

**3GPP2 C.S0098-100-0**

**Version 1.0**

**Date: January 2011**



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

---

## ***Introduction to cdma2000 Extended Cell High Rate Packet Data Air Interface Specification***

### ***Revision 0***

#### ***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.

## Revision History

---

<b>Revision</b>	<b>Description of Changes</b>	<b>Date</b>
Rev 0 v1.0	Initial Publication	January 2011

**CONTENTS**

1	FOREWORD.....	ix
2	REFERENCES.....	xi
3	1 Overview .....	1-1
4	1.1 Scope of This Document .....	1-1
5	1.2 Requirements Language.....	1-1
6	1.3 Architecture Reference Model.....	1-1
7	1.4 Protocol Architecture .....	1-2
8	1.4.1 Layers.....	1-2
9	1.5 Physical Layer Channels .....	1-3
10	1.6 Protocols .....	1-4
11	1.6.1 Interfaces.....	1-4
12	1.6.2 States .....	1-5
13	1.6.3 InUse and InConfiguration Protocol/Application Instances.....	1-6
14	1.6.3.1 InConfiguration Instantiation.....	1-6
15	1.6.3.1.1 Protocol Instantiation.....	1-6
16	1.6.3.1.2 Application Instantiation.....	1-6
17	1.6.3.2 Protocol Initialization.....	1-6
18	1.6.3.3 Procedures and Messages .....	1-6
19	1.6.3.3.1 Commit Procedures .....	1-7
20	1.6.4 Common Commands.....	1-7
21	1.6.5 Protocol Negotiation .....	1-7
22	1.6.6 Protocol Overview.....	1-7
23	1.7 Default Applications .....	1-11
24	1.8 Streams.....	1-12
25	1.9 Sessions and Connections.....	1-12
26	1.10 Security.....	1-12
27	1.11 Terms .....	1-12
28	1.12 Notation .....	1-16
29	1.13 Malfunction Detection.....	1-17
30	1.14 CDMA System Time .....	1-17
31	1.15 Revision Number .....	1-20
32	2 Common Algorithms and Data Structures.....	2-1

**CONTENTS**

1 2.1 Channel Record..... 2-1

2 2.2 Access Terminal Identifier Record ..... 2-1

3 2.3 Attribute Record..... 2-2

4 2.4 Hash Function ..... 2-4

5 2.5 Pseudorandom Number Generator ..... 2-4

6 2.5.1 General Procedures ..... 2-4

7 2.5.2 Initialization ..... 2-5

8 2.6 Sequence Number Validation..... 2-5

9 2.7 Generic Configuration Protocol ..... 2-5

10 2.7.1 Introduction ..... 2-5

11 2.7.2 Procedures ..... 2-6

12 2.7.2.1 Configuration Negotiation..... 2-6

13 2.7.3 Message Formats..... 2-7

14 2.7.3.1 ConfigurationRequest..... 2-7

15 2.7.3.2 ConfigurationResponse ..... 2-7

16 2.8 Session State Information Record ..... 2-8

17 2.9 SectorID Provisioning ..... 2-10

18 2.9.1 Overview of Relevant Formats ..... 2-10

19 2.9.1.1 Global Unicast IPv6 Address Format..... 2-10

20 2.9.1.2 Site-Local Unicast IPv6 Address Format ..... 2-10

21 2.9.1.3 Link-Local Unicast IPv6 Address Format ..... 2-10

22 2.9.1.4 Reserved IPv6 Address Format ..... 2-11

23 2.9.1.5 Modified EUI-64 Format ..... 2-11

24 2.9.2 SectorID Construction ..... 2-12

25 2.9.2.1 Construction of Globally Unique SectorID..... 2-12

26 2.9.2.1.1 SectorID Based On an IPv6 Unique Identifier ..... 2-12

27 2.9.2.1.2 SectorID Not Based On an IPv6 Unique Identifier ..... 2-13

28 2.9.2.1.2.1 ANSI-41 Method ..... 2-14

29 2.9.2.1.2.2 GSM/UMTS Method ..... 2-14

30 2.9.2.1.2.3 IPv4 Unique Identifier ..... 2-15

31 2.9.2.2 Construction of Locally Unique SectorID..... 2-15

32 2.10 Generic Attribute Update Protocol..... 2-15

**CONTENTS**

1        2.10.1 Introduction.....2-15

2        2.10.2 Procedures .....2-16

3            2.10.2.1 Initiator Requirements .....2-16

4            2.10.2.2 Responder Requirements .....2-16

5        2.10.3 Message Formats .....2-17

6            2.10.3.1 AttributeUpdateRequest .....2-17

7            2.10.3.2 AttributeUpdateAccept.....2-17

8            2.10.3.3 AttributeUpdateReject .....2-18

9        2.10.4 Protocol Numeric Constants .....2-18

10       2.11 Linear Interpolation .....2-19

11       2.12 Bi-linear Interpolation .....2-20

12       2.13 IIR filter implementation .....2-21

13       2.14 ReverseChannel Record.....2-21

14       3 Assigned Names And Numbers .....3-1

15            3.1 Protocols .....3-1

16

17

**CONTENTS**

- 1 This page intentionally left blank

**FIGURES**

1	Figure 1.3-1. Architecture Reference Model.....	1-2
2	Figure 1.4.1-1. Air Interface Layering Architecture .....	1-2
3	Figure 1.5-1. Forward Channel Structure .....	1-3
4	Figure 1.5-2. Reverse Channel Structure .....	1-4
5	Figure 1.6.6-1. Default Protocols .....	1-8
6	Figure 1.6.6-2. Non-Default Protocols .....	1-9
7	Figure 1.14-1. CDMA System Time Line.....	1-19
8	Figure 2.9.1.1-1. Global Unicast IPv6 Address Format.....	2-10
9	Figure 2.9.1.2-1. Site-Local Unicast IPv6 Address Format .....	2-10
10	Figure 2.9.1.3-1. Link-Local Unicast IPv6 Address Format.....	2-11
11	Figure 2.9.1.4-1. Format of the Reserved IPv6 Addresses.....	2-11
12	Figure 2.9.1.4-2. IPv6 Values That Are to be Avoided.....	2-11
13	Figure 2.9.1.5-1. Universally Unique Modified EUI-64 .....	2-12
14	Figure 2.9.1.5-2. Locally Unique Modified EUI-64 .....	2-12
15	Figure 2.9.2.1.2-1. “S” bits in the Site-Local Unicast IPv6 Address Format.....	2-13
16	Figure 2.9.2.1.2-2. “S” bits in the Link-Local Unicast IPv6 Address Format.....	2-13
17	Figure 2.9.2.1.2-3. “S” bits in the Reserved IPv6 Address Format.....	2-13
18	Figure 2.9.2.1.2-4. Sub-fields of the “S” bits.....	2-13
19	Figure 2.9.2.1.2.1-1. Assignment of the “T” Bits, the “N” Bits, and the “X” Bits for	
20	the ANSI-41 Method.....	2-14
21	Figure 2.9.2.1.2.2-1. Assignment of the “T” Bits, the “N” Bits, and the “X” Bits for	
22	the GSM/UMTS Method.....	2-14
23	Figure 2.9.2.1.2.3-1. Assignment of the “T” Bits, the “N” Bits, and the “X” Bits for	
24	the IPv4 Method.....	2-15
25	Figure 2.9.2.2-1. Format of the Locally Unique SectorID.....	2-15

26

27

**FIGURES**

- 1 This page intentionally left blank.

**TABLES**

1 Table 2.1-1. SystemType Encoding .....2-1  
2 Table 2.2-1. ATType Field Encoding .....2-2  
3 Table 2.8-1. The Format of the Session State Information Record .....2-8  
4 Table 2.8-2. Encoding of the ParameterType Field .....2-9  
5 Table 3.1-1. Protocol Type and Subtypes.....3-2  
6

**TABLES**

- 1
- 2 This page intentionally left blank.

**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 Introduction part of the high  
5 rate packet data air interface. Other parts of this standard are:

- 6 • Physical Layer for cdma2000 Extended Cell High Rate Packet Data Air Interface  
7 Specification
- 8 • Upper Layers for cdma2000 Extended Cell High Rate Packet Data Air Interface  
9 Specification

10

11

---

<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.”

**FOREWORD**

- 1 This page intentionally left blank.

## REFERENCES

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

### Normative References:

- [1] 3GPP2 C.S0024-100-C, Introduction to cdma2000 High Rate Packet Data Air Interface Specification, April 2010.
- [2] 3GPP2 C.S0024-200-C, Physical Layer for cdma2000 High Rate Packet Data Air Interface Specification, April 2010.
- [3] 3GPP2 C.S0024-300-C, Medium Access Control Layer for cdma2000 High Rate Packet Data Air Interface Specification, April 2010.
- [4] 3GPP2 C.S0024-400-C, Connection and Security Layers for cdma2000 High Rate Packet Data Air Interface Specification, April 2010.
- [5] 3GPP2 C.S0024-500-C, Application, Stream and Session Layers for cdma2000 High Rate Packet Data Air Interface Specification, April 2010.
- [6] Reserved.
- [7] 3GPP2 C.S0098-200-0, Physical Layer for Extended Cell cdma2000 High Rate Packet Data Air Interface Specification, January 2011.
- [8] 3GPP2 C.S0098-300-0, Upper Layers for Extended Cell cdma2000 High Rate Packet Data Air Interface Specification, January 2011.
- [9] 3GPP2 P.S0001-B v2.0, Wireless IP Network Standard, October 2004.
- [10] 3GPP2 C.S0002-E v2.0, Physical Layer Standard for cdma2000 Spread Spectrum Systems, June 2010.
- [11] 3GPP2 C.S0005-E v2.0, Upper Layer (Layer 3) Signaling Specification for cdma2000 Spread Spectrum Systems, June 2010.
- [12] 3GPP2, Recommended Minimum Performance Standards for Extended Cell High Rate Packet (xHRPD) Data Access Network<sup>2</sup>.
- [13] 3GPP2, Recommended Minimum Performance Standards for Extended Cell High Rate Packet Data (xHRPD) Access Terminal<sup>2</sup>.
- [14] FIPS PUB 180-1, Federal Information Processing Standards Publication 180-1.

---

<sup>2</sup> Editor's Note: The above documents are work in progress and should not be referenced unless and until they are approved and published. Until such time as this Editor's Note is removed, the inclusion of the above documents is for informational purposes only.

## REFERENCES

- 1 [15] Internet Engineering Task Force (IETF) RFC 2409, The Internet Key Exchange  
2 (IKE).
- 3 [16] 3GPP2 A.S0009-C v3.0, IOS for HRPD Radio Access Network Interfaces with  
4 Session Control in the Packet Control Function, June 2010.
- 5 [17] 3GPP2 A.S0008-C v3.0, IOS for HRPD Radio Access Network Interfaces with  
6 Session Control in the Access Network, June 2010.
- 7 [18] 3GPP2 C.R1001-0 v1.0, Administration of Parameter Value Assignments for  
8 cdma2000 Spread Spectrum Standards. (Informative), December 1999.
- 9 [19] IETF RFC 2373, IP Version 6 Addressing Architecture.
- 10 [20] ITU-T Recommendation E.212, Identification Plan for Land Mobile Stations, 1988.
- 11 [21] IETF RFC 3056, Connection of IPv6 Domains via IPv4 Clouds, February 2001.
- 12 [22] 3GPP2 C.S0054-A v1.0, cdma2000 High Rate Broadcast-Multicast Packet Data Air  
13 Interface Specification, March 2006.
- 14 [23] 3GPP2 C.S0057-E v1.0, Band Class Specification for cdma2000 Spread Spectrum  
15 Systems, October 2010.
- 16 [24] 3GPP2 C.S0004-E v2.0, Signaling Link Access Control (LAC) standard for  
17 cdma2000 Spread Spectrum Systems, June 2010.
- 18 [25] IETF RFC 1662, PPP in HDLC-like Framing.
- 19 [26] 3GPP2 X.S0011-E, cdma2000 Wireless IP Network Standard, November 2009.
- 20 [27] 3GPP2 C.S0072-0 v1.0, Mobile Station Equipment Identifier (MEID) Support for  
21 cdma2000 Spread Spectrum Systems, August 2005.
- 22 [28] 3GPP2 C.S0063-B v1.0, cdma2000 High Rate Packet Data Supplemental Services,  
23 May 2010.

