

Application Program Interface Supplement
to the
Software Communications Architecture Specification

APPENDIX F

**Media Access Control
Building Block Service Definition**

Revision Summary

1.0	Initial Release
2.2.1	Document numbering change for consistency with SCA main document numbering. Incorporate approved Change Proposals, numbers 46, 47
3.0	No change.

Table of Contents

F.1	INTRODUCTION.....	F-1
F.1.1	OVERVIEW.....	F-1
F.1.2	SERVICE LAYER DESCRIPTION.....	F-1
F.1.3	MODES OF SERVICE.....	F-2
F.1.4	SERVICE STATES.....	F-2
F.1.5	REFERENCED DOCUMENTS.....	F-3
F.2	UUID.....	F-3
F.3	SERVICES.....	F-4
F.3.1	MAC COMMON UTILITY BUILDING BLOCK.....	F-5
F.3.1.1	Activate Channel Service.....	F-5
F.3.1.2	Set RF Power Control.....	F-5
F.3.1.3	Minimum Transmission Unit Attribute.....	F-6
F.3.1.4	Maximum Transmission Unit Attribute.....	F-6
F.3.1.5	MAC Common Utility States.....	F-7
F.3.2	TRANSEC BUILDING BLOCK.....	F-7
F.3.2.1	Load Fill Service.....	F-8
F.3.2.2	Load Fill ID.....	F-8
F.3.2.3	Read Fill ID Service.....	F-8
F.3.2.4	Load Seed Service.....	F-9
F.3.2.5	Read Seed Service.....	F-10
F.3.2.6	Zeroize Service.....	F-10
F.3.2.7	TRANSEC Service Group States.....	F-11
F.3.3	CHANNEL ERROR CONTROL BUILDING BLOCK.....	F-13
F.3.3.1	Set Channel Error Control Service.....	F-13
F.3.3.2	Set Channel Error Control Service States.....	F-14
F.3.4	CHANNEL ACCESS BUILDING BLOCK.....	F-14
F.3.4.1	Set Channel Access Parameters Service.....	F-14
F.3.4.2	Channel Access States.....	F-15
F.3.5	MAC ADDRESS BUILDING BLOCK.....	F-16
F.3.5.1	Bind Address Service.....	F-17
F.3.5.2	MAC Bind Address Service States.....	F-17
RX TERMINATION BUILDING BLOCK.....		F-18
Terminate Reception Service.....		F-19
Terminate ReceptionService State Influence.....		F-19
F.3.7	QUALITY OF SERVICE (QOS) BUILDING BLOCK.....	F-20
F.3.7.1	Read QOS Service.....	F-21
F.3.7.2	Set QOS Parameters Service.....	F-21
F.3.7.3	Enable QOS Service.....	F-21
F.3.7.4	Disable QOS Service.....	F-21
F.3.7.5	Quality of Service States.....	F-21
F.3.8	SIMPLE PACKET BB (INHERITED).....	F-22
F.3.8.1	Parameters Filled in by Inheriting Class.....	F-22

F.3.8.2	Attributes.....	F-23
F.3.9	EXAMPLES.....	F-23
F.3.9.1	MAC Common Utility BB.....	F-24
F.3.9.2	TRANSEC BB.....	F-25
F.3.9.3	Channel Error Control BB.....	F-25
F.3.9.4	Channel Access BB.....	F-27
F.3.9.5	MAC Addressing BB.....	F-28
Rx Termination.....		F-28
F.3.9.7	Quality of Service BB.....	F-29
F.4	SERVICE PRIMITIVES.....	F-29
F.4.1	MAC COMMON UTILITY SERVICE PRIMITIVES.....	F-29
F.4.1.1	Activate Channel Service.....	F-29
F.4.1.2	Set RF Power Control Service.....	F-30
F.4.1.3	Minimum Transmission Unit Length Attribute.....	F-31
F.4.1.4	Maximum Transmission Unit Length Attribute.....	F-31
F.4.2	TRANSEC SERVICE PRIMITIVES.....	F-32
F.4.2.1	Load Fill Service.....	F-32
F.4.2.2	Load Fill ID Service.....	F-32
F.4.2.3	Read Fill ID Service.....	F-33
F.4.2.4	Load Seed Service.....	F-34
F.4.2.5	Read Seed Service.....	F-34
F.4.2.6	Zeroize Service.....	F-35
F.4.3	CHANNEL ERROR CONTROL SERVICE PRIMITIVES.....	F-36
F.4.3.1	Set Channel Error Control Service.....	F-36
F.4.4	CHANNEL ACCESS SERVICE PRIMITIVES.....	F-36
F.4.4.1	Set Access Parameters Service.....	F-36
F.4.5	MAC ADDRESS PRIMITIVES.....	F-37
F.4.5.1	Register Address Service.....	F-37
TERMINATE RECEPTION SERVICE PRIMITIVES.....		F-38
Terminate Reception Service.....		F-38
F.4.7	QUALITY OF SERVICE(QOS) PRIMITIVES.....	F-38
F.4.7.1	Read QOS Service.....	F-38
F.4.7.2	Set QOS Parameters Service.....	F-39
F.4.7.3	Enable QOS Service.....	F-39
F.4.7.4	Disable QOS Service.....	F-40
F.5	ALLOWABLE SEQUENCE OF SERVICE PRIMITIVES.....	F-40
F.6	UTILIZATION OF MAC BUILDING BLOCKS.....	F-41
F.7	PRECEDENCE OF SERVICE PRIMITIVES.....	F-42
F.8	SERVICE USER GUIDELINES.....	F-42
F.9	SERVICE PROVIDER-SPECIFIC INFORMATION.....	F-42

