

3GPP2 C.S0084-000-0

Version 3.0

Date: August, 2008



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.....	1-5
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	1-7
14	1.3.2.5 Security Functions.....	1-7
15	1.3.2.6 Connection Control Plane	1-8
16	1.3.2.7 Session Control Plane	1-8
17	1.3.2.8 Route Control Plane.....	1-9
18	1.3.2.9 Broadcast-Multicast Service (BCMCS) Upper Layer Protocols.....	1-9
19	1.4 Protocol Interfaces	1-10
20	1.5 Protocol Data.....	1-11
21	1.6 Protocol States	1-11
22	1.7 Protocol Set Identifier	1-12
23	1.8 Protocol Instances and Personalities.....	1-12
24	1.9 Sessions and Connections	1-13
25	1.10 Security.....	1-13
26	1.11 Requirements Language	1-13
27	1.12 Notation	1-14
28	1.13 Malfunction Detection	1-15
29	2 Common Procedures	2-1
30	2.1 Protocol Initialization for the InConfiguration Protocol Instance.....	2-1
31	2.2 Protocol Initialization for the InUse Protocol Instance	2-1

CONTENTS

1 2.3 Hard Commit Procedures 2-2

2 2.4 Soft Commit Procedures..... 2-2

3 2.5 Hash Function 2-2

4 2.6 Pseudorandom Number Generator 2-3

5 2.6.1 General Procedures 2-3

6 2.6.2 Initialization 2-3

7 2.7 Sequence Number Validation 2-3

8 2.8 AttributeID numbering..... 2-4

9 3 Common Data Structures 3-1

10 3.1 Channel Record 3-1

11 3.1.1 SystemTypeSpecificFields when SystemType is 0x00 3-1

12 3.1.2 SystemTypeSpecificFields when SystemType is 0x01, 0x02, or 0x03 3-3

13 3.2 Neighbor Technology Record 3-4

14 3.3 Access Terminal Identifier Record 3-5

15 3.4 Attribute Record..... 3-6

16 3.5 Non-attribute Data Record 3-8

17 3.6 Session State Information Record..... 3-8

18 4 Assigned Names and Numbers..... 4-1

19 5 ANID, SectorID, and UATI provisioning 5-1

20 5.1 ANID Construction 5-1

21 5.2 SectorID Construction..... 5-1

22 5.3 UATI Construction 5-1

23

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	1-7
9	Figure 1-9. Security Protocols	1-7
10	Figure 1-10. Connection Control Plane Protocols.....	1-8
11	Figure 1-11. Session Control Plane Protocols	1-8
12	Figure 1-12. Route Control Plane Protocols	1-9
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. ATIType Field Encoding3-6

3 Table 3-3. The Format of the Session State Information Record.....3-9

4 Table 3-4. Encoding of the ParameterType Field3-10

5 Table 4-1. Protocol Type and Subtype for InUse Instance4-2

6 Table 4-2. Protocol Type and Subtype for InConfiguration Instance.....4-3

7 Table 4-3. Application Layer Protocol ID.....4-4

8 Table 4-4. Initial Protocol Set Identifiers.....4-5

9 Table 4-5. BCMCS Initial Protocol Set Identifiers.....4-5

10 Table 4-6. Protocol Set Identifier4-5

11

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.

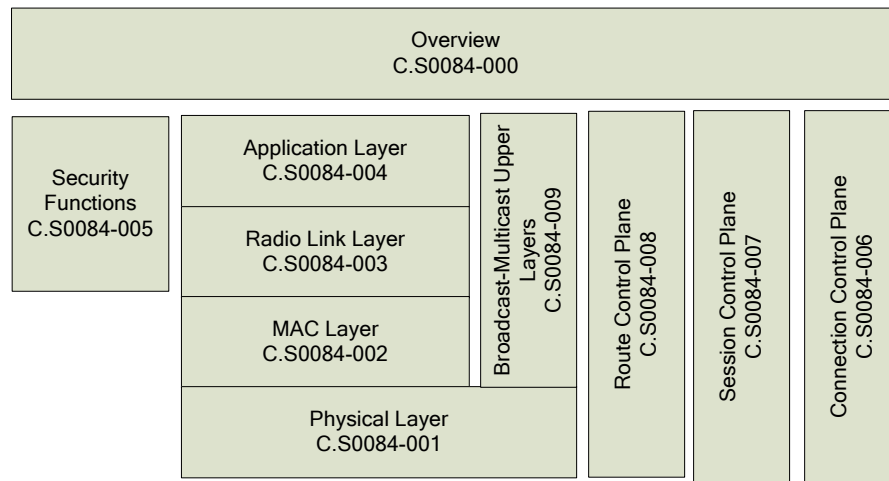


Figure 1-1. UMB Air Interface Specification Document Structure

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.

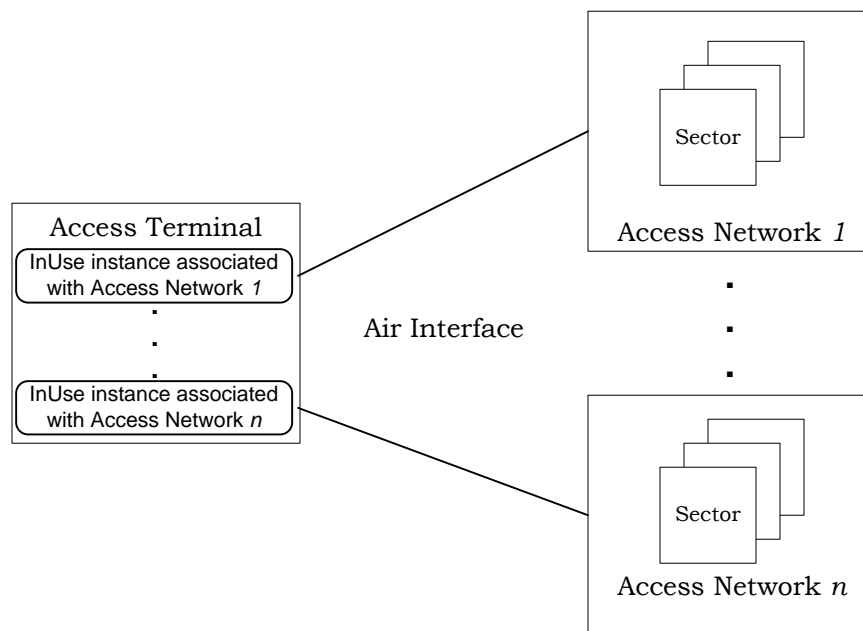


Figure 1-2. Architecture Reference Model

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. An access network may not support a radio link. An access network that does not have direct radio communication with the access terminal (either because the access network does not support a radio link or because the access network is not the current serving access network) can communicate with the access terminal by sending/receiving Route Protocol packets through the serving access network using the Inter-Route Tunneling Protocol of the serving access network.