## 1 OVERVIEW

### 1.1 Scope of This Document

These technical requirements form a compatibility standard for cdma2000 extended cell 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.

This compatibility standard is based upon spectrum allocations that have been defined by various governmental administrations. Those wishing to deploy systems compliant with this standard should also take notice of the requirement to be compliant with the applicable rules and regulations of local administrations. Those wishing to deploy systems compliant with this standard should also take notice of the electromagnetic exposure criteria for the general public and for radio frequency carriers with low frequency amplitude modulation.

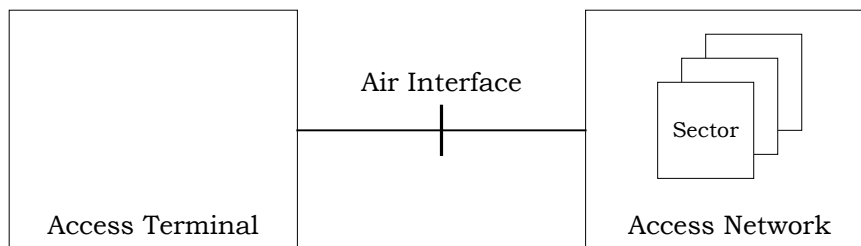
### 1.2 Requirements Language

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 1.11.

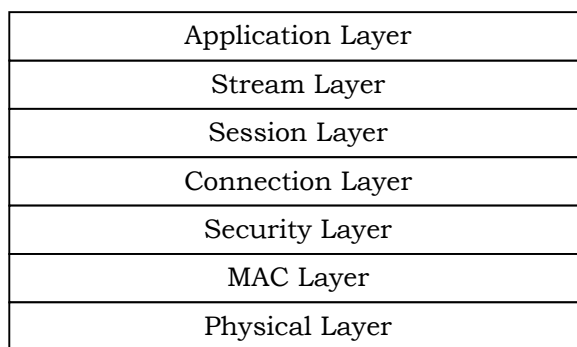
The reference model includes the air interface between the access terminal and the access network. The protocols used over the air interface are defined in this document.

#### 1.4 Protocol Architecture

The air interface has been layered, with interfaces defined for each layer (and for each protocol within each layer). This allows future modifications to a layer or to a protocol to be isolated.

##### 1.4.1 Layers

Figure 1.4.1-1 describes the layering architecture for the air interface. Each layer consists of one or more protocols that perform the layer's functionality. Each of these protocols can be individually negotiated.



**Figure 1.4.1-1. Air Interface Layering Architecture**

The protocols and layers specified in Figure 1.4.1-1 are:

**Application Layer:** The Application Layer provides multiple applications. It provides the Default Signaling Application for transporting air interface protocol messages. The Default Signaling Application is defined in [8]. It also provides the Default Packet Application for transporting user data. The Default Packet Application is defined in [8].

1 Stream Layer: The Stream Layer provides multiplexing of distinct application streams.  
 2 The Default Stream Protocol provides four streams. Stream 0 is dedicated to  
 3 signaling and defaults to the Default Signaling Application (see [8]). Stream 1,  
 4 Stream 2, and Stream 3 are not used by default. The Stream Layer is defined in [8].  
 5 The Generic Virtual Stream Protocol provides 255 virtual streams to which  
 6 applications may be bound.

7 Session Layer: The Session Layer provides address management, protocol negotiation,  
 8 protocol configuration and state maintenance services. The Session Layer is defined  
 9 in [8].

10 Connection Layer: The Connection Layer provides air link connection establishment and  
 11 maintenance services. The Connection Layer is defined in [8].

12 Security Layer: The Security Layer provides authentication and encryption services. The  
 13 Security Layer is defined in [8].

14 MAC Layer: The Medium Access Control (MAC) Layer defines the procedures used to  
 15 receive and to transmit over the Physical Layer. The MAC Layer is defined in [8].

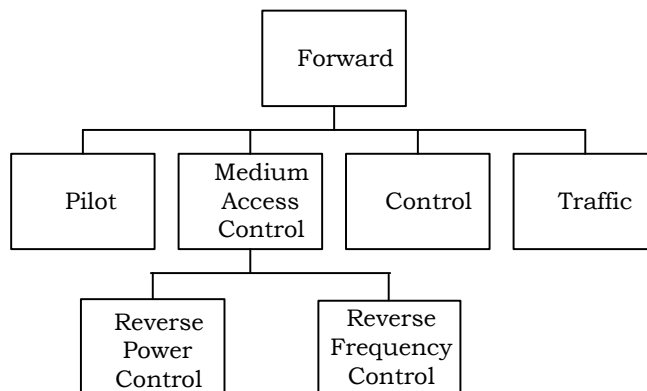
16 Physical Layer: The Physical Layer provides the channel structure, frequency, power  
 17 output, modulation, and encoding specifications for the Forward and Reverse  
 18 Channels. The Physical Layer is defined in [7].

19 Each layer may contain one or more protocols. Protocols use signaling messages or headers  
 20 to convey information to their peer protocols at the other side of the air-link. When  
 21 protocols and applications send messages, they use the Signaling Network Protocol (SNP) to  
 22 transmit these messages.

23 **1.5 Physical Layer Channels**

24 The Physical Layer defines the Physical Layer Channels and the Forward and Reverse  
 25 Channel hierarchies shown in Figure 1.5-1 and Figure 1.5-2. Channel  $x$  is part of Channel  
 26  $y$  if  $y$  is an ancestor of  $x$ . The specific channels are defined in 1.11. When the context is  
 27 clear, the complete qualified name is usually omitted (e.g., Pilot Channel as opposed to  
 28 Forward Pilot Channel or Data Channel as opposed to Reverse Traffic Data Channel).

29



30

31 **Figure 1.5-1. Forward Channel Structure**

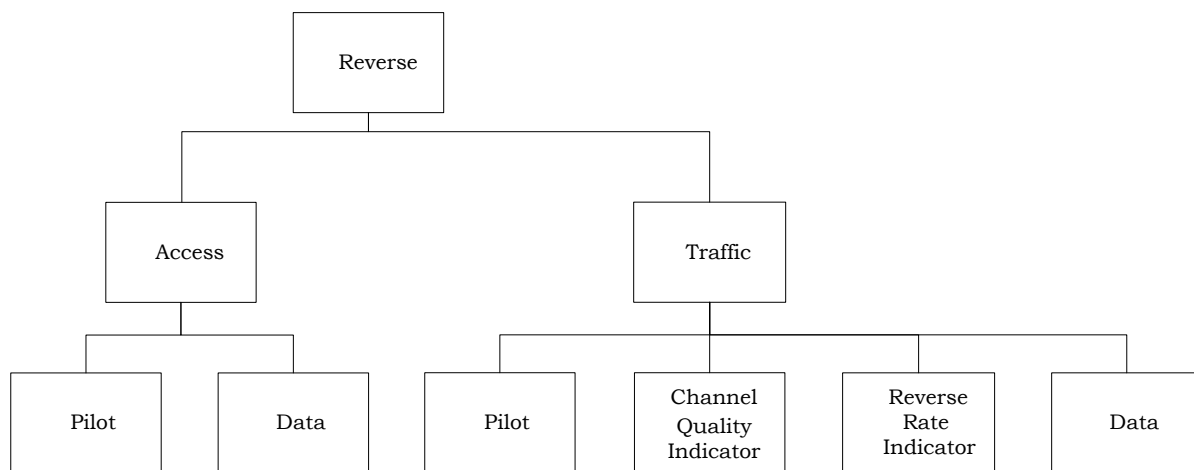


Figure 1.5-2. Reverse Channel Structure

## 1.6 Protocols

### 1.6.1 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. For example, *AccessChannelMAC.Abort* causes the Access Channel MAC Protocol to abort any access attempt currently in progress.
- 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. For example, the access terminal Reverse Traffic Channel MAC Protocol returns a “Reverse Link Acquired” indication when it gets a message from its peer protocol at the access network that it has acquired the Reverse Traffic Channel. This notification is then used by Connection Layer protocols to continue with the handshake leading to the establishment of the connection.

1 • Public Data is used to share information in a controlled way between  
2 protocols/applications. Public data is shared between protocols/applications in the  
3 same layer, as well as between protocols/applications in different layers. The public  
4 data of the InUse protocol/application is created when an InUse instance (see 1.6.3) of a  
5 protocol/application is created. An example of this is the MinimumProtocolRevision  
6 made public by the Connection Layer Initialization State Protocol after the protocol  
7 receives it in the Sync message. All configurable attributes of the InConfiguration  
8 instance of a protocol or application are also public data of that protocol or application.

9 Commands and indications are written in the form of *Protocol.Command* and  
10 *Protocol.Indication*. For example, *AccessChannelMAC.Activate* is a command activating the  
11 Access Channel MAC, and *IdleState.ConnectionOpened* is an indication provided by the  
12 Connection Layer Idle State Protocol that the connection is now open. When the context is  
13 clear, the *Protocol* part is dropped (e.g., within the Idle State Protocol, *Activate* refers to  
14 *IdleState.Activate*).

15 Commands are always written in the imperative form, since they direct an action.  
16 Indications are always written in the past tense since they notify of events that happened  
17 (e.g., *OpenConnection* for a command and *ConnectionOpened* for an indication).

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

## 22 1.6.2 States

23 When protocols exhibit different behavior as a function of the environment (e.g., if a  
24 connection is opened or not, if a session is opened or not, etc.), this behavior is captured in  
25 a set of states and the events leading to a transition between states.

26 Unless otherwise specifically mentioned, the state of the access network refers to the state  
27 of a protocol engine in the access network as it applies to a particular access terminal.  
28 Since the access network communicates with multiple access terminals, multiple  
29 independent instantiations of a protocol will exist in the access network, each with its own  
30 independent state machine.

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

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

36 When a protocol is not functional at a particular time (e.g., the Access Channel MAC  
37 protocol at the access terminal when the access terminal has an open connection) the  
38 protocol is placed in a state called the Inactive state. This state is common for most  
39 protocols.

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

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

#### 4 1.6.3 InUse and InConfiguration Protocol/Application Instances

5 A protocol/application instance can be either an InUse instance or an InConfiguration  
6 instance.

##### 7 1.6.3.1 InConfiguration Instantiation

8 An InConfiguration instance of each protocol is created by the Session Configuration  
9 Protocol once the session configuration is initiated (e.g., in the Default Session  
10 Configuration Protocol this occurs once entering the AT Initiated state).

##### 11 1.6.3.1.1 Protocol Instantiation

12 InConfiguration protocol instances can be changed by the Session Configuration Protocol.  
13 Once the access terminal and access network agree upon using a new protocol subtype for  
14 a certain protocol Type, an InConfiguration protocol instance associated with the newly  
15 negotiated protocol (specified by its protocol subtype) is created and the existing  
16 InConfiguration protocol instance for that protocol Type is replaced by the newly negotiated  
17 one.

##### 18 1.6.3.1.2 Application Instantiation

19 InConfiguration application instances are created by the Stream Layer protocol. Once the  
20 access terminal and access network agree upon using a new application subtype for a  
21 certain stream, an InConfiguration application instance associated with the newly  
22 negotiated application (specified by its application subtype) is created, and the existing  
23 InConfiguration application instance for that stream is replaced by the newly negotiated  
24 one.

##### 25 1.6.3.2 Protocol Initialization

26 The initialization procedures for an InUse protocol/application instance are invoked upon  
27 creation of the InUse protocol/application instance.

28 The initialization procedures for an InConfiguration protocol/application instance are  
29 invoked upon creation of the InConfiguration protocol/application instance.

##### 30 1.6.3.3 Procedures and Messages

31 Each protocol/application specifies procedures and messages corresponding to the InUse  
32 and InConfiguration protocol/application instances. In general, the InConfiguration  
33 protocol/application instances process messages that are related to parameter  
34 configuration for each protocol/application, while non-configuration procedures and  
35 messages are processed by the InUse protocol/application instances.

#### 1.6.3.3.1 Commit Procedures

Each InConfiguration protocol/application defines a set of Commit procedures. The Commit procedures for a protocol/application are invoked by the InUse instance of the Session Configuration Protocol.

