

3GPP2 C.P0084-000-0

Version 1.0

Date: April, 2007



3RD GENERATION
PARTNERSHIP
PROJECT 2
"3GPP2"

Overview for Ultra Mobile Broadband (UMB) Air Interface Specification

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 <mailto:secretariat@3gpp2.org>. Requests to reproduce individual Organizational Partner's documents should be directed to that Organizational Partner. See <http://www.3gpp2.org> for more information.

No text.

CONTENTS

1 FOREWORD ix

2 NOTES 1-1

3 REFERENCES 1-1

4 1 Overview 1-1

5 1.1 Scope of This Document 1-1

6 1.2 Architecture Reference Model 1-2

7 1.3 Protocol Architecture 1-3

8 1.3.1 Layers 1-3

9 1.3.2 Protocols 1-4

10 1.3.2.1 Physical Layer 1-5

11 1.3.2.2 MAC Layer 1-5

12 1.3.2.3 Radio Link Layer 1-6

13 1.3.2.4 Application Layer 1-6

14 1.3.2.5 Security Functions 1-7

15 1.3.2.6 Connection Control Plane 1-7

16 1.3.2.7 Session Control Plane 1-8

17 1.3.2.8 Route Control Plane 1-8

18 1.3.2.9 Broadcast-Multicast Service (BCMCS) Upper Layer Protocols 1-9

19 1.4 Protocol Interfaces 1-9

20 1.5 Protocol States 1-10

21 1.6 Protocol Set Identifier 1-11

22 1.7 Protocol Instances and Personalities 1-11

23 1.8 Sessions and Connections 1-12

24 1.9 Security 1-12

25 1.10 Requirements Language 1-12

26 1.11 Notation 1-12

27 1.12 Malfunction Detection 1-14

28 2 Common Procedures 2-1

29 2.1 Protocol Initialization for the InConfiguration Protocol Instance 2-1

30 2.2 Protocol Initialization for the InUse Protocol Instance 2-1

31 2.3 Hard Commit Procedures 2-2

CONTENTS

1	2.4 Soft Commit Procedures.....	2-2
2	2.5 Hash Function	2-2
3	2.6 Pseudorandom Number Generator	2-2
4	2.6.1 General Procedures	2-2
5	2.6.2 Initialization	2-3
6	2.7 Sequence Number Validation	2-3
7	3 Common Data Structures	3-1
8	3.1 Channel Record	3-1
9	3.2 Access Terminal Identifier Record	3-2
10	3.3 Attribute Record.....	3-2
11	3.4 Session State Information Record.....	3-5
12	4 Assigned Names and Numbers.....	4-1
13	5 ANID and SectorID provisioning.....	5-1
14	5.1 ANID Construction.....	5-1
15	5.2 SectorID Construction.....	5-1
16	5.3 UATI Construction	5-1
17		

FIGURES

1	Figure 1-1. UMB Air Interface Specification Document Structure	1-1
2	Figure 1-2. Architecture Reference Model	1-2
3	Figure 1-3. Unicast Route Layering Architecture	1-3
4	Figure 1-4. BCMCS Route Layering Architecture	1-4
5	Figure 1-5. Physical Layer Protocols	1-5
6	Figure 1-6. MAC Layer Protocols	1-5
7	Figure 1-7. Radio Link Layer Protocols	1-6
8	Figure 1-8. Application Layer Protocols	1-6
9	Figure 1-9. Security Protocols	1-7
10	Figure 1-10. Connection Control Plane Protocols.....	1-7
11	Figure 1-11. Session Control Plane Protocols	1-8
12	Figure 1-12. Route Control Plane Protocols	1-8
13	Figure 1-13. BCMCS Upper Layer Protocols	1-9
14	Figure 5-1. ANID Construction as a 6to4 IPv6 Address.....	5-1

15

FIGURES

- 1 No text.

TABLES

1 Table 3-1. SystemType Encoding3-1
2 Table 3-2. ATType Field Encoding3-2
3 Table 3-3. The Format of the Session State Information Record.....3-5
4 Table 3-4. Encoding of the ParameterType Field3-6
5 Table 4-1. Protocol Type and Subtype4-1
6 Table 4-2. Application Layer Protocol ID.....4-2
7 Table 4-3. Initial Protocol Set Identifier4-2
8 Table 4-4. Protocol Set Identifier4-3
9

TABLES

- 1 No text.

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

This Standard was prepared by Technical Specification Group C of the Third Generation Partnership Project 2 (3GPP2). This Standard is the Overview part of the Ultra Mobile Broadband™ (UMB™)¹ air interface. Other parts of this Standard are:

- Physical Layer for Ultra Mobile Broadband (UMB) Air Interface Specification
- MAC Layer for Ultra Mobile Broadband (UMB) Air Interface Specification
- Radio Link Layer for Ultra Mobile Broadband (UMB) Air Interface Specification
- Application Layer for Ultra Mobile Broadband (UMB) Air Interface Specification
- Security Functions for Ultra Mobile Broadband (UMB) Air Interface Specification
- Connection Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification
- Session Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification
- Route Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification
- Broadcast-Multicast Upper Layers for Ultra Mobile Broadband (UMB) Air Interface Specification

Other Standards may be required to implement this system and are listed in the References section of each part.

This standard provides a specification for land mobile wireless systems based upon cellular principles. This Standard is one part of the IMT-2000 CDMA Multi-Carrier, IMT-2000 CDMA MC, also known as cdma2000®².

¹ Ultra Mobile Broadband™ and (UMB™) are trade and service marks owned by the CDMA Development Group (CDG).

² cdma2000® 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® is a registered trademark of the Telecommunications Industry Association (TIA-USA) in the United States.

FOREWORD

- 1 No text.

