub>), shall be initialized to 127.
- 3           – The transmit pointer for DataOverSignaling message validation on Route A,  
4           V(S<sub>A</sub>), shall be initialized to zero.
- 5           – The transmit pointer for DataOverSignaling message validation on Route B,  
6           V(S<sub>B</sub>), shall be initialized to zero.
- 7           – Forward and reverse link Reservations with ReservationLabel 0xff shall enter  
8           the Open state. All other Reservations shall enter the Close state.<sup>2</sup>
- 9           – The ServiceNetworkID parameter of the Flow Control Protocol shall be  
10          initialized to NULL.
- 11          – The Route Selection Protocol shall enter the A Open B Setting state (See Figure  
12          2.4.4.1.2-1).
- 13          – The InConfiguration instance of the Packet Application at the access terminal  
14          and access network shall become the InUse instance for the Packet Application  
15          (corresponding to the stream or virtual stream to which this application is  
16          bound).
- 17      • All the public data not defined by this application shall be removed from the public data  
18      of the InUse application.

### 19 2.3.3 Message Formats

#### 20 2.3.3.1 ConfigurationRequest

21 The ConfigurationRequest message format is as follows:

22

Field	Length (bits)
MessageID	8
TransactionID	8

Zero or more instances of the following record

AttributeRecord	Attribute dependent
-----------------	---------------------

- 23 **MessageID**           The sender shall set this field to 0x50.
- 24 **TransactionID**       The sender shall increment this value for each new  
25 ConfigurationRequest message sent.
- 26 **AttributeRecord**      The format of this record is specified in [8].  
27

---

<sup>2</sup> Forward and reverse link Reservations 0xff initialized in the Open state so that data can be sent without having to perform a state transition.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	unicast	<b>Priority</b>	40

1    2.3.3.2 ConfigurationResponse

2    The ConfigurationResponse message format is as follows:

3

<b>Field</b>	<b>Length (bits)</b>
MessageID	8
TransactionID	8

Zero or more instances of the following record

AttributeRecord	Attribute dependent
-----------------	---------------------

4    **MessageID**                    The sender shall set this field to 0x51.

5    **TransactionID**                The sender shall set this value to the TransactionID field of the  
6    corresponding ConfigurationRequest message.

7    **AttributeRecord**            An attribute record containing a single attribute value. If this  
8    message selects a complex attribute, only the ValueID field of the  
9    complex attribute shall be included in the message. The format of the  
10   AttributeRecord is given in [8]. The sender shall not include more  
11   than one attribute record with the same attribute identifier.

12

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	unicast	<b>Priority</b>	40

## 2.4 Route Selection Protocol

### 2.4.1 Overview

The Route Selection Protocol provides means to select either instance A or instance B of the Route Protocol. The Route Selection Protocol routes Flow Protocol SDUs to the selected instance of the Route Protocol. Instance A of the Route Protocol is bound to Route A of the Link Flow. Instance B of the Route Protocol is bound to Route B of the Link Flow. The Route Selection Protocol is a protocol associated with the Enhanced Multi-Flow Packet Application. The application subtype for this application is defined in [3].

### 2.4.2 Primitives and Public Data

#### 2.4.2.1 Commands

This protocol does not define any commands.

#### 2.4.2.2 Return Indications

This protocol does not return any indications.

### 2.4.3 Protocol Data Unit

The Route Selection Protocol routes Flow Protocol SDUs to the Route Protocol without modifying them. Hence, the transmission unit of this protocol is a Flow Protocol SDU. The Flow Protocol for a forward Link Flow *NN* is identified by the ProtocolID field of the FlowNNFlowProtocolFwd attribute. The Flow Protocol for a reverse Link Flow *NN* is identified by the ProtocolID field of the FlowNNFlowProtocolRev attribute.

### 2.4.4 Procedures and Messages for the InUse Instance of the Protocol

#### 2.4.4.1 Procedures

##### 2.4.4.1.1 General Requirements

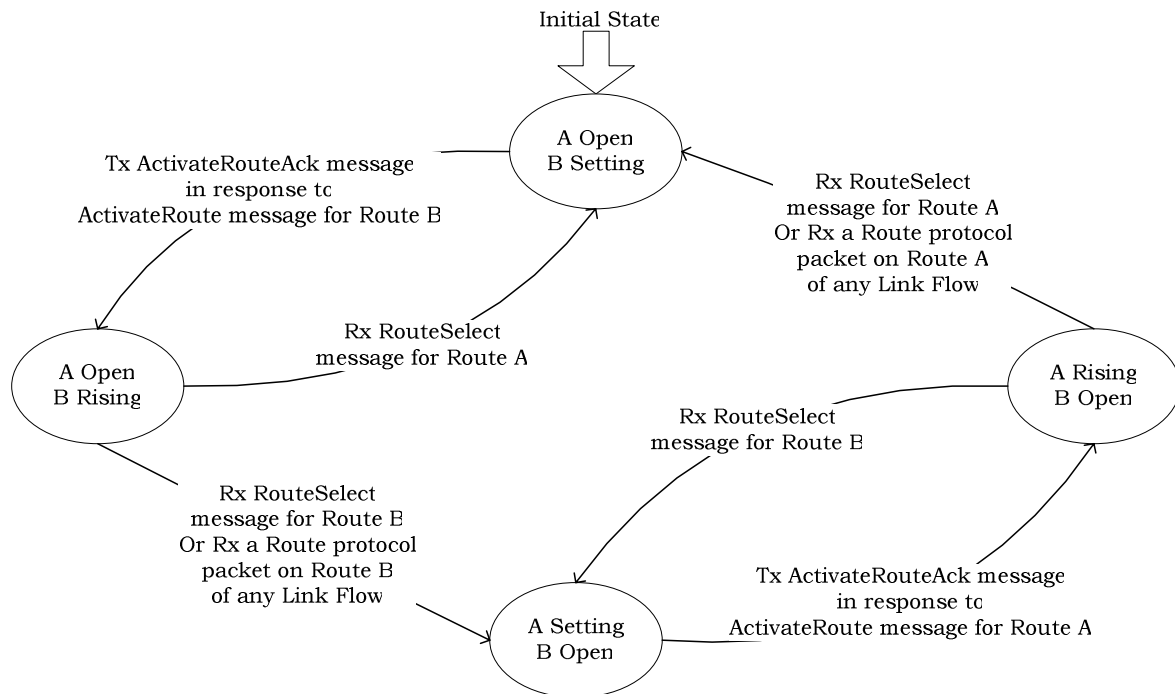
If the FlowNNFlowProtocolSDUFwd attribute of forward Link Flow *NN* is 0x0000, then forward Link Flow *NN* provides an octet stream to the Flow Protocol. If the FlowNNFlowProtocolSDUFwd attribute of forward Link Flow *NN* is 0x0001, then forward Link Flow *NN* provides a packet stream to the Flow Protocol.

If the FlowNNOutOfOrderDeliveryToFlowProtocolFwd attribute of forward Link Flow *NN* is 0x0000, then forward Link Flow *NN* delivers Flow Protocol SDUs in order. If the FlowNNOutOfOrderDeliveryToFlowProtocolFwd attribute of forward Link Flow *NN* is 0x0001, then forward Link Flow *NN* may deliver Flow Protocol SDUs out of order.

If the FlowNNFlowProtocolSDURev attribute of reverse Link Flow *NN* is 0x0000, then reverse Link Flow *NN* provides an octet stream to the Flow Protocol. If the FlowNNFlowProtocolSDURev attribute of reverse Link Flow *NN* is 0x0001, then reverse Link Flow *NN* provides a packet stream to the Flow Protocol.

#### 2.4.4.1.2 Access Terminal Requirements

The Route Selection Protocol associated with an activated Link Flow can be in one of four states: A Open B Setting, A Open B Rising, A Setting B Open, or A Rising B Open. The Route Selection Protocol instance associated with all activated Link Flows shall be in the same state at any time. When a Link Flow is activated, the Route Selection Protocol shall enter the state that the Route Selection Protocols of other activated Link Flows are in. If no other Link Flows are activated when a Link Flow is activated, then the Route Selection Protocol shall enter the A Open B Setting state. Figure 2.4.4.1.2-1 shows the state diagram for the Route Selection Protocol at the access terminal.



**Figure 2.4.4.1.2-1. Route Selection Protocol State Diagram (Access Terminal)**

#### 2.4.4.1.2.1 A Open B Setting State

##### 2.4.4.1.2.1.1 State Transitions

Upon receiving an ActivateRoute message requesting to activate Route B, the access terminal shall perform the following:

- The Route Selection Protocol shall issue a *RadioLinkProtocol.InitializeRoute* command with Route B as the argument.
- The access terminal shall initialize the Route Protocol bound to Route B.
- After the Radio Link Protocol and the Route Protocol are initialized, the access terminal shall send an ActivateRouteAck message, and shall transition to the A Open B Rising state.

1 Upon receiving a RouteSelect message for Route A, the access terminal shall respond with a  
2 RouteSelectAck message.

#### 3 2.4.4.1.2.1.2 Transmitter Requirements

4 The access terminal shall route Flow Protocol SDUs to Route A. The access terminal shall  
5 not route Flow Protocol SDUs to Route B.

#### 6 2.4.4.1.2.1.3 Receiver Requirements

7 The access terminal shall pass Flow Protocol SDUs received on Route A to the Flow  
8 Protocol.

9 If the FlowNNOutOfOrderDeliveryToFlowProtocolFwd attribute for Link Flow *NN* is 0x0001,  
10 then the access terminal shall pass Flow Protocol SDUs received on Route B of the Link  
11 Flow to the Flow Protocol if the access terminal has not received an ActivateRoute message  
12 requesting to activate Route B since the last time it entered this state.

13 If the FlowNNOutOfOrderDeliveryToFlowProtocolFwd attribute for Link Flow *NN* is 0x0000,  
14 then the access terminal shall pass Flow Protocol SDUs received on Route B of the Link  
15 Flow to the Flow Protocol if the access terminal has not passed Flow Protocol SDUs received  
16 on Route A of the Link Flow to the Flow Protocol since the last time the access terminal  
17 entered this state and if the access terminal has not received an ActivateRoute message  
18 requesting to activate Route B since the last time it entered this state. If the  
19 FlowNNOutOfOrderDeliveryToFlowProtocolFwd attribute for Link Flow *NN* is 0x0000, then  
20 the access terminal shall discard Flow Protocol SDUs received on Route B of the Link Flow  
21 if the access terminal has passed Flow Protocol SDUs received on Route A of the Link Flow  
22 to the Flow Protocol since the last time the access terminal entered this state.

#### 23 2.4.4.1.2.2 A Open B Rising State

##### 24 2.4.4.1.2.2.1 State Transitions

25 Upon receiving a RouteSelect message requesting to select Route B, the access terminal  
26 shall respond with a RouteSelectAck message, and shall transition to the A Setting B Open  
27 state. Upon receiving Flow Protocol SDU on Route B of any Link Flow, the access terminal  
28 shall store the Flow Protocol SDU received from Route B for processing in the A Setting B  
29 Open state and shall transition to the A Setting B Open state.

30 Upon receiving a RouteSelect message requesting to select Route A, the access terminal  
31 shall respond with a RouteSelectAck message, and shall transition to the A Open B Setting  
32 state.

##### 33 2.4.4.1.2.2.2 Transmitter Requirements

34 The access terminal shall route Flow Protocol SDUs to Route A. The access terminal shall  
35 not route Flow Protocol SDUs to Route B.

#### 2.4.4.1.2.2.3 Receiver Requirements

The access terminal shall pass Flow Protocol SDUs received on Route A to the Flow Protocol.

#### 2.4.4.1.2.3 A Setting B Open State

##### 2.4.4.1.2.3.1 State Transitions

Upon receiving an *ActivateRoute* message requesting to activate Route A, the access terminal shall perform the following:

- The Route Selection Protocol shall issue a *RadioLinkProtocol.InitializeRoute* command with Route A as the argument.
- The access terminal shall initialize the Route Protocol bound to Route A.
- After the Radio Link Protocol and the Route Protocol are initialized, the access terminal shall send an *ActivateRouteAck* message, and shall transition to the A Rising B Open state.

Upon receiving a *RouteSelect* message for Route B, the access terminal shall respond with a *RouteSelectAck* message.

##### 2.4.4.1.2.3.2 Transmitter Requirements

The access terminal shall route Flow Protocol SDUs to Route B. The access terminal shall not route Flow Protocol SDUs to Route A.

##### 2.4.4.1.2.3.3 Receiver Requirements

The access terminal shall pass Flow Protocol SDUs received on Route B to the Flow Protocol.

If the *FlowNNOOutOfOrderDeliveryToFlowProtocolFwd* attribute for Link Flow *NN* is 0x0001, then the access terminal shall pass Flow Protocol SDUs received on Route A of the Link Flow to the Flow Protocol if the access terminal has not received an *ActivateRoute* message requesting to activate Route A since the last time it entered this state.

If the *FlowNNOOutOfOrderDeliveryToFlowProtocolFwd* attribute for Link Flow *NN* is 0x0000, then the access terminal shall pass Flow Protocol SDUs received on Route A of the Link Flow to the Flow Protocol if the access terminal has not passed Flow Protocol SDUs received on Route B of the Link Flow to the Flow Protocol since the last time the access terminal entered this state and if the access terminal has not received an *ActivateRoute* message requesting to activate Route A since the last time it entered this state. If the *FlowNNOOutOfOrderDeliveryToFlowProtocolFwd* attribute for Link Flow *NN* is 0x0000, then the access terminal shall discard Flow Protocol SDUs received on Route A of the Link Flow if the access terminal has passed Flow Protocol SDUs received on Route B of the Link Flow to the Flow Protocol since the last time the access terminal entered this state.

#### 2.4.4.1.2.4 A Rising B Open State

##### 2.4.4.1.2.4.1 State Transitions

Upon receiving a RouteSelect message requesting to select Route A, the access terminal shall respond with a RouteSelectAck message, and shall transition to the A Open B Setting state. Upon receiving Flow Protocol SDU on Route A of any Link Flow, the access terminal shall store the Flow Protocol SDU received on Route A for processing in the A Open B Setting state and shall transition to the A Open B Setting state.

Upon receiving a RouteSelect message requesting to select Route B, the access terminal shall respond with a RouteSelectAck message, and shall transition to the A Setting B Open state.

##### 2.4.4.1.2.4.2 Transmitter Requirements

The access terminal shall route Flow Protocol SDUs to Route B. The access terminal shall not route Flow Protocol SDUs to Route A.

##### 2.4.4.1.2.4.3 Receiver Requirements

The access terminal shall pass Flow Protocol SDUs received on Route B to the Flow Protocol.

##### 2.4.4.1.3 Access Network Requirements

Upon sending an ActivateRoute message requesting to activate Route A, the access network shall issue a *RadioLinkProtocol.InitializeRoute* command with Route A as the argument and initialize the Route Protocol bound to Route A.

Upon sending an ActivateRoute message requesting to activate Route B, the access network shall issue a *RadioLinkProtocol.InitializeRoute* command with Route B as the argument and initialize the Route Protocol bound to Route B.

#### 2.4.4.2 Message Formats

##### 2.4.4.2.1 RouteSelect

The access network sends this message to transition the access terminal to the A Open B Setting or the A Setting B Open state.

Field	Length (bits)
MessageID	8
TransactionID	8
Route	1
Reserved	7

MessageID                      The access network shall set this field to 0x1e.

- 1 TransactionID            The access network shall set this field to one more (modulo 256) than  
2 the TransactionID field of the last RouteSelect message sent by the  
3 access network.
- 4 Route                    The access network shall set this field to '0' to transition the access  
5 terminal to the A Open B Setting state. The access network shall set  
6 this field to '1' to transition the access terminal to the A Setting B  
7 Open state.
- 8 Reserved                The access network shall set this field to '0000000'. The access  
9 terminal shall ignore this field.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast	<b>Priority</b>	40

11 2.4.4.2.2 RouteSelectAck

12 The access terminal sends this message to acknowledge the receipt of a RouteSelect  
13 message.

Field	Length (bits)
MessageID	8
TransactionID	8

- 15 MessageID                The access terminal shall set this field to 0x1f.
- 16 TransactionID            The access terminal shall set this field to the TransactionID field of  
17 the RouteSelect message whose receipt is being acknowledged by this  
18 message.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast	<b>Priority</b>	40

20 2.4.4.2.3 ActivateRoute

21 The access network sends this message to transition the access terminal to the A Rising B  
22 Open state or the A Open B Rising state.

23

<b>Field</b>	<b>Length (bits)</b>
MessageID	8
TransactionID	8
Route	1
Reserved	7

- 1 **MessageID** The access network shall set this field to 0x20.
- 2 **TransactionID** The access network shall set this field to one more (modulo 256) than  
3 the TransactionID field of the last ActivateRoute message sent by the  
4 access network.
- 5 **Route** The access network shall set this field to '0' to transition the access  
6 terminal to the A Rising B Open state. The access network shall set  
7 this field to '1' to transition the access terminal to the A Open B  
8 Rising state.
- 9 **Reserved** The access network shall set this field to '0000000'. The access  
10 terminal shall ignore this field.  
11

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	unicast	<b>Priority</b>	40

#### 12 2.4.4.2.4 ActivateRouteAck

13 The access terminal sends this message to acknowledge the receipt of an ActivateRoute  
14 message.  
15

<b>Field</b>	<b>Length (bits)</b>
MessageID	8
TransactionID	8

- 16 **MessageID** The access terminal shall set this field to 0x21.
- 17 **TransactionID** The access terminal shall set this field to the TransactionID field of  
18 the ActivateRoute message whose receipt is being acknowledged by  
19 this message.  
20

<b>Channels</b>	RTC
<b>Addressing</b>	unicast

<b>SLP</b>	Reliable
<b>Priority</b>	40

1 2.4.4.3 Interface to Other Protocols

2 2.4.4.3.1 Commands

3 This protocol issue the following commands:

- 4 • *RadioLinkProtocol.InitializeRoute* with argument indicating which Route is to be  
5 initialized.

6 2.4.4.3.2 Indications

7 This protocol does not register to receive any indications.

8 2.4.5 Protocol Numeric Constants

9 This protocol does not define any protocol numeric constants.

## 2.5 Radio Link Protocol

### 2.5.1 Overview

The Radio Link Protocol (RLP) provides one or more octet or packet streams with an acceptably low erasure rate for efficient operation of higher layer protocols (e.g., TCP). When used as part of the Enhanced Multi-Flow Packet Application, the protocol carries one or more octet or packet streams from the higher layer. RLP is a protocol associated with the Enhanced Multi-Flow Packet Application. The application subtype for this application is defined in [3].