If the Commit procedures for an InConfiguration protocol instance set the state of the protocol instance to a particular initial state and the InConfiguration protocol instance becomes the InUse protocol instance, the procedures associated with entering the initial state are to be executed at that time

If the Commit procedures for an InConfiguration protocol instance set the state of the InUse protocol instance to a particular initial state, the procedures associated with entering the initial state are executed upon entering the initial state.

#### 1.6.4 Common Commands

Most protocols support the following two commands:

- *Activate*, which commands the protocol to transition from the Inactive state to some other state.
- *Deactivate*, which commands the protocol to transition to the Inactive state. Some protocols do not transition immediately to the Inactive state, due to requirements on orderly cleanup procedures.

Other common commands are *Open* and *Close*, which command protocols to perform session open / close or connection open / close related functions.

#### 1.6.5 Protocol Negotiation

Most protocols can be negotiated and can be configured when the session is set-up (see 1.9 for a discussion of sessions). Protocols are associated with a Type that denotes the type of the protocol (e.g., Access Channel MAC Protocol) and with a Subtype that denotes a specific instance of a protocol (e.g., the Default Access Channel MAC Protocol).

The negotiation and configuration processes are part of the Session Layer.

#### 1.6.6 Protocol Overview

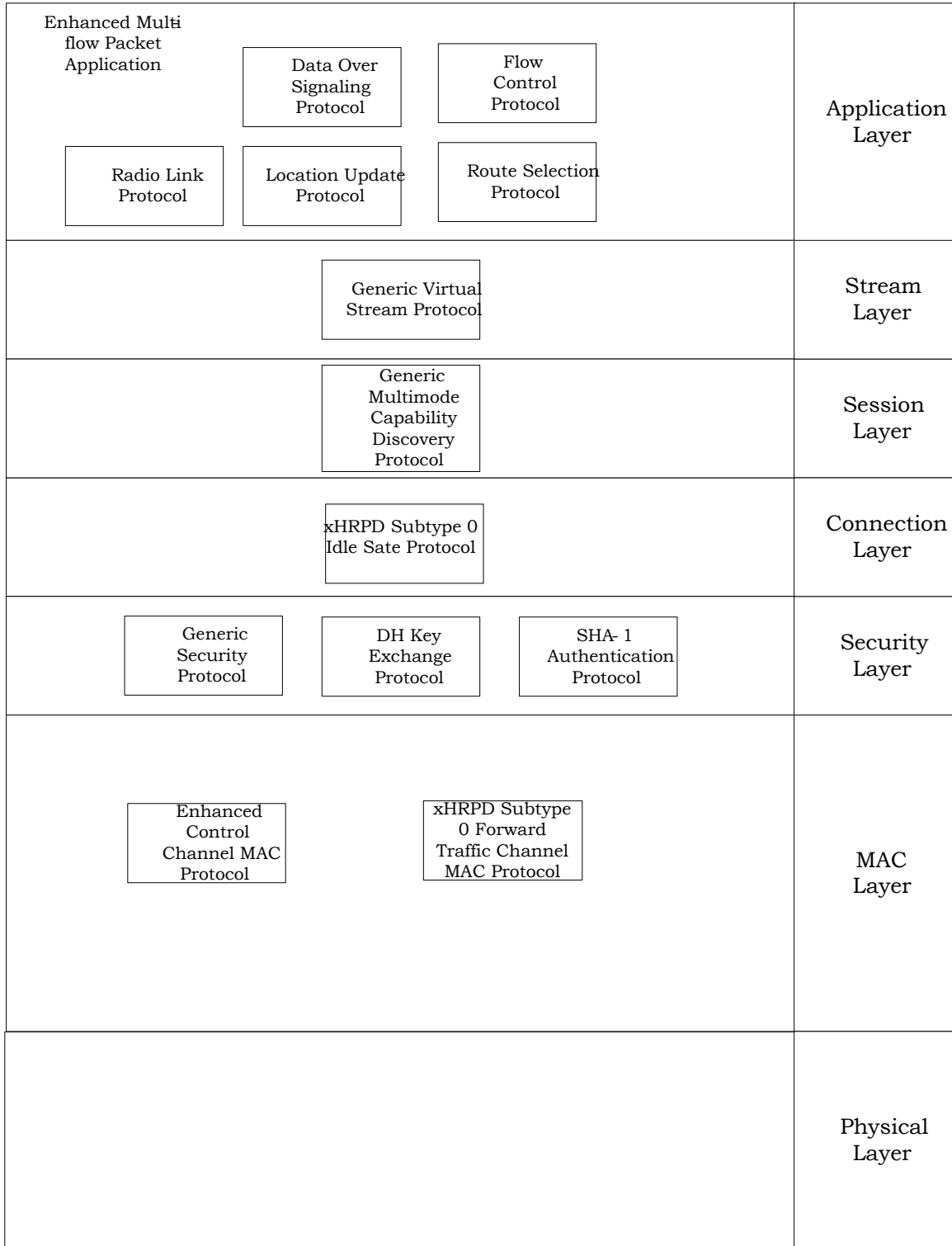
Figure 1.6.6-1 presents the default protocols defined for each one of the layers shown in Figure 1.4.1-1. The following is a brief description of each protocol. A more complete description is provided in the Introduction section of each layer.

Figure 1.6.6-2 presents the non-default protocols defined in this specification for each one of the layers shown in Figure 1.4.1-1.

Default Signaling Application Signaling Network Protocol Signaling Link Protocol		Default Packet Application Flow Control Protocol Radio Link Protocol Location Update Protocol		Application Layer
Stream Protocol				Stream Layer
Session Management Protocol	Address Management Protocol		Session Configuration Protocol	Session Layer
Air Link Management Protocol	Initialization State Protocol	xHRPD Subtype 1 Idle State Protocol	Connected State Protocol	Connection Layer
Packet Consolidation Protocol	xHRPD Subtype 0 Route Update Protocol	Overhead Messages Protocol		
Security Protocol	Key Exchange Protocol	Authentication Protocol	Encryption Protocol	Security Layer
Control Channel MAC Protocol	xHRPD Subtype 1 Forward Traffic Channel MAC Protocol	xHRPD Subtype 0 Access Channel MAC Protocol	xHRPD Subtype 0 Reverse Traffic Channel MAC Protocol	MAC Layer
xHRPD Subtype 0 Physical Layer Protocol				Physical Layer

**Figure 1.6.6-1. Default Protocols**

1  
2  
3  
4



1  
2  
3  
4

**Figure 1.6.6-2. Non-Default Protocols**

- Application Layer:
  - Default Signaling Application:

- 1           + Signaling Network Protocol: The Signaling Network Protocol (SNP) provides  
2           message transmission services for signaling messages.
- 3           + Signaling Link Protocol: The Signaling Link Protocol (SLP) provides  
4           fragmentation mechanisms, along with reliable and best-effort delivery  
5           mechanisms for signaling messages. When used in the context of the Default  
6           Signaling Application, SLP carries SNP packets.
- 7           – Default Packet Application:
- 8           + Radio Link Protocol: The Radio Link Protocol (RLP) provides retransmission and  
9           duplicate detection for an octet data stream.
- 10          + Location Update Protocol: The Location Update Protocol defines location update  
11          procedures and messages in support of mobility management for the Default  
12          Packet Application.
- 13          + Flow Control Protocol: The Flow Control Protocol defines flow control  
14          procedures to enable and disable the Default Packet Application data flow.
- 15      • Stream Layer:
  - 16          – Stream Protocol: Adds the stream header to application packets prior to  
17          transmission; and, after reception, removes the stream header and forwards  
18          application packets to the correct application.
- 19      • Session Layer:
  - 20          – Session Management Protocol: provides means to control the activation and  
21          the deactivation of the Address Management Protocol and the Session  
22          Configuration Protocol. It also provides a session keep alive mechanism.
  - 23          – Address Management Protocol: Provides access terminal identifier (ATI)  
24          management.
  - 25          – Session Configuration Protocol: Provides negotiation and configuration of the  
26          protocols used in the session.
- 27      • Connection Layer:
  - 28          – Air Link Management Protocol: Provides the overall state machine management  
29          that an access terminal and an access network follow during a connection.
  - 30          – Initialization State Protocol: Provides the procedures that an access terminal  
31          follows to acquire a network and that an access network follows to support  
32          network acquisition.
  - 33          – Idle State Protocol: Provides the procedures that an access terminal and an  
34          access network follow when a connection is not open.
  - 35          – Connected State Protocol: Provides the procedures that an access terminal and  
36          an access network follow when a connection is open.
  - 37          – Route Update Protocol: Provides the means to maintain the route between the  
38          access terminal and the access network.

- 1           – Overhead Messages Protocol: Provides broadcast messages containing  
2           information that is mostly used by Connection Layer protocols.
- 3           – Packet Consolidation Protocol: Provides transmit prioritization and packet  
4           encapsulation for the Connection Layer.
- 5   • Security Layer:
  - 6           – Key Exchange Protocol: Provides the procedures followed by the access  
7           network and the access terminal to exchange security keys for authentication  
8           and encryption.
  - 9           – Authentication Protocol: Provides the procedures followed by the access  
10           network and the access terminal for authenticating traffic.
  - 11           – Encryption Protocol: Provides the procedures followed by the access network  
12           and the access terminal for encrypting traffic.
  - 13           – Security Protocol: Provides procedures for generation of a cryptosync that can  
14           be used by the Authentication Protocol and Encryption Protocol.
- 15   • MAC Layer:
  - 16           – Control Channel MAC Protocol: Provides the procedures followed by the access  
17           network to transmit, and by the access terminal to receive the Control  
18           Channel.
  - 19           – Access Channel MAC Protocol: Provides the procedures followed by the access  
20           terminal to transmit, and by the access network to receive the Access Channel.
  - 21           – Forward Traffic Channel MAC Protocol: Provides the procedures followed by the  
22           access network to transmit, and by the access terminal to receive the Forward  
23           Traffic Channel.
  - 24           – Reverse Traffic Channel MAC Protocol: Provides the procedures followed by the  
25           access terminal to transmit, and by the access network to receive the Reverse  
26           Traffic Channel.
- 27   • Physical Layer:
  - 28           – Physical Layer Protocol: Provides channel structure, frequency, power output  
29           and modulation specifications for the forward and reverse links.

### 30 **1.7 Default Applications**

31 This document defines two default applications that all compliant access terminals and  
32 access networks support:

- 33   • Default Signaling Application, which provides the means to carry messages between a  
34    protocol in one entity and the same protocol in the other entity. The Default Signaling  
35    Application consists of a messaging protocol (Signaling Network Protocol) and a link  
36    layer protocol that provides message fragmentation, retransmission and duplicate  
37    detection (Signaling Link Protocol).

- Default Packet Application. The Default Packet Application consists of a link layer protocol that provides octet retransmission and duplicate detection (Radio Link Protocol), a location update protocol that provides mobility between data service networks and a flow control protocol that provides flow control of data traffic.

The applications used and the streams upon which they operate are negotiated as part of session negotiation.

### 1.8 Streams

The air interface can support up to four parallel application streams. The first stream (Stream 0) always carries Signaling, and the other three can be used to carry applications with different Quality of Service (QoS) requirements or other applications.

### 1.9 Sessions and Connections

A session refers to a shared state between the access terminal and the access network. This shared state stores the protocols and protocol configurations that were negotiated and are used for communications between the access terminal and the access network.

Other than to open a session, an access terminal cannot communicate with an access network without having an open session.

A connection is a particular state of the air-link in which the access terminal is assigned a Forward Traffic Channel, a Reverse Traffic Channel and associated MAC Channels.

During a single session the access terminal and the access network can open and can close a connection multiple times.

### 1.10 Security

The air interface supports a security layer, which can be used for authentication and encryption of access terminal traffic transported by the Control Channel, the Access Channel, the Forward Traffic Channel and the Reverse Traffic Channel.

### 1.11 Terms

**Access Network (AN).** The network equipment providing data connectivity between a packet switched data network (typically the Internet) and the access terminals. An access network is equivalent to a base station in [10].

**Access Terminal (AT).** A device providing data connectivity to a user. An access terminal may be connected to a computing device such as a laptop personal computer or it may be a self-contained data device such as a personal digital assistant. An access terminal is equivalent to a mobile station in [10].