F.10 IDL..... F-42

F.11 UML..... F-42

List of Figures

Figure 1. JTRS Service Layers..... F-2

Figure 2. MAC Common Utility Building Block F-5

Figure 3. Sequence Diagram, MAC Common BB..... F-6

Figure 4. State Diagram, MAC Utility BB..... F-7

Figure 5. TRANSEC Building Block..... F-8

Figure 6. Sequence Diagram, loadFill and readFillID F-9

Figure 7. Sequence Diagram, loadSeed and readSeed F-10

Figure 8. Sequence Diagram, zeroize..... F-11

Figure 9. State Diagram, TRANSEC BB Fill and Seed..... F-11

Figure 10. State Diagram, TRANSEC Stream..... F-12

Figure 11. Channel Error Control Building Block..... F-13

Figure 12. Sequence Diagram, setChannelErrorControl..... F-13

Figure 13. State Diagram, Set Channel Error Control F-14

Figure 14. Channel Access Building Block F-14

Figure 15. Sequence Diagrams, setChannelAccessParameters..... F-15

Figure 16. State Diagram, Channel Access..... F-16

Figure 17. MAC Address Building Block..... F-16

Figure 18. Sequence Diagram, bindAddress..... F-17

Figure 19. State Diagram, MAC Address BB..... F-18

Figure 20. Rx Termination Building Block F-19

Figure 21. Sequence Diagram, Terminate Reception F-19

Figure 22. State Diagram, Terminate Reception BB F-20

Figure 23. Quality of Service Building Block..... F-21

Figure 24. Sequence Diagram, QOS BB..... F-21

Figure 25. State Diagram, Quality of Service BB..... F-22

List of Tables

Table 1. Cross-Reference of Services and Primitives F-4

F.1 INTRODUCTION.

F.1.1 Overview.

Media Access Control (MAC) application-program interfaces (APIs) provide standardized interfaces to the MAC layer. MAC Services are grouped into Service Groups or Building Blocks (BBs) to define abstract services to foster software reuse and commonality between different Service Definition implementations. MAC Services provide Service Users with methods to send non-real-time control and data between software resources and methods to signal the Service User that an event has occurred. Real-time control and signals are communicated via the packet interface. The MAC API is intended to support at least the following Waveform APIs:

- Demand Assigned Single Access/Demand Assigned Multiple Access (DASA/DAMA)
- Have Quick
- HF Automate Link Establishment (ALE)
- Line Of Sight (LOS)
- SINCGARS
- VRC-99
- Wideband Data Waveform (WDW)

Location of the MAC layer with respect to other layers is shown in Figure 1.

Providing a common service definition for each JTRS Service Layer is attainable for all layers above the MAC. However, due to the complexity and variety of waveforms, defining a single MAC API capable of satisfying all waveform requirements would result in significant processing and memory inefficiencies. For these reasons, MAC Building Blocks are abstract classes, which must be used with concrete classes that are waveform-specific. A concrete class is one that may have direct instances.

F.1.2 Service Layer Description.

The primary Service used by the MAC is the Physical layer. The MAC Service Layer is subdivided into the following Service Groups/BBs:

- MAC Common Utility
- TRANSEC
- Channel Error Control
- Channel Access
- MAC Address
- Rx Termination
- Quality of Service (QOS)
- Packet (Provided by the Packet BB)