REFERENCE

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

- 7
- 8 [1] Reserved.
 - 9 [2] C.S0084-001-0, Physical Layer for Ultra Mobile Broadband (UMB) Air Interface
10 Specification.
 - 11 [3] C.S0084-002-0, MAC Layer for Ultra Mobile Broadband (UMB) Air Interface
12 Specification.
 - 13 [4] C.S0084-003-0, Radio Link Layer for Ultra Mobile Broadband (UMB) Air Interface
14 Specification.
 - 15 [5] C.S0084-004-0, Application Layer for Ultra Mobile Broadband (UMB) Air Interface
16 Specification.
 - 17 [6] C.S0084-005-0, Security Functions for Ultra Mobile Broadband (UMB) Air
18 Interface Specification.
 - 19 [7] C.S0084-006-0, Connection Control Plane for Ultra Mobile Broadband (UMB) Air
20 Interface Specification.
 - 21 [8] C.S0084-007-0, Session Control Plane for Ultra Mobile Broadband (UMB) Air
22 Interface Specification.
 - 23 [9] C.S0084-008-0, Route Control Plane for Ultra Mobile Broadband (UMB) Air
24 Interface Specification.
 - 25 [10] C.S0084-009-0, Broadcast-Multicast Upper Layers for Ultra Mobile Broadband
26 (UMB) Air Interface Specification
 - 27 [11] C.R1001, Administration of Parameter Value Assignments for cdma2000 Spread
28 Spectrum Standards. (Informative)
 - 29 [12] IETF RFC 3095, Robust Header Compression (ROHC): Framework and four
30 profiles: RTP, UDP, ESP, and uncompressed
 - 31 [13] RFC 791, Internet Protocol, Sept. 1981
 - 32 [14] RFC 2460, Deering, Hindin, Internet Protocol, Version 6 (IPv6) Specification,
33 December 1998
 - 34 [15] RFC 3748, Extensible Authentication Protocol (EAP)
 - 35 [16] C.S0001, Introduction to cdma2000 Standards for Spread Spectrum Systems
 - 36 [17] C.S0024, cdma2000 High Rate Packet Data Air Interface Specification
 - 37 [18] RFC 3056, Connection of IPv6 Domains via IPv4 Clouds

REFERENCE

- 1 No text.

1 OVERVIEW

1.1 Scope of This Document

The set of documents that form the compatibility specification for UMB air interface systems is shown in Figure 1-1.

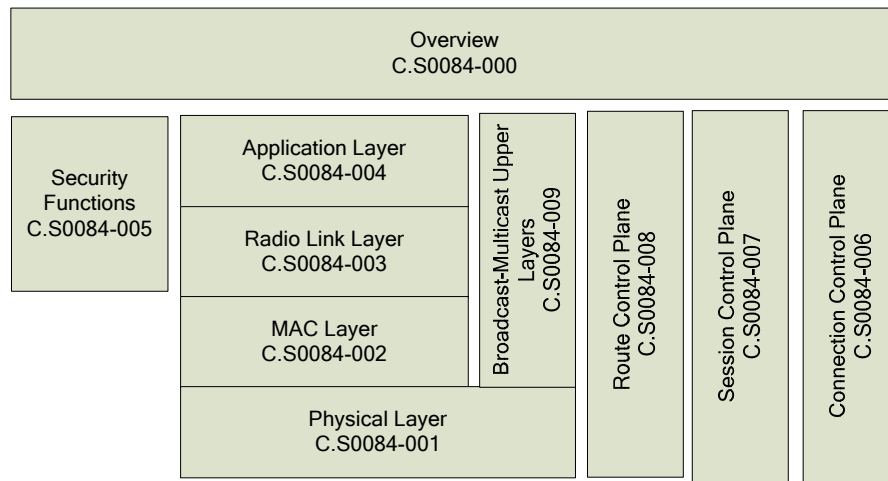


Figure 1-1. UMB Air Interface Specification Document Structure

In the following, “this specification” refers to the set of documents shown in Figure 1-1.

The requirements in this specification ensure that a compliant access terminal can obtain service through any access networks conforming to this specification. 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 specification is based upon spectrum allocations that have been defined by various governmental administrations. Those wishing to deploy systems compliant with this specification 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 specification 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 Architecture Reference Model

The architecture reference model is presented in Figure 1-2.

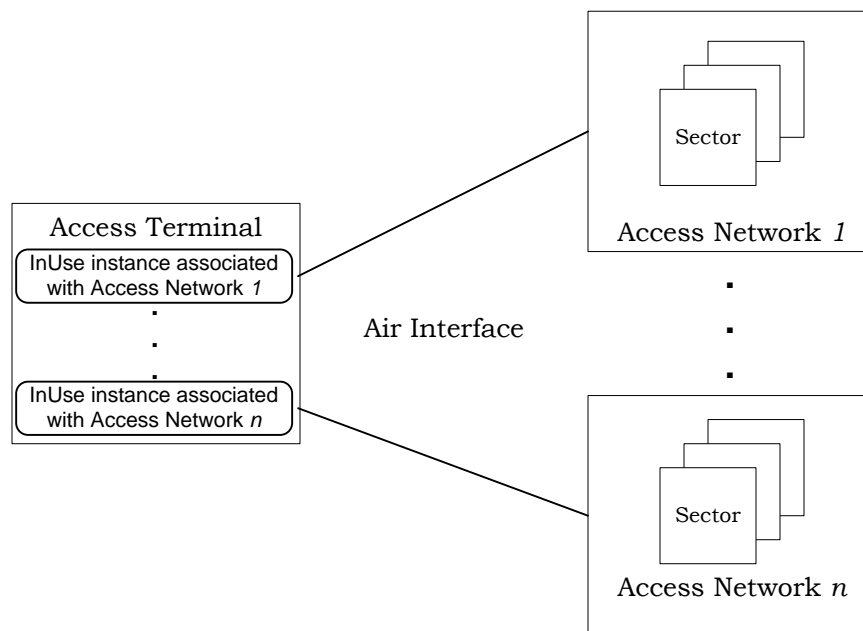


Figure 1-2. Architecture Reference Model

The reference model includes the air interface between the access terminal and the access networks. The access terminal communicates with one or more access networks over the air interface.

The access terminal maintains an InUse instance of the protocol stack associated with each access network that it is in communication with. Each InUse protocol stack instance is called a Route. The protocols used over the air interface are defined in this specification. This specification describes the interaction between one InUse instance at the access terminal and a corresponding InUse instance at one access network. Unless specified otherwise, access terminal and access network procedures are applicable only to the InUse instance being referred to. For example,

- “The access terminal shall transition to Inactive State” shall be interpreted to mean that “This InUse instance of the access terminal shall transition to the Inactive State”.