**ATI.** Access Terminal Identifier.

**BATI.** Broadcast Access Terminal Identifier.

**BPSK.** Binary Phase Shift Keying.

**Cell.** A physical grouping of one or more sectors that transmit the same power control command to an access terminal.

1 **CDMA System Time in Slots.** An integer value  $s$  such that:  $s = \lfloor t \times 600 \rfloor$ , where  $t$   
2 represents CDMA System Time in seconds. Whenever the document refers to the CDMA  
3 System Time in slots, it is referring to the value  $s$ .

4 **CDMA System Time.** The time reference used by the system. CDMA System Time is  
5 synchronous to UTC time except for leap seconds and uses the same time origin as GPS  
6 time. Access terminals use the same CDMA System Time, offset by the propagation delay  
7 from the access network to the access terminal.

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

11 **Connection Layer.** The Connection Layer provides air link connection establishment and  
12 maintenance services. The Connection Layer is defined in [4].

13 **Dedicated Resource.** An access network resource required to provide any data service to  
14 the access terminal, e.g., Wireless IP Service (see [9]) that is granted to the access terminal  
15 only after access terminal authentication has completed successfully. Power control and  
16 rate control are not considered dedicated resources.

17 **Effective Isotropically Radiated Power (EIRP).** The product of the power supplied to the  
18 antenna and the antenna gain in a direction relative to an isotropic antenna.

19 **Effective Radiated Power (ERP).** The product of the power supplied to the antenna and its  
20 gain relative to a half-wave dipole in a given direction.

21 **FDD-Paired.** A forward CDMA channel and reverse CDMA channel pair where the [23]  
22 specification specifies the association between the forward CDMA channel and reverse  
23 CDMA channel.

24 **Forward CDMA Traffic Channel.** A type of Forward Traffic Channel that uses CDMA  
25 waveform.

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

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

30 **Forward MAC Channel.** The portion of the Forward Channel dedicated to Medium Access  
31 Control activities. The Forward MAC Channel consists of the RPC, and RFC Channels.

32 **Forward MAC Reverse Power Control (RPC) Channel.** The portion of the Forward MAC  
33 Channel that controls the power of the Reverse Channel for one particular access terminal.

34 **Forward MAC Reverse Frequency Control (RFC) Channel.** The portion of the Forward  
35 MAC Channel that controls the frequency of the Reverse Channel for one particular access  
36 terminal.

37 **Forward Pilot Channel.** The portion of the Forward Channel that carries the pilot.

38 **Forward Traffic Channel.** The portion of the Forward Channel that carries information for  
39 a specific access terminal. The Forward Traffic Channel can be used as either a Dedicated

1 Resource or a non-Dedicated Resource. Prior to successful access terminal authentication,  
2 the Forward Traffic Channel serves as a non-Dedicated Resource. Only after successful  
3 access terminal authentication can the Forward Traffic Channel be used as a Dedicated  
4 Resource for the specific access terminal.

5 **Frame.** The duration of time specified by 16 slots or 26.66... ms.

6 **FCS.** Frame Check Sequence.

7 **Global Positioning System (GPS).** A US government satellite system that provides location  
8 and time information to users. See Navstar GPS Space Segment/Navigation User Interfaces  
9 ICD-GPS-200 for specifications.

10 **MAC Layer.** The MAC Layer defines the procedures used to receive and to transmit over the  
11 Physical Layer. The MAC Layer is defined in [8].

12 **MATI.** Multicast Access Terminal Identifier.

13 **Multi-User Packet.** A single physical layer packet composed of zero or more security layer  
14 packets addressed to one or more access terminals.

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

16 **Physical Layer Protocol.** The Physical Layer Protocol provides the channel structure,  
17 frequency, power output, modulation, and encoding specifications for the forward and  
18 reverse links. The Physical Layer is defined in [7].

19 **QPSK.** Quadrature Phase Shift Keying.

20 **QAM.** Quadrature Amplitude Modulation.

21 **RATI.** Random Access Terminal Identifier.

22 **Reservation.** Air interface resources set up by the access network to carry a higher layer  
23 flow. A Reservation is identified by its ReservationLabel. ReservationLabels are bound to  
24 RLP Flows that carry higher layer flows. A Reservation can be either in the Open or Close  
25 state.

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

29 **Reverse Access Data Channel.** The portion of the Access Channel that carries data.

30 **Reverse Access Pilot Channel.** The portion of the Access Channel that carries the pilot.

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

33 **Reverse Traffic Channel.** The portion of the Reverse Channel that carries information from  
34 a specific access terminal to the access network. The Reverse Traffic Channel can be used  
35 as either a Dedicated Resource or a non-Dedicated Resource. Prior to successful access  
36 terminal authentication, the Reverse Traffic Channel serves as a non-Dedicated Resource.  
37 Only after successful access terminal authentication can the Reverse Traffic Channel be  
38 used as a Dedicated Resource for the specific access terminal.

- 1 **Reverse Traffic Data Channel.** The portion of the Reverse Traffic Channel that carries user  
2 data.
- 3 **Reverse Traffic MAC Channel.** The portion of the Reverse Traffic Channel dedicated to  
4 Medium Access Control activities. The Reverse Traffic MAC Channel consists of the RRI and  
5 CQI Channels.
- 6 **Reverse Traffic MAC Channel Quality Indicator (CQI) Channel.** The portion of the  
7 Reverse Traffic Channel that indicates the rate at which the access terminal can receive the  
8 Forward Traffic Channel and the sector from which the access terminal wishes to receive  
9 the Forward Traffic Channel.
- 10 **Reverse Traffic MAC Reverse Rate Indicator (RRI) Channel.** The portion of the Reverse  
11 Traffic Channel that indicates the rate of the Reverse Traffic Data Channel.
- 12 **Reverse Traffic Pilot Channel.** The portion of the Reverse Traffic Channel that carries the  
13 pilot.
- 14 **RLP.** Radio Link Protocol provides retransmission and duplicate detection for an octet-  
15 aligned data stream.
- 16 **Rx.** Receive.
- 17 **Sector.** The part of the access network that is identified by (SectorID, CDMA Channel).
- 18 **Sector-CDMA Channel.** A CDMA Channel between a sector and the access terminal.
- 19 **Security Layer.** The Security Layer provides authentication and encryption services. The  
20 Security Layer is defined in [8].
- 21 **Session Layer.** The Session Layer provides protocol negotiation, protocol configuration, and  
22 state maintenance services. The Session Layer is defined in [8].
- 23 **Single User Packet.** A single physical layer packet consisting of one or more security layer  
24 packets addressed to one access terminal.
- 25 **Slot.** A duration of time specified by 1.66... ms.
- 26 **SLP.** Signaling Link Protocol provides best-effort and reliable-delivery mechanisms for  
27 signaling messages. SLP is defined in [8].
- 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.
- 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 **Subnet Mask (of length  $n$ ).** A 128-bit value whose binary representation consists of  $n$   
35 consecutive '1's followed by 128- $n$  consecutive '0's.
- 36 **Tx.** Transmit.
- 37 **TxT2P.** Transmitted Traffic Channel to Pilot Channel transmit power ratio.
- 38 **T2P.** Traffic Channel to Pilot Channel transmit power ratio.

1 **UATI.** Unicast Access Terminal Identifier.

2 **Universal Coordinated Time (UTC).** An internationally agreed-upon time scale maintained  
3 by the Bureau International de l'Heure (BIH) used as the time reference by nearly all  
4 commonly available time and frequency distribution systems.

5 **UTC.** Universal Temps Coordine. See Universal Coordinated Time.

## 6 **1.12 Notation**

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

8  **$\langle e_1, e_2, \dots, e_n \rangle$**  A *structure* with elements ' $e_1$ ', ' $e_2$ ', ..., ' $e_n$ '.  
9 Two structures  $E = \langle e_1, e_2, \dots, e_n \rangle$  and  $F = \langle f_1, f_2, \dots, f_m \rangle$  are equal if  
10 and only if ' $m$ ' is equal to ' $n$ ' and  $e_i$  is equal to  $f_i$  for  $i=1, \dots, n$ .  
11 Given  $E = \langle e_1, e_2, \dots, e_n \rangle$  and  $F = \langle f_1, f_2, \dots, f_m \rangle$ , the assignment " $E =$   
12  $F$ " denotes the following set of assignments:  $e_i = f_i$ , for  $i=1, \dots, n$ .

13 **S.e** The member of the structure 'S' that is identified by 'e'.

14 **M[i:j]** Bits  $i^{\text{th}}$  through  $j^{\text{th}}$  inclusive ( $i \geq j$ ) of the binary representation of  
15 variable M. M[0:0] denotes the least significant bit of M.

16 **|** Concatenation operator. (A | B) denotes variable A concatenated with  
17 variable B.

18  **$\times$**  Indicates multiplication.

19  **$\lfloor x \rfloor$**  Indicates the largest integer less than or equal to x:  $\lfloor 1.1 \rfloor = 1$ ,  $\lfloor 1.0 \rfloor =$   
20  $1$ .

21  **$\lceil x \rceil$**  Indicates the smallest integer greater or equal to x:  $\lceil 1.1 \rceil = 2$ ,  $\lceil 2.0 \rceil =$   
22  $2$ .

23  **$|x|$**  Indicates the absolute value of x:  $|-17| = 17$ ,  $|17| = 17$ .

24  **$\oplus$**  Indicates exclusive OR (modulo-2 addition).

25  **$\otimes$**  Indicates bitwise logical AND operator.

26  **$\min(x, y)$**  Indicates the minimum of x and y.

27  **$\max(x, y)$**  Indicates the maximum of x and y.

28  **$x \bmod y$**  Indicates the remainder after dividing x by y:  $x \bmod y = x - (y \times \lfloor x/y \rfloor)$ .

29  **$x^y$**  Indicates the result of x raised to the power y, also denoted as  $x^y$ .

30  **$x^y$**  Indicates the result of x raised to the power y, also denoted as  $x^y$ .

1 Unless otherwise specified, the format of field values is unsigned binary.

2 Unless indicated otherwise, this standard presents numbers in decimal form. Binary  
3 numbers are distinguished in the text by the use of single quotation marks. Hexadecimal  
4 numbers are distinguished by the prefix '0x'.

5 Unless specified otherwise, each field of a packet shall be transmitted in sequence such  
6 that the most significant bit (MSB) is transmitted first and the least significant bit (LSB) is  
7 transmitted last. The MSB is the left-most bit in the figures in this document. If there are  
8 multiple rows in a table, the top-most row is transmitted first. If a table is used to show the  
9 sub-fields of a particular field or variable, the top-most row consists of the MSBs of the  
10 field. Within a row in a table, the left-most bit is transmitted first. Notations of the form  
11 "repetition factor of N" or "repeated N times" mean that a total of N versions of the item are  
12 used.

### 13 **1.13 Malfunction Detection**

14 The access terminal shall have a malfunction timer that is separate from and independent  
15 of all other functions and that runs continuously whenever power is applied to the  
16 transmitter of the access terminal. The timer shall expire if the access terminal detects a  
17 malfunction. If the timer expires, the access terminal shall be inhibited from transmitting.  
18 The maximum time allowed for expiration of the timer is two seconds.