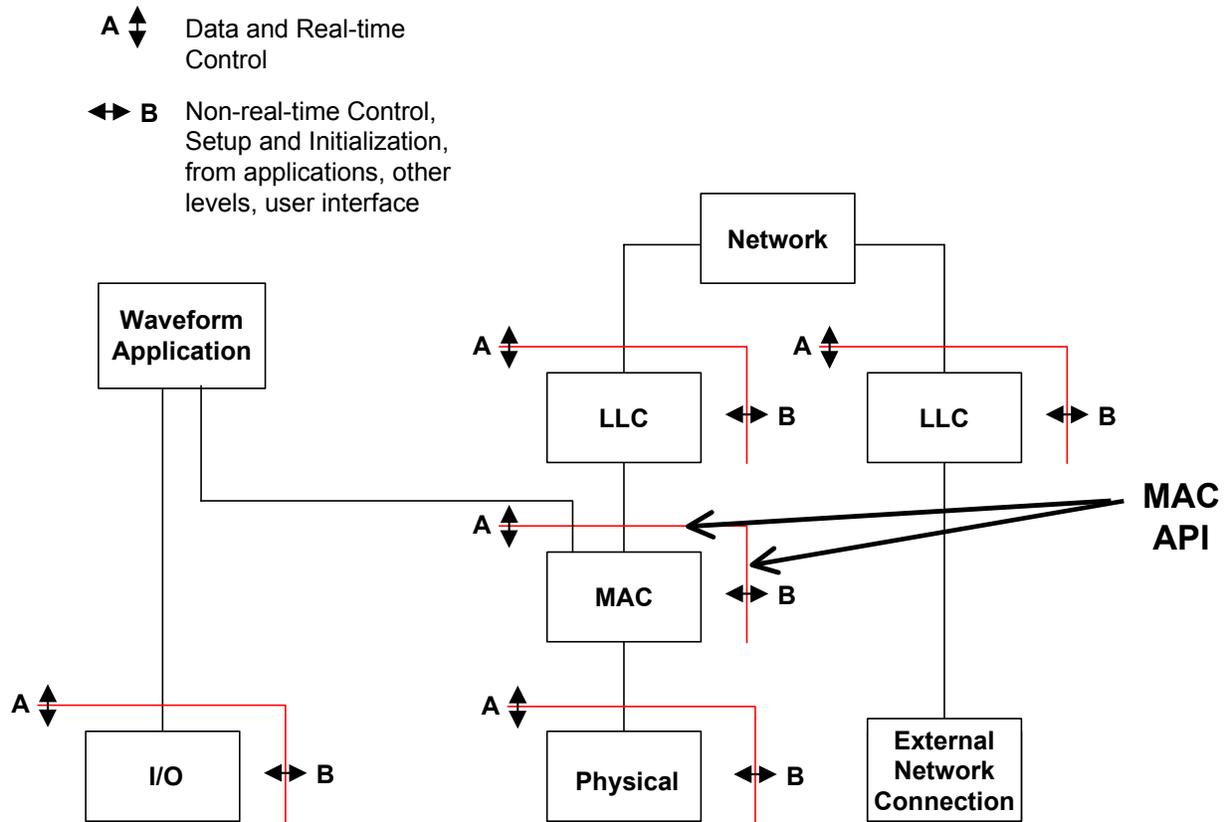


Figure 1. JTRS Service Layers

Each BB is an abstract Service Definition that must be instantiated with a concrete type to be realized. MAC layer concrete types are waveform-specific. Therefore, when a BB is realized, its use is limited to a specific waveform (e.g., SINCGARS or HF ALE). Some Waveform (WF) APIs can use all BBs while others will only need one or two. The relationships between BBs and WF APIs are illustrated in Section 6, Utilization of MAC Building Blocks.

F.1.3 Modes of Service.

There are no specific Modes of Service. This document describes a set of SCA building blocks that provide a generalization of the MAC layer interface. To form a complete service layer or application-programming interface, several MAC building blocks may need to be combined. (See section 6) When used in this manner, the BB combination may be considered as constituting a MAC layer control-data service available to other layers of the SCA model.

MAC mode services are operation-oriented and support control-data transfer in self-contained units with no logical relationships required between BBs.

F.1.4 Service States.

Each building block realization is an instantiation of a parameterized building block, which subsumes the states enumerated in a waveform-specific concrete building block.

F.1.5 Referenced Documents.

JTRS-5000SCA, “Software Communications Architecture Specification (SCA)”, V2.2.1, April 30, 2004
Interface Specification for Mode 2 and Mode 3 Fill, A3191133, Rev -, 94/03/10
SCA Service Definition Description for Packet Building Blocks

F.2 UUID.

Not applicable.

F.3 SERVICES.

A cross-reference of Services and Primitives provided by the MAC Layer Building Blocks is shown in the following table and described fully in the remainder of this section.