When a reference to more than one InUse instance is required, the reference is explicitly stated. For example

- “If ExampleVariable is set to zero in all InUse instances at the access terminal, the access terminal shall issue an *Example* command”, shall be interpreted to mean that the instance of the protocol currently being referred to shall check ExampleVariable across all InUse instances, and issue the *Example* command if the condition is satisfied.

1.3 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.

1.3.1 Layers

Figure 1-3 describes the layering architecture for each Route of the air interface that carries unicast data. Each layer or plane consists of one or more protocols that perform the functions of the layer.

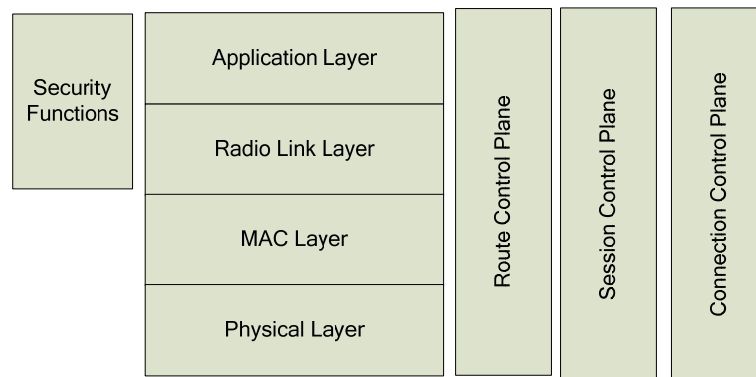


Figure 1-3. Unicast Route Layering Architecture

The layers specified in Figure 1-3 are:

Physical Layer: The Physical Layer provides the channel structure, frequency, power output, modulation, and encoding specifications for the Forward and Reverse Channels. The Physical Layer protocols are defined in [2].

MAC Layer: The Medium Access Control (MAC) Layer defines the procedures used to receive and to transmit over the Physical Layer. The MAC Layer protocols are defined in [3].

Radio Link Layer: Protocols in the Radio Link Layer provide services such as reliable and in-sequence delivery of Application Layer packets, multiplexing of Application Layer packets, and QoS negotiation in support of applications. The Radio Link Layer protocols are defined in [4].

Application Layer: The Application Layer provides multiple applications. It provides the Signaling Protocol for transporting air interface protocol messages. It also provides the Inter-Route Tunneling Protocol for transporting packets to/from other Routes. These Application Layer protocols are defined in [5]. Other examples of Application Layer protocols include the Extensible Authentication Protocol (EAP) (see [15]) for support of authentication, the Internet Protocol (IP) (see [13] and [14]) for transporting user data, the Robust Header Compression (RoHC) (see [12]) for compressing packet headers, and protocols for transporting packets from other air interfaces.

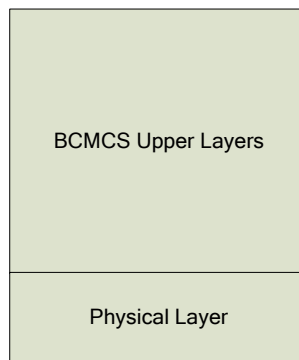
1 Connection Control Plane: The Connection Control Plane provides air link connection
 2 establishment and maintenance services. The Connection Control Plane is defined
 3 in [7].

4 Session Control Plane: The Session Control Plane provides protocol negotiation and
 5 protocol configuration services. The Session Control Plane is defined in [8].

6 Route Control Plane: The Route Control Plane provides creation, maintenance, and
 7 deletion of Routes. The Route Control Plane is defined in [9].

8 Security Functions: Security functions include functions for key exchange, ciphering,
 9 and message integrity protection. Security functions are defined in [6].

10 Figure 1-4 describes the layering architecture for a Route of the air interface that carries
 11 broadcast-multicast (BCMCS) data. Each layer or plane consists of one or more protocols
 12 that perform the functions of the layer.



13
 14 **Figure 1-4. BCMCS Route Layering Architecture**

15 Physical Layer: The Physical Layer provides the channel structure, frequency, power
 16 output, modulation, and encoding specifications for the BCMCS channel. The
 17 Physical Layer protocols are defined in [2].

18 Broadcast-Multicast Upper Layers: Besides the unicast Routes the air interface also
 19 defines a BCMCS Route which contains protocols in support of Broadcast-Multicast
 20 Service (BCMCS). BCMCS upper layer protocols are defined in [10]

21 Each layer contains one or more protocols. Protocols use signaling messages or headers to
 22 convey information to their peer protocols at the other side of the air-link. When protocols
 23 and applications send messages, they use the Signaling Protocol to transmit these
 24 messages.

25 1.3.2 Protocols

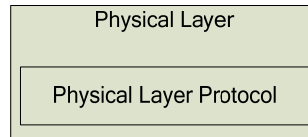
26 Each Layer specifies one or more protocol Types. Protocols are associated with a Type that
 27 denotes the type of the protocol (e.g., Access Channel MAC Protocol) and with a Subtype
 28 that denotes a specific instance of a protocol (e.g., the Basic Access Channel MAC Protocol).

29 The following is a brief description of protocols in each layer. A more complete description
 30 is provided in the Introduction section of each layer.

1 1.3.2.1 Physical Layer

2 The Physical Layer defines the Physical Layer protocol.

3



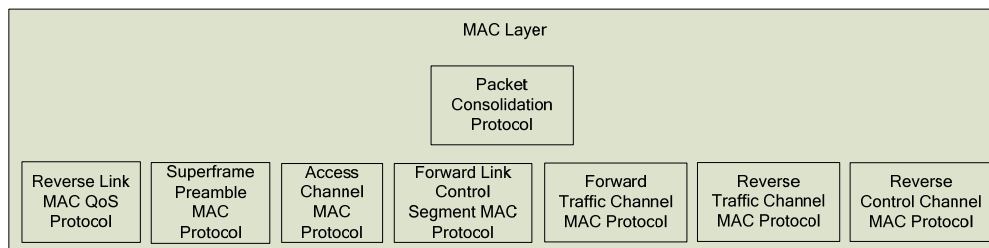
4

5 **Figure 1-5. Physical Layer Protocols**

- 6 • Physical Layer Protocol: Provides channel structure, frequency, power output and
7 modulation specifications for the forward and reverse links.