### 19 **1.14 CDMA System Time**

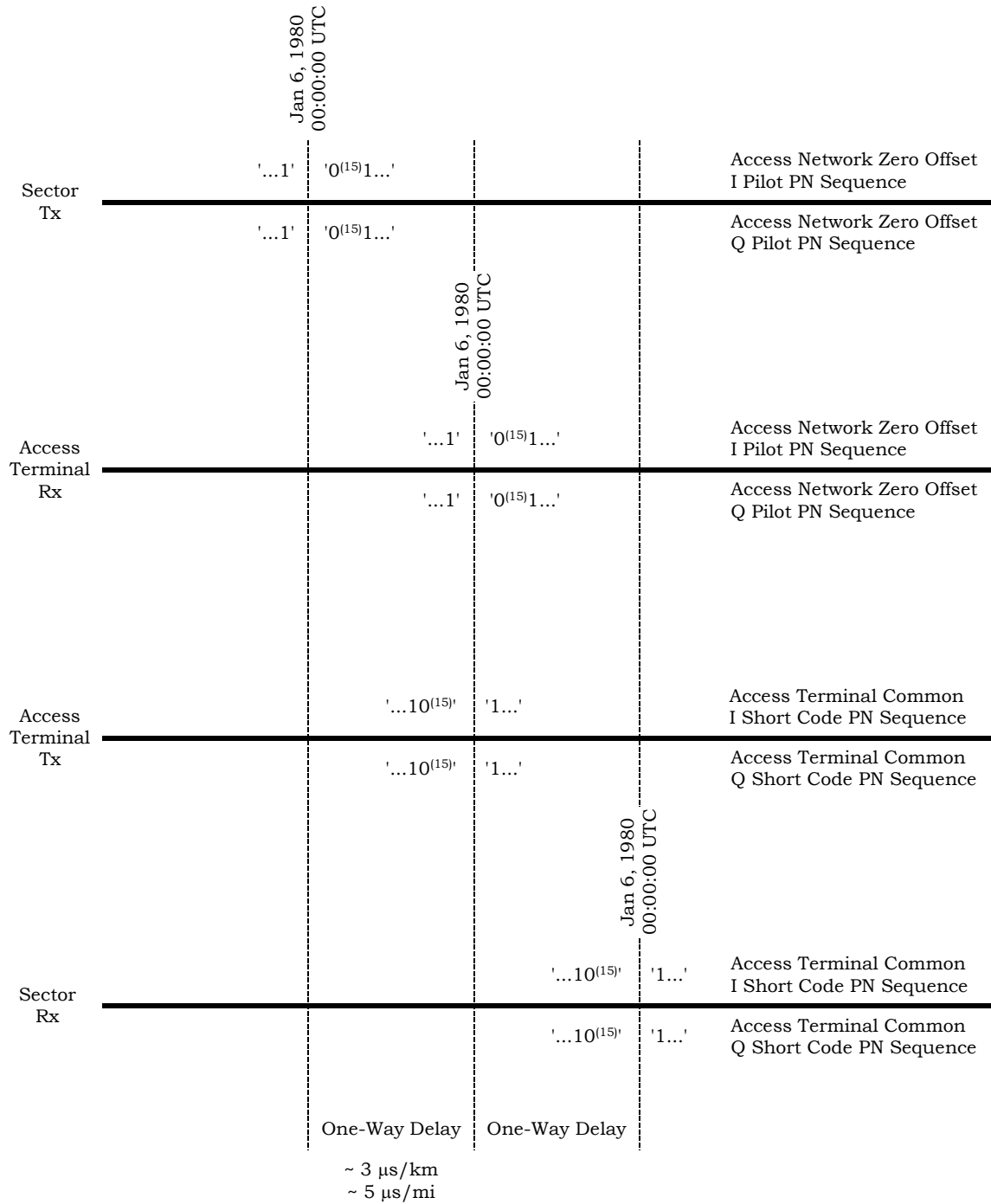
20 All sector air interface transmissions are referenced to a common system-wide timing  
21 reference that uses the Global Positioning System (GPS) time, which is traceable to and  
22 synchronous with Universal Coordinated Time (UTC). GPS and UTC differ by an integer  
23 number of seconds, specifically the number of leap second corrections added to UTC since  
24 January 6, 1980. The start of CDMA System Time is January 6, 1980 00:00:00 UTC, which  
25 coincides with the start of GPS time.

26 CDMA System Time keeps track of leap second corrections to UTC but does not use these  
27 corrections for physical adjustments to the CDMA System Time clocks.

28 Figure 1.14-1 shows the relation of CDMA System Time at various points in the system.  
29 The access network zero offset pilot PN sequences (as defined in [2]) and the access  
30 terminal common short code PN sequences (as defined in [2]) for the I and Q branches are  
31 shown in their initial states at the start of CDMA System Time. The initial state of the  
32 access network zero offset pilot PN sequences, both I and Q, is that state in which the next  
33 15 outputs of the pilot PN sequence generator are '0'. The initial state of the access terminal  
34 common short code PN sequences, both I and Q, is that state in which the output of the  
35 short code PN sequence generator is the '1' following 15 consecutive '0' outputs.

36 From Figure 1.14-1, note that the CDMA System Time at various points in the transmission  
37 and the reception processes is the absolute time referenced at the access network antenna  
38 offset by the one-way or round-trip delay of the transmission, as appropriate. Time  
39 measurements are referenced to the transmit and receive antennas of the access network  
40 and the RF connector of the Access Terminal. The precise zero instant of CDMA System

- 1 Time is the midpoint between the '1' prior to the 15 consecutive '0' outputs and the
- 2 immediate succeeding '0' of the access network zero offset pilot PN sequences.



- Notes:**
- (1) Time measurements are made at the antennas of Sectors and the RF connectors of the Access Terminals.
  - (2) 0<sup>(n)</sup> denotes a sequence of n consecutive zeroes.

1  
2

**Figure 1.14-1. CDMA System Time Line**

1 **1.15 Revision Number**

- 2 Access terminals and access networks complying with the requirements of this specification  
3 shall set their revision number to 0x01.

## 2 COMMON ALGORITHMS AND DATA STRUCTURES

### 2.1 Channel Record

The Channel record defines an access network channel frequency and the type of system on that frequency. This record contains the following fields:

Field	Length (bits)
SystemType	8
BandClass	5
ChannelNumber	11

SystemType The access network shall set this field to the following value:

**Table 2.1-1. SystemType Encoding**

Field value	Meaning
0x00, 0x02	Not available
0x01	System compliant to [10]
0x03	System compliant to this specification. ChannelNumber field specifies only the forward CDMA channel.
0x04-0xff	Reserved

BandClass The access network shall set this field to the band class number corresponding to the frequency assignment of the channel specified by this record for the forward CDMA channel only.

ChannelNumber The access network shall set this field to the channel number corresponding to the frequency assignment of the channel specified by this record for the forward CDMA channel only.

### 2.2 Access Terminal Identifier Record

The Access Terminal Identifier record provides a unicast, multicast, or broadcast access terminal address. This record contains the following fields:

Field	Length (bits)
ATIType	2
ATI	0 or 32

ATIType Access Terminal Identifier Type. This field shall be set to the type of the ATI, as shown in Table 2.2-1:

**Table 2.2-1. ATIType Field Encoding**

<b>ATIType</b>	<b>ATIType Description</b>	<b>ATI Length (bits)</b>
'00'	Broadcast ATI (BATI)	0
'01'	Multicast ATI (MATI)	32
'10'	Unicast ATI	32
'11'	Random ATI (RATI)	32

ATI Access Terminal Identifier. The field is included only if ATIType is not equal to '00'. This field shall be set as shown in Table 2.2-1.

### 2.3 Attribute Record

The attribute record defines a set of suggested values for a given attribute. The attribute record format is defined, such that if the recipient does not recognize the attribute, it can discard it and parse attribute records that follow this record.

An attribute can be one of the following three types:

- Simple attribute, if it contains a single value,
- Attribute list, if it contains multiple single values which are to be interpreted as different suggested values for the same attribute identifier (e.g., a list of possible protocol Subtypes for the same protocol Type), or
- Complex attribute, if it contains multiple values that together form a complex value for a particular attribute identifier (e.g., a set of parameters for the Route Update Protocol).

Simple attributes are a special case of an attribute list containing a single value.

The type of the attribute is determined by the attribute identifier.

The sender of a ConfigurationResponse message (see 2.7) selects an attribute-value from a ConfigurationRequest message by sending the attribute value if it is a simple attribute or a selected value out of an attribute list. Selection of complex-attributes is done by sending the value identifier which identifies the complex value.

The format of a simple attribute and attribute list is given by

<b>Field</b>	<b>Length (bits)</b>
Length	8
AttributeID	Protocol Specific
One or more instances of the following record	
AttributeValue	Attribute dependent
Reserved	variable

- 1    Length                      Length in octets of the attribute record, excluding the Length field.
- 2    AttributeID                 Attribute identifiers are unique in the context of the protocol being
- 3                                    configured.
- 4    AttributeValue              A suggested value for the attribute. Attribute value lengths are, in
- 5                                    general, an integer number of octets. Attribute values have an explicit
- 6                                    or implicit length indication (e.g., fixed length or null terminated
- 7                                    strings) so that the recipient can successfully parse the record when
- 8                                    more than one value is provided.
- 9    Reserved                      The length of this field is the smallest value that will make the
- 10                                    attribute record octet aligned. The sender shall set this field to zero.
- 11                                    The receiver shall ignore this field.

12    The format of a complex attribute is given by

<b>Field</b>	<b>Length (bits)</b>
Length	8
AttributeID	Protocol Specific
One or more instances of the following fields	
ValueID	Protocol Specific
An appropriate number of instances of the following	
record for each instance of the ValueID field	
AttributeValue	Attribute dependent
Reserved	variable

- 14    Length                      Length in octets of the attribute record, excluding the Length field.
- 15    AttributeID                 Attribute identifiers are unique in the context of the protocol being
- 16                                    configured.

1	ValueID	It identifies the set of attribute values following this field. The sender shall increment this field for each new set of values for this complex attribute.
2		
3		
4	AttributeValue	A suggested value for the attribute. Attribute value lengths are in general an integer number of octets. Attribute values have an explicit or implicit length indication (e.g., fixed length or null terminated strings) so that the recipient can successfully parse the record when more than one value is provided.
5		
6		
7		
8		
9	Reserved	The length of this field is the smallest value that will make the attribute record octet aligned. The sender shall set this field to zero. The receiver shall ignore this field.
10		
11		

## 12 2.4 Hash Function

13 The hash function takes three arguments, *Key* (typically the access terminal's ATI), *N* (the number of resources), and *Decorrelate* (an argument used to de-correlate values obtained for different applications for the same access terminal).

16 Define:

- 17 • Word *L* to be bits 0-15 of *Key*
- 18 • Word *H* to be bits 16-31 of *Key*

19 where bit 0 is the least significant bit of *Key*.

20 The hash value is computed as follows<sup>3</sup>:

$$21 \quad R = \lfloor N \times ((40503 \times (L \oplus H \oplus \text{Decorrelate})) \bmod 2^{16}) / 2^{16} \rfloor.$$

## 22 2.5 Pseudorandom Number Generator

### 23 2.5.1 General Procedures

24 When an access terminal is required to use the pseudo random number generator described in this section, then the access terminal shall implement the linear congruential generator defined by

$$27 \quad z_n = a \times z_{n-1} \bmod m$$

28 where  $a = 7^5 = 16807$  and  $m = 2^{31} - 1 = 2147483647$ .  $z_n$  is the output of the generator.<sup>4</sup>

---

<sup>3</sup> This formula is adapted from Knuth, D. N., *Sorting and Searching*, vol. 3 of *The Art of Computer Programming*, 3 vols., (Reading, MA: Addison-Wesley, 1973), pp. 508-513. The symbol  $\oplus$  represents bitwise exclusive-or function (or modulo 2 addition) and the symbol  $\lfloor \rfloor$  represents the "largest integer smaller than" function.

<sup>4</sup> This generator has full period, ranging over all integers from 1 to  $m-1$ ; the values 0 and  $m$  are never produced. Several suitable implementations can be found in Park, Stephen K. and Miller, Keith W.,

1 The access terminal shall initialize the random number generator as defined in 2.5.2.

2 The access terminal shall compute a new  $z_n$  for each subsequent use.

3 The access terminal shall use the value  $u_n = z_n / m$  for those applications that require a  
4 binary fraction  $u_n$ ,  $0 < u_n < 1$ .

5 The access terminal shall use the value  $k_n = \lfloor N \times z_n / m \rfloor$  for those applications that require  
6 a small integer  $k_n$ ,  $0 \leq k_n \leq N-1$ .

### 7 2.5.2 Initialization

8 The access terminal shall initialize the random number generator by setting  $z_0$  to

$$9 \quad z_0 = (\text{HardwareID} \oplus \chi) \bmod m$$

10 where HardwareID is the least 32 bits of the hardware identifier associated with the access  
11 terminal, and  $\chi$  is a time-varying physical measure available to the access terminal. If the  
12 initial value so produced is found to be zero, the access terminal shall repeat the procedure  
13 with a different value of  $\chi$ .

## 14 2.6 Sequence Number Validation

15 When the order in which protocol messages are delivered is important, air interface  
16 protocols use a sequence number to verify this order.

17 The sequence number has  $s$  bits. The sequence space is  $2^s$ . All operations and comparisons  
18 performed on sequence numbers shall be carried out in unsigned modulo  $2^s$  arithmetic. For  
19 any message sequence number  $N$ , the sequence numbers in the range  $[N+1, N+2^{s-1}-1]$  shall  
20 be considered greater than  $N$ , and the sequence numbers in the range  $[N-2^{s-1}, N-1]$  shall be  
21 considered smaller than  $N$ .