Table 1. Cross-Reference of Services and Primitives

Service Group	Service	Primitives
MAC Common Utility BB	Activate Channel	activateChannel (PresetNum : in short) : boolean
	Set RF Power Control	setRFPowerControl (PowerMode : in PowerModeType) : boolean
	Minimum Transmission Unit Length	readonly attribute in unsigned short MinTU
	Maximum Transmission Unit Length	readonly attribute in unsigned short MaxTU
TRANSEC BB	Load Fill	loadFill (presetNum : in short, Fill : in FillType, FillInfo : in FillInfoType) : boolean
	Load Fill ID	LoadFillID (PresetNum : in short, Fill : in FillType, FillID : in FillIDType) boolean
	Read Fill ID	readFillID (PresetNum : in short, Fill : in FillType, FillID : out FillIDType) : boolean
	Load Seed	loadSeed (SeedNum : in short, Seed : in SeedType, SeedInfo : in SeedInfoType) : boolean
	Read Seed	readSeed (SeedNum : in short, Seed : in SeedType, SeedInfo : out SeedInfoType) : boolean
	Zeroize Fill/Seed	zeroize () : boolean
Channel Error Control BB	Enable/Disable Error Control	setChannelErrorControl (ErrorControl : in ErrorControlType) : void
Channel Access BB	Set Access Parameters	setAccessParameters(Parameters : in AccessParameterType) : boolean
MAC Address BB	Bind Own Address	bindAddress (address : in AddressType) : boolean
Rx Termination BB	Enable/Disable	terminateReception () : boolean
QOS BB	Read QOS	readQOS (QOS : in QOSType) : void
	Set QOS Parameters	setQOSParameters (QOSParameters ; in QOSType) : void
	Enable QOS	enableQOS () : void

Service Group	Service	Primitives
	Disable QOS	disableQOS () : void

F.3.1 MAC Common Utility Building Block.

The MAC Common Utility Service Group provides generalized interfaces to the MAC layer for methods required by more than one MAC Building Block. For a typical radio, interfaces could include:

- active or selected channel – e.g., Channel 1, Emergency, Guard, Primary, etc; and
- RF power level – e.g., Hi, Med, Lo, 10 dBm, Max, Min, etc.

To accommodate a variety of descriptors, these interface parameters are enumerated types.

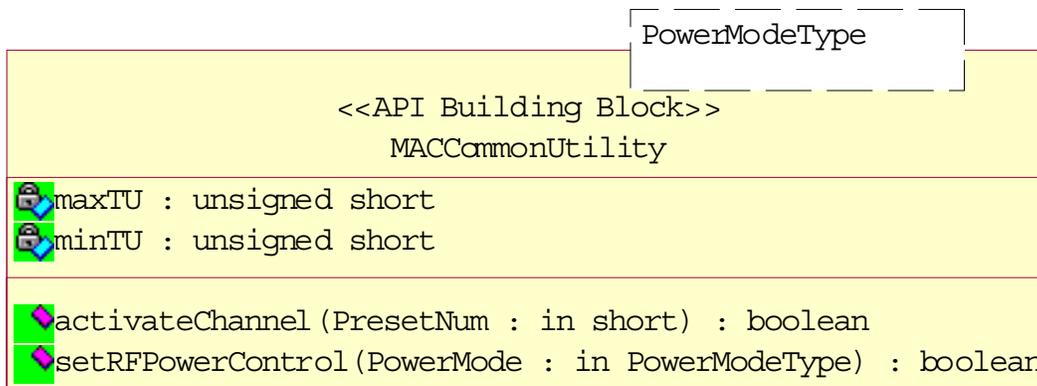


Figure 2. MAC Common Utility Building Block

F.3.1.1 Activate Channel Service.

The Activate Channel Service provides Service Users with a method to communicate to the MAC Layer the active channel (e.g., 1, 2, Guard, Emergency, etc.).

Typically, this service would be used by the Human Control Interface to allow a radio operator to designate a Preset Channel to be the active or operational channel. When the PresetNumber specified by the Service User is a channel number known by the Common Utility BB and the channel designated by the PresetNumber has been activated, the returned Boolean is set to True to indicate the channel has been Activated. When the Boolean is returned set to False, it indicates the PresetNumber was Unknown and/or the channel was not activated. Valid Channel Numbers may range from 0 to $2^{16} - 1$.

F.3.1.2 Set RF Power Control.

The Set RF Power Control Service provides Service Users with a method to communicate to the MAC Layer the selected RF Power Control mode (e.g., Auto, Lo, Med, Hi, Max, etc.).

Typically, this service would be used by the Human Control Interface to allow a radio operator to designate the RF power mode for the operational RF Channel. When the RFPowerMode specified by the Service User is a power mode known by the Common Utility BB and the power mode has been set to the selected mode, the Boolean returned by the method is set to True.

Otherwise, the Boolean is set to False to indicate the Power RF Mode Unknown and/or the power mode was not set to the selected value (i.e., operation failed). Power Mode is an enumerated type that is specific to a given waveform API.

Power mode can be used to designate a specific level of RF power output or to designate the RF power level will be controlled automatically by algorithms contained within the MAC Layer or within data packets.

F.3.1.3 Minimum Transmission Unit Attribute.

This attribute provides Service Users with the minimum Transmission Unit (TU) length the Service Provider will accept as being a valid packet. Min TU is declared by the inheriting API.

F.3.1.4 Maximum Transmission Unit Attribute.

This attribute provides Service Users with the maximum Transmission Unit (TU) length the Service Provider will accept as being a valid packet. MaxTU is declared by the inheriting API.