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.

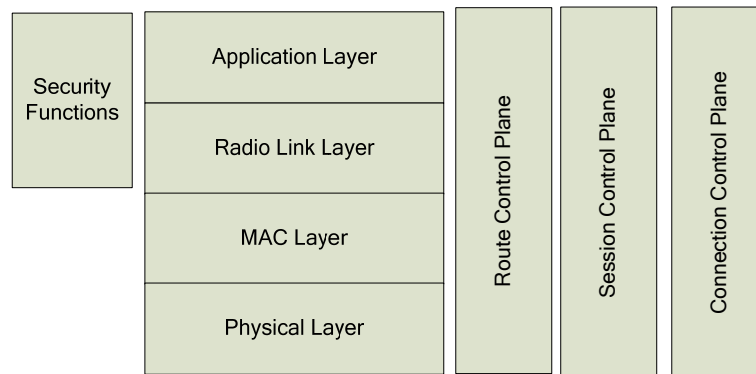


Figure 1-3. Unicast Route Layering Architecture

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 Quality of Service (QoS) negotiation in support of applications. The Radio Link Layer protocols are defined in [4].

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

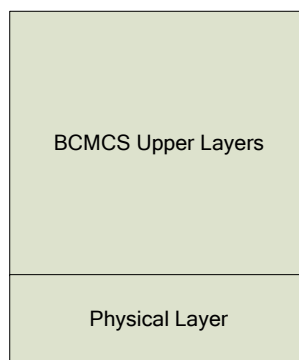
9 Connection Control Plane: The Connection Control Plane provides air link connection
 10 establishment and maintenance services. The Connection Control Plane is defined
 11 in [7].

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

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

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

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



21
 22 **Figure 1-4. BCMCS Route Layering Architecture**

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

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

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

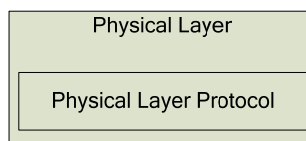
1 1.3.2 Protocols

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

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

7 1.3.2.1 Physical Layer

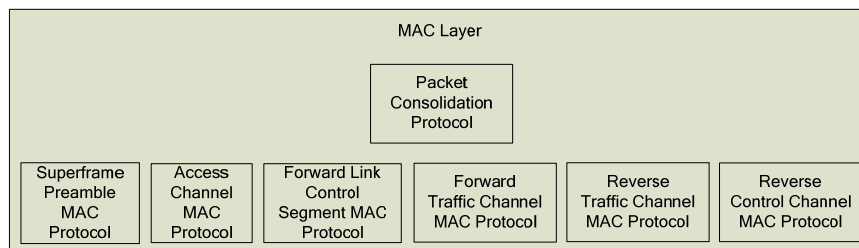
8 The Physical Layer defines the Physical Layer protocol.



10
11 **Figure 1-5. Physical Layer Protocols**

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

14 1.3.2.2 MAC Layer

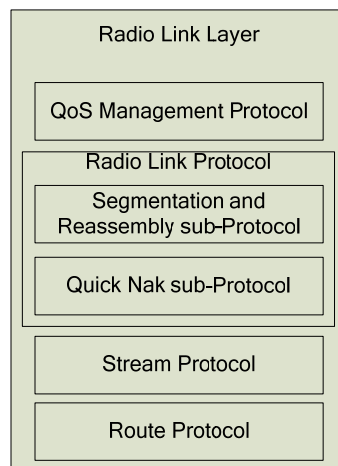


16
17 **Figure 1-6. MAC Layer Protocols**

- 18 • Packet Consolidation Protocol: Provides transmit prioritization and packet
 19 encapsulation for upper layer packets.
- 20 • Superframe Preamble MAC Protocol: Provides procedures followed by an access
 21 network to transmit and access terminal to receive the superframe preamble.
- 22 • Access Channel MAC Protocol: Provides the procedures followed by the access terminal
 23 to transmit, and by an access network to receive the Access Channel.
- 24 • Forward Link Control Segment MAC Protocol: Provides the procedures followed by an
 25 access network to transmit, and by the access terminal to receive the Forward Control
 26 Channel.

- 1 • Forward Traffic Channel MAC Protocol: Provides the procedures followed by an access
2 network to transmit, and by the access terminal to receive the Forward Data Channel.
- 3 • Reverse Control Channel MAC Protocol: Provides the procedures followed by the access
4 terminal to transmit, and by an access network to receive the Reverse Control Channel.
- 5 • Reverse Traffic Channel MAC Protocol: Provides the procedures followed by the access
6 terminal to transmit, and by an access network to receive the Reverse Data Channel.

7 1.3.2.3 Radio Link Layer

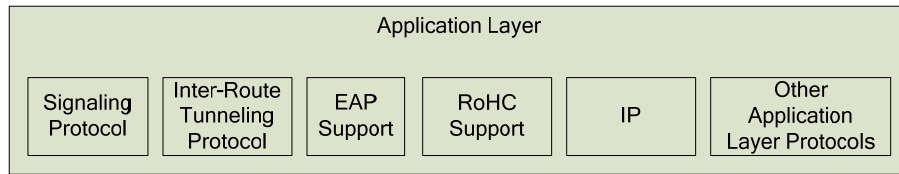


9
10 **Figure 1-7. Radio Link Layer Protocols**

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

1 1.3.2.4 Application Layer

2



3

4

Figure 1-8. Application Layer Protocols

5

6

7

8

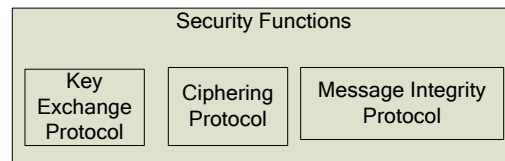
9

10

- Signaling Protocol: The Signaling Protocol provides message transmission services for signaling messages.
- Inter-Route Tunneling Protocol: The Inter-Route Tunneling Protocol provides transport of packets from other Routes.
- Other Application Layer Protocols such as IP, EAP Support Protocol, RoHC Support Protocol etc. may create payload to be carried over the UMB air interface.