### 2.5.2 Primitives and Public Data

#### 2.5.2.1 Commands

This protocol defines the following commands:

- *InitializeRoute* with argument indicating which Route is to be initialized.

#### 2.5.2.2 Return Indications

This protocol does not return any indications.

### 2.5.3 Protocol Data Unit

The transmission unit of this protocol is an RLP packet.

### 2.5.4 Procedures and Messages for the InUse Instance of the Protocol

A forward Link Flow *NN* is defined to be activated if the *FlowNNActivatedFwd* attribute is set to 0x0001, where *NN* is the hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}} - 1$  inclusive. The number of activated Link Flows on the forward link shall not exceed the value of the *LinkFlowCountMaxFwd* attribute.

A reverse Link Flow *NN* is defined to be activated if the *FlowNNActivatedRev* attribute is set to 0x0001. The number of activated Link Flows on the reverse link shall not exceed the value of the *LinkFlowCountMaxRev* attribute.

A Link Flow is defined to be deactivated if it is not activated.

This section defines the procedures and messages for the in-use instance of each forward or reverse Link Flow.

#### 2.5.4.1 Procedures

Each Route of the Link Flow receives octets or packets for transmission from the corresponding instance of the Route Protocol and forms an RLP packet by prepending the RLP packet header defined in 2.5.4.3 with a number of received contiguous octets.

The Route Protocol for a forward Link Flow *NN* is identified by the *FlowNNRouteProtocolIDFwd* attribute. The Route Protocol for a reverse Link Flow *NN* is identified by the *FlowNNRouteProtocolIDRev* attribute.

1 If the Route Protocol is NULL<sup>3</sup>, then the transmitter shall set Route Protocols octets or  
2 packets to Flow Protocol octets or packets routed along the Route. If the Route Protocol is  
3 NULL, then the receiver shall set Flow Protocols octets or packets to Route Protocol packets  
4 or octets received on the Route.

5 If the FlowNNRouteProtocolSDUFwd attribute of forward Link Flow *NN* is 0x0000, then each  
6 Route of forward Link Flow *NN* provides an octet stream to the corresponding instance of  
7 the Route Protocol. If the FlowNNRouteProtocolSDUFwd attribute of forward Link Flow *NN*  
8 is 0x0001, then each Route of forward Link Flow *NN* provides a packet stream to the  
9 corresponding instance of the Route Protocol.

10 If the FlowNNOutOfOrderDeliveryToRouteProtocolFwd attribute of forward Link Flow *NN* is  
11 0x0000, then each Route of forward Link Flow *NN* delivers payload to the corresponding  
12 instance of the Route Protocol in order. If the  
13 FlowNNOutOfOrderDeliveryToRouteProtocolFwd attribute of forward Link Flow *NN* is  
14 0x0001, then each Route of forward Link Flow *NN* may deliver payload to the corresponding  
15 instance of the Route Protocol out of order.

16 If the FlowNNRouteProtocolSDURev attribute of reverse Link Flow *NN* is 0x0000, then each  
17 Route of reverse Link Flow *NN* provides an octet stream to the corresponding instance of the  
18 Route Protocol. If the FlowNNRouteProtocolSDURev attribute of reverse Link Flow *NN* is  
19 0x0001, then each Route of reverse Link Flow *NN* provides a packet stream to the  
20 corresponding instance of the Route Protocol.

21 If the FlowNNOutOfOrderDeliveryToRouteProtocolRev attribute of reverse Link Flow *NN* is  
22 0x0000, then each Route of reverse Link Flow *NN* delivers payload to the corresponding  
23 instance of the Route Protocol in order. If the  
24 FlowNNOutOfOrderDeliveryToRouteProtocolRev attribute of reverse Link Flow *NN* is 0x0001,  
25 then each Route of reverse Link Flow *NN* may deliver payload to the corresponding instance  
26 of the Route Protocol out of order.

27 The policy RLP follows in determining the number of octets to send in an RLP packet is  
28 beyond the scope of this specification. It is subject to the following requirements:

- 29 • the size of an RLP packet shall not exceed the maximum payload length that can be  
30 carried by the Stream Layer given the target channel and current transmission rate on  
31 that channel,
- 32 • if the Link Flow is carrying a packet stream, then an RLP packet shall contain octets  
33 from no more than one Route Protocol packet, and
- 34 • the RLP packet shall contain all octets of the Route Protocol packet if all of the following  
35 conditions are true:
  - 36 – the size of the RLP packet carrying all octets of the Route Protocol packet does  
37 not exceed the maximum payload length that can be carried by the Stream  
38 Layer given the target channel and the current transmission rate on that  
39 channel,

---

<sup>3</sup> <sup>3</sup> Route Protocol being NULL means that a Route Protocol has not been negotiated.

- 1           – the Link flow is carrying a packet stream, and
- 2           – the data unit for the Link Flow is RLP packet payloads.

3 For the purpose of binding Link Flows to lower layer flows (such as MAC flows in the  
4 Subtype 3 Reverse Traffic Channel MAC Protocol specified in [8]), each Link Flow shall be  
5 treated as a substream with the substream number set to the Link Flow number.

6 When forward Link Flow *NN* is activated, the access network and the access terminal shall  
7 not update the following attributes:

- 8 • FlowNNFlowProtocolFwd,
- 9 • FlowNNRouteProtocolFwd,
- 10 • FlowNNSequenceLengthFwd,
- 11 • FlowNNFlowProtocolSDUFwd,
- 12 • FlowNNRouteProtocolSDUFwd,
- 13 • FlowNNDataUnitFwd,
- 14 • FlowNNOutOfOrderDeliveryToFlowProtocolFwd, and
- 15 • FlowNNOutOfOrderDeliveryToRouteProtocolFwd.

16 The access terminal shall not initiate activation of a deactivated forward Link Flow *NN*  
17 unless the access terminal initiated deactivation of forward Link Flow *NN* since the last time  
18 the Link Flow was activated, and the deactivation was successful. The access network shall  
19 not initiate activation of a deactivated forward Link Flow *NN* unless the access network  
20 initiated deactivation of forward Link Flow *NN* since the last time the Link Flow was  
21 activated, and the deactivation was successful.

22 When reverse Link Flow *NN* is activated, the access network and the access terminal shall  
23 not update the following attributes:

- 24 • FlowNNFlowProtocolRev,
- 25 • FlowNNRouteProtocolRev,
- 26 • FlowNNSequenceLengthRev,
- 27 • FlowNNFlowProtocolSDURev,
- 28 • FlowNNRouteProtocolSDURev,
- 29 • FlowNNDataUnitRev, and
- 30 • FlowNNOutOfOrderDeliveryToRouteProtocolRev.

31 The access terminal shall not initiate activation of a deactivated reverse Link Flow *NN*  
32 unless the access terminal initiated deactivation of reverse Link Flow *NN* since the last time  
33 the Link Flow was activated, and the deactivation was successful. The access network shall  
34 not initiate activation of a deactivated reverse Link Flow *NN* unless the access network  
35 initiated deactivation of reverse Link Flow *NN* since the last time the Link Flow was  
36 activated, and the deactivation was successful.

1 The ProtocolID field of the FlowNNFlowProtocolFwd attribute shall be set to a value that is  
2 included as ProtocolIdentifier in the SupportedFlowProtocolIDs attribute. The ProtocolID  
3 field of the FlowNNFlowProtocolRev attribute shall be set to a value that is included as  
4 ProtocolIdentifier in the SupportedFlowProtocolIDs attribute. The ProtocolID field of the  
5 FlowNNRouteProtocolFwd attribute shall be set to a value that is included as  
6 ProtocolIdentifier in the SupportedRouteProtocolIDs attribute. The ProtocolID field of the  
7 FlowNNRouteProtocolRev attribute shall be set to a value that is included as  
8 ProtocolIdentifier in the SupportedRouteProtocolIDs attribute.

9 The access terminal shall not initiate negotiation of the ANSupportedQoSProfiles attribute.  
10 The access network shall not initiate negotiation of the ATSupportedQoSProfiles attribute.  
11 The access terminal shall include all supported values of ProfileValue with ProfileType equal  
12 to 0x01 in the ATSupportedQoSProfiles attribute during the AT Initiated state of the Session  
13 Configuration Protocol.

14 If the FlowNNDataUnitFwd attribute of forward Link Flow NN is 0x0000, then the data unit  
15 for the Link Flow shall be octets. Otherwise the data unit for the Link Flow shall be RLP  
16 packet payloads. If the FlowNNDataUnitRev attribute reverse Link Flow NN is 0x0000, then  
17 the data unit for the Link Flow shall be octets. Otherwise the data unit for the Link Flow  
18 shall be RLP packet payloads.

#### 19 2.5.4.1.1 Initialization and Reset

20 The RLP initialization procedure initializes the RLP variables and data structures in one end  
21 of the link. The RLP reset procedure guarantees that RLP state variables on both sides are  
22 synchronized. The reset procedure includes initialization.

23 The access terminal and the access network shall perform the initialization procedures  
24 defined in 2.5.4.1.1.1.1 and 2.5.4.1.1.1.2 for both routes of all activated Link Flows if the  
25 protocol receives an *IdleState.ConnectionOpened* indication. The access network shall  
26 perform the initialization procedure defined in 2.5.4.1.1.1.1 for both routes of forward Link  
27 Flow NN when forward Link Flow NN is activated. The access terminal shall perform the  
28 initialization procedure defined in 2.5.4.1.1.1.2 for both routes of forward Link Flow NN  
29 when forward Link Flow NN is activated. The access terminal shall perform the initialization  
30 procedure defined in 2.5.4.1.1.1.1 for both routes of reverse Link Flow NN when reverse  
31 Link Flow NN is activated. The access network shall perform the initialization procedure  
32 defined in 2.5.4.1.1.1.2 for both routes of reverse Link Flow NN when reverse Link Flow NN  
33 is activated. Upon receiving an *InitializeRoute* command, the access terminal shall perform  
34 the initialization procedures defined in 2.5.4.1.1.1 for the specified Route of all activated  
35 Link Flows. Upon receiving an *InitializeRoute* command, the access network shall perform  
36 the initialization procedures defined in 2.5.4.1.1.1 for the specified Route of all activated  
37 Link Flows.

#### 38 2.5.4.1.1.1 Initialization Procedure

##### 39 2.5.4.1.1.1.1 Initialization Procedure for the RLP Transmitter

40 When RLP transmitter performs the initialization procedure it shall:

- 1 • Reset the send state variable  $V(S)_{NN,P}$  to zero, where  $NN$  indicates the Link Flow, and  $P$   
2 indicates the Route which is being initialized, and  
3 • clear the retransmission queues.

#### 4 2.5.4.1.1.1.2 Initialization Procedure for the RLP Receiver

5 When RLP receiver performs the initialization procedure it shall:

- 6 • Reset the receive state variables  $V(R)_{NN,P}$  and  $V(N)_{NN,P}$  to zero, and  
7 • clear the resequencing buffer.

#### 8 2.5.4.1.1.2 Reset Procedure

##### 9 2.5.4.1.1.2.1 Reset Procedure for the Initiating Side when it is an RLP Transmitter

10 If the side initiating a reset procedure is an RLP transmitter for the Route of the Link Flow  
11 (or of all Link Flows) being reset, then it shall send a ResetTxIndication message and enter  
12 the RLP Reset State.

13 Upon entering the RLP Reset state RLP transmitter shall:

- 14 • Perform the RLP transmitter initialization procedure defined in 2.5.4.1.1.1.1 for the  
15 Route being reset.  
16 • If a ReceiverStatus message is received for the Route of the Link Flow being reset while  
17 in the RLP Reset state, the message shall be ignored.  
18 • If the RLP transmitter that initiated the reset procedure is an access terminal, and if a  
19 *PhysicalLayer.ReverseTrafficPacketsMissed* indication is received for the Route of the  
20 Link Flow being reset while RLP is in the Reset state, then the indication shall be  
21 ignored.  
22 • The RLP transmitter should not transmit RLP packets while in the RLP Reset state.  
23 • If the RLP transmitter receives a ResetTxIndicationAck message for the Route of the  
24 Link Flow being reset while in the RLP Reset state, the RLP transmitter shall send a  
25 ResetTxComplete message back and leave the RLP Reset state.

26 If a ResetTxIndicationAck message is received for a Route while that Route is not in the RLP  
27 Reset state, the message shall be ignored.

##### 28 2.5.4.1.1.2.2 Reset Procedure for Initiating Side when it is an RLP Receiver

29 If the side initiating a reset procedure is an RLP receiver for the Route of the Link Flow  
30 being reset, then it shall send a ResetRxIndication message and enter the RLP Reset State.

31 Upon entering the RLP Reset state, the RLP receiver shall:

- 32 • Perform the RLP receiver initialization procedure defined in 2.5.4.1.1.1.2 for the Route  
33 being reset.  
34 • Ignore all RLP data units received for the Route of the Link Flow being reset while in the  
35 RLP Reset state.

- When the RLP receiver receives a ResetRxComplete message for the Route of the Link Flow being reset, the RLP receiver shall leave the RLP Reset state.

If a ResetRxComplete is received for a Route while the Route is not in the RLP Reset state, the message shall be ignored.

2.5.4.1.1.2.3 Reset Procedure for the Responding Side when it is an RLP Receiver

If the side responding to a reset procedure is an RLP receiver for the Route of the Link Flow being reset, then it shall respond with a ResetTxIndicationAck message upon receiving a ResetTxIndication message. After sending the message it shall enter the RLP Reset state for the Route being reset, if it was not already in the RLP Reset state. Upon entering the RLP Reset state RLP shall:

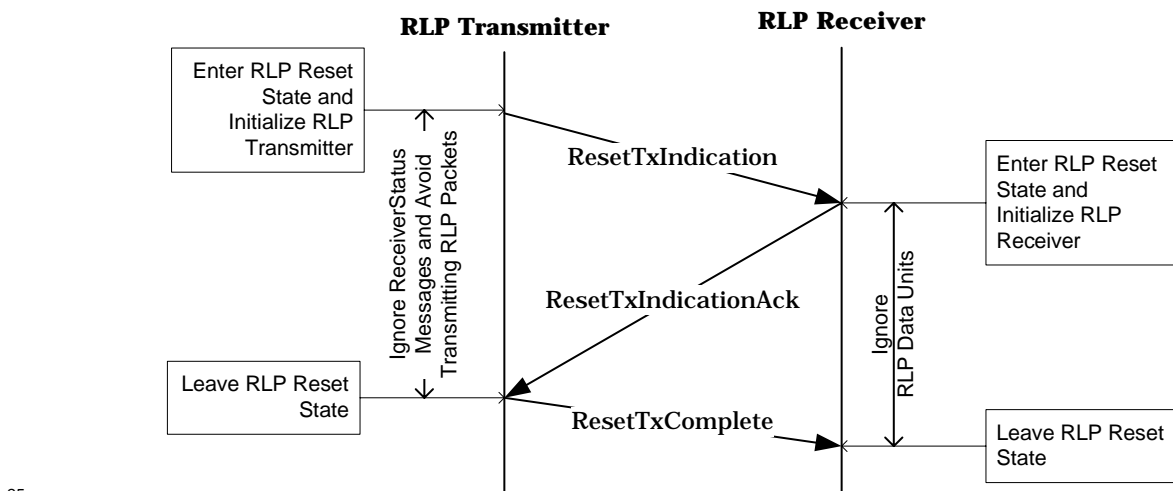
- Perform the RLP receiver initialization procedure defined in 2.5.4.1.1.1.2 for the Route being reset.
- Ignore all RLP data units received for the Route of the Link Flow being reset while in the RLP Reset state.
- When RLP receives a ResetTxComplete message for the Route of the Link Flow being reset, it shall leave the RLP Reset state.

If a ResetTxComplete message is received for a Route while the Route is not in the RLP Reset state, the message shall be ignored.

2.5.4.1.1.2.4 Reset Procedure for the Responding Side when it is an RLP transmitter

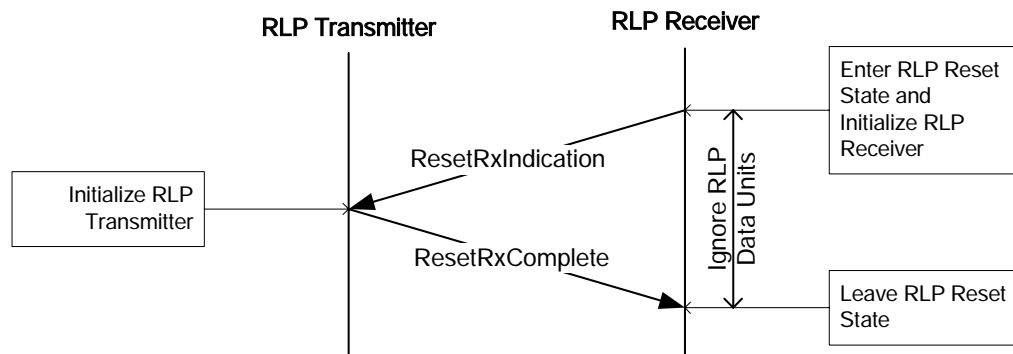
If the side responding to a reset procedure is an RLP transmitter for the Route being reset, then it shall respond with a ResetRxComplete message upon receiving a ResetRxIndication message. After sending the message, it shall perform the RLP transmitter initialization procedure defined in 2.5.4.1.1.1.1 for the Route being reset.

2.5.4.1.1.2.5 RLP Reset Message Flows



25  
26

Figure 2.5.4.1.1.2.5-1. RLP Reset Procedure Initiated by RLP Transmitter



1  
2 **Figure 2.5.4.1.1.2.5-2. RLP Reset Procedure Initiated by RLP Receiver**

3 **2.5.4.2 Data Transfer**

4 RLP is an Ack and/or Nak-based protocol with a sequence space of SequenceLength bits,  
5 where SequenceLength is indicated by the FlowNNSequenceLengthFwd and  
6 FlowNNSequenceLengthRev attribute for forward and reverse Link Flow NN, respectively.