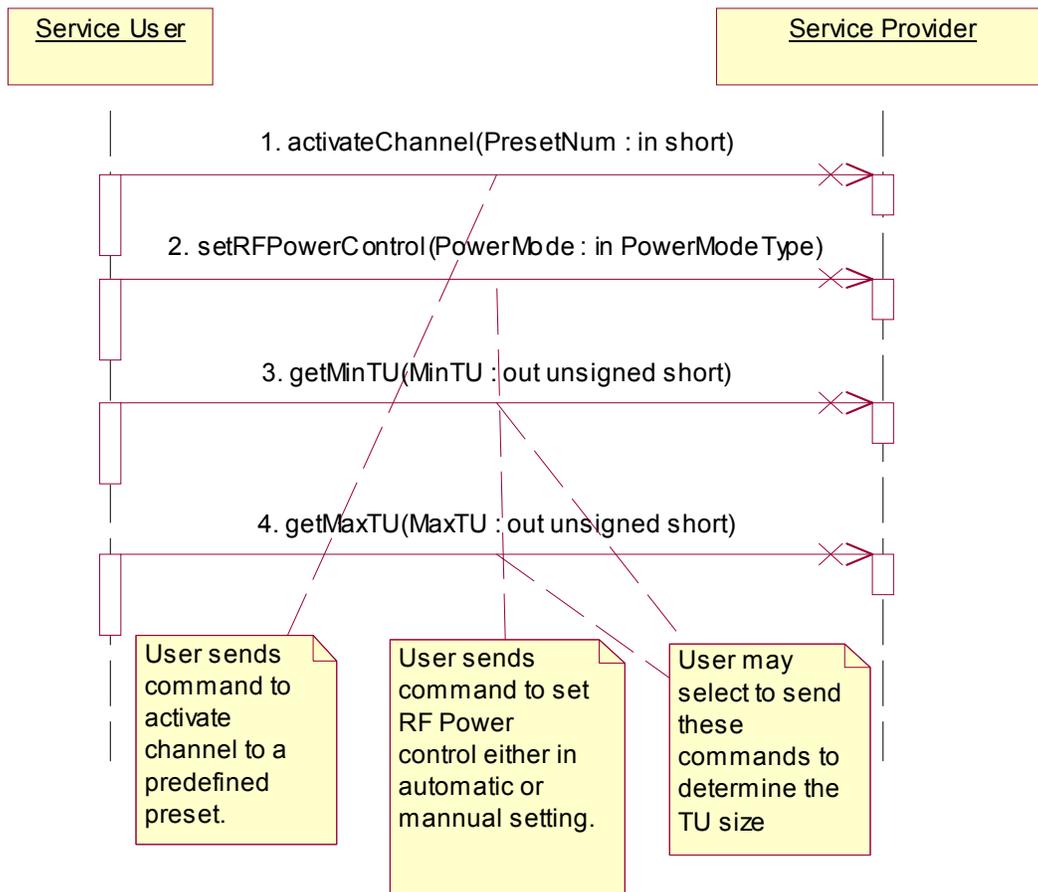


Figure 3. Sequence Diagram, MAC Common BB

F.3.1.5 MAC Common Utility States.

MAC Common Utility BB states and state transitions that occur as a result of the Service User setting RF power level and/or activating a channel are depicted in the following state diagram.