11 1.3.2.5 Security Functions

12



13

14

Figure 1-9. Security Protocols

15

16

17

18

19

20

21

- Key Exchange Protocol: Provides the procedures followed by an access network and the access terminal to generate security keys for message integrity protection and cipherring.
- Message Integrity Protocol: Provides the procedures followed by an access network and the access terminal for integrity protection of signaling messages.
- Cipherring Protocol: Provides the procedures followed by an access network and the access terminal for cipherring traffic.

1.3.2.6 Connection Control Plane

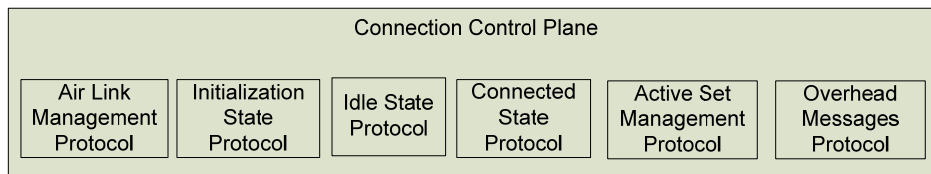


Figure 1-10. Connection Control Plane Protocols

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

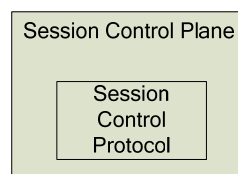
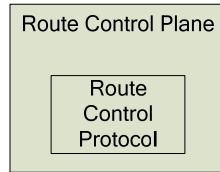


Figure 1-11. Session Control Plane Protocols

- Session Control Protocol: The Session Control Protocol provides negotiation and configuration of the protocols used in the session.

1 1.3.2.8 Route Control Plane

2

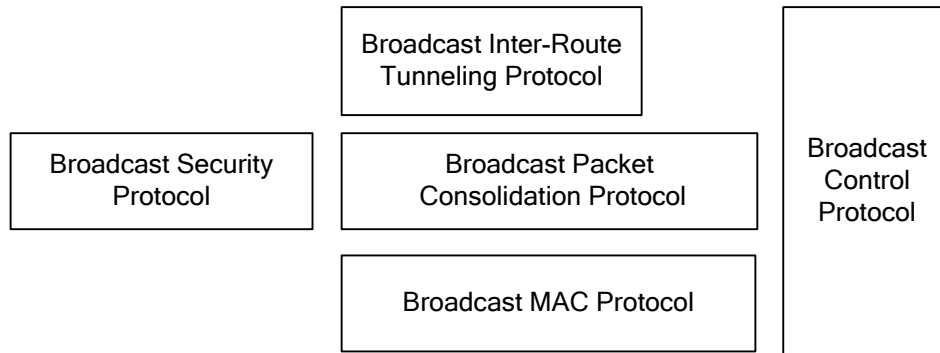


3

4 **Figure 1-12. Route Control Plane Protocols**

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

8 1.3.2.9 Broadcast-Multicast Service (BCMCS) Upper Layer Protocols



9

10 **Figure 1-13. BCMCS Upper Layer Protocols**

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

- 13 • Broadcast MAC Protocol defines the procedures followed by an access network to
 14 transmit, and by the access terminal to receive the BCMCS channel.
- 15 • Broadcast Packet Consolidation Protocol provides framing of BCMCS content packets
 16 and multiplexing of packets to be carried on the BCMCS channel.
- 17 • Broadcast Inter-Route Tunneling Protocol provides transport for packets from other
 18 Routes.
- 19 • Broadcast Security Protocol provides ciphering of BCMCS content.
- 20 • Broadcast Control Protocol defines control procedures such as registration related to
 21 the BCMCS service.

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

1.4 Protocol Interfaces

This specification 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 an access terminal and an access network.

Protocols in this specification have four types of interfaces:

- Headers and/or Trailers are added by a protocol to payload received from an upper layer protocol, and are used for communications between a protocol executing in one entity and the same protocol executing in the other entity.
- Messages are generated by a protocol and are transported using the Signaling Protocol. Messages are used for communications between a protocol executing in one entity and the same protocol executing in the other entity. Messages can be transmitted using Reliable, or Best Effort (unreliable) delivery.
- Commands are used by a protocol to obtain a service from another protocol within the same access network or access terminal.
- Indications are used by a protocol to convey information regarding the occurrence of an event to another protocol within the same access network or access terminal. Any protocol can register to receive these indications.
- Public Data is used to share information in a controlled way between protocols. Public data is shared between protocols in the same layer, between protocols in different layers, as well as between protocols in different Routes.
- Commands and indications are written in the form of *Protocol.Command* and *Protocol.Indication*. For example, *AccessChannelMAC.Activate* is a command activating the Access Channel MAC, and *IdleState.ConnectionOpened* is an indication provided by the Idle State Protocol that the connection is now open. When the context is clear, the *Protocol* part is dropped (e.g., within the Idle State Protocol, *Activate* refers to *IdleState.Activate*).

Unless specified otherwise, a protocol registers for indications from other protocols in the same Route. Unless specified otherwise, a protocol sends/receives commands to/from other protocols in the same Route.

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 or Route open/close related functions.

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

4 Headers, trailers, and messages are binding on all implementations. Commands,
5 indications, protocol data, and procedures that are internal to the access network or access
6 terminal are used purely as a device for a clear and precise specification. Access terminals
7 and access networks can be compliant with this specification while choosing a different
8 implementation that exhibits identical over-the-air behavior.

9 **1.5 Protocol Data**

10 Each protocols owns and acts on parameters that are called protocol data. Protocol data
11 can have various properties:

12 Public: Protocol data is said to be public if the value of the data is visible outside the
13 protocol. Value of public data may be read by other protocols within the same Route as well
14 as by protocols in other Routes.