7 All operations and comparisons performed on RLP packet sequence numbers shall be  
8 carried out in unsigned modulo  $2^S$  arithmetic, where S represents the value of  
9 SequenceLength. For any RLP sequence number N, the sequence numbers in the range  
10  $[N+1, N+2^{S-1}-1]$  shall be considered greater than N and the sequence numbers in the range  
11  $[N-2^{S-1}, N-1]$  shall be considered smaller than N.

12 **2.5.4.2.1 RLP Transmit Procedures**

13 The RLP transmitter shall maintain a SequenceLength-bit variable  $V(S)_{NN,P}$  for all  
14 transmitted RLP data units (see Figure 2.5.4.2.1-1), where NN is the two-digit hexadecimal  
15 Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive, and P is the Route indicator  
16 that takes values of either A or B.  $V(S)_{NN,P}$  is the sequence number of the next RLP data unit  
17 to be sent on Route P of Link Flow NN. The sequence number field (SEQ) in each new RLP  
18 packet transmitted shall be set to  $V(S)_{NN,P}$ , corresponding to the sequence number of the  
19 first data unit in the packet. If the data unit is octets, then the sequence number of the  $i^{\text{th}}$   
20 octet in the packet (with the first octet being octet 0) is implicitly given by  $\text{SEQ}+i$ .  $V(S)_{NN,P}$   
21 shall be incremented for each data unit contained in the packet. The RLP transmitter  
22 should allow sufficient time before deleting an RLP packet payload transmitted for the first  
23 time.

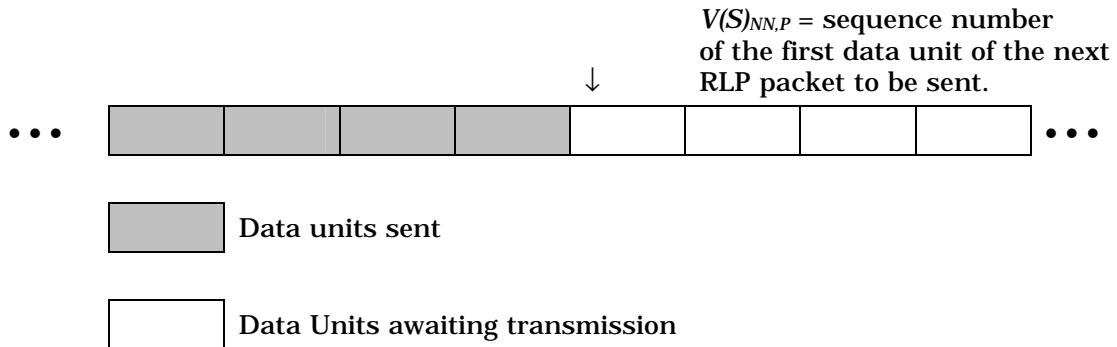
24 Upon receiving a ReceiverStatus message, RLP shall transmit the missing data unit(s) (if  
25 any) conveyed by the ReceiverStatus message if those data units are available and if those  
26 data units have not been retransmitted before in response to a ReceiverStatus message.

27 If the RLP transmitter is the access network and if FlowNNAckEnableFwd is not 0x0000,  
28 then the access network may determine that transmitted data units have been lost if it does  
29 not receive a ReceiverStatus message acknowledging the receipt of the data units within an  
30 implementation dependent time interval based on the AckTimer and an estimate of the  
31 round-trip delay. If the RLP transmitter is the access network and if FlowNNAckEnableFwd  
32 is not 0x0000 and the access network determines that transmitted data units were lost,

1 then the access network shall re-transmit the data units if they have not been re-  
 2 transmitted in response to a ReceiverStatus message.

3 If the RLP transmitter is the access terminal and if FlowNNAckEnableRev is not 0x0000,  
 4 then the access terminal may determine that transmitted data units have been lost if it  
 5 does not receive a ReceiverStatus message acknowledging the receipt of the data units  
 6 within an implementation dependent time interval based on the AckTimer and an estimate  
 7 of the round-trip delay. If the RLP transmitter is the access terminal and if  
 8 FlowNNAckEnableRev is not 0x0000 and the access terminal determines that transmitted  
 9 data units were lost, then the access terminal shall re-transmit the data units if they have  
 10 not been re-transmitted in response to a ReceiverStatus message.

11 If the RLP transmitter is the access network, and the ReceiverStatus message includes any  
 12 sequence number greater than or equal to  $V(S)_{NN,P}$ , RLP shall perform the reset procedures  
 13 specified in 2.5.4.1.1.2.1 for Route  $P$  of forward Link Flow  $NN$ . If the RLP transmitter is the  
 14 access terminal, and the ReceiverStatus message includes any sequence number greater  
 15 than or equal to  $V(S)_{NN,P}$ , RLP shall perform the reset procedures specified in 2.5.4.1.1.2.1  
 16 for Route  $P$  of reverse Link Flow  $NN$ . If the ReceiverStatus message does not include any  
 17 sequence number greater than or equal to  $V(S)_{NN,P}$  but the requested data units are not  
 18 available for retransmissions, RLP shall ignore the ReceiverStatus message for data units  
 19 that are not available.  
 20



21 **Figure 2.5.4.2.1-1. RLP Transmit Sequence Number Variable**

22 Upon receiving a *PhysicalLayer.ReverseTrafficPacketsMissed* indication for reverse Link  
 23 Flow  $NN$ , the RLP transmitter in the access terminal shall transmit the requested data  
 24 units(s) if and only if all of the following conditions are satisfied:

- 25 • FlowNNPhysicalLayerNakEnableRev attribute is set to 0x01,
- 26 • the requested data units have not been retransmitted before, and
- 27 • the requested data units are available.

28 If FlowNNNakEnableFwd is 0x0001, then the transmitter at the access network for each  
 29 Route of Link Flow  $NN$  shall meet the following requirements:

- 30 • After transmitting a packet, the RLP transmitter shall start an RLP flush timer for time  
 31 FlushTimer, where FlushTimer is a parameter of the FlowNNTimersFwd attribute.

- 1 • If the RLP transmitter sends another packet before the RLP flush timer expires, the RLP  
2 transmitter shall reset and restart the timer.
- 3 • If the timer expires, the RLP transmitter shall disable the flush timer and the RLP  
4 transmitter should send an RLP packet that contains at least the data unit with  
5 sequence number  $V(S)_{NN,P}-1$ .

6 If  $FlowNNNakEnableRev$  is 0x0001, then the transmitter at the access terminal for each  
7 Route of Link Flow  $NN$  shall meet the following requirements:

- 8 • After transmitting a packet, the RLP transmitter shall start an RLP flush timer for time  
9  $FlushTimer$ , where  $FlushTimer$  is a parameter of the  $FlowNNTimersRev$  attribute.
- 10 • If the RLP transmitter sends another packet before the RLP flush timer expires, the RLP  
11 transmitter shall reset and restart the timer.
- 12 • If the timer expires, the RLP transmitter shall disable the flush timer and the RLP  
13 transmitter should send an RLP packet that contains at least the data unit with  
14 sequence number  $V(S)_{NN,P}-1$ .

15 The RLP transmitter should not transmit more than  $2^{SequenceLength-1}$  first-time data units in  
16 any  $AbortTimer$  interval, where  $SequenceLength$  is the length of the SEQ field in the RLP  
17 header for the corresponding Link Flow.

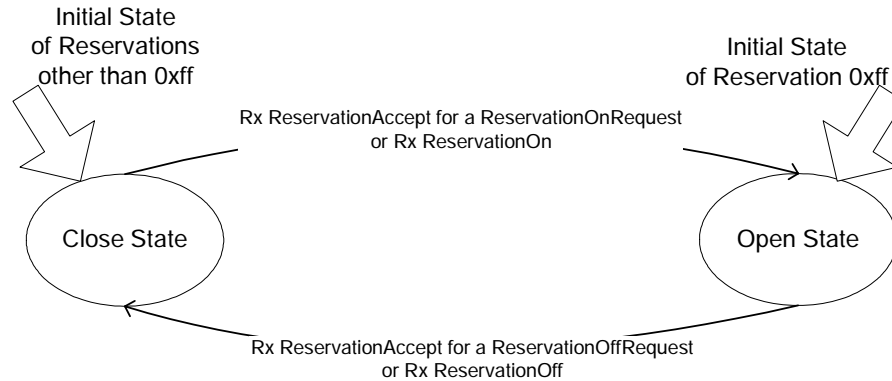
18 The  $ReservationLabel$  parameter of the  $FlowNNReservationFwd$  or  $FlowNNReservationRev$   
19 attribute identifies the higher layer flows associated with Link Flow  $NN$ . Each  
20  $ReservationLabel$  shall be associated with no more than one forward Link Flow. Each  
21  $ReservationLabel$  shall be associated with no more than one reverse Link Flow.

22 Each Reservation can be in one of the following two states:

- 23 • Close State
- 24 • Open State

25 The transmitter should transmit higher layer octets or packets using the Link Flow  
26 associated with the higher layer flow if the associated Link Flow is activated and if the  
27 Reservation is in the Open state. The transmitter should transmit higher layer octets or  
28 packets belonging to a higher layer flow that is not associated with any Link Flow using the  
29 Link Flow with  $ReservationLabel$  0xff. The transmitter may transmit higher layer octets or  
30 packets belonging to a higher layer flow identified by a Reservation that is in the Close state  
31 using the Link Flow with  $ReservationLabel$  0xff.

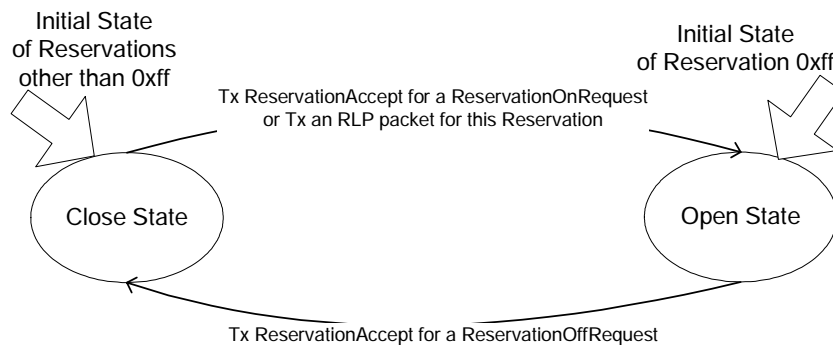
32 Figure 2.5.4.2.1-2 and Figure 2.5.4.2.1-3 show the state transition diagram at the access  
33 terminal and the access network. State transitions that may be caused by  
34 *ConnectedState.ConnectionClosed* and *RouteUpdate.ConnectionLost* indications are not  
35 shown.



1

2

**Figure 2.5.4.2.1-2. Reverse Link Reservation State Diagram (Access Terminal)**



3

4

**Figure 2.5.4.2.1-3. Forward Link Reservation State Diagram (Access Network)**

5

#### 2.5.4.2.1.1 Close State

6

##### 2.5.4.2.1.1.1 Access Terminal Requirements

7 The access terminal shall not transmit SDUs from higher layer flows belonging to the  
 8 Reservation in Close state using any Link Flow other than the Link Flow associated with  
 9 ReservationLabel 0xff. The access terminal may send a ReservationOnRequest message to  
 10 request transition of the Reservation to the Open state. The access terminal may re-send a  
 11 ReservationOnRequest message if it does not receive a corresponding ReservationAccept or  
 12 ReservationReject message within the time period specified by  $T_{\text{Response}}$  of sending the  
 13 ReservationOnRequest message. If the ReservationOnRequest message contains a  
 14 Reservation bound to a reverse Link Flow, then the Reservation shall transition to the Open  
 15 state when the access terminal receives the corresponding ReservationAccept message.

16 Upon receiving a ReservationOn message, the access terminal shall respond with a  
 17 ReservationAccept message and transition the Reservation to the Open state.

##### 2.5.4.2.1.1.2 Access Network Requirements

19 If the Reservation entered this state as a result of any condition other than the following  
 20 conditions, then the access network shall send a FwdReservationOff message upon entering  
 21 this state:

- 1 • the access network transmitted a ReservationAccept message in response to a  
2 ReservationOffRequest message requesting to transition the Reservation to the Close  
3 state, or
- 4 • ReservationKKIdleState attribute of the Reservation is 0x01, and the Reservation  
5 transitioned to the Close state because the Connection was closed or lost.

6 If the access network receives a ReservationOnRequest message, it shall

- 7 • Send either a ReservationAccept message or a ReservationReject message within the  
8 time period specified by  $T_{\text{Response}}$  of reception of the ReservationOnRequest message.
- 9 • Set the TransactionID field of the ReservationAccept or ReservationReject message to  
10 that of the ReservationOnRequest message.
- 11 • Transition the Reservation to the Open State if the access network sent a  
12 ReservationAccept message for a Reservation bound to a forward Link flow in response  
13 to a ReservationOnRequest message.

14 The access network may transmit SDUs from higher layer flows belonging to this  
15 Reservation using the Link Flow to which the Reservation is bound. Upon doing so, the  
16 access network shall transition the Reservation to the Open State.

#### 17 2.5.4.2.1.2 Open State

##### 18 2.5.4.2.1.2.1 Access Terminal Requirements

19 The access terminal may transmit SDUs from higher layer flows belonging to the  
20 Reservation in Open state using the Link Flow to which the Reservation is bound.

21 The access terminal may send a ReservationOffRequest message to request the transition of  
22 a Reservation to the Close state. The access terminal may re-send a ReservationOffRequest  
23 message if it does not receive a ReservationAccept or ReservationReject message within the  
24 time period specified by  $T_{\text{Response}}$  of sending the ReservationOffRequest message. If the  
25 ReservationOffRequest message contains a Reservation bound to a reverse Link Flow, then  
26 the access terminal shall transition the Reservation to the Close state when the access  
27 terminal receives a ReservationAccept message.

28 Upon receiving a ReservationOff message, the access terminal shall respond with a  
29 ReservationAccept message and transition the Reservation to the Close state. Upon  
30 receiving a *ConnectedState.ConnectionClosed* or *RouteUpdate.ConnectionLost* indication, the  
31 access terminal shall transition to the Close state Reservations whose corresponding  
32 ReservationKKIdleStateRev attribute is 0x01, where *KK* is the two-digit hexadecimal  
33 ReservationLabel.

##### 34 2.5.4.2.1.2.2 Access Network Requirements

35 The access network may transmit SDUs from higher layer flows belonging to this  
36 Reservation using the Link Flow to which the Reservation is bound.

37 If the access network receives a ReservationOffRequest message, it shall

- 1 • Send a ReservationAccept or a ReservationReject message within the time period  
2 specified by  $T_{Response}$  of reception of the ReservationOffRequest message.
- 3 • Set the TransactionID field of the ReservationAccept or ReservationReject message to  
4 that of the ReservationOffRequest message.
- 5 • Transition the Reservation to the Close State if it sent a ReservationAccept message for  
6 a Reservation bound to a forward Link Flow in response to a ReservationOffRequest  
7 message.

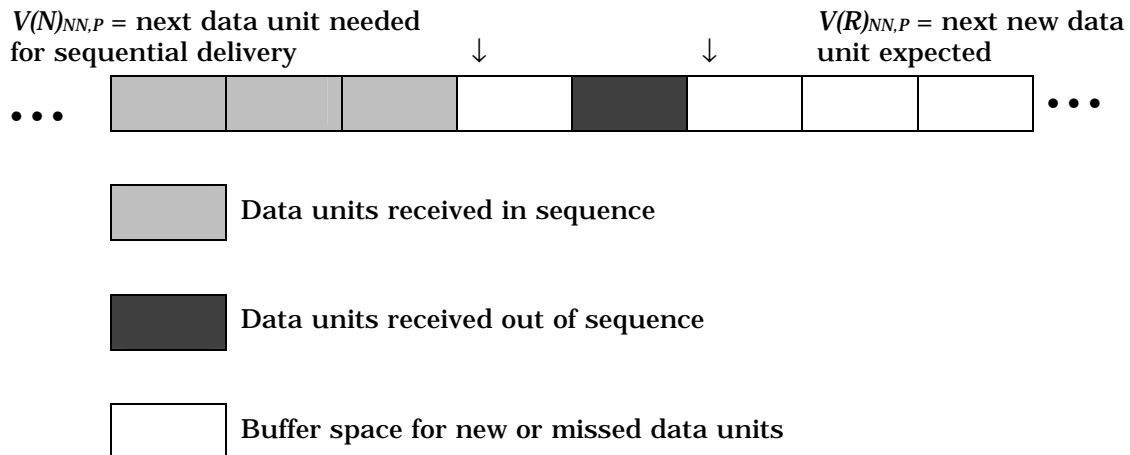
8 Upon receiving a *ConnectedState.ConnectionClosed* or *RouteUpdate.ConnectionLost*  
9 indication, the access network shall transition to the Close state Reservations whose  
10 corresponding ReservationKKIdleStateFwd attribute is 0x01, where KK is the two-digit  
11 hexadecimal ReservationLabel.

12 **2.5.4.2.2 RLP Receive Procedures**

13 The RLP receiver shall maintain two SequenceLength-bit variables for receiving,  $V(R)_{NN,P}$  and  
14  $V(N)_{NN,P}$  (see Figure 2.5.4.2.2-1), where NN is the two-digit hexadecimal Link Flow number in  
15 the range 0x00 to  $N_{LinkFlowMax}-1$  inclusive, and P is the Route indicator that takes values of  
16 either A or B.  $V(R)_{NN,P}$  contains the sequence number of the next data unit expected to  
17 arrive.  $V(N)_{NN,P}$  contains the sequence number of the first missing data unit, as described  
18 below.

19 In addition, the RLP receiver shall keep track of the status of each data unit in its  
20 resequencing buffer indicating whether the data unit was received or not. Use of this status  
21 is implied in the following procedures. The RLP receiver informs the RLP transmitter of the  
22 status of data units in its receive buffer by sending a ReceiverStatus message. The  
23 ReceiverStatus message shall convey status of all missing data units from  $V(N)_{NN,P}$  onwards  
24 for which the status has not been conveyed in a previous ReceiverStatus message and  
25  $V(R)_{NN,P}$ . The ReceiverStatus message may convey status of missing data units for which  
26 status has been conveyed in previous ReceiverStatus messages. The ReceiverStatus  
27 message shall not convey status of data units with sequence number less than  $V(N)_{NN,P}$ .