8 1.3.2.2 MAC Layer

9



10

11 **Figure 1-6. MAC Layer Protocols**

- 12 • Packet Consolidation Protocol: Provides transmit prioritization and packet
13 encapsulation for upper layer packets.
- 14 • Reverse Link QoS MAC Protocol: Provides MAC parameters to meet the QoS granted to
15 upper layer packets.
- 16 • Superframe Preamble MAC Protocol: Provides procedures followed by an access
17 network to transmit and access terminal to receive the superframe preamble.
- 18 • Access Channel MAC Protocol: Provides the procedures followed by the access terminal
19 to transmit, and by an access network to receive the Access Channel.
- 20 • Forward Link Control Segment MAC Protocol: Provides the procedures followed by an
21 access network to transmit, and by the access terminal to receive the Forward Control
22 Channel.
- 23 • Forward Traffic Channel MAC Protocol: Provides the procedures followed by an access
24 network to transmit, and by the access terminal to receive the Forward Data Channel.
- 25 • Reverse Control Channel MAC Protocol: Provides the procedures followed by the access
26 terminal to transmit, and by an access network to receive the Reverse Control Channel.

- Reverse Traffic Channel MAC Protocol: Provides the procedures followed by the access terminal to transmit, and by an access network to receive the Reverse Data Channel.

1.3.2.3 Radio Link Layer

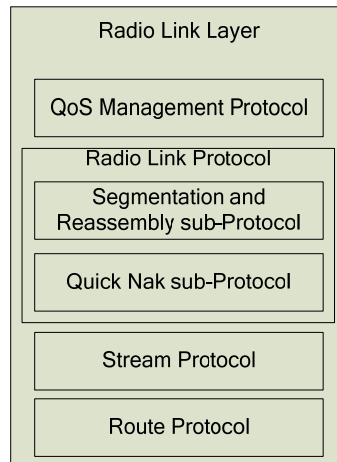


Figure 1-7. Radio Link Layer Protocols

- QoS Management Protocol: The QoS Management Protocol provides negotiation of flow and filter specifications to provide appropriate Quality of Service (QoS) to application layer packets.
- Radio Link Protocol: The Radio Link Protocol (RLP) provides fragmentation and re-assembly, retransmission and duplicate detection for upper layer packets.
- Stream Protocol: The Stream Protocol identifies the stream on which the upper layer fragments are being carried.
- Route Protocol: The Route Protocol routes Stream Protocol packets over through the serving Route between the access terminal and an access network.

1.3.2.4 Application Layer

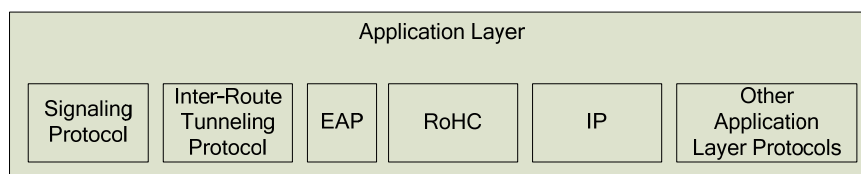


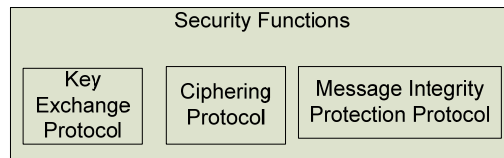
Figure 1-8. Application Layer Protocols

- Signaling Protocol: The Signaling Protocol provides message transmission services for signaling messages.

- 1 • Inter-Route Tunneling Protocol: The Inter-Route Tunneling Protocol provides transport
- 2 of packets from other Routes.
- 3 • Other Application Layer Protocols may create payload to be carried over the UMB air
- 4 interface.

5 1.3.2.5 Security Functions

6



7

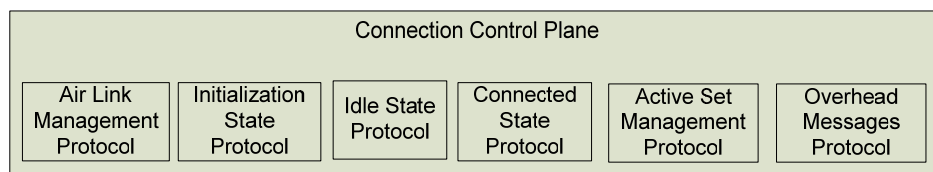
8

Figure 1-9. Security Protocols

- 9 • Key Exchange Protocol: Provides the procedures followed by an access network and the
- 10 access terminal to generate security keys for message integrity and ciphering.
- 11 • Message Integrity Protection Protocol: Provides the procedures followed by an access
- 12 network and the access terminal for integrity protection of signaling messages.
- 13 • Ciphering Protocol: Provides the procedures followed by an access network and the
- 14 access terminal for ciphering traffic.

15 1.3.2.6 Connection Control Plane

16



17

18

Figure 1-10. Connection Control Plane Protocols

- 19 • Air Link Management Protocol: Provides the overall state machine management that an
- 20 access terminal and an access network follow during a connection.
- 21 • Initialization State Protocol: Provides the procedures that an access terminal follows to
- 22 acquire a network and that an access network follows to support network acquisition.
- 23 • Idle State Protocol: Provides the procedures that an access terminal and an access
- 24 network follow when a connection is not open.
- 25 • Connected State Protocol: Provides the procedures that an access terminal and an
- 26 access network follow when a connection is open.
- 27 • Active Set Management Protocol: Provides the means to maintain the air link between
- 28 the access terminal and an access network.

- Overhead Messages Protocol: Provides broadcast messages containing information that is mostly used by Connection Layer protocols.

1.3.2.7 Session Control Plane

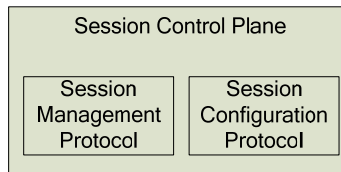


Figure 1-11. Session Control Plane Protocols

- Session Management Protocol: The Session Management Protocol provides means to control the activation and the deactivation of the Session Configuration Protocol. It also provides a session keep alive mechanism.
- Session Configuration Protocol: The Session Configuration Protocol provides negotiation and configuration of the protocols used in the session.