15 Static (Dynamic): Protocol data is said to be static if the data must take the same value
16 across the same Protocol Type in different Routes and Personalities. Since the value of
17 static data is the same across Routes and Personalities, all static data must be public.
18 Protocol data is said to be dynamic if the data may take different values in different Routes
19 or Personalities.

20 Local Common data: Local common data is static data that is maintained only by the
21 access terminal. All local common data is static and , and all local common data is public.

22 Attribute: Protocol data is called an attribute if the value of the data can be negotiated
23 between the access terminal and the access network using configuration services provided
24 by the InUse instance of the Session Control Protocol. Attributes may be either soft or hard
25 committable. An attribute is said to be soft committable if changes to values of the data can
26 take effect without closing the Route. In contrast, an attribute is said to be hard
27 committable if the Route must be closed and re-opened for changes in data values to take
28 effect.

29 Session data: Protocol data is said to be session data if it is part of the air interface session
30 that could be copied from one access network to another. All attributes are session data.
31 Session data is conveyed between access networks through Session State Information
32 Records.

33 **1.6 Protocol States**

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

37 Unless otherwise specifically mentioned, the state of an access network refers to the state
38 of a protocol engine in the access network as it applies to a particular access terminal.
39 Since an access network communicates with multiple access terminals, multiple

1 independent instantiations of a protocol will exist in the access network, each with its own
2 independent state machine.

3 Unless otherwise specifically mentioned, the state of the access terminal refers to the state
4 of a protocol engine in the access terminal as it applies to a particular access network.
5 Since the access terminal communicates with multiple access networks, multiple
6 independent instantiations of a protocol will exist in the access terminal (one in each
7 Route), each with its own independent state machine.

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

10 Typical events leading to a transition from one state to another are the receipt of a
11 message, a command, an indication, or the expiration of a timer.

12 When a protocol is not functional at a particular time the protocol is placed in a state that
13 is usually called the Inactive state. This state is common for most protocols.

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

17 If a protocol has a single state other than the Inactive state, that state is usually called the
18 Active state. If a protocol has more than one state other than the Inactive state, all of these
19 states are considered active, and are given individual names.

20 **1.7 Protocol Set Identifier**

21 Protocol Set Identifier is used to identify subtype of each of the protocol in the protocol
22 stack. Initial Protocol Set Identifier is used to identify default protocol stack for each of the
23 major revisions of the air interface specification. Access network includes values of
24 supported Initial Protocol Set Identifier in overhead messages. Multiple Protocol Set
25 Identifiers may map to single Initial Protocol Set Identifier.

26 **1.8 Protocol Instances and Personalities**

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

28 Each protocol specifies procedures and messages corresponding to the InUse and
29 InConfiguration protocol instances. In general, the InConfiguration protocol instances use
30 the services of the Session Control Protocol to perform attribute configuration. Typically,
31 non-configuration procedures and messages are processed by the InUse protocol instances.

32 An InConfiguration instance of each protocol is created by the Session Control Protocol
33 once the session configuration is initiated (e.g., in the Basic Session Control Protocol this
34 occurs upon entering the AT Initiated state). The initialization procedures for an
35 InConfiguration protocol instance are invoked upon creation of the InConfiguration
36 protocol instance. InConfiguration protocol instances can be changed by the Session
37 Control Protocol.

1 The configured InConfiguration instances can be stored. Stored protocol stack
2 configurations are called Personalities. The access terminal and access network can store
3 multiple Personalities as part of the session (see 1.9 for a discussion of sessions).

4 The access terminal and the access network determine which Personality to use for every
5 Route. Once the access terminal and access network agree upon using a Personality, then
6 that Personality is Committed to the InUse instance. Each InUse protocol defines a set of
7 Hard Commit and Soft Commit procedures. The Commit procedures for a protocol are
8 invoked by the InUse instance of the Session Control Protocol. Commit procedures are
9 used to commit a stored Personality to the InUse instance. The initialization procedures for
10 an InUse protocol instance are invoked upon creation of the InUse protocol instance.

11 **1.9 Sessions and Connections**

12 A session refers to a shared state between the access terminal and an access network. This
13 shared state stores one or more Personalities. Each Personality contains the protocols and
14 protocol configurations that were negotiated and are used for communications between the
15 access terminal and an access network.

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

18 A connection is a particular state of the air-link in which the access terminal is assigned
19 dedicated communication channels.

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

22 **1.10 Security**

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

27 **1.11 Requirements Language**

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

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

1 **1.12 Notation**

2	A[i]	The i^{th} element of array A. The first element of the array is A[0].
3	<e₁, e₂, ..., e_n>	A <i>structure</i> with elements 'e ₁ ', 'e ₂ ', ..., 'e _n '.
4		Two structures E = <e ₁ , e ₂ , ..., e _n > and F = <f ₁ , f ₂ , ..., f _m > are equal if
5		and only if 'm' is equal to 'n' and e _i is equal to f _i for i=1, ...n.
6		Given E = <e ₁ , e ₂ , ..., e _n > and F = <f ₁ , f ₂ , ..., f _m >, the assignment "E =
7		F" denotes the following set of assignments: e _i = f _i , for i=1, ...n.
8	S.e	The member of the structure 'S' that is identified by 'e'.
9	M[i:j]	Bits i^{th} through j^{th} inclusive ($i \geq j$) of the binary representation of
10		variable M. M[0:0] denotes the least significant bit of M.
11		Concatenation operator. (A B) denotes variable A concatenated with
12		variable B.
13	×	Indicates multiplication.
14	⌊x⌋	Indicates the largest integer less than or equal to x: ⌊1.1⌋ = 1, ⌊1.0⌋ =
15		1.
16	⌈x⌉	Indicates the smallest integer greater or equal to x: ⌈1.1⌉ = 2, ⌈2.0⌉ =
17		2.
18	x	Indicates the absolute value of x: -17 =17, 17 =17.
19	⊕	Indicates exclusive OR (modulo-2 addition).
20	⊗	Indicates bitwise logical AND operator.
21	min (x, y)	Indicates the minimum of x and y.
22	max (x, y)	Indicates the maximum of x and y.
23	x mod y	Indicates the remainder after dividing x by y: x mod y = x - (y ×
24		⌊x/y⌋).
25	x ^y	Indicates the result of x raised to the power y, also denoted as x ^y .
26	x ^y	Indicates the result of x raised to the power y, also denoted as x ^y .
27	N!	Indicates the factorial of N, i.e., N! = 1×2×3× ... ×N.
28	Rounding	Indicates the nearest integer indicates the nearest integer: 1 for 1.4, -
29		1 for -1.4, 2 for 1.5, and -2 for -1.5.