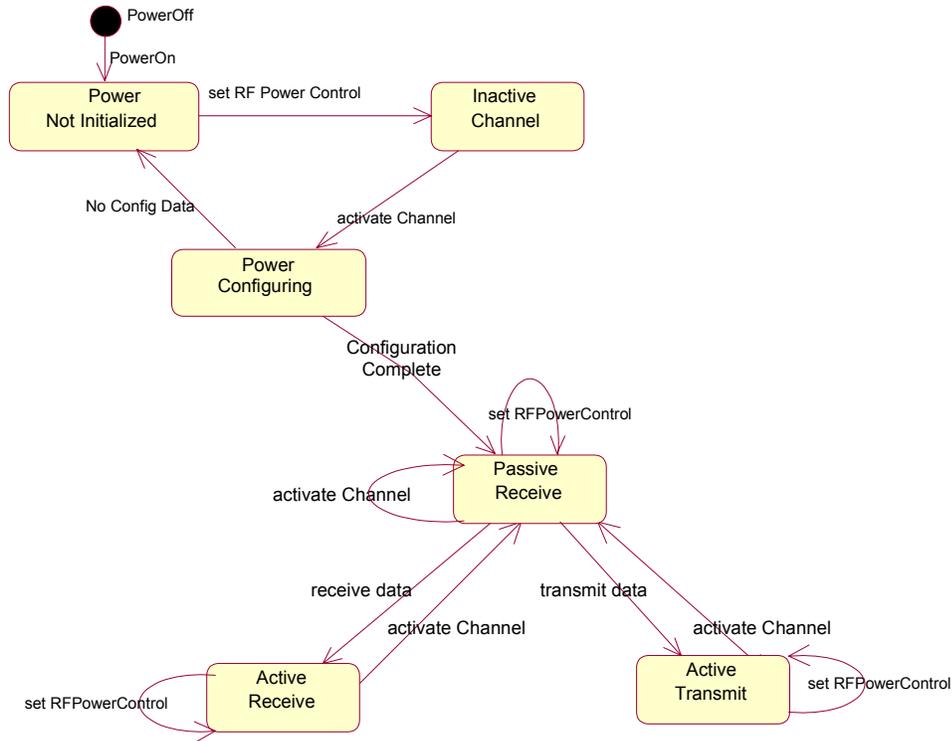


Figure 4. State Diagram, MAC Utility BB

F.3.2 TRANSEC Building Block.

The Transmission Security (TRANSEC) Service Group provides a generalized interface to the MAC layer for non-Type 1 TRANSEC functions, which may reside in either the Red or Black security domain. TRANSEC services vary as a function of the type of over-the-air waveform being realized. Therefore, this Building Block must be instantiated with a waveform-specific base class. The BB is shown as a Template Class, in the following figure. This is to accommodate waveform unique differences in the data formats of Fill and Seed data.

The TRANSEC service generates a TRANSEC Stream that is a function of the TRANSEC algorithm, TRANSEC Variable, and TRANSEC Seed(s).

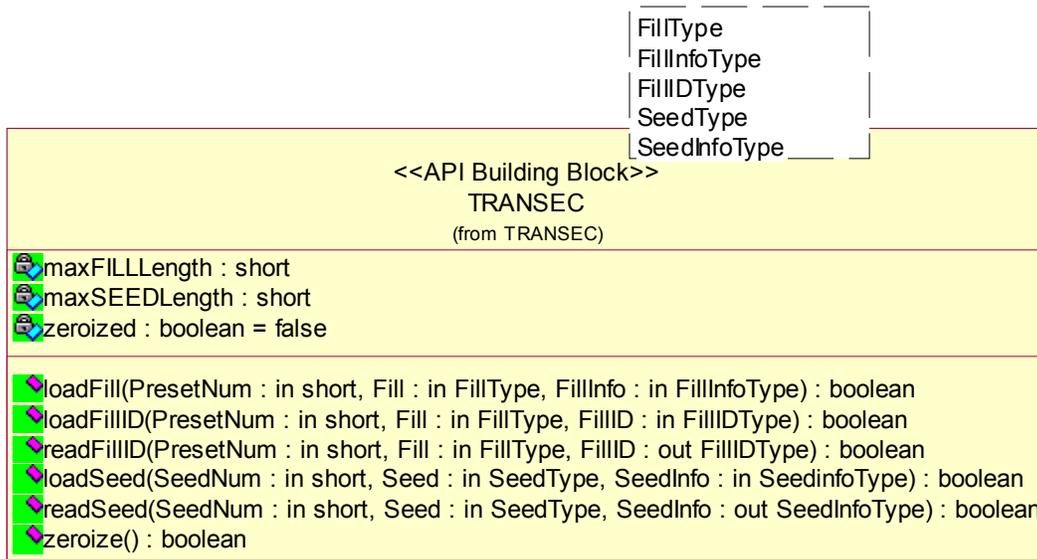


Figure 5. TRANSEC Building Block

Interfaces provided by the MAC BB are intended to support the following functions:

- TRANSEC Stream Generation
- Net Time Management
- Tx/Rx Frequency Management

MAC Layer Service Users also interface with this BB via a Uses relationship with the MAC Common Utility Service to transfer radio mode and active channel to the TRANSEC BB.

F.3.2.1 Load Fill Service.

The Load Fill Service provides a method to transfer TRANSEC Fill data from the Service User to the MAC layer, but never in the reverse direction. It returns the status of each transfer, Good or Bad. Fill types may include Hopsets, Lockouts, TRANSEC Keys, etc.

F.3.2.2 Load Fill ID

The Load Fill ID Service provides Service Users a method to edit/change a Fill ID specified by a Preset Number by transferring a new ID, which replaces an existing ID. It returns the status of each ID change, Good or Bad. Hopsets are an example of a Fill type for which ID edits may be desired.

F.3.2.3 Read Fill ID Service.

The Read Fill ID Service provides Service Users with a method of reading the identity of a fill element but not the actual Fill data. As an example, readFill will return a Hopset's ID but not the Hopset data. For a TRANSEC Variable it will return the variables ID, but not the variable.