1.3.2.8 Route Control Plane

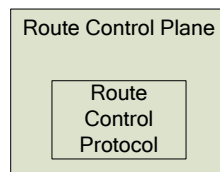
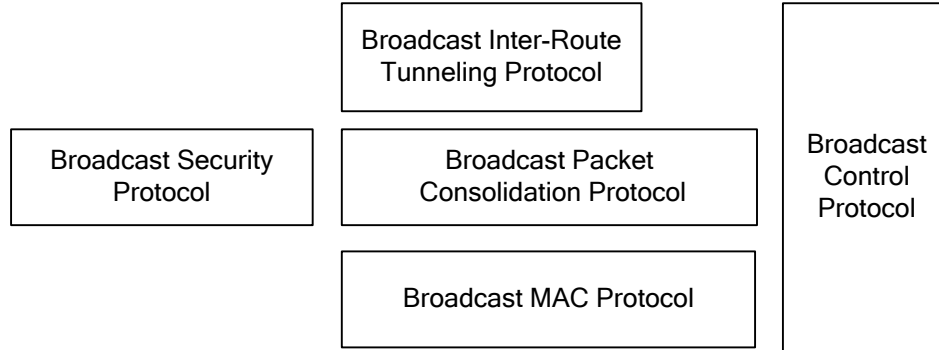


Figure 1-12. Route Control Plane Protocols

- Route Control Protocol: The Route Control Protocol performs creation, maintenance, and deletion of Routes. The Route Control Protocol also performs management of access terminal identifier (ATI).

1 1.3.2.9 Broadcast-Multicast Service (BCMCS) Upper Layer Protocols



2

3

Figure 1-13. BCMCS Upper Layer Protocols

4 Protocols in the BCMCS Upper Layer Protocol Suite provide functions that offer broadcast-
5 multicast service.

- 6 • Broadcast MAC Protocol defines the procedures followed by an access network to
7 transmit, and by the access terminal to receive the BCMCS channel.
- 8 • Broadcast Packet Consolidation Protocol provides framing of BCMCS content packets
9 and multiplexing of packets to be carried on the BCMCS channel.
- 10 • Broadcast Inter-Route Tunneling Protocol provides transport for packets from other
11 Routes.
- 12 • Broadcast Security Protocol provides ciphering of BCMCS content.
- 13 • Broadcast Control Protocol defines control procedures such as registration related to
14 the BCMCS service.

15 Broadcast Physical Layer is defined by the Physical Layer Protocol.

16 **1.4 Protocol Interfaces**

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

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

22 Protocols in this specification have four types of interfaces:

- 23 • Headers and messages are used for communications between a protocol executing in
24 one entity and the same protocol executing in the other entity.
- 25 • Commands are used by a protocol to obtain a service from another protocol within the
26 same access network or access terminal.
- 27 • Indications are used by a protocol to convey information regarding the occurrence of an
28 event to another protocol within the same access network or access terminal. Any
29 protocol can register to receive these indications.

- 1 • Public Data is used to share information in a controlled way between protocols. Public
2 data is shared between protocols in the same layer, between protocols in different
3 layers, as well as between protocols in different Routes. The public data of the InUse
4 protocol is created when an InUse instance (see 1.7) of a protocol is created. All
5 configuration attributes of the InConfiguration instance of a protocol are also public
6 data of that protocol. If public data is defined as static, then the public data has the
7 same value across the same Protocol Type in different Routes. Otherwise, the public
8 data may have different values in different Routes. Public Data also includes Local
9 Common Data. Only the access terminal maintains Local Common Data, and has a
10 single instance of a Local Common Data across all routes.
- 11 • Commands and indications are written in the form of *Protocol.Command* and
12 *Protocol.Indication*. For example, *AccessChannelMAC.Activate* is a command activating
13 the Access Channel MAC, and *IdleState.ConnectionOpened* is an indication provided by
14 the Idle State Protocol that the connection is now open. When the context is clear, the
15 *Protocol* part is dropped (e.g., within the Idle State Protocol, *Activate* refers to
16 *IdleState.Activate*). Most protocols support the following two commands:
 - 17 – *Activate*, which commands the protocol to transition from the Inactive state to
18 some other state.
 - 19 – *Deactivate*, which commands the protocol to transition to the Inactive state.
20 Some protocols do not transition immediately to the Inactive state, due to
21 requirements on orderly cleanup procedures.

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

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

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

31 **1.5 Protocol States**

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

35 Unless otherwise specifically mentioned, the state of an access network refers to the state
36 of a protocol engine in the access network as it applies to a particular access terminal.
37 Since an access network communicates with multiple access terminals, multiple
38 independent instantiations of a protocol will exist in the access network, each with its own
39 independent state machine.

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

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

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

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

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

12 **1.6 Protocol Set Identifier**

13 Protocol Set Identifier is used to identify subtype of each of the protocol in the protocol
14 stack. Initial Protocol Set Identifier is used to identify default protocol stack for each of the
15 major revisions of the air interface specification. Access network includes values of
16 supported Initial Protocol Set Identifier in overhead messages. Multiple Protocol Set
17 Identifiers may map to single Initial Protocol Set Identifier.

18 **1.7 Protocol Instances and Personalities**

19 A protocol instance can be either an InUse instance or an InConfiguration instance.

20 Each protocol specifies procedures and messages corresponding to the InUse and
21 InConfiguration protocol instances. In general, the InConfiguration protocol instances use
22 the services of the Session Configuration Protocol to perform attribute configuration.
23 Typically, non-configuration procedures and messages are processed by the InUse protocol
24 instances.

25 An InConfiguration instance of each protocol is created by the Session Configuration
26 Protocol once the session configuration is initiated (e.g., in the Basic Session Configuration
27 Protocol this occurs upon entering the AT Initiated state). The initialization procedures for
28 an InConfiguration protocol instance are invoked upon creation of the InConfiguration
29 protocol instance. InConfiguration protocol instances can be changed by the Session
30 Configuration Protocol.

31 The configured InConfiguration instances can be stored. Stored protocol stack
32 configurations are called Personalities. The access terminal and access network can store
33 multiple Personalities as part of the session (see 1.8 for a discussion of sessions).