1 ${}^n C_r$ $n! / [(n-r)! \times r!]$

2 ${}^n P_r$ $n! / (n-r)!$

3 (a, b) Complex number $a + jb$, where $j = \text{sqrt}(-1)$

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

5 Unless indicated otherwise, this specification presents numbers in decimal form. Binary
6 numbers are distinguished in the text by the use of single quotation marks. Hexadecimal
7 numbers are distinguished by the prefix '0x'.

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

14 Notations of the form "repetition factor of N" or "repeated N times" mean that a total of N
15 versions of the item are used.

16 **1.13 Malfunction Detection**

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

- 1 No text.

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:

- If one or more stored personalities exist then ~~the~~ the access terminal and the access network shall set the value of the static data associated with the InConfiguration protocol instance to the corresponding values for the protocol instance of any of the stored ~~personalities~~ Personalities. Otherwise, the access terminal and the access network shall set the value of the static attributes associated with the InConfiguration instance to the default values specified for each static attribute.
- If the InConfiguration instance of this protocol is created from a stored personality Personality³, 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 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 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 stored personality.

³ The procedure that causes the creation of the InConfiguration instance specifies whether the InConfiguration instance is to be created from a stored Personality or a ProtocolSetIdentifier. For example, the ConfigurationRequest message specified in [8] includes fields that provide such information.

- 1 • If the InUse instance of this protocol is created from an InitialProtocolSetIdentifier, then
 - 2 – The access terminal and the access network shall set values of the dynamic
 - 3 attributes associated with the InUse protocol instance to the default values
 - 4 specified for each dynamic attribute.

5 **2.3 Hard Commit Procedures**

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

- 9 • The current InUse protocol instance shall be purged.
- 10 • A new InUse protocol instance shall be created from the stored Personality
- 11 corresponding to the PersonalityIndex field of the SwitchPersonality message.
- 12 • Protocol Initialization for the InUse Protocol Instance shall be performed as specified in
- 13 2.2.

14 **2.4 Soft Commit Procedures**

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

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

21 **2.5 Hash Function**

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

25 Define:

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

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

29 The hash value is computed as follows⁴:

$$30 \quad R = \lfloor N \times ((40503 \times (L \oplus H \oplus \text{Decorrelate})) \bmod 2^{16}) / 2^{16} \rfloor.$$

⁴ 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.

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

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

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

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

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

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

⁵ 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 is considered a valid message and $V(R)$ is set to this sequence number; otherwise, the
2 message is considered an invalid message.

3 **2.8 AttributeID numbering**

4 AttributeIDs are 16 bits long. AttributeID assignment follows the below rules:

- 5 • **0x0000 to 0x00FF** – Simple attributes with one instance. 256 attributes supported.
6 Example: ConnectionFailureReportingEnabled, which is a Boolean attribute of the
7 Basic Air Link Management Protocol (in the Connection Control Plane).
- 8 • **0x01NN to 0x7FNN** – Simple attributes with NN or KK instances. 127 types, with 256
9 instances per type.
10 Example ReservationKKStreamFwd, which is an attribute of Basic QoS Management
11 Protocol (in the Radio Link Layer).
- 12 • **0x8000 to 0x80FF** – Complex attributes with one instance. 256 attributes.
13 Example: ANSupportedQoSProfiles which is an attribute of the Basic QoS Management
14 Protocol (in the Radio Link Layer).
- 15 • **0x81NN to 0xFFNN** – Complex attributes with NN or KK instances, etc. 127 types, with
16 256 instances per type.
17 Example: ReservationKKQoSRequestFwd which is an attribute of the Basic QoS
18 Management Protocol (Radio Link Layer).

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
Length	8
SystemTypeSpecificFields	8 × Length

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] ⁶ .
0x02	System compliant to [17]. ChannelNumber field specifies forward channel and reverse channel that are FDD-paired.
0x03	System compliant to [17] ChannelNumber field specifies only the forward channel
0x04-0xff	Reserved

Length This field shall be set to the length, in units of octets, of SystemTypeSpecificFields.

SystemTypeSpecificFields

This field shall be set according to the value of SystemType as specified below.

3.1.1 SystemTypeSpecificFields when SystemType is 0x00

⁶ SystemType of 0x01 applies to [16] and all of its predecessors.

Field	Length (bits)
BandClass	8
ChannelNumber	16
HalfDuplexIncluded	1
HalfDuplexSupported	0 or 1
ReverseChannelIncluded	0 or 1
ReverseChannelBandClass	0 or 8
ReverseChannelNumber	0 or 16
CyclicPrefixIncluded	1
CyclicPrefixLength	0 or 2
FFTSIZEIncluded	1
FFTSIZE	0 or 3
NumGuardSubcarriersIncluded	1
NumGuardSubcarriers	0 or 6
Reserved	Variable

1

2 BandClass This field shall be set to the band class number corresponding to the
3 frequency assignment of the forward channel specified by this record.

4 ChannelNumber This field shall be set to the channel number corresponding to the
5 frequency assignment of the forward channel specified by this record.

6 HalfDuplexIncluded This field shall be set to '1' if the HalfDuplexSupported and
7 ReverseChannelIncluded fields are included in this record.

8 HalfDuplexSupported
9 This field shall be omitted if HalfDuplexIncluded is equal to '0'.
10 Otherwise, this field shall be set to '1' if half-duplex operation is
11 supported on this channel, and to '0' if half-duplex operation is not
12 supported on this channel.

13 ReverseChannelIncluded
14 This field shall be omitted if HalfDuplexIncluded is set to '0'.
15 Otherwise, this field shall be included and set as follows:

16 This field shall be set to '1' if ReverseChannelBandClass and
17 ReverseChannelNumber fields are included. Otherwise, this field
18 shall be set to '0'. If this field is set to '0', the reverse channel is as
19 specified by the frequency division duplex pair reverse channel
20 corresponding to the forward channel.