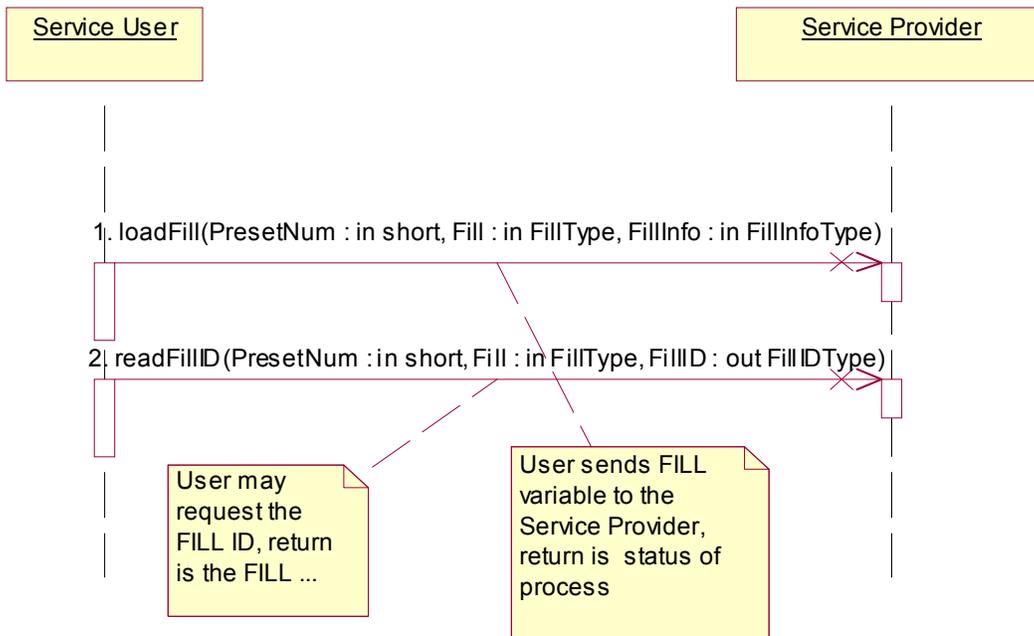


Figure 6. Sequence Diagram, loadFill and readFillID

F.3.2.4 Load Seed Service.

The Load Seed Service provides for transfer of TRANSEC Seed data from the Service User to the MAC layer. It returns the status of each transfer, Good or Bad. Seed types include Base Time of Day (TOD), Net TOD, etc.

F.3.2.5 Read Seed Service.

The Read Seed Service provides Service Users with a method of reading the actual Seed data. As an example, readSeed will return the Time of Day (TOD) or Word of Day (WOD).

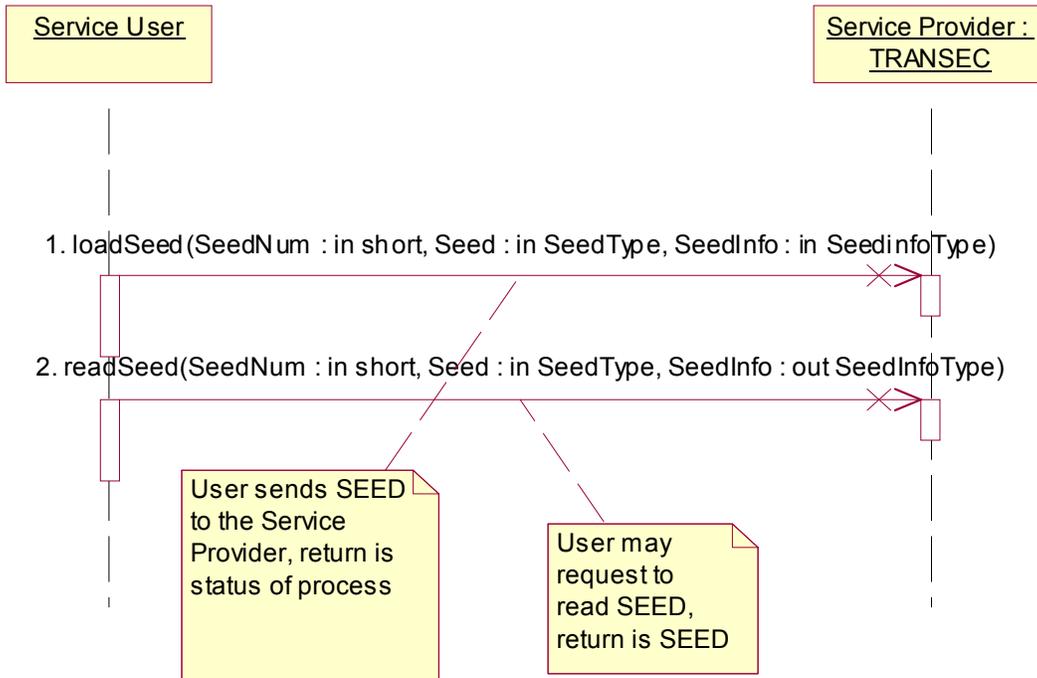


Figure 7. Sequence Diagram, loadSeed and readSeed

F.3.2.6 Zeroize Service.