28



29

**Figure 2.5.4.2.2-1. RLP Receive Sequence Number Variables**

1 In the following,  $X$  denotes the sequence number of a received data unit. For each received  
2 data unit, RLP shall perform the following procedures:

- 3 • If the RLP receiver is an access terminal and FlowNNAckEnableFwd is 0x0002 and the  
4 LastDataUnit field of the RLP header is '1', then RLP shall send a ReceiverStatus  
5 message upon receiving the data unit.
- 6 • If the RLP receiver is an access terminal and FlowNNAckEnableFwd is 0x0002 and the  
7 LastDataUnit field of the RLP header is '0', then RLP shall send a ReceiverStatus  
8 message within an interval specified by AckTimer of receiving the data unit.
- 9 • If the RLP receiver is an access terminal and FlowNNAckEnableFwd is 0x0001, then RLP  
10 shall send a ReceiverStatus message within AckTimer interval of receiving the data unit.
- 11 • If the RLP receiver is an access network and FlowNNAckEnableRev is 0x0002 and the  
12 LastDataUnit field of the RLP header is '1', then RLP shall send a ReceiverStatus  
13 message upon receiving the data unit.
- 14 • If the RLP receiver is an access network and FlowNNAckEnableRev is 0x0002 and the  
15 LastDataUnit field of the RLP header is '0', then RLP shall send a ReceiverStatus  
16 message within AckTimer interval of receiving the data unit.
- 17 • If the RLP receiver is an access network and FlowNNAckEnableRev is 0x0001, then RLP  
18 shall send a ReceiverStatus message within AckTimer interval of receiving the data unit.
- 19 • If  $X < V(N)_{NN,P}$ , the data unit shall be discarded as a duplicate.
- 20 • If  $V(N)_{NN,P} \leq X < V(R)_{NN,P}$ , and the data unit is not already stored in the resequencing  
21 buffer, then:
  - 22 – RLP shall store the received data unit in the resequencing buffer.
  - 23 – If  $X = V(N)_{NN,P}$ , and if the Link Flow is carrying an octet stream, then RLP shall  
24 pass all contiguous octets in the resequencing buffer, from  $V(N)_{NN,P}$  upward, to  
25 the Route Protocol. RLP shall then set  $V(N)_{NN,P}$  to (LAST+1) where LAST is the  
26 sequence number of the last contiguous octet (i.e., the octet with the highest  
27 sequence number) passed to the Route Protocol from the resequencing buffer.
  - 28 – If  $X = V(N)_{NN,P}$ , and if the Link Flow is carrying a packet stream, and if in-order  
29 delivery of Route Protocol packets is required, then RLP shall pass all  
30 contiguous complete Route Protocol packets in the resequencing buffer, that  
31 have not been passed to the Route Protocol, from the beginning of the  
32 resequencing buffer upward, to the Route Protocol. RLP shall then set  $V(N)_{NN,P}$   
33 to (LAST+1) where LAST is the sequence number of the last contiguous data  
34 unit in the resequencing buffer.
  - 35 – If  $X = V(N)_{NN,P}$ , and if the Link Flow is carrying a packet stream, and if in-order  
36 delivery of Route Protocol packets is not required, then RLP shall pass all  
37 complete Route Protocol packets in the resequencing buffer, that have not been  
38 passed to the Route Protocol, from the beginning of the resequencing buffer  
39 upward, to the Route Protocol. RLP shall then set  $V(N)_{NN,P}$  to (LAST+1) where  
40 LAST is the sequence number of the last contiguous data unit in the  
41 resequencing buffer.

- 1 • If  $V(N)_{NN,P} < X < V(R)_{NN,P}$ , and the data unit has already been stored in the resequencing  
2 buffer, then the data unit shall be discarded as a duplicate.
- 3 • If  $X = V(R)_{NN,P}$ , then:
- 4 – If  $V(R)_{NN,P} = V(N)_{NN,P}$  and if the Link Flow is carrying an octet stream, then RLP  
5 shall increment  $V(N)_{NN,P}$  and  $V(R)_{NN,P}$  and shall pass the octet to the Route  
6 Protocol.
- 7 – If  $V(R)_{NN,P} = V(N)_{NN,P}$ , and if the Link Flow is carrying a packet stream, then RLP  
8 shall increment  $V(N)_{NN,P}$  and  $V(R)_{NN,P}$  and shall pass all complete Route Protocol  
9 packets in the resequencing buffer, that have not been passed to the Route  
10 Protocol, from the beginning of the resequencing buffer upward, to the Route  
11 Protocol.
- 12 – If  $V(R)_{NN,P} \neq V(N)_{NN,P}$ , RLP shall increment  $V(R)_{NN,P}$  and shall store the data unit  
13 in the resequencing buffer. If the Link Flow is carrying a packet stream, and if  
14 in-order delivery of Route Protocol packets is not required, then RLP shall pass  
15 all complete Route Protocol packets in the resequencing buffer, that have not  
16 been passed to the Route Protocol, from the beginning of the resequencing  
17 buffer upward, to the Route Protocol.
- 18 • If  $X > V(R)_{NN,P}$ , then:
- 19 – RLP shall store the data unit in the resequencing buffer.
- 20 – If the Link Flow is carrying a packet stream, and if in-order delivery of Route  
21 Protocol packets is not required, then RLP shall pass all complete Route  
22 Protocol packets in the resequencing buffer, that have not been passed to the  
23 Route Protocol, from the beginning of the resequencing buffer upward, to the  
24 Route Protocol.
- 25 – If the RLP receiver is an access network, then RLP shall set an RLP abort timer  
26 to AbortTimer, where AbortTimer is a parameter of the FlowNNTimersRev  
27 attribute, for each missing RLP data unit from  $V(R)_{NN,P}$  to  $X-1$ , inclusive. If the  
28 RLP receiver is an access terminal, then RLP shall set an RLP abort timer to  
29 AbortTimer, where AbortTimer is a parameter of the FlowNNTimersFwd  
30 attribute, for each missing RLP data unit from  $V(R)_{NN,P}$  to  $X-1$ , inclusive.
- 31 – RLP shall set  $V(R)_{NN,P}$  to  $X+1$ .
- 32 – If the RLP receiver is an access terminal, and if the FlowNNNakEnableFwd  
33 attribute is set to 0x0001, then RLP shall send a ReceiverStatus message. If  
34 the RLP receiver is an access network, and if the FlowNNNakEnableRev  
35 attribute is set to 0x0001, then RLP shall send a ReceiverStatus message.

36 If a missing data unit has not arrived when its RLP abort timer expires and if the Link Flow  
37 is carrying an octet stream, RLP shall pass all octets in the resequencing buffer up to the  
38 next missing octet, in order of sequence number, to the Route Protocol. RLP shall skip any  
39 missing octets.

40 If a missing data unit has not arrived when its RLP abort timer expires and if the Link Flow  
41 is carrying a packet stream, and if in-order delivery of Route Protocol packets is required,

1 then RLP shall pass all complete Route Protocol packets, that have not been passed to the  
 2 Route Protocol, from the beginning of the resequencing buffer upward up to the next  
 3 missing data unit, to the Route Protocol. RLP may pass to the Route Protocol partially  
 4 received packets with an indication of partial packet delivery.

5 RLP shall set  $V(N)_{NN,P}$  to the sequence number of the next missing data unit, or to  $V(R)_{NN}$  if  
 6 there are no remaining missing data units. Further recovery is the responsibility of higher  
 7 layer protocols.

#### 8 2.5.4.3 RLP Packet Header

9 The RLP packet header, which precedes the RLP payload, has the following format:  
 10

Field	Length (bits)
LinkFlowID	5
Route	1
FirstDataUnit	1
LastDataUnit	1
SEQ	SequenceLength

11 **LinkFlowID** The identifier for this Link Flow.

12 **Route** If this RLP packet is sent on Route A, then the sender shall set this  
 13 field to '0'. Otherwise, the sender shall set this field to '1'.

14 **FirstDataUnit** If the Link Flow is carrying an octet stream, then the sender shall set  
 15 this field to '0'. Otherwise, the sender shall set this field as follows:

16 If the payload of this RLP packet is the first segment of a Route  
 17 Protocol packet, then the sender shall set this field to '1'. Otherwise,  
 18 the sender shall set this field to '0'.

19 **LastDataUnit** If the Link Flow is carrying an octet stream, then the sender shall set  
 20 this field to '0'. Otherwise, the sender shall set this field as follows:

21 If the payload of this RLP packet is the last segment of a Route  
 22 Protocol packet, then the sender shall set this field to '1'. Otherwise,  
 23 the sender shall set this field to '0'.

24 **SEQ** The RLP sequence number of the first data unit in the RLP payload. If  
 25 this RLP packet is being sent on the forward link, the length of this  
 26 field is indicated by the SequenceLength field in the  
 27 FlowNNSequenceLengthFwd attribute corresponding to this flow. If  
 28 this RLP packet is being sent on the reverse link, the length of this  
 29 field is indicated by the SequenceLength field in the  
 30 FlowNNSequenceLengthRev attribute corresponding to this flow.

1 2.5.4.4 Message Formats

2 The messages described in this section control the function of the RLP. These messages are  
 3 exchanged between the access terminal and the access network using the SNP.

4 2.5.4.4.1 ResetTxIndication

5 The RLP transmitter in the access terminal or the access network sends the  
 6 ResetTxIndication message to reset its peer RLP receiver.

Field	Length (bits)
MessageID	8
LinkFlowID	5
Route	1
Reserved	2

8 MessageID The sender shall set this field to 0x00.

9 LinkFlowID The sender shall set this field to the Link Flow that is reset. The  
 10 sender shall set this field to '11111' to reset all Link Flows.

11 Route If Route A is reset, then the sender shall set this field to '0'. If Route B  
 12 is reset, then the sender shall set this field to '1'.

13 Reserved The sender shall set this field to '00'. The receiver shall ignore this  
 14 field.

<b>Channels</b>	FTC	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast		<b>Priority</b>	50

16 2.5.4.4.2 ResetRxIndication

17 The RLP receiver in the access terminal or the access network sends the ResetRxIndication  
 18 message to reset its peer RLP transmitter.

Field	Length (bits)
MessageID	8
LinkFlowID	5
Route	1
Reserved	2

20 MessageID The sender shall set this field to 0x01.

- 1 **LinkFlowID** The sender shall set this field to the Link Flow that is reset. The  
2 sender shall set this field to '11111' to reset all Link Flows.
- 3 **Route** If Route A is reset, then the sender shall set this field to '0'. If Route B  
4 is reset, then the sender shall set this field to '1'.
- 5 **Reserved** The sender shall set this field to '00'. The receiver shall ignore this  
6 field.

<b>Channels</b>	FTC RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Priority</b>	50

#### 8 2.5.4.4.3 ResetTxIndicationAck

9 The RLP receiver in the access terminal or the access network sends the  
10 ResetTxIndicationAck message in response to a ResetTxIndication message.

Field	Length (bits)
MessageID	8
LinkFlowID	5
Route	1
Reserved	2

- 12
- 13 **MessageID** The sender shall set this field to 0x0d.
- 14 **LinkFlowID** The sender shall set this field to the Link Flow that is reset. If this  
15 message is being sent in response to a ResetTxIndication message  
16 that required reset of all Link Flows, then the sender shall set this  
17 field to '11111'.
- 18 **Route** If Route A is reset, then the sender shall set this field to '0'. If Route B  
19 is reset, then the sender shall set this field to '1'.
- 20 **Reserved** The sender shall set this field to '00'. The receiver shall ignore this  
21 field.

<b>Channels</b>	FTC RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Priority</b>	50

#### 23 2.5.4.4.4 ResetTxComplete

24 The RLP transmitter in the access terminal or the access network sends the  
25 ResetTxComplete message to complete the RLP reset procedure.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

Field	Length (bits)
MessageID	8
LinkFlowID	5
Route	1
Reserved	2

**MessageID** The sender shall set this field to 0x0e.

**LinkFlowID** The sender shall set this field to the Link Flow that is reset. If all Link Flows were reset, then the sender shall set this field to '11111'.

**Route** If Route A is reset, then the sender shall set this field to '0'. If Route B is reset, then the sender shall set this field to '1'.

**Reserved** The sender shall set this field to '00'. The receiver shall ignore this field.

<b>Channels</b>	FTC RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Priority</b>	50

2.5.4.4.5 ResetRxComplete

The RLP transmitter in the access terminal or the access network sends the ResetRxComplete message to complete the RLP reset procedure.

Field	Length (bits)
MessageID	8
LinkFlowID	5
Route	1
Reserved	2

**MessageID** The sender shall set this field to 0x0f.

**LinkFlowID** The sender shall set this field to the Link Flow that is reset. If all Link Flows were reset, then the sender shall set this field to '11111'.

**Route** If Route A is reset, then the sender shall set this field to '0'. If Route B is reset, then the sender shall set this field to '1'.

**Reserved** The sender shall set this field to '00'. The receiver shall ignore this field.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	unicast	<b>Priority</b>	50

#### 2.5.4.4.6 ReceiverStatus

The access terminal and the access network send the ReceiverStatus message to acknowledge the receipt of one or more RLP data units or to request the retransmission of one or more RLP data units.

Field	Length (bits)
MessageID	8
LinkFlowID	5
Route	1
LatestDataUnit	1
TimeStampIncluded	1
TimeStamp	0 or 12
SequenceLength	8
ReportCount	8

ReportCount occurrences of the following two fields:

FirstErasedDataUnit	SequenceLength
WindowLen	SequenceLength

VR	SequenceLength
----	----------------

Reserved	0 - 7 (as needed)
----------	-------------------

- 7    **MessageID**                    The sender shall set this field to 0x02.
- 8    **LinkFlowID**                    The sender shall set this field to the Link Flow for which this  
9    ReceiverStatus is being sent.
- 10   **Route**                            If this ReceiverStatus is being sent for Route A, then the sender shall  
11   set this field to '0'. Otherwise, the sender shall set this field to '1'.
- 12   **LatestDataUnit**                If the latest data unit in the receive buffer is the data unit with  
13   sequence number  $V(R) - 1$ , then the sender shall set this field to '1'.  
14   Otherwise, the sender shall set this field to '0'.
- 15   **TimeStampIncluded**            If the value of the FlowNNAckEnableFwd attribute is 0x0001, then  
16   the access terminal shall set this field to '1'. Otherwise, the access

- 1 terminal shall set this field to '0'. If the value of the  
 2 FlowNNAckEnableRev attribute is 0x0001, then the access network  
 3 shall set this field to '1'. Otherwise, the access network shall set this  
 4 field to '0'. *NN* is the two-digit hexadecimal Link Flow number.
- 5 **TimeStamp** If TimeStampIncluded is '0', then the sender shall omit this field.  
 6 Otherwise, the sender shall set this field to the 12 least significant  
 7 bits of the CDMA system time, in units of slots, when the latest data  
 8 unit (in order or receive time) in the receive buffer was received.
- 9 **SequenceLength** The sender shall set this field to the size of the sequence number for  
 10 this Link Flow in units of bits.
- 11 **ReportCount** The sender shall set this field to the number of Report records  
 12 included in this message. The sender shall include ReportCount  
 13 occurrences of the following two fields with the message.
- 14 **FirstErasedDataUnit** The sender shall set this field to the sequence number of the first RLP  
 15 data unit erased in a sequence of erased data units.
- 16 **WindowLen** The sender shall set this field to the length of the erased window in  
 17 units of data units.
- 18 **VR** The sender shall set this field to  $V(R)_{NN,P}$ .
- 19 **Reserved** The sender shall add reserved bits to make the length of the entire  
 20 message an integer number of octets. The sender shall set these bits  
 21 to '0'. The receiver shall ignore this field.

<b>Channels</b>	FTC	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	50

23 2.5.4.4.7 ReservationOnRequest

24 The access terminal sends this message to request transition of one or more Reservations to  
 25 the Open State.

26

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8
ReservationCount occurrences of the following two fields:	
Link	1
ReservationLabel	8
Reserved	0 - 7 (as needed)

- 1    **MessageID**                    The access terminal shall set this field to 0x16.
  
- 2    **TransactionID**                The access terminal shall set this field to one more (modulo 256) than  
 3    the TransactionID field of the last ReservationOnRequest or  
 4    ReservationOffRequest message sent by the access terminal. If this is  
 5    the first ReservationOnRequest or ReservationOffRequest message  
 6    sent by the access terminal, then the access terminal shall set this  
 7    field to zero.
  
- 8    **ReservationCount**            The access terminal shall set this field to the number of  
 9    ReservationLabel fields in this message.
  
- 10   **Link**                            If this request is for a forward link Reservation, then the access  
 11   terminal shall set this field to '1'. If this request is for a reverse link  
 12   Reservation, then the access terminal shall set this field to '0'.
  
- 13   **ReservationLabel**            The access terminal shall set this field to the ReservationLabel for  
 14   which this request is generated.
  
- 15   **Reserved**                        The access terminal shall add reserved bits to make the length of the  
 16   entire message an integer number of octets. The access terminal shall  
 17   set these bits to zero. The access network shall ignore this field.

<b>Channels</b>	AC	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

19    2.5.4.4.8 ReservationOffRequest

20    The access terminal sends this message to request transition of one or more Reservations to  
 21    the Close State.

22

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following two fields:

Link	1
ReservationLabel	8

Reserved	0 - 7 (as needed)
----------	-------------------

- 1    **MessageID**                    The access terminal shall set this field to 0x17.
  
- 2    **TransactionID**                The access terminal shall set this field to one more (modulo 256) than
- 3                                    the TransactionID field of the last ReservationOnRequest or
- 4                                    ReservationOffRequest message sent by the access terminal. If this is
- 5                                    the first ReservationOnRequest or ReservationOffRequest message
- 6                                    sent by the access terminal, then the access terminal shall set this
- 7                                    field to zero.
  
- 8    **ReservationCount**            The access terminal shall set this field to the number of
- 9                                    ReservationLabel fields in this message.
  
- 10   **Link**                            If this request is for a forward link Reservation, then the access
- 11                                    terminal shall set this field to '1'. If this request is for a reverse link
- 12                                    Reservation, then the access terminal shall set this field to '0'.
  
- 13   **ReservationLabel**            The access terminal shall set this field to the Reservation for which
- 14                                    this request is generated.
  
- 15   **Reserved**                        The access terminal shall add reserved bits to make the length of the
- 16                                    entire message an integer number of octets. The access terminal shall
- 17                                    set these bits to zero. The access network shall ignore this field.