- 1 ReverseChannelBandClass
2 This field shall be omitted if ReverseChannelIncluded is omitted or is
3 set to '0'. Otherwise, this field shall be set to the BandClass of the
4 reverse channel.
- 5 ReverseChannelNumber
6 This field shall be omitted if ReverseChannelIncluded is omitted or is
7 set to '0'. Otherwise, this field shall be set to the channel number of
8 the reverse channel.
- 9 CyclicPrefixIncluded This field shall determine if the length of the cyclic prefix is included
10 in this record.
- 11 CyclicPrefixLength This field shall be omitted if CyclicPrefixIncluded is equal to '0'.
12 Otherwise, this field shall be set to one less than the CyclicPrefix of
13 the forward channel. The interpretation of this field shall be as
14 defined by the ~~Overhead Messages Protocol~~Physical Layer.
- 15 FFTSizeIncluded This field shall determine if the FFT size is included in this record.
- 16 FFTSize This field shall be omitted if FFTSizeIncluded is equal to '0'.
17 Otherwise, this field shall be set to $\log_2(N_{FFT}/128)$.
- 18 NumGuardSubcarriersIncluded
19 This field shall determine if the number of guard carriers is included
20 in this record.
- 21 NumGuardSubcarriers
22 This field shall be omitted if NumGuardSubCarriersIncluded is equal
23 to '0'. Otherwise, this field shall be set to the NumGuardSubCarriers
24 of the forward channel. The interpretation of this field shall be as
25 defined by the Overhead Messages Protocol.
- 26 Reserved The length of this field shall be set to octet align this record. This
27 field shall be set to zero.

28 3.1.2 SystemTypeSpecificFields when SystemType is 0x01, 0x02, or 0x03
29

Field	Length (bits)
BandClass	5
ChannelNumber	11

- 30 BandClass If the SystemType field is set to 0x01 or 0x02, the access network
31 shall set this field to the band class number corresponding to the
32 frequency assignment of the channel specified by this record for both
33 the forward channel and the reverse channel. If the SystemType is

1 set to 0x03, then access network shall set this field to the band class
 2 number corresponding to the frequency assignment of the channel
 3 specified by this record for the forward channel only.

4 ChannelNumber If the SystemType is set to 0x01 or 0x02, the access network shall set
 5 this field to the channel number corresponding to the frequency
 6 assignment of the channel specified by this record for both the
 7 forward channel and the reverse channel. If the SystemType is set
 8 to 0x03, this access network shall set this field to the channel
 9 number corresponding to the frequency assignment of the channel
 10 specified by this record for the forward channel only.

11 3.2 Neighbor Technology Record

12 The Neighbor Technology Record defines an parameters for a neighboring access network
 13 belonging to another technology.

14 If TechnologyType⁷ is 0x00 (CDMA2000 1x), then this record contains the following fields:
 15

Field	Length
ChannelRecord	24
SIDNIDIncluded	1
SID	0 or 15
NID	0 or 16
PacketZoneIDIncluded	1
PacketZoneID	0 or 8
Reserved	Variable

16 ChannelRecord This field shall be set to the ChannelRecord for the CDMA2000 1x
 17 system, as specified in the CDMA2000 1x specification.

18 SIDNIDIncluded This field shall determine if the SID and NID fields are included.
 19

20 SID The access network shall omit this field if SIDNIDIncluded is '0'.
 21 Otherwise, the field shall be set to the System Identifier for the
 22 CDMA2000 1x system.

⁷ [Neighbor Technology Record is described in a TechnologyType-NeighborTechnologyRecordLength-NeighborTechnologyRecord format. The fields of the Neighbor Technology Record are dependent on the associated TechnologyType. For example, see inclusion of this record in the SectorParameters message \[7\].](#)

- 1 NID The access network shall omit this field if SIDNIDIncluded is '0'.
 2 Otherwise, the field shall be set to the Network Identifier for the
 3 CDMA2000 1x system. The access network may set this field to all
 4 ones to indicate an unspecified NID value.
- 5 PacketZoneIDIncluded
 6 This field shall determine if a PacketZoneID is included.
- 7 PacketZoneID The access network shall omit this field if PacketZoneIDIncluded is
 8 '0'. Otherwise, this field shall be set to the packet zone ID for the
 9 CDMA2000 1x system.
- 10 Reserved The length of this field shall be set to octet align this record. This
 11 field shall be set to zero. The access terminal shall ignore this field.
- 12 If TechnologyType is 0x01 (CDMA2000 High Rate Packet Data), then this record contains
 13 the following fields:
 14

Field	Length
ChannelRecord	24
SubnetLength	8
Subnet	SubnetLength
Reserved	Variable

- 15 ChannelRecord This field shall be set to the ChannelRecord for the HRPD system, [as](#)
 16 [specified in \[17\]](#).
- 17 SubnetLength This field shall be set to the length of the Subnet of the HRPD
 18 system.
- 19 Subnet This field shall be omitted if SubnetLength is zero. Otherwise, this
 20 field shall be set to the Subnet for the HRPD system.
- 21 Reserved This field shall be set to zero. The access terminal shall ignore this
 22 field.

23 3.3 Access Terminal Identifier Record

- 24 The Access Terminal Identifier record provides a unicast, multicast, or broadcast access
 25 terminal address. This record contains the following fields:
 26

Field	Length (bits)
ATIType	2
ATI	0 or 128

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

3 **Table 3-2. ATIType Field Encoding**

ATIType	ATIType Description	ATI Length (bits)
'00'	Broadcast ATI	0
'01'	Multicast ATI	128
'10'	Unicast ATI	128
'11'	Random ATI (RATI)	128

4 ATI Access Terminal Identifier. This field shall be set as shown in Table
5 3-2.

6 **3.4 Attribute Record**

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

10 An attribute can be one of the following three types:

- 11 • Simple attribute, if it contains a single value,
- 12 • Attribute list, if it contains multiple single values which are to be interpreted as
13 different suggested values for the same attribute identifier (e.g., a list of possible
14 protocol Subtypes for the same protocol Type), or
- 15 • Complex attribute, if it contains multiple values that together form a complex value for
16 a particular attribute identifier.