34 The access terminal and the access network determine which Personality to use for every
35 Route. Once the access terminal and access network agree upon using a Personality, then
36 that Personality is Committed to the InUse instance. Each InUse protocol defines a set of
37 Hard Commit and Soft Commit procedures. The Commit procedures for a protocol are
38 invoked by the InUse instance of the Session Configuration Protocol. Commit procedures
39 are used to commit a stored Personality to the InUse instance. The initialization procedures
40 for an InUse protocol instance are invoked upon creation of the InUse protocol instance.

1.8 Sessions and Connections

A session refers to a shared state between the access terminal and an access network. This shared state stores one or more Personalities. Each Personality contains the protocols and protocol configurations that were negotiated and are used for communications between the access terminal and an 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 an access network can open and can close a connection multiple times.

1.9 Security

The air interface supports security functions, which can be used for integrity protection of signaling messages and encryption of signaling messages and other traffic transported between the access terminal and an access network. Unless specified otherwise, all signaling messages shall be integrity protected.

1.10 Requirements Language

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

“Shall” and “shall not” identify requirements to be followed strictly to conform to the specification 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 specification. “Can” and “cannot” are used for statements of possibility and capability, whether material, physical, or causal.

Unless specified otherwise, requirements on access terminal or access network behavior refer to requirements on the relevant Route in the access terminal or access network.

1.11 Notation

A[i] The i^{th} element of array A. The first element of the array is A[0].

<e₁, e₂, ..., e_n> A *structure* with elements ‘e₁’, ‘e₂’, ..., ‘e_n’.

Two structures E = <e₁, e₂, ..., e_n> and F = <f₁, f₂, ..., f_m> are equal if and only if ‘m’ is equal to ‘n’ and e_i is equal to f_i for i=1, ...,n. Given E = <e₁, e₂, ..., e_n> and F = <f₁, f₂, ..., f_m>, the assignment “E = F” denotes the following set of assignments: e_i = f_i, for i=1, ...,n.

1	S.e	The member of the structure ‘S’ that is identified by ‘e’.
2	M[i:j]	Bits i th through j th inclusive (i ≥ j) of the binary representation of
3		variable M. M[0:0] denotes the least significant bit of M.
4		Concatenation operator. (A B) denotes variable A concatenated with
5		variable B.
6	×	Indicates multiplication.
7	⌊x⌋	Indicates the largest integer less than or equal to x: ⌊1.1⌋ = 1, ⌊1.0⌋ =
8		1.
9	⌈x̄⌉	Indicates the smallest integer greater or equal to x: ⌈1.1⌉ = 2, ⌈2.0⌉ =
10		2.
11	x	Indicates the absolute value of x: -17 =17, 17 =17.
12	⊕	Indicates exclusive OR (modulo-2 addition).
13	⊗	Indicates bitwise logical AND operator.
14	min (x, y)	Indicates the minimum of x and y.
15	max (x, y)	Indicates the maximum of x and y.
16	x mod y	Indicates the remainder after dividing x by y: x mod y = x - (y ×
17		⌊x/y⌋).
18	x ^y	Indicates the result of x raised to the power y, also denoted as x ^y .
19	x ^y	Indicates the result of x raised to the power y, also denoted as x ^y .

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

21 Unless indicated otherwise, this specification presents numbers in decimal form. Binary
 22 numbers are distinguished in the text by the use of single quotation marks. Hexadecimal
 23 numbers are distinguished by the prefix ‘0x’.

24 Unless specified otherwise, each field of a packet shall be transmitted in sequence such
 25 that the most significant bit (MSB) is transmitted first and the least significant bit (LSB) is
 26 transmitted last. The MSB is the left-most bit in the figures in this document. If there are
 27 multiple rows in a table, the top-most row is transmitted first. If a table is used to show the
 28 sub-fields of a particular field or variable, the top-most row consists of the MSBs of the
 29 field. Within a row in a table, the left-most bit is transmitted first.

30 Notations of the form “repetition factor of N” or “repeated N times” mean that a total of N
 31 versions of the item are used.

1 1.12 Malfunction Detection

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

2 COMMON PROCEDURES

2.1 Protocol Initialization for the InConfiguration Protocol Instance

Upon invocation of the initialization procedures of the InConfiguration instance of a protocol, the access terminal and the access network shall perform the following in the order specified:

- The access terminal and the access network shall set the value of the static data and static attributes associated with the InConfiguration protocol instance to the corresponding values for the protocol instance of any of the stored personalities.
- If the InConfiguration instance of this protocol is created from a stored personality, then
 - The access terminal and the access network shall set the values of the dynamic attributes associated with the InConfiguration protocol instance to the values of the corresponding dynamic attributes in the stored personality.
- If the InConfiguration instance of this protocol is created from a ProtocolSetIdentifier, then
 - The access terminal and the access network shall set value of the dynamic attributes associated with the InConfiguration protocol instance to the default values specified for each dynamic attribute.

2.2 Protocol Initialization for the InUse Protocol Instance

Upon invocation of the initialization procedures of the InUse instance of a protocol, the access terminal and access network shall perform the following in the order specified:

- If one or more stored personalities exist, then the access terminal and the access network shall set the value of the static data and static attributes associated with the InUse protocol instance to the corresponding values in the stored personalities. Otherwise, the access terminal and the access network shall initialize the static data and static attributes associated with the InUse protocol instance to the default values specified for each of them.
- If the InUse instance of this protocol is created from a stored personality, then
 - The access terminal and the access network shall set value of the dynamic attributes associated with the InUse protocol instance to the values of the corresponding dynamic attributes in the that stored personality.
- If the InUse instance of this protocol is created from an InitialProtocolSetIdentifier, then
 - The access terminal and the access network shall set values of the dynamic attributes associated with the InUse protocol instance to the default values specified for each dynamic attribute.

2.3 Hard Commit Procedures

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

- The current InUse protocol instance shall be purged.
- A new InUse protocol instance shall be created from the stored Personality corresponding to the PersonalityIndex field of the SwitchPersonality message.
- Protocol Initialization for the InUse Protocol Instance shall be performed as specified in 2.2.

2.4 Soft Commit Procedures

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

- The access terminal shall set the static data, static attributes, and soft-committable dynamic attributes of the InUse instance to the corresponding values in the stored personality corresponding to the InUse personality.