<b>Channels</b>	AC                    RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast	<b>Priority</b>	40

19    2.5.4.4.9 ReservationAccept

20    The access network sends this message to acknowledge reception of and allow the state  
 21    transition requested by a ReservationOnRequest or ReservationOffRequest message. The  
 22    access terminal sends this message to acknowledge reception of and accept the state  
 23    transition requested by a ReservationOn or ReservationOff message.

24

Field	Length (bits)
MessageID	8
TransactionID	8

1 MessageID The sender shall set this field to 0x18.

2 TransactionID The access network shall set this field to the TransactionID field of  
3 the ReservationOnRequest or ReservationOffRequest message to  
4 which the access network is responding. The access terminal shall set  
5 this field to the TransactionID field of the ReservationOn or  
6 ReservationOff message to which the access terminal is responding.  
7

<b>Channels</b>	CC	FTC	RTC	AC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast			<b>Priority</b>	40	

#### 8 2.5.4.4.10 ReservationReject

9 The access network sends this message to acknowledge reception of and deny the state  
10 transition requested by a ReservationOnRequest or ReservationOffRequest message.  
11

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following two fields:

AllowableLink	1
AllowableReservationLabel	8

Reserved	0 – 7 (as needed)
----------	-------------------

12 MessageID The access network shall set this field to 0x19.

13 TransactionID The access network shall set this field to the TransactionID field of  
14 the ReservationOnRequest or ReservationOffRequest message to  
15 which the access network is responding.

16 ReservationCount The access network shall set this field to the number of  
17 ReservationLabel fields in this message.

18 AllowableLink If the Reservation for which the access network would have allowed  
19 the state transition requested in the ReservationOnRequest or  
20 ReservationOffRequest message is a forward link Reservation, then

1 the access network shall set this field to '1'. If the Reservation for  
 2 which the access network would have allowed the state transition  
 3 requested in the ReservationOnRequest or ReservationOffRequest  
 4 message is a reverse link Reservation, then the access network shall  
 5 set this field to '0'.

6 **AllowableReservationLabel**

7 The access network shall set this field to the ReservationLabel for  
 8 which the access network would have allowed the state transition  
 9 requested in the ReservationOnRequest or ReservationOffRequest  
 10 message.

11 **Reserved**

12 The access network shall add reserved bits to make the length of the  
 13 entire message an integer number of octets. The access network shall  
 14 set these bits to zero. The access terminal shall ignore this field.

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

15 **2.5.4.4.11 ReservationOn**

16 The access network sends this message to transition a reverse link Reservation to the Open  
 17 state.

<b>Field</b>	<b>Length (bits)</b>
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following field:

ReservationLabel	8
------------------	---

19 **MessageID** The access network shall set this field to 0x1a.

20 **TransactionID** The access network shall set this field to one more (modulo 256) than  
 21 the TransactionID field of the last ReservationOn or ReservationOff  
 22 message sent by the access network. If this is the first ReservationOn  
 23 or ReservationOff message sent by the access network, then the  
 24 access network shall set this field to zero.

25 **ReservationCount** The access network shall set this field to the number of  
 26 ReservationLabel fields in this message.

27 **ReservationLabel** The access network shall set this field to the Reservation which is to  
 28 be transitioned to the Open state.  
 29

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

#### 2.5.4.4.12 ReservationOff

The access network sends this message to transition a reverse link Reservation to the Close state.

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8
ReservationCount occurrences of the following field:	
ReservationLabel	8

**MessageID** The access network shall set this field to 0x1b.

**TransactionID** The access network shall set this field to one more (modulo 256) than the TransactionID field of the last ReservationOn or ReservationOff message sent by the access network. If this is the first ReservationOn or ReservationOff message sent by the access network, then the access network shall set this field to zero.

**ReservationCount** The access network shall set this field to the number of ReservationLabel fields in this message.

**ReservationLabel** The access network shall set this field to the Reservation that is to be transitioned to the Close state.

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

#### 2.5.4.4.13 FwdReservationOff

The access network sends this message to inform the access terminal when a forward Reservation transitions to the Close state.

Field	Length (bits)
MessageID	8
ReservationCount	8

ReservationCount occurrences of the following field:

ReservationLabel	8
------------------	---

- 1 MessageID The access network shall set this field to 0x22.
- 2 ReservationCount The access network shall set this field to the number of  
3 ReservationLabel fields in this message.
- 4 ReservationLabel The access network shall set this field to the Reservation transitioned  
5 to the Close state.

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

7 2.5.4.4.14 AttributeUpdateRequest

8 The sender sends an AttributeUpdateRequest message to offer a set of attribute values for a  
9 given attribute.

Field	Length (bits)
MessageID	8
TransactionID	8

10 One or more instances of the following record

AttributeRecord	Attribute dependent
-----------------	---------------------

- 11 MessageID The sender shall set this field to 0x52.
- 12 TransactionID The sender shall increment this value for each new  
13 AttributeUpdateRequest message sent.
- 14 AttributeRecord The format of this record is specified in [8].

<b>Channels</b>	CC	AC	FTC	RTC	<b>SLP</b>	Reliable on FTC and RTC Best Effort on CC and AC
<b>Addressing</b>	unicast			<b>Priority</b>	40	

16 2.5.4.4.15 AttributeUpdateAccept

17 The sender sends an AttributeUpdateAccept message in response to an  
18 AttributeUpdateRequest message to select an attribute value from a list of offered values.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The sender shall set this field to 0x53.

TransactionID The sender shall set this value to the TransactionID field of the corresponding AttributeUpdateRequest message.

<b>Channels</b>	CC AC FTC RTC	<b>SLP</b>	Reliable on FTC and RTC Best Effort on CC and AC
<b>Addressing</b>	unicast	<b>Priority</b>	40

#### 2.5.4.4.16 AttributeUpdateReject

The access network sends an AttributeUpdateReject message to reject the set of attribute values proposed by the access terminal in the corresponding AttributeUpdateRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access network shall set this field to 0x54.

TransactionID The sender shall set this value to the TransactionID field of the corresponding AttributeUpdateRequest message.

<b>Channels</b>	FTC CC	<b>SLP</b>	Reliable on FTC Best Effort on CC
<b>Addressing</b>	unicast	<b>Priority</b>	40

#### 2.5.4.5 Interface to Other Protocols

##### 2.5.4.5.1 Commands

This protocol does not issue any commands.

##### 2.5.4.5.2 Indications

This protocol registers to receive the following indications:

- *IdleState.ConnectionOpened*

- 1 • *PhysicalLayer.ReverseTrafficPacketsMissed* along with parameters indicating the Link  
 2 Flow number and missing data units.
- 3 • *ConnectedState.ConnectionClosed*
- 4 • *RouteUpdate.ConnectionLost*

5 **2.5.4.6 RLP Packet Priorities**

6 The sender shall assign priority between 60 and 70 inclusive to RLP packets. For a given  
 7 Link Flow, the sender shall assign higher priority (lower number) to packets containing  
 8 retransmitted application traffic than packets containing only first time transmissions.

9

<b>Type of RLP Packet</b>	<b>Channel</b>	<b>Addressing</b>	<b>Priority</b>
Packet containing only First Time Transmissions	FTC, RTC	unicast	Between 60 and 70 inclusive
Packet containing re-transmitted application traffic	FTC, RTC	unicast	Between 60 and 70 inclusive

10 **2.5.5 Protocol Numeric Constants**

11

<b>Constant</b>	<b>Meaning</b>	<b>Value</b>
$N_{\text{LinkFlowMax}}$	Maximum total number of activated and deactivated Link Flows.	31
$T_{\text{Response}}$	Time period within which the access network is to respond to ReservationOnRequest and ReservationOffRequest messages.	1 second

## 2.6 Data Over Signaling Protocol

### 2.6.1 Overview

The Data Over Signaling Protocol provides transmission and duplicate detection of higher layer packets using signaling messages. Each Link flow provides two instances of the Data Over Signaling Protocol, one associated with Route A of the Link flow, and the other associated with Route B. A higher layer packet is carried in a DataOverSignaling message. The Data Over Signaling Protocol uses message sequence numbers in the DataOverSignaling message to provide duplicate detection. Data Over Signaling Protocol is associated with the Enhanced Multi-Flow Packet Application. The application subtype for this application is defined in [3].

### 2.6.2 Primitives and Public Data

#### 2.6.2.1 Commands

This protocol does not define any commands.

#### 2.6.2.2 Return Indications

This protocol does not return any indications.

### 2.6.3 Protocol Data Unit

The transmission unit of this protocol is a DataOverSignaling message. The DataOverSignaling message carries payload on behalf of the higher layer. This protocol uses the Signaling Application to transmit and receive messages.

### 2.6.4 Procedures and Messages for the InUse Instance of the Protocol

#### 2.6.4.1 Procedures

The sender shall set the MessageSequence field of a DataOverSignaling message to  $V(S_P)$  value maintained by the sender for the Route  $P$  on which the DataOverSignaling message was sent. Each time the sender sends a new DataOverSignaling message, it shall increment the value of  $V(S_P)$ . If the sender does not receive a DataOverSignalingAck message within an implementation specific time interval in response to a DataOverSignaling message requiring an acknowledgment, then the sender may retransmit the DataOverSignaling message containing the same higher layer packet and the same MessageSequence an implementation specific number of times.

The access terminal or the access network shall not send a DataOverSignaling message if the associated Link Flow for which the DataOverSignaling message is carrying payload is deactivated, or if the associated Reservation is in the Close state.

Upon receiving a DataOverSignaling message, the receiver shall perform the following:

- If Reset is set to '1' and the receiver is the access terminal, the receiver shall perform the following:

- 1           – If Route is set to '0', the receiver shall set  $V(R_A)$  to  $(\text{MessageSequence} - 1) \bmod$   
2           128.
- 3           – If Route is set to '1', the receiver shall set  $V(R_B)$  to  $(\text{MessageSequence} - 1) \bmod$   
4           128.
- 5       • The receiver shall validate the message using the procedure defined in the Sequence  
6       Number Validation Procedure of [8] by setting the variable  $V(S)$  defined in [8] to the  
7       MessageSequence field  $V(S_P)$  of the DataOverSignaling message, setting the variable  $V(R)$   
8       defined in [8] to the  $V(R_P)$  value maintained by the receiver for the Route  $P$  on which the  
9       DataOverSignaling message was received, and  $S = 7$ .
  - 10      • The receiver shall discard the DataOverSignaling message if it is invalid. If the  
11      DataOverSignaling message is valid, then the receiver shall pass the HigherLayerPacket  
12      field of the DataOverSignaling message to the higher layer. If the receiver is an access  
13      terminal, then the higher layer is indicated by the ProtocolID field of the  
14      FlowNNRouteProtocolFwd attribute, where  $NN$  is the Link Flow with which the  
15      DataOverSignaling message is associated. If the receiver is an access network, then the  
16      higher layer is indicated by the ProtocolID field of the FlowNNFlowRouteRev attribute,  
17      where  $NN$  is the Link Flow with which the DataOverSignaling message is associated.
  - 18      • If the AckRequired field of the DataOverSignaling message is '1', then the receiver shall  
19      respond with a DataOverSignalingAck message with AckSequence field set to the  
20      MessageSequence field of the DataOverSignaling message.

#### 21   2.6.4.2 Message Formats

22   The messages described in this section are exchanged between the access terminal and the  
23   access network using the Signaling Application.

##### 24   2.6.4.2.1 DataOverSignaling

25   The access network or the access terminal sends the DataOverSignaling message to  
26   transmit a higher layer packet.

27

Field	Length (bits)
MessageID	8
LinkFlowID	5
Route	1
AckRequired	1
Reset	1
Reserved	1
MessageSequence	7
HigherLayerPacket	Variable Length

28   MessageID                   The sender shall set this field to 0x14.

- 1 **LinkFlowID** The sender shall set this field to the Link Flow with which this  
2 DataOverSignaling message is associated.
- 3 **Route** If this DataOverSignaling message is associated with Route A, then  
4 the sender shall set this field to '0'. If this DataOverSignaling message  
5 is associated with Route B, then the sender shall set this field to '1'.
- 6 **AckRequired** The sender shall set this field to '1' if the receiver is required to  
7 acknowledge the receipt of this message. Otherwise, the sender shall  
8 set this field to '0'.
- 9 **Reset** The access terminal shall set this field to '0'. The access network may  
10 set this field to '1' to indicate that the access terminal is to reset its  
11 V(R) for the indicated route. The access network may set this field to  
12 '0' to indicate that the access terminal is not required to reset its  
13 V(R).
- 14 **Reserved** The sender shall set this field to '0'. The receiver shall ignore this  
15 field.
- 16 **MessageSequence** The sender shall set this field to the V(S<sub>P</sub>) value maintained by the  
17 sender for the Route *P* on which the DataOverSignaling message was  
18 sent.
- 19 **HigherLayerPacket** The sender shall set this field to an entire higher layer packet<sup>4</sup>. The  
20 length of the higher layer packet shall be an integer number of octets.  
21

<b>Channels</b>	CC	AC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast			<b>Priority</b>	20 to 50 (inclusive)

22 The sender shall assign message priority in the range 20 to 50, inclusive, depending on the  
23 priority of the higher layer packet carried as payload in this message.

#### 24 2.6.4.2.2 DataOverSignalingAck

25 The access network or the access terminal sends a DataOverSignalingAck message to  
26 acknowledge receipt of a DataOverSignaling message.  
27

---

<sup>4</sup> For example, if the higher layer packet is an HDLC frame, then the entire HDLC frame is included.

<b>Field</b>	<b>Length (bits)</b>
MessageID	8
Route	1
AckSequence	7

- 1 **MessageID**            The sender shall set this field to 0x15.
- 2 **Route**                If this message is acknowledging a DataOverSignaling received on
- 3                            Route A, then the sender shall set this field to '0'. Otherwise the
- 4                            sender shall set this field to '1'.
- 5 **AckSequence**        The sender shall set this field to the MessageSequence field of the
- 6                            DataOverSignaling message whose receipt is being acknowledged.
- 7

<b>Channels</b>	CC	AC	FTC	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast			<b>Priority</b>	40	

8 **2.6.4.3 Interface to Other Protocols**

9 **2.6.4.3.1 Commands**

10 This protocol does not issue any commands.

11 **2.6.4.3.2 Indications**

12 This protocol does not register to receive any indications.

## 2.7 Location Update Protocol

### 2.7.1 Overview

The Location Update Protocol defines location update procedures and messages for mobility management for the Enhanced Multi-Flow Packet Application.

The Location Update Protocol is a protocol associated with the Enhanced Multi-Flow Packet Application. The application subtype for this application is defined in [3].

### 2.7.2 Primitives and Public Data

#### 2.7.2.1 Commands

This protocol does not define any commands.

#### 2.7.2.2 Return Indications

This protocol does not return any indications.

### 2.7.3 Protocol Data Unit

The transmission unit of this protocol is a message. This is a control protocol; and, therefore, it does not carry payload on behalf of other layers or protocols.

### 2.7.4 Procedures and Messages for the InUse Instance of the Protocol

#### 2.7.4.1 Procedures

##### 2.7.4.1.1 Access Network Requirements

If the protocol receives an *AddressManagement.SubnetChanged* indication, the access network:

- May send a *LocationRequest* message to query the Location information.
- May send a *LocationAssignment* message to update the Location information.
- May send a *ServiceNetworkIDRequest* message to query the service network identifier.
- May send a *ServiceNetworkIDAssignment* message to update the service network identifier.

##### 2.7.4.1.2 Access Terminal Requirements

If the access terminal receives a *LocationRequest* message, it shall send a *LocationNotification* message. If the access terminal has a stored value for the *LocationValue* parameter, the access terminal shall set the *LocationType*, *LocationLength*, and *LocationValue* fields in this message to its stored values of these fields. If the access terminal does not have a stored value for the *LocationValue* parameter, the access terminal shall omit the *LocationLength* and *LocationValue* fields in this message.

1 If the access terminal receives a LocationAssignment message, it shall send a  
 2 LocationComplete message and the access terminal shall store the value of the  
 3 LocationType, LocationLength, and LocationValue fields of the LocationAssignment message  
 4 in LocationType, LocationLength, and LocationValue variables, respectively.

5 If the access terminal receives a ServiceNetworkIDRequest message, it shall send a  
 6 ServiceNetworkIDNotification message. If the access terminal has a stored value for the  
 7 ServiceNetworkID parameter, the access terminal shall set the ServiceNetworkIDLength and  
 8 ServiceNetworkID fields in this message to its stored values of these fields. If the access  
 9 terminal does not have a stored value for the ServiceNetworkID parameter, the access  
 10 terminal shall set the ServiceNetworkIDLength field to zero and shall omit the  
 11 ServiceNetworkID fields in this message.

12 If the access terminal receives a ServiceNetworkIDAssignment message, it shall send a  
 13 ServiceNetworkIDComplete message and the access terminal shall store the value of the  
 14 ServiceNetworkIDLength and ServiceNetworkID fields of the ServiceNetworkIDAssignment  
 15 message in ServiceNetworkIDLength and ServiceNetworkID variables, respectively.

16 **2.7.4.2 Message Formats**

17 **2.7.4.2.1 LocationRequest**

18 The access network uses this message to query the access terminal of its Location  
 19 information.

Field	Length (bits)
MessageID	8

21 **MessageID** The access network shall set this field to 0x03.

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

23 **2.7.4.2.2 LocationNotification**

24 The access terminal sends the LocationNotification message either in response to the  
 25 LocationRequest message or in an unsolicited manner as specified in [11] if the configured  
 26 value of the RANHandoff attribute is 0x01.

27

<b>Field</b>	<b>Length (bits)</b>
MessageID	8
LocationType	8
LocationLength	0 or 8
LocationValue	0 or 8 × LocationLength

- 1 MessageID The access terminal shall set this field to 0x04.
- 2 LocationType The access terminal shall set this field to zero if the value of its stored  
3 LocationValue is NULL; otherwise, the access terminal shall set this  
4 field to the stored value of LocationType.
- 5 LocationLength The access terminal shall not include this field if the value of its  
6 stored LocationValue is NULL; otherwise, the access terminal shall  
7 set this field to the stored value of LocationLength.
- 8 LocationValue The access terminal shall not include this field if the value of its  
9 stored LocationValue is NULL; otherwise, the access terminal shall  
10 set this field to the stored value of LocationValue.