22 The receiver of the message maintains a receive pointer  $V(R)$  whose initialization is defined  
23 as part of the protocol. When a message arrives, the receiver compares the sequence  
24 number of the message with  $V(R)$ . If the sequence number is greater than  $V(R)$ , the message  
25 is considered a valid message and  $V(R)$  is set to this sequence number; otherwise, the  
26 message is considered an invalid message.

## 27 2.7 Generic Configuration Protocol

### 28 2.7.1 Introduction

29 The Generic Configuration Protocol provides a means to negotiate protocol parameters. The  
30 procedure consists of the initiator sending an attribute and one or more allowed values.  
31 The responder then selects one of the offered values. Each attribute must have a well  
32 known fall-back value; if the responder does not select any of the offered values, the fall-  
33 back value is selected.

---

“Random Number Generators: Good Ones are Hard to Find,” *Communications of the ACM*, vol. 31, no. 10, October 1988, pp. 1192-1201.

## 2.7.2 Procedures

### 2.7.2.1 Configuration Negotiation

The protocol uses a ConfigurationRequest message and a ConfigurationResponse message to negotiate a mutually acceptable configuration. The initiator uses the ConfigurationRequest message to provide the responder with a list of acceptable attribute values for each attribute. The responder uses the ConfigurationResponse message to provide the initiator with the accepted attribute value for each attribute, choosing the accepted attribute value from the initiator's acceptable attribute value list.

The initiator shall order the acceptable attribute values for each attribute in descending order of preference. The initiator shall send these ordered attribute-value lists to the responder using one or more ConfigurationRequest messages. If the ordered attribute value lists fit within one ConfigurationRequest message, then the initiator should use one ConfigurationRequest message. If the ordered attribute value lists do not fit within one ConfigurationRequest message, then the initiator may use more than one ConfigurationRequest message. Each ConfigurationRequest message shall contain one or more complete ordered attribute value lists; an ordered attribute value list for an attribute shall not be split within a ConfigurationRequest message and shall not be split across multiple ConfigurationRequest messages.

After sending a ConfigurationRequest message, the sender shall set the value of all parameters that were listed in the message to NULL.

After receiving a ConfigurationRequest message, the responder shall respond within  $T_{\text{Turnaround}}$ , where  $T_{\text{Turnaround}} = 2$  seconds, unless specified otherwise. For each attribute included in the ConfigurationRequest message, the responder shall choose an acceptable attribute value from the associated acceptable attribute value list. If the responder does not recognize an attribute or does not find an acceptable attribute value in the associated attribute list, then the responder shall skip the attribute. The responder shall send the accepted attribute value for each attribute within one ConfigurationResponse message. The value included for each attribute shall be one of the values listed in the ConfigurationRequest message. After receiving a ConfigurationResponse message, the initiator shall pair the received message with the associated ConfigurationRequest message. If the ConfigurationResponse message does not contain an attribute found in the associated ConfigurationRequest message, then the initiator shall assume that the missing attribute is using the fall-back value.

If the initiator requires no further negotiation of protocols or configuration of negotiated protocols and if the value of the any of the parameters for which the initiator has sent a ConfigurationRequest message is NULL, then the sender shall declare a failure.

The initiator and the responder shall use the attribute values in the ConfigurationResponse messages as the configured attribute values, provided that the attribute values were also present in the associated ConfigurationRequest message.

### 2.7.3 Message Formats

The receiver shall discard all unrecognized messages. The receiver shall discard all unrecognized fields following the fields defined herein. The receiver may log the message for diagnostic reasons.

The specification of the Physical Layer channels on which the following messages are to be carried; and, whether the messages are to be sent reliably or as best-effort, is provided in the context of the protocols in which these messages are used.

#### 2.7.3.1 ConfigurationRequest

The sender sends the ConfigurationRequest message to offer a set of attribute-values for a given attribute.

Field	Length (bits)
MessageID	Protocol dependent
TransactionID	8

Zero or more instances of the following record

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

**MessageID** The value of this field is specified in the context of the protocol using this message. The value 0x50 is recommended.

**TransactionID** The sender shall increment this value for each new ConfigurationRequest message sent.

**AttributeRecord** The format of this record is specified in 2.3.

#### 2.7.3.2 ConfigurationResponse

The sender sends a ConfigurationResponse message to select an attribute-value from a list of offered values.

Field	Length (bits)
MessageID	Protocol dependent
TransactionID	8

Zero or more instances of the following record

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

**MessageID** The value of this field is specified in the context of the protocol using this message. The value 0x51 is recommended.

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

1 AttributeRecord An attribute record containing a single attribute value. If this  
 2 message selects a complex attribute, only the ValueID field of the  
 3 complex attribute shall be included in the message. The format of the  
 4 AttributeRecord is given in 2.3. The sender shall not include more  
 5 than one attribute record with the same attribute identifier.

## 6 2.8 Session State Information Record

7 The Session State Information is to be used in [16][17] for transferring the session  
 8 parameters corresponding to the InUse protocol instances from a source access network to  
 9 a target access network. Session parameters are the attributes and the internal parameters  
 10 that define the state of each protocol. The format of this record is shown in Table 2.8-1. If  
 11 an attribute is not contained in the Session State Information record, the target access  
 12 network shall assume that the missing attributes have the default values (specified for each  
 13 attribute in each protocol). The sender shall include all the Parameter Records associated  
 14 with the ProtocolType and ProtocolSubtype in the same Session State Information Record.

15 **Table 2.8-1. The Format of the Session State Information Record**

Field	Length (bits)
FormatID	8
Reserved	1
ProtocolType	7 or 15
ProtocolSubtype	16

One or more instances of the following Parameter Record:

ParameterType	8
ParameterType-specific record	Variable

16 FormatID This field identifies the format of the rest of the fields in this record  
 17 and shall be set to zero.

18 Reserved This field shall be set to zero.

19 ProtocolType This field has the following format:  
 20

Sub-Field	Length (bits)
Type1	7
Type2	0 or 8

21 Type1 This sub-field shall be set to the seven most significant bits of the  
 22 Type value for the protocol (as defined in [18]) associated with the  
 23 encapsulated parameter.

- 1 Type2 If the length of the Type value for the protocol associated with the  
 2 encapsulated parameter is 7 bits, then this sub-field shall be omitted.  
 3 Otherwise, this field shall be set to the 8 least significant bits of the  
 4 Type value for the protocol associated with the encapsulated  
 5 parameter.<sup>5</sup>
- 6 ProtocolSubtype This field shall be set to the protocol subtype value (see Table 3.1-1)  
 7 for the protocol associated with the encapsulated session parameters.
- 8 ParameterType This field shall be set according to Table 2.8-2.

9 **Table 2.8-2. Encoding of the ParameterType Field**

Field Value	Meaning
0x00	The ParameterType-specific record consists of a Complex or a Simple Attribute as defined in 2.3. The ValueID field of the complex attribute shall be set to zero.
All other values	ParameterType-specific record are protocol dependent

- 10 ParameterType-specific record
- 11 If the ParameterType field is set to 0x00, then this record shall be set  
 12 to the simple or complex attribute (see 2.3) associated with the  
 13 protocol identified by the (ProtocolType, ProtocolSubtype) pair.  
 14 Otherwise, the structure of this record shall be as specified by the  
 15 protocol which is identified by the (ProtocolType, ProtocolSubtype)  
 16 pair.  
 17

---

<sup>5</sup> For example, if Type1 is '0011010', then Type2 shall be 8 bits long.

## 2.9 SectorID Provisioning

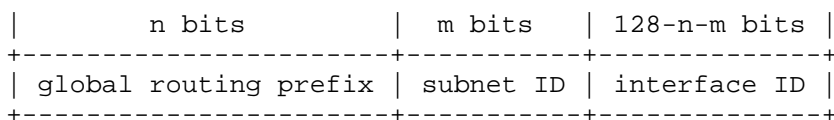
The SectorID is an IPv6 address from one of the following four address pools: Global Unicast, Site-Local Unicast, Link-Local Unicast and Reserved

This section describes the rules for assigning SectorID values to sectors in order to ensure that the value of the SectorID is unique across operator networks, when the SectorID is a Global Unicast address, Site-Local Unicast address, a Link-Local Unicast address or a Reserved address. If the SectorID is Global Unicast address, then the value of the SectorID is globally unique.

### 2.9.1 Overview of Relevant Formats

#### 2.9.1.1 Global Unicast IPv6 Address Format

Global Unicast addresses have the following format:

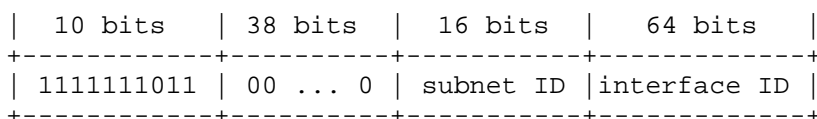


**Figure 2.9.1.1-1. Global Unicast IPv6 Address Format**

For all Global Unicast addresses, except those that start with binary 000, the Interface ID is required to be 64 bits long and to be constructed in Modified EUI-64 format.

#### 2.9.1.2 Site-Local Unicast IPv6 Address Format

Addresses that start with binary 1111111011 are Site-Local Unicast addresses. However, only Site-Local Unicast addresses of the following format have been defined.

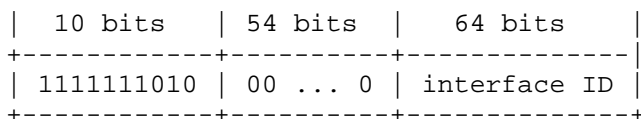


**Figure 2.9.1.2-1. Site-Local Unicast IPv6 Address Format**

The Interface ID is required to be 64 bits long and to be constructed in Modified EUI-64 format.

#### 2.9.1.3 Link-Local Unicast IPv6 Address Format

Addresses that start with binary 1111111010 are Link-Local Unicast addresses. However, only Link-Local addresses of the following format have been defined.



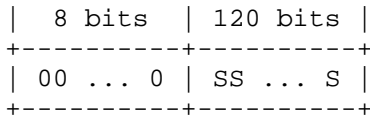
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41

**Figure 2.9.1.3-1. Link-Local Unicast IPv6 Address Format**

The Interface ID is required to be 64 bits long and to be constructed in Modified EUI-64 format.

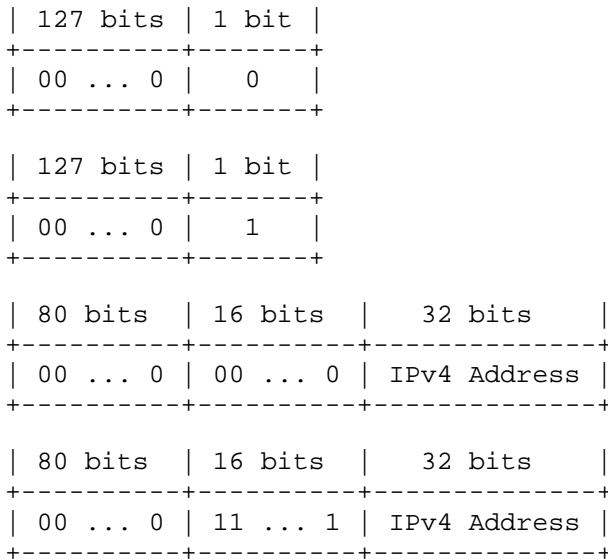
2.9.1.4 Reserved IPv6 Address Format

Reserved addresses have the following format



**Figure 2.9.1.4-1. Format of the Reserved IPv6 Addresses**

However, the Unspecified address, the Loopback address, and the Embedded IPv4 addresses have been chosen from the Reserved Address pool. Therefore, the following values shall be excluded from the Reserved IPv6 address category for SectorID values.

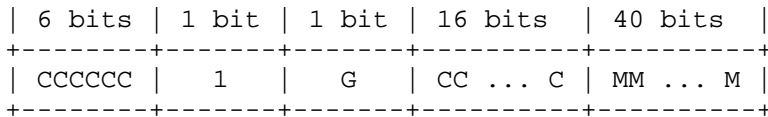


**Figure 2.9.1.4-2. IPv6 Values That Are to be Avoided**

2.9.1.5 Modified EUI-64 Format

The Modified EUI-64 Format may take on one of two formats: the universally unique format and the locally unique (non-universally unique) format.