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

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

19 The sender of a ConfigurationResponse message or FastConfigurationAccept message (see
20 Session Control Protocol) selects an attribute-value from a ConfigurationRequest message
21 or FastConfigurationRequest message respectively by sending the attribute value if it is a
22 simple attribute or a selected value out of an attribute list. Selection of complex-attributes
23 is done by sending the value identifier which identifies the complex value.

24 The format of a simple attribute and attribute list is given by
25

Field	Length (bits)
AttributeID	16
NumValues	48
Reserved	4

~~One or more~~[NumValues](#) instances of the following record

LengthLength	1
Length	7 or 15
AttributeValue	8 × Length

1 AttributeID The sender shall set this field to the attribute identifier of this
2 attribute.

3 [NumValues](#) The sender shall set this field to the number of instances of
4 [AttributeValue](#) in this attribute record.

5 [Reserved](#) The sender shall set this field to zero.

6 LengthLength This field shall be set to '0' if the length of the Length field is seven
7 bits. Otherwise, this field shall be set to '1'.

8 Length This field shall be set to the length of the AttributeValue field in units
9 of octets.

10 AttributeValue A suggested value for the attribute. Attribute value lengths are, in
11 general, an integer number of octets.

12 The format of a complex attribute is given by

13

Field	Length (bits)
AttributeID	16
NumValues	48
Reserved	4

~~One or more~~[NumValues](#) instances of the following fields

LengthLength	1
Length	7 or 15
ValueID	8
AttributeValue	8 × [Length - 1]

14 AttributeID The sender shall set this field to the attribute identifier of this
15 attribute.

1	<u>NumValues</u>	<u>The sender shall set this field to the number of instances of</u>
2		<u>AttributeValue in this attribute record.</u>
3	<u>Reserved</u>	<u>The sender shall set this field to zero.</u>
4	LengthLength	This field shall be set to '0' if the length of the Length field is seven
5		bits. Otherwise, this field shall be set to '1'.
6	Length	This field shall be set to the length of the ValueID field plus the
7		length of the AttributeValue field in units of octets.
8	ValueID	It identifies the set of attribute values following this field. The sender
9		shall increment this field for each new set of values for this complex
10		attribute.
11	AttributeValue	A suggested value for the attribute. Attribute value lengths are in
12		general an integer number of octets.

3.5 Non-attribute Data Record

Non-attribute data could be included in a Session State Information Record. Format of the non-attribute data is as follows:

Field	Length (bits)
DataID	16
DataValue	Data dependent
Reserved	Variable

17	DataID	The sender shall set this field to the non-attribute data identifier of
18		this data.
19	DataValue	A value for the data. Data value lengths are in general an integer
20		number of octets.
21	Reserved	The length of this field is the smallest value that will make the Data
22		record octet aligned. The sender shall set this field to zero. The
23		receiver shall ignore this field.

3.6 Session State Information Record

The Session State Information is to be used for transferring the session parameters from a source access network to a target access network. Session parameters are the attributes of the Personalities and the internal parameters that define the state of each InUse instance. 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

1 shall include all the Parameter Records associated a (PersonalityID, ProtocolType) pair in
 2 the same Session State Information Record.

3 **Table 3-3. The Format of the Session State Information Record**

Field	Length (bits)
FormatID	8
PersonalityID	4
Reserved	4
ProtocolSetIdentifier	16
ProtocolType	8or 16
ProtocolSubtype	8

ZeroOne or more instances of the following
 Parameter Record:

ParameterType	8
Length	16
ParameterType-specific record	$8 \times \text{Length}$

- 4 FormatID This field identifies the format of the rest of the fields in this record
 5 and shall be set to zero.
- 6 PersonalityID This field shall be set to the identifier associated with the Personality
 7 whose Session State Information is being conveyed.
- 8 Reserved This field shall be set to zero.
- 9 ProtocolSetIdentifier This field shall be set to the Protocol Set Identifier associated with
 10 this Personality.
- 11 ProtocolType If length of this field is 8 bits, then the most significant bit of this
 12 field shall be set to '0'; otherwise, the most significant bit of this field
 13 shall be set to '1'. This field shall be set to the protocol to which this
 14 attribute belongs.
- 15 ProtocolSubtype This field shall be set to the protocol subtype value (see Table 4-1
 16 and Table 4-2) for the protocol associated with the encapsulated
 17 session parameters.
- 18 ParameterType This field shall be set according to Table 3-4.

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 3.4. 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

Length Length in octets of the ParameterType-specific-record following this field.

ParameterType-specific record

If the ParameterType field is set to 0x00, then this record shall be set to the simple or complex attribute (see 3.4) 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 The Protocol Type may be 8 or 16 bits long. If a Protocol Type is 8 bits long, its MSB shall
3 be set to '0'. Otherwise, its MSB shall be set to '1'.

4 Following rules are used for Protocol Type assignment:

- 5 • '0'|'ppppppp' is used as Protocol Type for InUse instance of non-Application Layer
6 protocols, where 'ppppppp' are seven binary digits.
- 7 • Protocol Types described by '100'|'pppppppppppppp' where 'pppppppppppppp' are
8 thirteen binary digits is reserved.
- 9 • '110'|'000000'|'ppppppp' bits is used as Protocol Type for InConfiguration instance of
10 non-Application Layer protocols, and 'ppppppp' are seven binary digits. The same value
11 of 'ppppppp' is used for the InUse and InConfiguration instance of any non-Application
12 Layer protocol.
- 13 • '101'|'ttttt'|'aaaaaaaa' is used as Protocol Type for InUse instance of Application Layer
14 protocols, where 'ttttt' is are five binary digits, and 'aaaaaaaa' are eight binary digits
15 representing the Application Layer Protocol ID as specified in Table 4-3.
- 16 • '111'|'ttttt'|'aaaaaaaa' is used as Protocol Type for InConfiguration instance of
17 Application Layer protocols, where 'ttttt' is are five binary digits, and 'aaaaaaaa' are
18 eight binary digits representing the Application Layer Protocol ID as specified in Table
19 4-3. The same value of 'ttttt' is used for the InUse and InConfiguration instance of a
20 given Application Layer protocol instantiation.