<b>Channels</b>	AC	RTC	<b>SLP</b>	Reliable <sup>5</sup>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40	

#### 12 2.7.4.2.3 LocationAssignment

13 The access network uses this message to update the Location information of the access  
14 terminal.

<b>Field</b>	<b>Length (bits)</b>
MessageID	8
TransactionID	8
LocationType	8
LocationLength	8
LocationValue	8 × LocationLength

- 15 MessageID The access network shall set this field to 0x05.
- 16 TransactionID The access network shall increment this value for each new  
17 LocationAssignment message sent.

---

<sup>5</sup> This message is sent reliably when it is sent over the Reverse Traffic Channel.

1 LocationType The access network shall set this field to the type of the location as  
 2 specified in Table 2.7.4.2-1.

3 **Table 2.7.4.2-1. LocationType Encoding**

LocationType	LocationLength	Meaning
0x00	N/A	No location is stored
0x01	0x05	Location compatible with [2] (see Table 2.7.4.2-2)
All other values	N/A	Reserved

4 LocationLength The access network shall set this field to the length of the  
 5 LocationValue field in octets as specified in Table 2.7.4.2-1.

6 LocationValue The access network shall set this field to the Location of type  
 7 specified by LocationType. If LocationType is set to 0x01, the access  
 8 network shall set this field as shown in Table 2.7.4.2-2, where SID,  
 9 NID, and PACKET\_ZONE\_ID correspond to the current access  
 10 network.

11 **Table 2.7.4.2-2. Subfields of LocationValue when LocationType = 0x01**

Sub-fields of LocationValue	# of bits
SID	15
Reserved	1
NID	16
PACKET_ZONE_ID	8

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

13 2.7.4.2.4 LocationComplete

14 The access terminal sends this message in response to the LocationAssignment message.  
 15

Field	Length (bits)
MessageID	8
TransactionID	8

16 MessageID The access terminal shall set this field to 0x06.

TransactionID The access terminal shall set this field the TransactionID field of the corresponding LocationAssignment message.

<b>Channels</b>	AC	RTC	<b>SLP</b>	Reliable <sup>6</sup>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40	

#### 2.7.4.2.5 ServiceNetworkIDRequest

The access network uses this message to query the access terminal of its ServiceNetworkID information.

Field	Length (bits)
MessageID	8

MessageID The access network shall set this field to 0x10.

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort	
<b>Addressing</b>	unicast		<b>Priority</b>	40	

#### 2.7.4.2.6 ServiceNetworkIDNotification

The access terminal sends the ServiceNetworkIDNotification message in response to the ServiceNetworkIDRequest message.

Field	Length (bits)
MessageID	8
ServiceNetworkIDLength	8
ServiceNetworkID	0 or 8 × ServiceNetworkIDLength

MessageID The access terminal shall set this field to 0x11.

#### ServiceNetworkIDLength

The access terminal shall set this field to zero if the value of its stored ServiceNetworkID is NULL; otherwise, the access terminal shall set this field to the stored value of ServiceNetworkIDLength.

---

<sup>6</sup> This message is sent reliably when it is sent over the Reverse Traffic Channel.

1 ServiceNetworkID The access terminal shall not include this field if the value of its  
 2 stored ServiceNetworkID is NULL; otherwise, the access terminal  
 3 shall set this field to the stored value of ServiceNetworkID.  
 4

<b>Channels</b>	AC	RTC	<b>SLP</b>	Reliable <sup>7</sup>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40	

5 2.7.4.2.7 ServiceNetworkIDAssignment

6 The access network uses this message to update the ServiceNetworkID information of the  
 7 access terminal.

Field	Length (bits)
MessageID	8
TransactionID	8
ServiceNetworkIDLength	8
ServiceNetworkID	8 × ServiceNetworkIDLength

8 MessageID The access network shall set this field to 0x12.

9 TransactionID The access network shall increment this value for each new  
 10 ServiceNetworkIDAssignment message sent.

11 ServiceNetworkIDLength

12 The access network shall set this field to the length of the  
 13 ServiceNetworkID field in octets. The access network shall set this  
 14 field to zero if ServiceNetworkID is NULL.

15 ServiceNetworkID The access network shall not include this field if the  
 16 ServiceNetworkID is NULL. Otherwise, the access network shall set  
 17 this field to the ServiceNetworkID.  
 18

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort	
<b>Addressing</b>	unicast		<b>Priority</b>	40	

19 2.7.4.2.8 ServiceNetworkIDComplete

20 The access terminal sends this message in response to the ServiceNetworkIDAssignment  
 21 message.  
 22

---

<sup>7</sup> This message is sent reliably when it is sent over the Reverse Traffic Channel.

<b>Field</b>	<b>Length (bits)</b>
MessageID	8
TransactionID	8

1 MessageID The access terminal shall set this field to 0x13.

2 TransactionID The access terminal shall set this field the TransactionID field of the  
3 corresponding ServiceNetworkIDAssignment message.  
4

<b>Channels</b>	AC	RTC	<b>SLP</b>	Reliable <sup>8</sup>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40	

### 5 2.7.4.3 Interface to Other Protocols

#### 6 2.7.4.3.1 Commands

7 This protocol does not issue any commands.

#### 8 2.7.4.3.2 Indications

9 This protocol registers to receive the following indications:

- 10 • *SessionManagement.SessionClosed*
- 11 • *AddressManagement.SubnetChanged*

---

<sup>8</sup> This message is sent reliably when it is sent over the Reverse Traffic Channel.

**2.8 Flow Control Protocol**

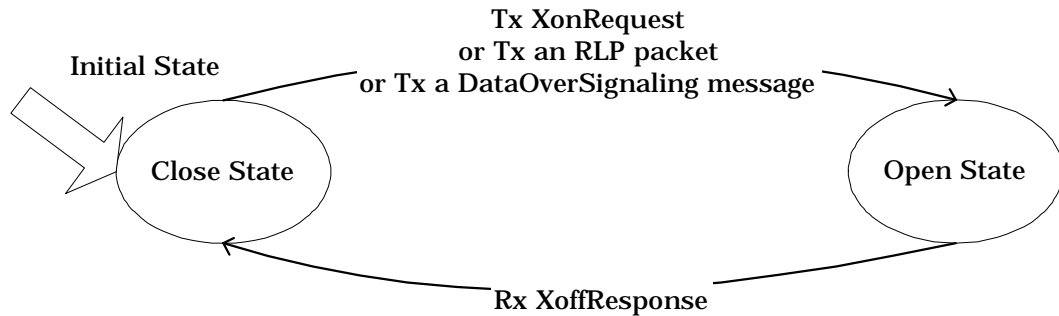
**2.8.1 Overview**

The Flow Control Protocol provides procedures and messages used by the access terminal and the access network to perform flow control for the Enhanced Multi-Flow Packet Application.

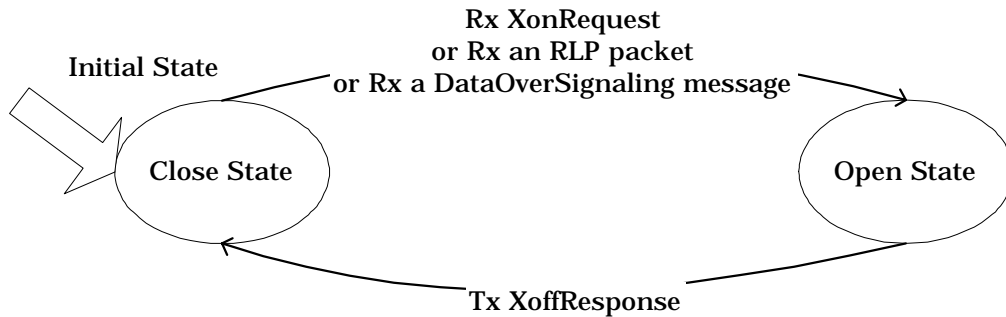
This protocol can be in one of the following states:

- Close State: in this state the Enhanced Multi-Flow Packet Application does not send or receive any RLP packets or DataOverSignaling messages.
- Open State: in this state the Enhanced Multi-Flow Packet Application can send or receive RLP packets and DataOverSignaling messages.

Figure 2.8.1-1 and Figure 2.8.1-2 show the state transition diagram at the access terminal and the access network.



**Figure 2.8.1-1. Flow Control Protocol State Diagram (Access Terminal)**



**Figure 2.8.1-2. Flow Control Protocol State Diagram (Access Network)**

The flow control protocol is a protocol associated with the Enhanced Multi-Flow Packet Application. The application subtype for this application is defined in [3].

## 2.8.2 Primitives and Public Data

### 2.8.2.1 Commands

This protocol does not define any commands.

### 2.8.2.2 Return Indications

This protocol does not return any indications.

## 2.8.3 Protocol Data Unit

The transmission unit of this protocol is a message. This is a control protocol and, therefore, it does not carry payload on behalf of other layers or protocols.

## 2.8.4 Procedures and Messages for the InUse Instance of the Protocol

### 2.8.4.1 Procedures

#### 2.8.4.1.1 Transmission and Processing of DataReady Message

The access network may send a DataReady message to indicate that there is data corresponding to this packet application waiting to be transmitted.

The access terminal shall send a DataReadyAck within the time period specified by  $T_{FCResponse}$  after reception of the DataReady message to acknowledge reception of the message.

#### 2.8.4.1.2 Transmission and Processing of RestartNetworkInterface Message

The access network may send a RestartNetworkInterface message to direct the access terminal to restart the interface between the packet application and the higher layer.

Upon receiving a RestartNetworkInterface message, the access terminal shall send a RestartNetworkInterfaceAck message and shall restart the interface between the packet application and the higher layer. The access terminal may also restart higher layer protocols.

#### 2.8.4.1.3 Close State

In this state, the access terminal and the access network shall not send or receive any RLP packets or DataOverSignaling messages.

#### 2.8.4.1.3.1 Access Terminal Requirements

The access terminal shall send an XonRequest message or an RLP packet (corresponding to this instance of the Enhanced Multi-Flow Packet Application) or a DataOverSignaling message (corresponding to this instance of the Enhanced Multi-Flow Packet Application) when it is ready to exchange RLP packets or DataOverSignaling messages with the access network. The access terminal should send an XonRequest message or an RLP packet (corresponding to this instance of the Enhanced Multi-Flow Packet Application) or a

1 DataOverSignaling message (corresponding to this instance of the Enhanced Multi-Flow  
2 Packet Application) when it receives a DataReady from the access network.

3 The access terminal shall transition to the Open State when it sends an XonRequest  
4 message or when it sends an RLP packet (corresponding to this instance of the Enhanced  
5 Multi-Flow Packet Application) or when it sends a DataOverSignaling message  
6 (corresponding to this instance of the Enhanced Multi-Flow Packet Application).

#### 7 2.8.4.1.3.2 Access Network Requirements

8 If the access network receives an XonRequest message, it shall

- 9 • Send an XonResponse message within the time period specified by  $T_{FCResponse}$  after  
10 reception of the XonRequest message to acknowledge reception of the message.
- 11 • Transition to the Open State.

12 The access network shall also transition to the Open State if it receives an RLP packet  
13 (corresponding to this instance of the Enhanced Multi-Flow Packet Application) or a  
14 DataOverSignaling message (corresponding to this instance of the Enhanced Multi-Flow  
15 Packet Application).

#### 16 2.8.4.1.4 Open State

17 In this state, the access terminal and the access network may send or receive any RLP  
18 packets or DataOverSignaling messages.

#### 19 2.8.4.1.4.1 Access Terminal Requirements

20 The access terminal may re-send an XonRequest message if it does not receive an  
21 XonResponse message or an RLP packet (corresponding to this instance of the Enhanced  
22 Multi-Flow Packet Application) or a DataOverSignaling message (corresponding to this  
23 instance of the Enhanced Multi-Flow Packet Application) within the time period specified by  
24  $T_{FCResponse}$  after sending the XonRequest message.

25 The access terminal may send an XoffRequest message to request the access network to  
26 stop sending RLP packets and DataOverSignaling messages. The access terminal shall  
27 transition to the Close state when it receives an XoffResponse message.

28 The access terminal may re-send an XoffRequest message if it does not receive an  
29 XoffResponse message within the time period specified by  $T_{FCResponse}$  after sending the  
30 XoffRequest message.

#### 31 2.8.4.1.4.2 Access Network Requirements

32 If the access network receives an XoffRequest message, it shall

- 33 • Send an XoffResponse message within the time period specified by  $T_{FCResponse}$  after  
34 reception of the XoffRequest message to acknowledge reception of the message.
- 35 • Transition to the Close State.

## 2.8.4.2 Message Formats

## 2.8.4.2.1 XonRequest

The access terminal sends this message to request transition to the Open State.

Field	Length (bits)
MessageID	8

MessageID The access terminal shall set this field to 0x07.

<b>Channels</b>	AC	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

## 2.8.4.2.2 XonResponse

The access network sends this message to acknowledge reception of the XonRequest message.

Field	Length (bits)
MessageID	8

MessageID The access network shall set this field to 0x08.

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

## 2.8.4.2.3 XoffRequest

The access terminal sends this message to request transition to the Close State.

Field	Length (bits)
MessageID	8

MessageID The access terminal shall set this field to 0x09.

<b>Channels</b>	AC	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

1 2.8.4.2.4 XoffResponse

2 The access network sends this message to acknowledge reception of the XoffRequest  
3 message.

4

Field	Length (bits)
MessageID	8

5 MessageID The access network shall set this field to 0x0a.

6

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

7 2.8.4.2.5 DataReady

8 The access network sends this message to indicate that there is data corresponding to this  
9 packet application awaiting to be transmitted.

10

Field	Length (bits)
MessageID	8
TransactionID	8

11 MessageID The access network shall set this field to 0x0b.

12 TransactionID The access network shall increment this value for each new  
13 DataReady message sent.

14

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

15 2.8.4.2.6 DataReadyAck

16 The access terminal sends this message to acknowledge reception of a DataReady message.

17

Field	Length (bits)
MessageID	8
TransactionID	8

18 MessageID The access terminal shall set this field to 0x0c.

TransactionID The access terminal shall set this value to the value of the TransactionID field of the corresponding DataReady message.

<b>Channels</b>	AC	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

#### 2.8.4.2.7 RestartNetworkInterface

The access network sends this message to request the access terminal to restart the network interface.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access network shall set this field to 0x1c.

TransactionID The access network shall increment this value for each new RestartNetworkInterface message sent.

<b>Channels</b>	CC	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

#### 2.8.4.2.8 RestartNetworkInterfaceAck

The access terminal sends this message to acknowledge reception of a RestartNetworkInterface message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access terminal shall set this field to 0x1d.

TransactionID The access terminal shall set this value to the value of the TransactionID field of the corresponding RestartNetworkInterface message.

<b>Channels</b>	AC	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	unicast		<b>Priority</b>	40

1 2.8.5 Protocol Numeric Constants

2

<b>Constant</b>	<b>Meaning</b>	<b>Value</b>
$T_{FCResponse}$	Time period within which the access terminal and access network are to respond to flow control messages.	1 second

## 2.9 Configuration Attributes for the Enhanced Multi-Flow Packet Application

The access terminal shall support default values of all attributes.

Unless specified otherwise, the access terminal and the access network shall not use the Generic Attribute Update Protocol to update configurable attributes belonging to the Enhanced Multi-Flow Packet Application. The access terminal and the access network shall support the use of the Generic Attribute Update Protocol to update values of the following configurable attributes belonging to the Enhanced Multi-Flow Packet Application:

- FlowNNTimersFwd,
- FlowNNTimersRev,
- FlowNNNakEnableFwd,
- FlowNNNakEnableRev,
- FlowNNPhysicalLayerNakEnableRev,
- FlowNNAckEnableFwd,
- FlowNNAckEnableRev,
- FlowNNFlowProtocolFwd,
- FlowNNFlowProtocolRev,
- FlowNNRouteProtocolFwd,
- FlowNNRouteProtocolRev,
- FlowNNActivatedFwd,
- FlowNNActivatedRev,
- FlowNNSequenceLengthFwd,
- FlowNNSequenceLengthRev,
- FlowNNFlowProtocolSDUFwd,
- FlowNNFlowProtocolSDURev,
- FlowNNRouteProtocolSDUFwd,
- FlowNNRouteProtocolSDURev,
- FlowNNOutOfOrderDeliveryToFlowProtocolFwd,
- FlowNNOutOfOrderDeliveryToRouteProtocolFwd,
- FlowNNOutOfOrderDeliveryToRouteProtocolRev,
- FlowNNDataUnitFwd,
- FlowNNDataUnitRev,
- FlowNNReservationFwd,
- FlowNNReservationRev,

- 1 • ReservationKKIdleStateFwd,
- 2 • ReservationKKIdleStateRev,
- 3 • ReservationKKQoSRequestFwd,
- 4 • ReservationKKQoSRequestRev,
- 5 • ReservationKKQoSResponseFwd,
- 6 • ReservationKKQoSResponseRev, and
- 7 • ANSupportedQoSProfiles,

8 where  $NN$  is the hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive,  
9 and  $KK$  is the hexadecimal Reservation number in the range from 0x00 to 0xff, inclusive.  
10 The updated values of the attributes shall be consistent with the value of the  
11 LinkFlowCountMaxFwd and LinkFlowCountMaxRev attributes.

12 The access network shall not initiate modification of the ReservationKKQoSRequest  
13 attribute. The access terminal shall not initiate modification of the  
14 ReservationKKQoSResponse attribute.

### 15 2.9.1 Simple Attributes

16 The negotiable simple attribute for this protocol is listed in Table 2.9-1. The access terminal  
17 and the access network shall use as defaults the values in Table 2.9-1 typed in ***bold***  
18 ***italics***.  
19

1

**Table 2.9-1. Configurable Values**

<b>Attribute ID</b>	<b>Attribute</b>	<b>Values</b>	<b>Meaning</b>
0xffff	RANHandoff	<b>0x0000</b>	The access terminal shall not send an unsolicited LocationNotification message. The access network does not switch between the radio access technologies as specified in [2] in a manner that preserves the state of all protocol layers at or above the data link layer (PPP) specified in [2].
		0x0001	The access terminal shall send an unsolicited LocationNotification message as specified in [2]. The access network switches between the radio access technologies specified in [2] in a manner that preserves the state of all protocol layers at or above the data link layer (PPP) specified in [2].
		All other values	Reserved
0xfffe	LinkFlowCountMaxFwd	<b>0x0003</b>	Maximum 3 simultaneous activated forward Link Flows supported.
		0x0003 to $N_{\text{LinkFlowMax}}$	Maximum number of simultaneous activated forward Link Flows supported.
		0x0000 to 0x0002	Reserved
		$(N_{\text{LinkFlowMax}} + 1)$ to 0xffff	Reserved
0xfffd	LinkFlowCountMaxRev	<b>0x0003</b>	Maximum 3 simultaneous activated reverse Link Flows supported.
		0x0003 to $N_{\text{LinkFlowMax}}$	Maximum number of simultaneous activated reverse Link Flows supported.
		0x0000 to 0x0002	Reserved
		$(N_{\text{LinkFlowMax}} + 1)$ to 0xffff	Reserved
0xfffc	MaxAbortTimer	<b>0x0005</b>	Maximum abort timer is 500 ms.