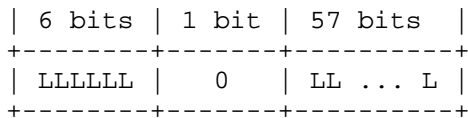
If the Modified EUI-64 value is universally unique, then it has the following format:



**Figure 2.9.1.5-1. Universally Unique Modified EUI-64**

The “C” bits are a company identifier assigned to the manufacturer. The “M” bits are the bits chosen by the manufacturer to ensure that the values assigned by the manufacturer are unique. The “G” bit is the group/individual bit.

If the Modified EUI-64 value is locally unique, then it has the format:

**Figure 2.9.1.5-2. Locally Unique Modified EUI-64**

where the L bits are local node identifier that is chosen such that it is unique on the link.

**2.9.2 SectorID Construction**

The access network shall construct the SectorID to be either a Globally Unique SectorID or a Locally Unique SectorID as described below.

If a Globally Unique SectorID is used, the SectorID is universally unique by construction.

If a Locally Unique SectorID is used, it is the responsibility of the network to ensure the uniqueness of the SectorID throughout the networks that the access terminal can visit.

**2.9.2.1 Construction of Globally Unique SectorID**

There are multiple methods by which a network can be uniquely identified. Networks connected to IPv6 networks are uniquely identified using an IPv6 subnet prefix. Networks connected to the ANSI-41 core are uniquely identified using a System Identifier (SID). Networks connected to the GSM/UMTS core are uniquely identified using a Mobile Country Code (MCC) and a Mobile Network Code (MNC). Networks connected to IPv4 networks are uniquely identified using an IPv4 subnet prefix.

It is likely that different operators will have different preferences when it comes to which type of unique identifier to use. Therefore, the following proposal allows the operator to use an IPv6 unique identifier, an ANSI-41 unique identifier, a GSM/UMTS unique identifier, or an IPv4 unique identifier, while ensuring that the SectorID is unique across operator networks.

**2.9.2.1.1 SectorID Based On an IPv6 Unique Identifier**

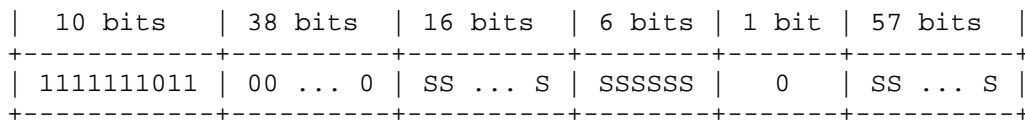
When the SectorID is based on an IPv6 unique identifier, the SectorID shall be any Global Unicast IPv6 Address that has been assigned to the operator and that does not start with binary 00000000. The Global Unicast IPv6 addresses that start with binary 00000000 are excluded because the conflict with the Reserved addresses.

An Operator that has not been assigned any IPv6 addresses but has been assigned at least one globally unique IPv4 address may construct a Global Unicast IPv6 address using the 6to4 method described in [21].

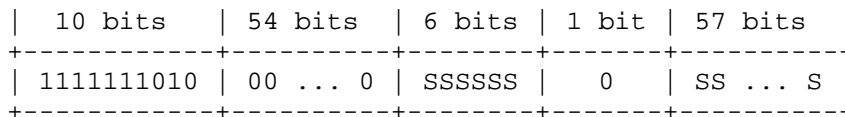
### 2.9.2.1.2 SectorID Not Based On an IPv6 Unique Identifier

When the SectorID is not based on an IPv6 unique identifier, the SectorID shall be a Site-Local Unicast IPv6 Address, a Link-Local Unicast IPv6 Address or a Reserved IPv6 Address. When the SectorID is a Site-Local Unicast IPv6 Address or a Link-Local Unicast IPv6 Address, the interface ID shall be a locally unique Modified EUI-64 value.

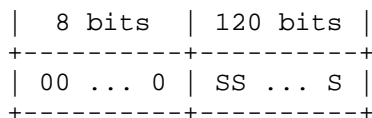
As is shown below for Site-Local Unicast, Link-Local Unicast and Reserved, there are certain bits of the addresses that must take on fixed values in order to meet the IPv6 address requirements. The remaining bits (denoted by “S”) are used to create unique SectorID values. Therefore, the number of bits available for creating the unique SectorID is 79, 63 and 120 bits for Site-Local Unicast, Link-Local Unicast and Reserved, respectively.



**Figure 2.9.2.1.2-1. “S” bits in the Site-Local Unicast IPv6 Address Format**

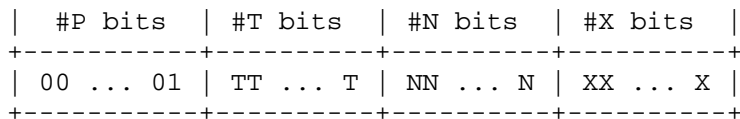


**Figure 2.9.2.1.2-2. “S” bits in the Link-Local Unicast IPv6 Address Format**



**Figure 2.9.2.1.2-3. “S” bits in the Reserved IPv6 Address Format**

The “S” bits are further broken down into the following sub-fields



**Figure 2.9.2.1.2-4. Sub-fields of the “S” bits**

where the “T” bits identify the type of unique identifier (IPv4, GSM/UMTS or ANSI-41), the “N” bits are the operator’s unique identifier, the “X” bits are operator selected bits (i.e., bits selected by the operator).

The “P” bits, which are a run of zero or more 0’s followed by one 1, allow for flexible positioning of the unique identifier within the IPv6 address. The number of “P” bits shall be less than or equal to 64. This is to ensure that the addresses in the Reserved IPv6 address

format category for SectorID do not collide with the Locally Unique SectorIDs (because the number of leading zeros in the SectorID in the Reserved IPv6 address format category is less than 72).

The “T” bits shall be chosen such that the values are prefix free.

The following sections specify how the “T” bits, the “N” bits, and the “X” bits are assigned for each of the unique identifier types defined in this document (that is, ANSI-41, GSM/UMTS, and IPv4).

#### 2.9.2.1.2.1 ANSI-41 Method

Ignoring bits in the SectorID that shall take on fixed values in order to meet IPv6 requirements, the SectorID format is as follows:

#P bits	2 bit	15 bits	#X bits
00 ... 01	00	SID	XX ... X
PP ... P	TT ... T	NN ... N	XX ... X

**Figure 2.9.2.1.2.1-1. Assignment of the “T” Bits, the “N” Bits, and the “X” Bits for the ANSI-41 Method**

The “T” bits shall be set to the binary value ‘00’. The “N” bits shall be set to “SID”, which is the ANSI-41 System Identifier that has been assigned to the operator. The “X” bits shall be set by the operator and shall be chosen to ensure that the SectorID values and corresponding UATI values are unique within the operator’s network. Therefore, there are up to 61, 45, and 102 operator settable bits for Site-Local Unicast, Link-Local Unicast and Reserved addresses, respectively.

#### 2.9.2.1.2.2 GSM/UMTS Method

Ignoring bits in the SectorID that must take on fixed values in order to meet IPv6 requirements, the SectorID format is as follows:

#P bits	2 bits	12 bits	12 bits	#X bits
00 ... 01	01	MCC	MNC	XX ... X
PP ... P	TT ... T	NN ... N		XX ... X

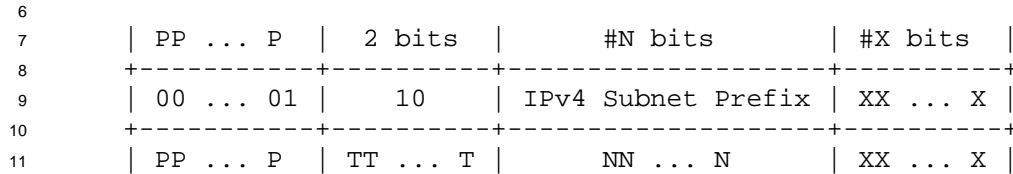
**Figure 2.9.2.1.2.2-1. Assignment of the “T” Bits, the “N” Bits, and the “X” Bits for the GSM/UMTS Method**

The “T” bits shall be set to the binary value ‘01’. The “N” bits shall be set to “MCC” and “MNC”, which are the binary coded decimal versions of a Mobile Country Code and Mobile Network Code pair that have been assigned to the operator. The “X” bits shall be set by the operator and shall be chosen to ensure that the SectorID values and corresponding UATI values are unique within the operator’s network. Therefore, there are up to 52, 36, and 93

1 operator settable bits for Site-Local Unicast, Link-Local Unicast and Reserved addresses,  
2 respectively.

### 3 2.9.2.1.2.3 IPv4 Unique Identifier

4 Ignoring bits in the SectorID that must take on fixed values in order to meet IPv6  
5 requirements, the SectorID format is as follows:

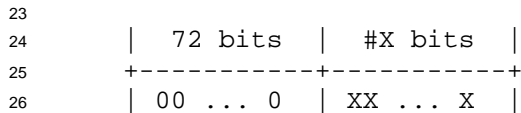


12 **Figure 2.9.2.1.2.3-1. Assignment of the “T” Bits, the “N” Bits, and the “X” Bits for**  
13 **the IPv4 Method**

14 The “T” bits shall be set to the binary value ‘10’. The “N” bits are set to “IPv4 Subnet Prefix”,  
15 which is a prefix of a globally unique IPv4 subnet assigned to the operator. The “X” bits  
16 shall be set by the operator and shall be chosen to ensure that the SectorID values and  
17 corresponding UATI values are unique within the operator’s network. Therefore, there are  
18 52, 36, and 93 operator settable bits for Site-Local Unicast, Link-Local Unicast and  
19 Reserved addresses, respectively, assuming that “IPv4 Subnet Prefix” is a 24-bit prefix  
20 identifying an IPv4 class C subnet.

### 21 2.9.2.2 Construction of Locally Unique SectorID

22 The format of the Locally Unique SectorID is as follows:



27 **Figure 2.9.2.2-1. Format of the Locally Unique SectorID**

28 The “X” bits shall be set by the network to ensure the uniqueness of the SectorID  
29 throughout the networks that the access terminal can visit.

## 30 2.10 Generic Attribute Update Protocol

### 31 2.10.1 Introduction

32 The Generic Attribute Update Protocol provides a means to update protocol attributes. The  
33 protocol uses an AttributeUpdateRequest message, an AttributeUpdateAccept message, and  
34 an AttributeUpdateReject message to negotiate a mutually acceptable configuration.

35 The initiator uses the AttributeUpdateRequest message to provide the responder with a  
36 proposed value for each attribute. The responder uses the AttributeUpdateAccept message  
37 to accept the proposed values. If the responder is an access network, and if any of the  
38 attribute values in the received AttributeUpdateRequest message is not acceptable to it,

1 then the access network sends the AttributeUpdateReject message, and the access terminal  
2 and access network continue to use the previously negotiated values for the attributes.

3 The access terminal is not allowed to send an AttributeUpdateReject message.

## 4 2.10.2 Procedures

### 5 2.10.2.1 Initiator Requirements

6 The access terminal and the access network shall not send an AttributeUpdateRequest  
7 message if the ConfigurationLock public data of the Session Configuration Protocol is set to  
8 Locked.

9 Unless indicated otherwise, the access terminal shall not send an AttributeUpdateRequest  
10 message on the Access Channel. Unless indicated otherwise, the access network shall not  
11 send an AttributeUpdateRequest message on the Control Channel.

12 The initiator shall include one attribute value for each attribute included in the  
13 AttributeUpdateRequest message.

14 After sending an AttributeUpdateRequest message, the initiator should continue to use  
15 previously negotiated values for attributes listed in the message until it receives either an  
16 AttributeUpdateAccept message or an AttributeUpdateReject message. However, the  
17 initiator should be prepared for the responder to begin using attribute values proposed by  
18 the initiator in the AttributeUpdateRequest message.

19 If the initiator receives an AttributeUpdateAccept message, then it shall pair the received  
20 message with the associated AttributeUpdateRequest message using the TransactionID  
21 field of the messages. The initiator shall use the attribute values in the  
22 AttributeUpdateRequest message as the configured attribute values. If the access terminal  
23 receives an AttributeUpdateReject message, then it shall use the previously configured  
24 values of the attributes included in the corresponding AttributeUpdateRequest message.