2.5 Hash Function

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).

Define:

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

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

The hash value is computed as follows³:

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

2.6 Pseudorandom Number Generator

2.6.1 General Procedures

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

³ 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.

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

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

The access terminal shall initialize the random number generator as defined in 2.6.2.

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

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

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

2.6.2 Initialization

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

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

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

2.7 Sequence Number Validation

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

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

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

⁴ 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., "Random Number Generators: Good Ones are Hard to Find," *Communications of the ACM*, vol. 31, no. 10, October 1988, pp. 1192-1201.

- 1 No text.

3 COMMON DATA STRUCTURES

3.1 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 one of the following values:

Table 3-1. SystemType Encoding

Field value	Meaning
0x00	System compliant to this specification. ChannelNumber field specifies forward channel and Reverse channel that are FDD-paired.
0x01	System compliant to [16] ⁵ .
0x02	System compliant to this specification. ChannelNumber field specifies only the forward channel.
0x03	System compliant to [17]. ChannelNumber field specifies forward channel and reverse channel that are FDD-paired.
0x04	System compliant to [17] ChannelNumber field specifies only the forward channel
0x05-0xff	Reserved

BandClass If the SystemType field is set to 0x00 or 0x01, 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 both the forward channel and the reverse channel. If the SystemType is set to 0x02, then 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 channel only.

ChannelNumber If the SystemType is set to 0x00 or 0x01, the access network shall set this field to the channel number corresponding to the frequency assignment of the channel specified by this record for both the forward channel and the reverse channel. If the SystemType is set to 0x02, this access network shall set this field to the channel

⁵ SystemType of 0x01 applies to [16] and all of its predecessors.

number corresponding to the frequency assignment of the channel specified by this record for the forward channel only.

3.2 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 128

ATIType Access Terminal Identifier Type. This field shall be set to the type of the ATI, as shown in Table 3-2:

Table 3-2. ATIType Field Encoding

ATIType	ATIType Description	ATI Length (bits)
'00'	Invalid	n/a
'01'	Multicast ATI	128
'10'	Unicast ATI	128
'11'	Random ATI (RATI)	128

ATI Access Terminal Identifier. This field shall be set as shown in Table 3-2.

3.3 Attribute Record

The attribute record defines a set of suggested values for a given attribute. The attribute record indicates the Protocol Type to which the attribute belongs. 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.

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.

1 The sender of a ConfigurationResponse message or FastConfigurationAccept message (see
 2 Session Configuration Protocol) selects an attribute-value from a ConfigurationRequest
 3 message or FastConfigurationRequest message respectively by sending the attribute value
 4 if it is a simple attribute or a selected value out of an attribute list. Selection of complex-
 5 attributes is done by sending the value identifier which identifies the complex value.

6 The format of a simple attribute and attribute list is given by
 7

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

One or more instances of the following record

AttributeValue	Attribute dependent
Reserved	variable

- 8 **Length** Length in octets of the attribute record, excluding the Length field.
- 9 **Reserved** This field shall be set to '0'.
- 10 **ProtocolType** If length of this field is 7 bits, then the sender shall set the most
 11 significant bit of this field to '0'; otherwise, the sender shall set the
 12 most significant bit of this field to '1'. The sender shall set this field to
 13 the protocol to which this attribute belongs
- 14 **AttributeID** If length of this field is 8 bits, then the sender shall set the most
 15 significant bit of this field to '0'; otherwise, the sender shall set the
 16 most significant bit of this field to '1'. The sender shall set this field to
 17 the attribute identifier of this attribute.
- 18 **AttributeValue** A suggested value for the attribute. Attribute value lengths are, in
 19 general, an integer number of octets. Attribute values have an
 20 explicit or implicit length indication (e.g., fixed length or null
 21 terminated strings) so that the recipient can successfully parse the
 22 record when more than one value is provided.
- 23 **Reserved** The length of this field is the smallest value that will make the
 24 attribute record octet aligned. The sender shall set this field to zero.
 25 The receiver shall ignore this field.

26 The format of a complex attribute is given by
 27

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

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
----------	----------

- 1 Length Length in octets of the attribute record, excluding the Length field.
- 2 Reserved This field shall be set to '0'.
- 3 ProtocolType If length of this field is 7 bits, then the sender shall set the most
4 significant bit of this field to '0'; otherwise, the sender shall set the
5 most significant bit of this field to '1'. The sender shall set this field to
6 the protocol to which this attribute belongs.
- 7 AttributeID If length of this field is 8 bits, then the sender shall set the most
8 significant bit of this field to '0'; otherwise, the sender shall set the
9 most significant bit of this field to '1'. The sender shall set this field to
10 the attribute identifier of this attribute.
- 11 ValueID It identifies the set of attribute values following this field. The sender
12 shall increment this field for each new set of values for this complex
13 attribute.
- 14 AttributeValue A suggested value for the attribute. Attribute value lengths are in
15 general an integer number of octets. Attribute values have an explicit
16 or implicit length indication (e.g., fixed length or null terminated
17 strings) so that the recipient can successfully parse the record when
18 more than one value is provided.
- 19 Reserved The length of this field is the smallest value that will make the
20 attribute record octet aligned. The sender shall set this field to zero.
21 The receiver shall ignore this field.

3.4 Session State Information Record

The Session State Information is to be used for transferring the session parameters corresponding to the InUse protocol instances from a source access network to a target access network. Session parameters are the attributes and the internal parameters that define the state of each protocol. The format of this record is shown in Table 3-3. If an attribute is not contained in the Session State Information Record, the target access network shall assume that the missing attributes have the default values (specified for each attribute in each protocol). The sender shall include all the Parameter Records associated with the ProtocolType and ProtocolSubtype in the same Session State Information Record.

Table 3-3. 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

FormatID	This field identifies the format of the rest of the fields in this record and shall be set to zero.
Reserved	This field shall be set to zero.
ProtocolType	If length of this field is 7 bits, then the most significant bit of this field shall be set to '0'; otherwise, the most significant bit of this field shall be set to '1'. This field shall be set to the protocol to which this attribute belongs.
ProtocolSubtype	This field shall be set to the protocol subtype value (see Table 4-1) for the protocol associated with the encapsulated session parameters.
ParameterType	This field shall be set according to Table 3-4.