Attribute ID	Attribute	Values	Meaning
		0x0000 to 0x0064	Maximum abort timer in units of 100 ms.
		All other values	Reserved
0xfffb	PANASupport	<b>0x0000</b>	Protocol for carrying Authentication for Network Access (PANA) [7] is not supported.
		0x0001	PANA is supported
		All other values	Reserved
0xfeNN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNNakEnableFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	RLP receivers associated with forward Link Flow NN do not transmit ReceiverStatus messages when missing data units are detected.
		<b>0x0001</b>	RLP receivers associated with forward Link Flow NN transmit a ReceiverStatus message when missing data units are detected.
		All other values	Reserved.
0xfdNN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNNakEnableRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	RLP receivers associated with reverse Link Flow NN do not transmit ReceiverStatus messages when missing data units are detected.
		<b>0x0001</b>	RLP receivers associated with reverse Link Flow NN transmit a ReceiverStatus message when missing data units are detected.
		All other values	Reserved.
0xfcNN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNPhysicalLayerNakEnableRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0001</b>	RLP is to retransmit data units when a <i>PhysicalLayer.ReverseTrafficPacketsMissed</i> indication is received.
		0x0000	RLP is to ignore <i>PhysicalLayer.ReverseTrafficPacketsMissed</i> indication.
		All other values	Reserved
0xfbKK	ReservationKKIdleStateFwd	<b>0x0000</b>	Reservation does not change states when a Connection is closed.

<b>Attribute ID</b>	<b>Attribute</b>	<b>Values</b>	<b>Meaning</b>
<i>KK</i> is the two-digit hexadecimal ReservationLabel.	<i>KK</i> is the two-digit hexadecimal forward link ReservationLabel, where hexadecimal digits A through F are specified in upper case letters.	0x0001	Reservation transitions to the Close state when a Connection is closed.
		All other values	Reserved
0xfa <i>KK</i> <i>KK</i> is the two-digit hexadecimal ReservationLabel.	Reservation <i>KK</i> IdleStateRev <i>KK</i> is the two-digit hexadecimal reverse link ReservationLabel, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Reservation does not change states when a Connection is closed.
		0x0001	Reservation transitions to the Close state when a Connection is closed.
		All other values	Reserved
0xf900	Flow00ActivatedFwd	<b>0x0001</b>	Forward Link Flow 0x00 is activated.
		0x0000	Forward Link Flow 0x00 is not activated.
		All other values	Reserved
0xf8 <i>NN</i> <i>NN</i> is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x01 to $N_{\text{LinkFlowMax}}-1$ inclusive.	Flow <i>NN</i> ActivatedFwd <i>NN</i> is the two-digit hexadecimal Link Flow number in the range 0x01 to $N_{\text{LinkFlowMax}}-1$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Forward Link Flow <i>NN</i> is not activated.
		0x0001	Forward Link Flow <i>NN</i> is activated.
		All other values	Reserved
0xf700	Flow00ActivatedRev	<b>0x0001</b>	Reverse Link Flow 0x00 is activated.
		0x0000	Reverse Link Flow 0x00 is not activated.
		All other values	Reserved
0xf7 <i>NN</i> <i>NN</i> is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x01 to $N_{\text{LinkFlowMax}}-1$ inclusive.	Flow <i>NN</i> ActivatedRev <i>NN</i> is the two-digit hexadecimal Link Flow number in the range 0x01 to $N_{\text{LinkFlowMax}}-1$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Reverse Link Flow <i>NN</i> is not activated.
		0x0001	Reverse Link Flow <i>NN</i> is activated.
		All other values	Reserved

<b>Attribute ID</b>	<b>Attribute</b>	<b>Values</b>	<b>Meaning</b>
0xf6NN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNSequenceLengthFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Forward Link Flow NN has a 6-bit sequence number.
		0x0001	Forward Link Flow NN has a 14-bit sequence number.
		<b>0x0002</b>	Forward Link Flow NN has a 22-bit sequence number.
		All other values	Reserved
0xf5NN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNSequenceLengthRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Reverse Link Flow NN has a 6-bit sequence number.
		0x0001	Reverse Link Flow NN has a 14-bit sequence number.
		<b>0x0002</b>	Reverse Link Flow NN has a 22-bit sequence number.
		All other values	Reserved
0xf4NN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNFlowProtocolSDUFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Forward Link Flow NN provides an octet stream to the Flow Protocol.
		0x0001	Forward Link Flow NN provides a packet stream to the Flow Protocol.
		All other values	Reserved.
0xf3NN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNFlowProtocolSDURev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Reverse Link Flow NN provides an octet stream to the Flow Protocol.
		0x0001	Reverse Link Flow NN provides a packet stream to the Flow Protocol.
		All other values	Reserved.
0xf2NN	FlowNNDataUnitFwd	<b>0x0000</b>	Data unit for forward Link Flow NN is octets.

Attribute ID	Attribute	Values	Meaning
<p><i>NN</i> is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to <math>N_{\text{LinkFlowMax}}-1</math> inclusive.</p>	<p><i>NN</i> is the two-digit hexadecimal Link Flow number in the range 0x00 to <math>N_{\text{LinkFlowMax}}-1</math> inclusive, where hexadecimal digits A through F are specified in upper case letters.</p>	0x0001	Data unit for forward Link Flow <i>NN</i> is RLP packet payload.
		All other values	Reserved.
<p>0xf1<i>NN</i></p> <p><i>NN</i> is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to <math>N_{\text{LinkFlowMax}}-1</math> inclusive.</p>	<p>Flow<i>NN</i>DataUnitRev</p> <p><i>NN</i> is the two-digit hexadecimal Link Flow number in the range 0x00 to <math>N_{\text{LinkFlowMax}}-1</math> inclusive, where hexadecimal digits A through F are specified in upper case letters.</p>	<b>0x0000</b>	Data unit for reverse Link Flow <i>NN</i> is octets.
		0x0001	Data unit for reverse Link Flow <i>NN</i> is RLP packet payload.
		All other values	Reserved.
<p>0xf0<i>NN</i></p> <p><i>NN</i> is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to <math>N_{\text{LinkFlowMax}}-1</math> inclusive.</p>	<p>Flow<i>NN</i>RouteProtocolSDUForward</p> <p><i>NN</i> is the two-digit hexadecimal Link Flow number in the range 0x00 to <math>N_{\text{LinkFlowMax}}-1</math> inclusive, where hexadecimal digits A through F are specified in upper case letters.</p>	<b>0x0000</b>	Each Route of Forward Link Flow <i>NN</i> provides an octet stream to the Route Protocol.
		0x0001	Each Route of Forward Link Flow <i>NN</i> provides a packet stream to the Route Protocol.
		All other values	Reserved.
<p>0xef<i>NN</i></p> <p><i>NN</i> is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to <math>N_{\text{LinkFlowMax}}-1</math> inclusive.</p>	<p>Flow<i>NN</i>RouteProtocolSDUReverse</p> <p><i>NN</i> is the two-digit hexadecimal Link Flow number in the range 0x00 to <math>N_{\text{LinkFlowMax}}-1</math> inclusive, where hexadecimal digits A through F are specified in upper case letters.</p>	<b>0x0000</b>	Each Route of Reverse Link Flow <i>NN</i> provides an octet stream to the Route Protocol.
		0x0001	Each Route of Reverse Link Flow <i>NN</i> provides a packet stream to the Route Protocol.
		All other values	Reserved.
<p>0xee<i>NN</i></p> <p><i>NN</i> is the two-digit hexadecimal</p>	<p>Flow<i>NN</i>OutOfOrderDeliveryToFlowProtocolFwd</p> <p><i>NN</i> is the two-digit hexadecimal Link Flow</p>	<b>0x0000</b>	Forward Link Flow <i>NN</i> delivers Flow Protocol payload in-order.
		0x0001	Forward Link Flow <i>NN</i> may deliver Flow Protocol payload out-of-order.

<b>Attribute ID</b>	<b>Attribute</b>	<b>Values</b>	<b>Meaning</b>
Link Flow number of the forward Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive.	number in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	All other values	Reserved.
0xedNN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive.	FlowNNOutOfOrderDeliveryToRouteProtocolFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Each Route of Forward Link Flow NN delivers Route Protocol payload in-order.
		0x0001	Each Route of Forward Link Flow NN may deliver Route Protocol payload out-of-order.
		All other values	Reserved.
0xecNN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive.	FlowNNOutOfOrderDeliveryToRouteProtocolRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Each Route of Reverse Link Flow NN delivers Route Protocol payload in-order.
		0x0001	Each Route of Reverse Link Flow NN may deliver Route Protocol payload out-of-order.
		All other values	Reserved.
0xebNN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive.	FlowNNAckEnableFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	RLP receivers associated with forward Link Flow NN do not transmit ReceiverStatus messages when data units are received.
		0x0001	RLP receivers associated with forward Link Flow NN send ReceiverStatus messages within Acktimer interval of receiving a data unit.
		0x0002	RLP receivers associated with forward Link Flow NN send ReceiverStatus messages within Acktimer interval of receiving a data unit. The receivers are required to send a ReceiverStatus message immediately upon receiving an RLP packet carrying the last segment of a higher layer packet.
		All other values	Reserved
0xeaNN NN is the two-	FlowNNAckEnableRev NN is the two-digit	<b>0x0000</b>	RLP receivers associated with reverse Link Flow NN do not transmit ReceiverStatus messages when data units are received.

Attribute ID	Attribute	Values	Meaning
digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive.	hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-1$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0001	RLP receivers associated with reverse Link Flow <i>NN</i> send ReceiverStatus messages within Acktimer interval of receiving a data unit.
		0x0002	RLP receivers associated with reverse Link Flow <i>NN</i> send ReceiverStatus messages within Acktimer interval of receiving a data unit. The receivers are required to send a ReceiverStatus message immediately upon receiving an RLP packet carrying the last segment of a higher layer packet.
		All other values	Reserved

## 2.9.2 Complex Attributes

The following complex attributes and default values are defined (see [8] for attribute record definition).

### 2.9.2.1 FlowNNTimersFwd Attribute

*NN* is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
AbortTimer	16	0x01f4
FlushTimer	16	0x012c
AckTimer	16	0x0064

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x01*NN*, where *NN* is the two-digit hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive.

**ValueID** The sender shall set this field to an identifier assigned to this complex value.

- 1 **AbortTimer**            The sender shall set this field to the value of the RLP abort timer for  
2 this forward Link Flow in units of ms. The sender shall not set this  
3 field to a value greater than MaxAbortTimer.
- 4 **FlushTimer**            The sender shall set this field to the value of the RLP flush timer for  
5 this forward Link Flow in units of ms. The value of the RLP flush  
6 timer shall be less than or equal to that of the corresponding abort  
7 timer.
- 8 **AckTimer**              The sender shall set this field to the value to the RLP Ack timer for  
9 this forward Link Flow in units of ms.

### 10 2.9.2.2 FlowNNTimersRev Attribute

11 *NN* is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range  
12 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive, where hexadecimal digits A through F are specified in upper  
13 case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
AbortTimer	16	0x01f4
FlushTimer	16	0x012c
AckTimer	16	0x0064

- 15 **Length**                Length of the complex attribute in octets. The sender shall set this  
16 field to the length of the complex attribute excluding the Length field.
- 17 **AttributeID**            The sender shall set this field to 0x02*NN*, where *NN* is the two-digit  
18 hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$   
19 inclusive.
- 20 **ValueID**                The sender shall set this field to an identifier assigned to this complex  
21 value.
- 22 **AbortTimer**            The sender shall set this field to the value of the RLP abort timer for  
23 this reverse Link Flow in units of ms. The sender shall not set this  
24 field to a value greater than MaxAbortTimer.
- 25 **FlushTimer**            The sender shall set this field to the value of the RLP flush timer for  
26 this reverse Link Flow in units of ms. The value of the RLP flush  
27 timer shall be less than or equal to that of the corresponding abort  
28 timer.

1 AckTimer The sender shall set this field to the value to the RLP Ack timer for  
2 this reverse Link Flow in units of ms.

### 3 2.9.2.3 FlowNNReservationFwd Attribute

4 NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range  
5 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive, where hexadecimal digits A through F are specified in upper  
6 case letters.

Field	Length (bits)	Default for NN = 0x00	Default for NN > 0x00
Length	8	N/A	N/A
AttributeID	16	N/A	N/A

One or more of the following record:

ValueID	8	N/A	N/A
ReservationCount	8	0x01	0x00

ReservationCount occurrences of the following field:

ReservationLabel	8	0xff	N/A
------------------	---	------	-----

8 Length Length of the complex attribute in octets. The sender shall set this  
9 field to the length of the complex attribute excluding the Length field.

10 AttributeID The sender shall set this field to 0x03NN, where NN is the two-digit  
11 hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$   
12 inclusive.

13 ValueID The sender shall set this field to an identifier assigned to this complex  
14 value.

15 ReservationCount The sender shall set this field to the number of reservations  
16 associated with this Link Flow.

17 ReservationLabel The sender shall set this field to the ReservationLabel of the  
18 reservation associated with this Link Flow.

### 19 2.9.2.4 FlowNNReservationRev Attribute

20 NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range  
21 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive, where hexadecimal digits A through F are specified in upper  
22 case letters.

23

Field	Length (bits)	Default for NN = 0x00	Default for NN >= 0x00
Length	8	N/A	N/A
AttributeID	16	N/A	N/A

One or more of the following record:

ValueID	8	N/A	N/A
ReservationCount	8	0x01	0x00

ReservationCount occurrences of the following field:

ReservationLabel	8	0xff	N/A
------------------	---	------	-----

- 1 Length Length of the complex attribute in octets. The sender shall set this
- 2 field to the length of the complex attribute excluding the Length field.
  
- 3 AttributeID The sender shall set this field to 0x04NN, where NN is the two-digit
- 4 hexadecimal Link Flow number in the range 0x00 to  $N_{LinkFlowMax}-1$
- 5 inclusive.
  
- 6 ValueID The sender shall set this field to an identifier assigned to this complex
- 7 value.
  
- 8 ReservationCount The sender shall set this field to the number of reservations
- 9 associated with this Link Flow.
  
- 10 ReservationLabel The sender shall set this field to the ReservationLabel of the
- 11 reservation associated with this Link Flow.

12 2.9.2.5 SupportedFlowProtocolIDs Attribute

13

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProtocolCount	8	0x02

ProtocolCount occurrences of the following field:

ProtocolIdentifier	8	See Table 2.9.2.5-1.
--------------------	---	----------------------

- 14 Length Length of the complex attribute in octets. The sender shall set this
- 15 field to the length of the complex attribute excluding the Length field.
  
- 16 AttributeID The sender shall set this field to 0x0000.

1	<b>ValueID</b>	The sender shall set this field to an identifier assigned to this complex value.
2		
3	<b>ProtocolCount</b>	The sender shall set this field to the number of occurrences of the ProtocolIdentifier field in this record.
4		
5	<b>ProtocolIdentifier</b>	The sender shall set this field to an identifier for the supported Flow Protocol as defined in [3].
6		

7 **Table 2.9.2.5-1. Default Values of ProtocolIdentifier in SupportedFlowProtocolIDs**  
8 **Attribute**

Profile	Description
0x0000	NULL
0x0001	Octet-based HDLC-like framing

9 **2.9.2.6 SupportedRouteProtocolIDs Attribute**  
10

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProtocolCount	8	0x01

ProtocolCount occurrences of the following field:

ProtocolIdentifier	8	0x0000 (NULL)
--------------------	---	---------------

11	<b>Length</b>	Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.
12		
13	<b>AttributeID</b>	The sender shall set this field to 0x0001.
14	<b>ValueID</b>	The sender shall set this field to an identifier assigned to this complex value.
15		
16	<b>ProtocolCount</b>	The sender shall set this field to the number of occurrences of the ProtocolIdentifier field in this record.
17		
18	<b>ProtocolIdentifier</b>	The sender shall set this field to an identifier for the supported Route Protocol as defined in [3].
19		

## 2.9.2.7 ATSupportedQoSProfiles Attribute

<b>Field</b>	<b>Length (bits)</b>	<b>Default</b>
Length	8	N/A
AttributeID	16	N/A

One occurrence of the following record:

ValueID	8	N/A
QoSProfileCount	8	0

QoSProfileCount of the following record:

ProfileType	8	N/A
ProfileLength	8	N/A
ProfileValue	ProfileLength × 8	N/A

- 3    **Length**                      Length of the complex attribute in octets. The sender shall set this  
4                                      field to the length of the complex attribute excluding the Length field.
- 5    **AttributeID**                The sender shall set this field to 0x0002.
- 6    **ValueID**                      The sender shall set this field to an identifier assigned to this complex  
7                                      value.
- 8    **QoSProfileCount**          The sender shall set this field to the number of QoS Profiles that are  
9                                      included in this message.
- 10   **ProfileType**                The sender shall set this field to indicate the profile type. The sender  
11                                      shall not set this field to 0x00. The sender shall set this field  
12                                      according to [3].
- 13   **ProfileLength**              The sender shall set this field to length of the ProfileValue field in  
14                                      units of octets.
- 15   **ProfileValue**                The sender shall set this field to the profile according to [3].

## 2.9.2.8 ANSupportedQoSProfiles Attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One occurrence of the following record:

ValueID	8	N/A
QoSProfileCount	8	0

QoSProfileCount of the following record:

ProfileType	8	N/A
ProfileLength	8	N/A
ProfileValue	ProfileLength × 8	N/A

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x0003.