25 If the initiator does not receive the corresponding AttributeUpdateAccept or  
26 AttributeUpdateReject message in response to the AttributeUpdateRequest message, it  
27 should re-transmit the AttributeUpdateRequest message.

28 While the initiator is waiting for a response to an AttributeUpdateRequest message, it shall  
29 not transmit another AttributeUpdateRequest message with a different TransactionID field  
30 that requests reconfiguration of an attribute included in the original  
31 AttributeUpdateRequest message.

### 32 2.10.2.2 Responder Requirements

33 After receiving an AttributeUpdateRequest message, the responder shall respond within  
34  $T_{\text{Turnaround}}$ , unless specified otherwise by the protocol which uses the Generic Attribute  
35 Update Protocol.

36 If the responder is an access terminal, then

- 37 • The responder shall send an AttributeUpdateAccept message.

- 1 • Upon sending an AttributeUpdateAccept message, the responder shall begin using the  
2 accepted attribute values.

3 If the responder is an access network, then

- 4 • If the responder finds the proposed value for each attribute in the  
5 AttributeUpdateRequest message to be acceptable, then the responder shall send an  
6 AttributeUpdateAccept message. Upon sending an AttributeUpdateAccept message, the  
7 responder shall begin using the accepted attribute values.
- 8 • If the responder does not recognize an attribute or does not find a proposed attribute  
9 value to be acceptable, then it shall send an AttributeUpdateReject message.
- 10 • If the responder sends an AttributeUpdateReject message, then it shall continue to use  
11 the previously configured values of the attributes found in the corresponding  
12 AttributeUpdateRequest message.

### 13 2.10.3 Message Formats

14 The specification of the Physical Layer channels on which the following messages are to be  
15 carried; and, whether the messages are to be sent reliably or as best-effort, is provided in  
16 the context of the protocols in which these messages are used.

#### 17 2.10.3.1 AttributeUpdateRequest

18 The sender sends an AttributeUpdateRequest message to offer an attribute-value for a given  
19 attribute.

20

Field	Length (bits)
MessageID	Protocol dependent
TransactionID	8

One or more instances of the following record

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

21 MessageID            The value of this field is specified in the context of the protocol using  
22 this message. The value 0x52 is recommended.

23 TransactionID        The sender shall increment this value for each new  
24 AttributeUpdateRequest message sent.

25 AttributeRecord      The format of this record is specified in 2.3.

#### 26 2.10.3.2 AttributeUpdateAccept

27 The sender sends an AttributeUpdateAccept message in response to an  
28 AttributeUpdateRequest message to accept the offered attribute values.

29

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

1 MessageID The value of this field is specified in the context of the protocol using  
2 this message. The value 0x53 is recommended.

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

### 5 2.10.3.3 AttributeUpdateReject

6 The access network sends an AttributeUpdateReject message in response to an  
7 AttributeUpdateRequest message to reject the offered attribute values.  
8

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

9 MessageID The value of this field is specified in the context of the protocol using  
10 this message. The value 0x54 is recommended.

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

### 13 2.10.4 Protocol Numeric Constants

14

<b>Constant</b>	<b>Meaning</b>	<b>Value</b>
$T_{\text{Turnaround}}$	Maximum time to respond to an AttributeUpdateRequest message.	2 sec

15

1 **2.11 Linear Interpolation**

2 The access terminal shall use the following procedure for linear interpolation:

- 3 1. Let  $f(x)$  be the one-dimensional function which is explicitly defined on some finite  
4 set of x-axis points  $\Sigma_x$ .
- 5 2. Let  $y' = f(x')$  be the interpolated value of the function at the input  $x'$ .
- 6 3. If  $f(x)$  is explicitly defined at only one point on the x-axis, then set  $y'$  equal to the  
7 value of the function at that point.
- 8 4. If  $f(x)$  is explicitly defined at two or more points on the x-axis, continue as follows:
- 9 – If  $x'$  is outside the range of  $\Sigma_x$ , then set  $x'$  equal to the nearest value of  $\Sigma_x$ .
- 10 – Let  $x_1, x_2$  be the points in  $\Sigma_x$  that are closest to  $x'$ , which satisfy the relation  $x_1$   
11  $\leq x' \leq x_2$ . Define  $y_1, y_2$  as follows:

<b>x</b>	<b>y = f(x)</b>
$x_1$	$y_1$
$x_2$	$y_2$

12 Then the value of  $y'$  is given by the equation:

13 
$$y' = y_1 + (y_2 - y_1) \times (x' - x_1) / (x_2 - x_1)$$

14 The access terminal shall compute  $y'$  with an error of no more than  $\pm 2\%$  of its true value.

15

## 2.12 Bi-linear Interpolation

The access terminal shall use the following procedure for bi-linear interpolation:

1. Let  $f(x,y)$  be the two-dimensional function which is explicitly defined on some finite set of x-axis and y-axis points, denoted  $\Sigma_x$  and  $\Sigma_y$  respectively.
2. Let  $z' = f(x',y')$  be the interpolated value of the function at inputs  $x'$  and  $y'$ .
3. If  $f(x,y)$  is explicitly defined at only one point on both the x-axis and y-axis, then set  $z'$  equal to the value of the function at that point.
4. If  $f(x,y)$  is explicitly defined at only one point on either the x-axis or the y-axis, then use the procedure of 2.11 on the other axis, and set  $z'$  to the result.
5. If  $f(x,y)$  is explicitly defined at two or more points for both the x-axis and the y-axis, continue as follows:
  - If  $x'$  is outside the range of  $\Sigma_x$ , then set  $x'$  equal to the nearest value of  $\Sigma_x$ .
  - If  $y'$  is outside the range of  $\Sigma_y$ , then set  $y'$  equal to the nearest value of  $\Sigma_y$ .
  - Let  $x_1, x_2$  be the points in  $\Sigma_x$  that are closest to  $x'$ , which satisfy the relation  $x_1 \leq x' \leq x_2$ . Let  $y_1, y_2$  be the points in  $\Sigma_y$  that are closest to  $y'$ , which satisfy the relation  $y_1 \leq y' \leq y_2$ . Define  $z_1, z_2, z_3,$  and  $z_4$  as follows:

$(x,y)$	$z = f(x,y)$
$(x_1,y_1)$	$z_1$
$(x_2,y_1)$	$z_2$
$(x_1,y_2)$	$z_3$
$(x_2,y_2)$	$z_4$

Then the value of  $z'$  is given by the equation:

$$z' = a \times ( b \times z_4 + (1-b) \times z_2 ) + (1-a) \times ( b \times z_3 + (1-b) \times z_1 )$$

where

$$a = (x' - x_1)/(x_2 - x_1)$$

$$b = (y' - y_1)/(y_2 - y_1)$$

The access terminal shall compute  $z'$  with an error of no more than  $\pm 2\%$  of its true value.

### 2.13 IIR filter implementation

The access terminal shall perform IIR filter implementation using the following equation:

$$y(n) = (1 - 1/\tau) \times y(n - 1) + (1/\tau) \times x(n)$$

where  $n$  denotes the time index in slots or sub-frames,  $\tau$  denotes the filter time constant,  $y$  denotes the IIR filter output and  $x$  denotes the IIR filter input. The filter shall be updated every slot or every sub-frame. The filter update rate is a function of the quantity filtered.

The access terminal shall compute  $y(n)$  with an error of no more than  $\pm 2\%$  of its true value.

### 2.14 ReverseChannel Record

The ReverseChannel record defines reverse link channel frequency. This record contains the following fields:

Field	Length (bits)
RevChannelNumber	11
NumOfNarrowBandChannels	4
NarrowBandRevChannelNumber	8
Reserved	1

RevChannelNumber

The access network shall set this field to the channel number corresponding to the center frequency of the 1.25 MHz channel specified by this record.

NumOfNarrowBandChannels

The access network shall set this field to the number of narrowband channels assigned by this record.

NarrowBandRevChannelNumber

The access network shall set this field to the narrowband channel number corresponding to the frequency of the first narrowband channel(s) specified by this record.

Reserved

This bit shall be set to zero.

1 This page intentionally left blank.

1   **3 ASSIGNED NAMES AND NUMBERS**

2   **3.1 Protocols**

3   Table 3.1-1 shows the Protocol Type and Protocol Subtypes assigned to the protocols  
4   defined in this specification. An updated list of Protocol Types and Protocol Subtypes is  
5   specified in [18].

1

**Table 3.1-1. Protocol Type and Subtypes**

Protocol Type			Protocol Subtype		Reference
Name	ID	Length (bits)	Name	ID	
Physical Layer	0x00	7	xHRPD Subtype 0 Physical Layer	0x0000	[7]
Control Channel MAC	0x01	7	Default Control Channel MAC	0x0000	[8]
Control Channel MAC	0x01	7	Enhanced Control Channel MAC	0x0001	[8]
Access Channel MAC	0x02	7	xHRPD Subtype 0 Access Channel MAC	0x0000	[8]
Forward Traffic Channel MAC	0x03	7	xHRPD Subtype 0 Forward Traffic Channel MAC	0x0000	[8]
Forward Traffic Channel MAC	0x03	7	xHRPD Subtype 1 Forward Traffic Channel MAC	0x0001	[8]
Reverse Traffic Channel MAC	0x04	7	xHRPD Subtype 0 Reverse Traffic Channel MAC	0x0000	[8]
Key Exchange	0x05	7	Default Key Exchange	0x0000	[8]
Key Exchange	0x05	7	DH Key Exchange	0x0001	[8]
Authentication	0x06	7	Default Authentication	0x0000	[8]
Authentication	0x06	7	SHA-1 Authentication	0x0001	[8]
Encryption	0x07	7	Default Encryption	0x0000	[8]
Security	0x08	7	Default Security	0x0000	[8]
Security	0x08	7	Generic Security	0x0001	[8]
Packet Consolidation	0x09	7	Default Packet Consolidation	0x0000	[8]
Air-Link Management	0x0a	7	Default Air-Link Management	0x0000	[8]
Initialization State	0x0b	7	Default Initialization State	0x0000	[8]
Idle State	0x0c	7	xHRPD Subtype 0 Idle State	0x0000	[8]
Idle State	0x0c	7	xHRPD Subtype 1 Idle State	0x0001	[8]
Connected State	0x0d	7	Default Connected State	0x0000	[8]
Route Update	0x0e	7	xHRPD Subtype 0 Route Update	0x0000	[8]
Overhead Messages	0x0f	7	Overhead Messages	0x0000	[8]
Session Management	0x10	7	Default Session Management	0x0000	[8]
Address Management	0x11	7	Default Address	0x0000	[8]

Protocol Type			Protocol Subtype		Reference
Name	ID	Length (bits)	Name	ID	
			Management		
Session Configuration	0x12	7	Default Session Configuration	0x0000	[8]
Multimode Capability Discovery	0x1b	7	Generic Multimode Capability Discovery	0x0001	[8]
Stream	0x13	7	Default Stream	0x0000	[8]
Virtual Stream	0x19	7	Generic Virtual Stream	0x0001	[8]
Stream 0 Application	0x14	7	Default Signaling Application	0x0000	[8]
Stream 1 Application	0x15	7	Default Packet Application bound to the radio network.	0x0001	[8]
Stream 1 Application	0x15	7	Default Packet Application bound to the service network	0x0002	[8]
Stream 1 Application	0x15	7	Enhanced Multi-Flow Packet Application bound to the radio network.	0x0008	[28]
Stream 1 Application	0x15	7	Enhanced Multi-Flow Packet Application bound to the service network.	0x0009	[28]
Stream 2 Application	0x16	7	Default Packet Application bound to the radio network	0x0001	[8]
Stream 2 Application	0x16	7	Default Packet Application bound to the service network	0x0002	[8]
Stream 2 Application	0x16	7	Enhanced Multi-Flow Packet Application bound to the radio network.	0x0008	[28]
Stream 2 Application	0x16	7	Enhanced Multi-Flow Packet Application bound to the service network.	0x0009	[28]
Stream 3 Application	0x17	7	Default Packet Application bound to the radio network	0x0001	[8]
Stream 3 Application	0x17	7	Default Packet Application bound to the service network	0x0002	[8]
Stream 3 Application	0x17	7	Enhanced Multi-Flow Packet Application bound to the radio network.	0x0008	[28]
Stream 3 Application	0x17	7	Enhanced Multi-Flow Packet Application bound to the service network.	0x0009	[28]

- 1 This page intentionally left blank.