1

Table 3-4. Encoding of the ParameterType Field

Field Value	Meaning
0x00	The ParameterType-specific record consists of a Complex or a Simple Attribute as defined in 3.3. The ValueID field of the complex attribute shall be set to zero.
0x01	The ParameterType-specific record consists of Non-Attribute Data as defined in by each protocol.
All other values	ParameterType-specific record are protocol dependent

2 ParameterType-specific record

3

4

5

6

7

8

9

10

If the ParameterType field is set to 0x00, then this record shall be set to the simple or complex attribute (see 3.3) associated with the protocol identified by the (ProtocolType, ProtocolSubtype) pair. If the ParameterType field is set to 0x01, then this record shall be set to the Non-Attribute Data associated with the protocol identified by the (ProtocolType, ProtocolSubtype) pair. Otherwise, the structure of this record shall be as specified by the protocol which is identified by the (ProtocolType, ProtocolSubtype) pair.

1 **4 ASSIGNED NAMES AND NUMBERS**

2 **Table 4-1. Protocol Type and Subtype**

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Physical Layer	0x4000	15	Basic Physical Layer	0x0000
Packet Consolidation	0x4001	15	Basic Packet Consolidation	0x0000
Superframe Preamble MAC	0x4002	15	Basic Superframe Preamble MAC	0x0000
Access Channel MAC	0x4003	15	Basic Access Channel MAC	0x0000
Forward Link Control Segment MAC	0x4004	15	Basic Forward Link Control Segment MAC	0x0000
Forward Traffic Channel MAC	0x00	7	Basic Forward Traffic Channel MAC	0x0000
Reverse Traffic Channel MAC	0x01	7	Basic Reverse Traffic Channel MAC	0x0000
Reverse Control Channel MAC	0x4005	15	Basic Reverse Control Channel MAC	0x0000
MAC RL QoS	0x4006	15	Basic MAC RL QoS	0x0000
Route	0x02	7	Basic Route	0x0000
Stream	0x4007	15	Basic Stream	0x0000
Radio Link corresponding to Stream NN ⁶	0x41NN	15	Basic Radio Link	0x0000
QoS Management	0x03	7	Basic QoS Management	0x0000
Application bound to Stream NN	0x42NN	15	NA	NA
Key Exchange	0x04	7	Basic Key Exchange	0x0000
Ciphering	0x4008	15	AES Ciphering	0x0000
Message Integrity	0x4009	15	Basic Message Integrity	0x0000
Air Link Management	0x400a	15	Basic Air Link Management	0x0000
Initialization State	0x400b	15	Basic Initialization State	0x0000
Idle State	0x05	7	Basic Idle State	0x0000
Connected State	0x06	7	Basic Connected State	0x0000

⁶ NN is the two-digit hexadecimal Stream number in the range 0x00 to 0x1f

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Active Set Management	0x07	7	Basic Active Set Management	0x0000
Overhead Message	0x400c	15	Basic Overhead Message	0x0000
Session Management	0x400d	15	Basic Session Management	0x0000
Session Configuration	0x400e	15	Basic Session Configuration	0x0000
Route Control	0x08	7	Basic Route Control	0x0000
BCMCS Protocol Suite	0x400f	15	Basic BCMCS Suite	0x0000

1 **Table 4-2. Application Layer Protocol ID**

Value	Application Layer Protocol
0x00	NULL
0x01	Reserved
0x02	Reserved
0x03	Reserved
0x04	Robust Header Compression (RoHC) as defined in [12]
0x05	Internet Protocol (IP) version 4 as defined in [7] and version 6 as defined in [14]
0x06	Signaling
0x07	Inter-Route Tunneling

2
3 **Table 4-3. Initial Protocol Set Identifier**

Value	Protocol Stack
0x0	Protocol stack with no BCMCS Protocol Suite, and with all other protocols with subtype 0x0000.
0xf	Protocol stack with only BCMCS Protocol Suite with subtype 0x0000
All other values	Reserved

1

Table 4-4. Protocol Set Identifier

Value	Protocol Stack
0x0000	Protocol stack with no BCMCS Protocol Suite, and with all other protocols with subtype 0x0000.
All other values	Reserved

2

- 1 No text.

5 ANID AND SECTORID PROVISIONING

The access network shall follow the procedures in this section to ensure that the ANID, SectorID, and UATI are globally unique.

5.1 ANID Construction

Access network shall construct a globally Unique ANIDs as follows:

Globally Unique ANID shall be set to a 6to4 IPv6 address as specified in Figure 5-1 (see [18]).

FP '001' (3 bits)	TLA '0000000000010' (13 bits)	V4ADDR (32 bits)	SLA ID (16 bits)	0x0000000000000001 (64 bits)
-------------------------	-------------------------------------	-------------------------	-------------------------	---------------------------------

Where:

V4ADDR is the globally routable IPv4 address of a group of access networks (group size may be one).

SLA ID is a locally unique ID given to this access network within the group of access networks sharing V4ADDR

Figure 5-1. ANID Construction as a 6to4 IPv6 Address

5.2 SectorID Construction

Access network shall construct a globally unique SectorID as follows:

64 MSBs of ANID to which this Sector belongs (64 bits)	ID of the Sector unique within the AN (64 bits)
--------------------------------------------------------------	--------------------------------------------------------

64 LSBs of SectorID shall not be set to any of the following:

- 0x0000000000000000, or
- 0x0000000000000001, or
- 64 LSBs of another SectorID with the same 64 MSBs as this SectorID, or
- 64 LSBs of another UATI with the same 64 MSBs as this SectorID.

5.3 UATI Construction

Access network shall construct a globally unique UATI as follows:

64 MSBs of ANID to which this UATI belongs (64 bits)	ID of the AT unique within the AN (64 bits)
------------------------------------------------------------	----------------------------------------------------

64 LSBs of UATI shall not be set to any of the following:

- 0x0000000000000000, or

- 1 • 0x0000000000000001, or
- 2 • 64 LSBs of another SectorID with the same 64 MSBs as this UATI, or
- 3 • 64 LSBs of another UATI with the same 64 MSBs as this UATI.
- 4