**ValueID** The sender shall set this field to an identifier assigned to this complex value.

**QoSProfileCount** The sender shall set this field to the number of QoS Profiles that are included in this message.

**ProfileType** The sender shall set this field to indicate the profile type. The sender shall not set this field to 0x00. The sender shall set this field according to [3].

**ProfileLength** The sender shall set this field to length of the ProfileValue field in units of octets.

**ProfileValue** The sender shall set this field to the profile according to [3].

## 2.9.2.9 ReservationKKQoSRequestFwd Attribute

**KK** is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProfileType	8	<b>0x00</b>
ProfileLength	16	<b>0x0000</b>
ProfileValue	ProfileLength × 8	N/A

- 1    **Length**                      Length of the complex attribute in octets. The sender shall set this  
2                                      field to the length of the complex attribute excluding the Length field.
- 3    **AttributeID**                The sender shall set this field to 0x05KK, where KK is the two-digit  
4                                      hexadecimal ReservationLabel.
- 5    **ValueID**                      The sender shall set this field to an identifier assigned to this complex  
6                                      value.
- 7    **ProfileType**                The sender shall set this field to indicate the profile type. If the profile  
8                                      type is NULL, then the sender shall set this field to 0x00. Otherwise,  
9                                      the sender shall set this field as defined in [3].
- 10   **ProfileLength**              The sender shall set this field to length of the ProfileValue field in  
11                                      units of octets. If ProfileType is equal to 0x00, then the sender shall  
12                                      set this field to 0x0000.
- 13   **ProfileValue**                The sender shall set this field to the profile. If ProfileType is equal to  
14                                      0x00, then the sender shall omit this field. If the corresponding  
15                                      ProfileType is set to 0x01 and this attribute is included in an  
16                                      AttributeUpdateRequest message, the sender shall set this field to a  
17                                      ProfileValue that is included in the ANSupportedQoSProfiles  
18                                      attribute.

#### 19    2.9.2.10 ReservationKKQoSRequestRev Attribute

20    **KK** is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F  
21    are specified in upper case letters.  
22

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProfileType	8	<b>0x00</b>
ProfileLength	16	<b>0x0000</b>
ProfileValue	ProfileLength × 8	N/A

- 1    **Length**                      Length of the complex attribute in octets. The sender shall set this  
2                                      field to the length of the complex attribute excluding the Length field.
- 3    **AttributeID**                The sender shall set this field to 0x06KK, where KK is the two-digit  
4                                      hexadecimal ReservationLabel.
- 5    **ValueID**                      The sender shall set this field to an identifier assigned to this complex  
6                                      value.
- 7    **ProfileType**                The sender shall set this field to indicate the profile type. If the profile  
8                                      type is NULL, then the sender shall set this field to 0x00. Otherwise,  
9                                      the sender shall set this field as defined in [3].
- 10   **ProfileLength**              The sender shall set this field to length of the ProfileValue field in  
11                                      units of octets. If ProfileType is equal to 0x00, then the sender shall  
12                                      set this field to 0x0000.
- 13   **ProfileValue**                The sender shall set this field to the profile. If ProfileType is equal to  
14                                      0x00, then the sender shall omit this field. If the corresponding  
15                                      ProfileType is set to 0x01 and this attribute is included in an  
16                                      AttributeUpdateRequest message, the sender shall set this field to a  
17                                      ProfileValue that is included in the ANSupportedQoSProfiles  
18                                      attribute.

#### 19    2.9.2.11 ReservationKKQoSResponseFwd Attribute

20    *KK* is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F  
21    are specified in upper case letters.  
22

<b>Field</b>	<b>Length (bits)</b>	<b>Default</b>
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProfileType	8	<b>0x00</b>
ProfileLength	16	<b>0x0000</b>
ProfileValue	ProfileLength × 8	N/A

- 1    **Length**                      Length of the complex attribute in octets. The sender shall set this  
2                                      field to the length of the complex attribute excluding the Length field.
- 3    **AttributeID**                The sender shall set this field to 0x07KK, where KK is the two-digit  
4                                      hexadecimal ReservationLabel.
- 5    **ValueID**                      The sender shall set this field to an identifier assigned to this complex  
6                                      value.
- 7    **ProfileType**                The sender shall set this field to indicate the profile type. If the profile  
8                                      type is NULL, then the sender shall set this field to 0x00. Otherwise,  
9                                      the sender shall set this field as defined in [3].
- 10   **ProfileLength**              The sender shall set this field to length of the ProfileValue field in  
11                                      units of octets. If ProfileType is equal to 0x00, then the sender shall  
12                                      set this field to 0x0000.
- 13   **ProfileValue**                The sender shall set this field to the profile. If ProfileType is equal to  
14                                      0x00, then the sender shall omit this field.

#### 15    2.9.2.12 ReservationKKQoSResponseRev Attribute

16    **KK** is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F  
17    are specified in upper case letters.

18

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProfileType	8	<b>0x00</b>
ProfileLength	16	<b>0x0000</b>
ProfileValue	ProfileLength × 8	N/A

- 1    **Length**                      Length of the complex attribute in octets. The sender shall set this  
2                                      field to the length of the complex attribute excluding the Length field.
- 3    **AttributeID**                The sender shall set this field to 0x08KK, where KK is the two-digit  
4                                      hexadecimal ReservationLabel.
- 5    **ValueID**                      The sender shall set this field to an identifier assigned to this complex  
6                                      value.
- 7    **ProfileType**                The sender shall set this field to indicate the profile type. If the profile  
8                                      type is NULL, then the sender shall set this field to 0x00. Otherwise,  
9                                      the sender shall set this field as defined in [3].
- 10   **ProfileLength**              The sender shall set this field to length of the ProfileValue field in  
11                                      units of octets. If ProfileType is equal to 0x00, then the sender shall  
12                                      set this field to 0x0000.
- 13   **ProfileValue**                The sender shall set this field to the profile. If ProfileType is equal to  
14                                      0x00, then the sender shall omit this field.

#### 15    2.9.2.13 FlowNNFlowProtocolFwd Attribute

16    NN is the two-digit hexadecimal forward Link flow identifier, where hexadecimal digits A  
17    through F are specified in upper case letters.

18

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProtocolID	8	0x00
ProtocolParametersLength	8	0x00
ProtocolParameters	ProtocolParametersLength × 8	N/A

1 Length Length of the complex attribute in octets. The sender shall set this  
2 field to the length of the complex attribute excluding the Length field.

3 AttributeID The sender shall set this field to 0x09NN, where NN is the two-digit  
4 hexadecimal forward Link flow number.

5 ValueID The sender shall set this field to an identifier assigned to this complex  
6 value.

7 ProtocolID The sender shall set this field to n identifier for the Flow Protocol  
8 according to Table 2.9.2.13-1.

9 **Table 2.9.2.13-1. ProtocolID for Flow Protocol**

Value	Protocol
0x00	NULL
0x01	Octet-based HDLC-like framing [4]
0x02	Internet Protocol (IP) version 4 [5]
0x03	IP version 6 [10]
0x04	The Flow Protocol is Robust Header Compression (ROHC) [6]
All other values	Defined in [3]

10 ProtocolParametersLength  
11 The sender shall set this field to the length of the ProtocolParameters  
12 record in units of octets.

13 ProtocolParameters

Unless specified otherwise by the document that defines the correspondence between the ProtocolID and the associated Flow Protocol, the sender shall omit this record. If ProtocolID is 0x04, then the sender shall set this record as defined in 2.9.2.13.1..

#### 2.9.2.13.1 Definition of ProtocolParameters record when the Flow Protocol is ROHC

Field	Length (bits)
MaxCID	16
LargeCIDs	1
FeedbackForIncluded	1
FeedbackFor	0 or 5
MRRU	16
ProfileCount	8
ProfileCount occurrences of the following field:	
Profile	16
Reserved	0 – 7 (as needed)

**MaxCID** The sender shall set this field to the MAX\_CID parameter for this ROHC Channel.

**LargeCIDs** If the LARGE\_CIDS parameter for this ROHC Channel is false, then the sender shall set this field to '0'. Otherwise, the sender shall set this field to '1'.

**FeedbackForIncluded** If ROHC feedback associated with another Link flow (ROHC channel) is sent on this Link flow (ROHC channel), then this field shall be set to '1'. Otherwise, this field shall be set to '0'.

**FeedbackFor** If FeedbackForIncluded is set to '0', then the sender shall omit this field. Otherwise, the sender shall set this field to the Link flow number (ROHC channel) to which ROHC feedback sent on this Link flow (ROHC channel) refers.

**MRRU** The sender shall set this field to the MRRU parameter for this ROHC channel.

**ProfileCount** The sender shall set this field to the number of ROHC profiles supported by the decompressor.

**Profile** The sender shall set this field to the ROHC profile supported by the decompressor according to [6].

1 Reserved The sender shall add reserved bits to make the length of the entire  
 2 record an integer number of octets. The sender shall set these bits to  
 3 '0'. The receiver shall ignore this field.

4 **2.9.2.14 FlowNNFlowProtocolRev Attribute**

5 *NN* is the two-digit hexadecimal reverse Link flow number, where hexadecimal digits A  
 6 through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

7 One or more of the following record:

ValueID	8	N/A
ProtocolID	8	0x00
ProtocolParametersLength	8	0x00
ProtocolParameters	ProtocolParametersLength × 8	N/A

8 **Length** Length of the complex attribute in octets. The sender shall set this  
 9 field to the length of the complex attribute excluding the Length field.

10 **AttributeID** The sender shall set this field to 0x0a*NN*, where *NN* is a two-digit  
 11 hexadecimal reverse Link flow number.

12 **ValueID** The sender shall set this field to an identifier assigned to this complex  
 13 value.

14 **ProtocolID** The sender shall set this field to an identifier for the Flow Protocol  
 15 according to Table 2.9.2.13-1.

16 **ProtocolParametersLength**  
 17 The sender shall set this field to the length of the ProtocolParameters  
 18 record in units of octets.

19 **ProtocolParameters** Unless specified otherwise by the document that defines the  
 20 correspondence between the ProtocolID and the associated Flow  
 21 Protocol, the sender shall omit this record. If ProtocolID is 0x04, then  
 22 the sender shall set this record as defined in as defined in 2.9.2.13.1.

23 **2.9.2.15 FlowNNRouteProtocolFwd Attribute**

24 *NN* is the two-digit hexadecimal forward Link flow number, where hexadecimal digits A  
 25 through F are specified in upper case letters.

26

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProtocolID	8	0x00
ProtocolParametersLength	8	0x00
ProtocolParameters	ProtocolParametersLength × 8	N/A

- 1 Length Length of the complex attribute in octets. The sender shall set this  
2 field to the length of the complex attribute excluding the Length field.
- 3 AttributeID The sender shall set this field to 0x0bNN, where NN is a two-digit  
4 hexadecimal forward Link flow number.
- 5 ValueID The sender shall set this field to an identifier assigned to this complex  
6 value.
- 7 ProtocolID The sender shall set this field to an identifier for the Route Protocol  
8 according to Table 2.9.2.15-1.

9 **Table 2.9.2.15-1. ProtocolID for Route Protocol**

Value	Protocol
0x00	NULL
0x01 to 0x03	Reserved
0x04	The Flow Protocol is Robust Header Compression (ROHC) [6]
All other values	Defined in [3]

- 10 ProtocolParametersLength  
11 The sender shall set this field to the length of the ProtocolParameters  
12 record in units of octets.
- 13 ProtocolParameters Unless specified otherwise by the document that defines the  
14 correspondence between the ProtocolID and the associated Route  
15 Protocol, the sender shall omit this record. If ProtocolID is 0x04, then  
16 the sender shall set this field as defined in as defined in 2.9.2.13.1.

### 2.9.2.16 FlowNNRouteProtocolRev Attribute

*NN* is the two-digit hexadecimal reverse Link flow number, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A

One or more of the following record:

ValueID	8	N/A
ProtocolID	8	0x00
ProtocolParametersLength	8	0x00
ProtocolParameters	ProtocolParametersLength × 8	N/A

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x0c*NN*, where *NN* is a two-digit hexadecimal reverse Link flow number.

**ValueID** The sender shall set this field to an identifier assigned to this complex value.

**ProtocolID** The sender shall set this field to an identifier for the Route Protocol according to Table 2.9.2.15-1.

**ProtocolParametersLength** The sender shall set this field to the length of the ProtocolParameters record in units of octets.

**ProtocolParameters** Unless specified otherwise by the document that defines the correspondence between the ProtocolID and the associated Route Protocol, the sender shall omit this record. If ProtocolID is 0x04, then the sender shall set this field as defined in as defined in 2.9.2.13.1.

## 2.10 Session State Information

The Session State Information record (see [8]) consists of parameter records.

This application defines the following parameter records in addition to the configuration attributes for this application.

## 2.10.1 Location Parameter

**Table 2.10.1-1. The Format of the Parameter Record for the Location Parameter**

<b>Field</b>	<b>Length (bits)</b>
ParameterType	8
Length	8
LocationType	8
LocationValue	$8 \times (\text{Length} - 2)$

ParameterType	This field shall be set to 0x01 for this parameter record.
Length	This field shall be set to the length of this parameter record in units of octets excluding the Length field.
LocationType	This field shall be set to the value of LocationType associated with the access terminal's session.
LocationValue	This field shall be set to the stored value of LocationValue associated with the access terminal's session.

## 2.10.2 FlowControlState Parameter

**Table 2.10.2-1. The Format of the Parameter Record for the FlowControlState Parameter**

<b>Field</b>	<b>Length (bits)</b>
ParameterType	8
Length	8
FlowControlState	8

ParameterType	This field shall be set to 0x02 for this parameter record.
Length	This field shall be set to the length of this parameter record in units of octets excluding the Length field.
FlowControlState	This field shall be set to 0x00 if the state of the Flow Control Protocol associated with the access terminal's session is Close. Otherwise, this field shall be set to 0x01. All the other values for this field are reserved.

## 2.10.3 DataOverSignalingMessageSequence Parameter

**Table 2.10.3-1. The Format of the Parameter Record for the DataOverSignalingMessageSequence Parameter**

Field	Length (bits)
ParameterType	8
Length	8
Reserved1	1
ReceivePointerA	7
Reserved2	1
TransmitPointerA	7
Reserved3	1
ReceivePointerB	7
Reserved4	1
TransmitPointerB	7

4	ParameterType	This field shall be set to 0x03 for this parameter record.
5	Length	This field shall be set to the length of this parameter record in units of octets excluding the Length field.
6		
7	Reserved1	The sender shall set this field to '0'. The receiver shall ignore this field.
8		
9	ReceivePointerA	This field shall be set to the value of the receive pointer for DataOverSignaling message validation on Route A, $V(R_A)$ .
10		
11	Reserved2	The sender shall set this field to '0'. The receiver shall ignore this field.
12		
13	TransmitPointerA	This field shall be set to the value of the transmit pointer for DataOverSignaling message validation on Route A, $V(S_A)$ .
14		
15	Reserved3	The sender shall set this field to '0'. The receiver shall ignore this field.
16		
17	ReceivePointerB	This field shall be set to the value of the receive pointer for DataOverSignaling message validation on Route B, $V(R_B)$ .
18		
19	Reserved4	The sender shall set this field to '0'. The receiver shall ignore this field.
20		

1 TransmitPointerB This field shall be set to the value of the transmit pointer for  
2 DataOverSignaling message validation on Route B,  $V(S_B)$ .

### 3 2.10.4 ServiceNetworkID Parameter

4 **Table 2.10.4-1. The Format of the Parameter Record for the ServiceNetworkID**  
5 **Parameter**

Field	Length (bits)
ParameterType	8
Length	8
ServiceNetworkIDLength	8
ServiceNetworkID	0 or $8 \times$ ServiceNetworkIDLength

6 ParameterType This field shall be set to 0x04 for this parameter record.

7 Length This field shall be set to the length of this parameter record in units  
8 of octets excluding the Length field.

#### 9 ServiceNetworkIDLength

10 This field shall be set to zero if the value of its stored  
11 ServiceNetworkID is NULL; otherwise, this field shall be set to the  
12 stored value of ServiceNetworkIDLength.

13 ServiceNetworkID This field shall be omitted if the value of the stored ServiceNetworkID  
14 is NULL; otherwise, this field shall be set to the stored value of  
15 ServiceNetworkID.

## 2.10.5 ReservationState Parameter

**Table 2.10.5-1. The Format of the Parameter Record for the ReservationState Parameter**

<b>Field</b>	<b>Length (bits)</b>
ParameterType	8
Length	8
OpenReservationCount	8
OpenReservationCount occurrences of the following record:	
Link	1
ReservationLabel	8
Reserved	0 – 7 (as needed)

- 4 **ParameterType** This field shall be set to 0x05 for this parameter record.
- 5 **Length** This field shall be set to the length of this parameter record in units  
6 of octets excluding the Length field.
- 7 **OpenReservationCount**
- 8 This field shall be set to the number of Reservations that are in the  
9 Open state.
- 10 **Link** This field shall be set to '1' for a forward link Reservation, and to '0'  
11 for a reverse link Reservation.
- 12 **ReservationLabel** This field shall be set to the ReservationLabel of the Open  
13 Reservation.
- 14 **Reserved** The sender shall add reserved bits to make the length of the entire  
15 parameter an integer number of octets. The sender shall set these  
16 bits to zero. The receiver shall ignore this field

## 2.10.6 RouteState Parameter

**Table 2.10.6-1. The Format of the Parameter Record for the RouteState Parameter**

Field	Length (bits)
ParameterType	8
Length	8
RouteSelectionProtocolState	2
NextRouteSelectTransactionID	8
NextActivateRouteTransactionID	8
Reserved	6

**ParameterType** This field shall be set to 0x06 for this parameter record.

**Length** This field shall be set to the length of this parameter record in units of octets excluding the Length field.

**RouteSelectionProtocolState**

This field shall be set to indicate the state of Route Selection Protocol according to Table 2.10.6-2.

**Table 2.10.6-2. RouteSelectionProtocolState Encoding**

State	Value
A Open B Setting	'00'
A Open B Rising	'01'
A Setting B Open	'10'
A Rising B Open	'11'

**NextRouteSelectTransactionID**

This field shall be set to the TransactionID field of the next RouteSelect message that will be sent.

**NextActivateRouteTransactionID**

This field shall be set to the TransactionID field of the next ActivateRoute message that will be sent.

**Reserved** This field shall be set to '000000'. The receiver shall ignore this field.