1

Table 4-1. Protocol Type and Subtype for InUse Instance

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Physical Layer	0x00	8	Basic Physical Layer	0x00
Packet Consolidation	0x01	8	Basic Packet Consolidation	0x00
Superframe Preamble MAC	0x02	8	Basic Superframe Preamble MAC	0x00
Access Channel MAC	0x03	8	Basic Access Channel MAC	0x00
Forward Link Control Segment MAC	0x04	8	Basic Forward Link Control Segment MAC	0x00
Forward Traffic Channel MAC	0x05	8	Basic Forward Traffic Channel MAC	0x00
Reverse Traffic Channel MAC	0x06	8	Basic Reverse Traffic Channel MAC	0x00
Reverse Control Channel MAC	0x07	8	Basic Reverse Control Channel MAC	0x00
Route	0x09	8	Basic Route	0x00
Stream	0x0a	8	Basic Stream	0x00
Radio Link for Stream NN ⁸	0x40 + NN	8	Basic Radio Link	0x00
QoS Management	0x0b	8	Basic QoS Management	0x00
Application	0xa0PP to 0xbfPP ⁹	16	NA	NA
Key Exchange	0x0c	8	Basic Key Exchange	0x00
Ciphering	0x0d	8	AES Ciphering	0x00
Message Integrity Protection	0x0e	8	Basic Message Integrity Protection	0x00
Air Link Management	0x0f	8	Basic Air Link Management	0x00
Initialization State	0x10	8	Basic Initialization State	0x00
Idle State	0x11	8	Basic Idle State	0x00
Connected State	0x12	8	Basic Connected State	0x00
Active Set Management	0x13	8	Basic Active Set Management	0x00

⁸ NN is the two-digit hexadecimal Stream number in the range 0x00 to 0x1f

⁹ PP is the two-digit hexadecimal Application Layer ProtocolID in the range 0x00 to 0xff.

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Overhead Messages	0x14	8	Basic Overhead Messages	0x00
Session Control	0x15	8	Basic Session Control	0x00
Route Control	0x17	8	Basic Route Control	0x00
BCMCS Protocol Suite	0x18	8	Basic BCMCS Protocol Suite	0x00

1

Table 4-2. Protocol Type and Subtype for InConfiguration Instance

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Physical Layer	0xc000	16	Basic Physical Layer	0x00
Packet Consolidation	0xc001	16	Basic Packet Consolidation	0x00
Superframe Preamble MAC	0xc002	16	Basic Superframe Preamble MAC	0x00
Access Channel MAC	0xc003	16	Basic Access Channel MAC	0x00
Forward Link Control Segment MAC	0xc004	16	Basic Forward Link Control Segment MAC	0x00
Forward Traffic Channel MAC	0xc005	16	Basic Forward Traffic Channel MAC	0x00
Reverse Traffic Channel MAC	0xc006	16	Basic Reverse Traffic Channel MAC	0x00
Reverse Control Channel MAC	0xc007	16	Basic Reverse Control Channel MAC	0x00
Route	0xc009	16	Basic Route	0x00
Stream	0xc00a	16	Basic Stream	0x00
Radio Link for Stream NN ¹⁰	0xc040 + NN	16	Basic Radio Link	0x00
QoS Management	0xc00b	16	Basic QoS Management	0x00
Application	0xeOPP to 0xffPP 11	16	NA	NA

¹⁰ NN is the two-digit hexadecimal Stream number in the range 0x00 to 0x1f

¹¹ PP is the two-digit hexadecimal Application Layer ProtocolID in the range 0x00 to 0xff.

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Key Exchange	0xc00c	16	Basic Key Exchange	0x00
Ciphering	0xc00d	16	AES Ciphering	0x00
Message Integrity Protection	0xc00e	16	Basic Message Integrity Protection	0x00
Air Link Management	0xc00f	16	Basic Air Link Management	0x00
Initialization State	0xc010	16	Basic Initialization State	0x00
Idle State	0xc011	16	Basic Idle State	0x00
Connected State	0xc012	16	Basic Connected State	0x00
Active Set Management	0xc013	16	Basic Active Set Management	0x00
Overhead Messages	0xc014	16	Basic Overhead Messages	0x00
Session Control	0xc015	16	Basic Session Control	0x00
Route Control	0xc017	16	Basic Route Control	0x00
BCMCS Protocol Suite	0xc018	16	Basic BCMCS Protocol Suite	0x00

1 **Table 4-3. Application Layer Protocol ID**

Value	Application Layer Protocol
0x00	NULL
0x01	Reserved
0x02	Reserved
0x03	Reserved
0x04	RoHC Support Protocol
0x05	Internet Protocol (IP) version 4 as defined in [7] and version 6 as defined in [14]
0x06	Basic Signaling Protocol
0x07	Basic Inter-Route Tunneling Protocol
0x08	EAP Support Protocol
<u>0x09</u>	<u>UMB Test Application</u>

2

1

Table 4-4. Initial Protocol Set Identifiers

IPSI Value	Protocol Stack
0x0	Protocol stack with all other protocols with subtype 0x00.
All other values	Reserved

2

Table 4-5. BCMCS Initial Protocol Set Identifiers

BCMCS IPSI Value	Protocol Stack
0x0	Protocol stack consisting of BCMCS Protocol Suite with subtype 0x00.
All other values	Reserved

3

4

Table 4-6. Protocol Set Identifier

Value	Protocol Stack
0x0000	Protocol stack all protocols with subtype 0x00.
0xZff0 – 0xZfff, where Z is any value of IPSI	Reserved for vendor-specific assignment.
All other values	Reserved

5

- 1 No text.

5 ANID, SECTORID, AND UATI PROVISIONING

The access network shall follow the procedures in this section to ensure that the ANID, SectorID, and UATI are globally unique. Unless specified otherwise, the access terminal shall not assume any structure in the ANID, SectorID, and UATI.

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:

- 1 • 0x0000000000000000, or
- 2 • 0x0000000000000001, or
- 3 • 64 LSBs of another SectorID with the same 64 MSBs as this UATI, or
- 4 • 64 LSBs of another UATI with the same 64 MSBs as this UATI.
- 5