The Zeroize Service provides Service Users with a method of removing all Fill and Seed data from memory space owned by the TRANSEC BB. This service over-writes all Fill and Seed data memory space with a variety of data patterns to ensure information is not retained and returns the status of the operation (i.e., successfully completed or failed). This service does not release this memory space back to the Operating Environment.

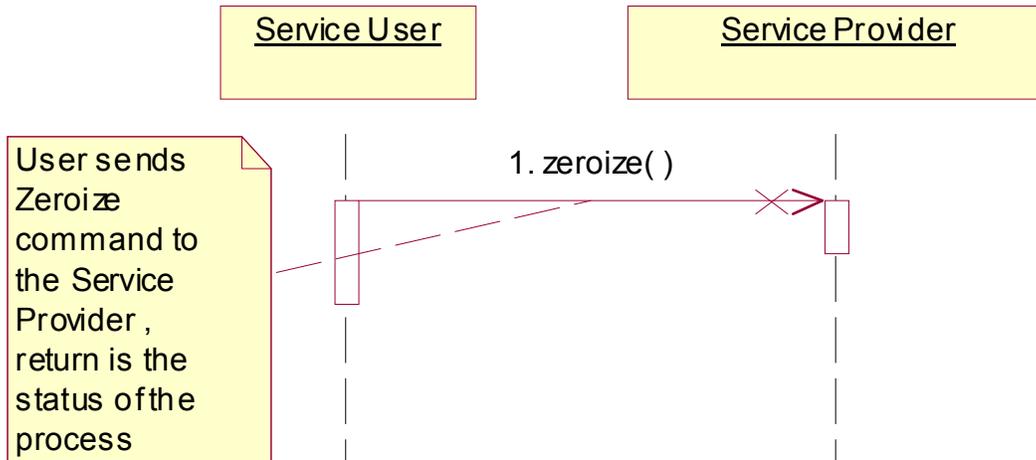


Figure 8. Sequence Diagram, zeroize

F.3.2.7 TRANSEC Service Group States.

The TRANSEC Service Group states are illustrated in the following diagrams.

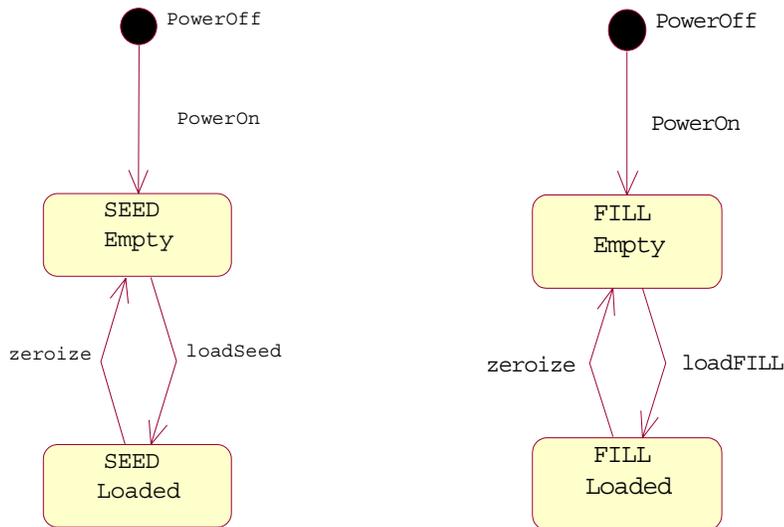


Figure 9. State Diagram, TRANSEC BB Fill and Seed

As depicted in Figure 9, the state diagram for TRANSEC BB Fill and Seed, at Power-On both Seed and Fill are in the Empty state. Reading these values when they are in this state will return an Empty indication (e.g., Boolean = False). Transitions from Empty to Loaded occur whenever a loadFill or a loadSeed operation is invoked by the Service User and the fill or seed has been successfully transferred to the TRANSEC BB, which is indicated by returning a Good indication (e.g., Boolean = True).

As shown in Figure 10, the state diagram for TRANSEC Stream, loading a Fill or Seed does not affect generation of the TRANSEC stream. To generate the TRANSEC Stream using a particular Fill/Seed, an activateChannel command (section F.3.3.1) is required to transfer Fill/Seed data into the TRANSEC algorithm. As shown in the state diagram, an activateChannel command disables the TRANSEC Stream, loads the TRANSEC algorithm with new Fill/Seed data and then enables the TRANSEC Stream. The ability to load the TRANSEC with Fill/Seed data without changing the current TRANSEC stream is required by some waveforms.

As an example for SINCGARS, changing a Preset Channel requires, as a minimum, a change in both Hopset (Fill) and Net TOD (Seed). The TRANSEC would implement the Hopset and Net TOD structures such that FillNum = 1 and SeedNum = 1 would be the Hopset and Net TOD for Preset = 1. Thus when a Preset change is recognized by the Service User (HCI or IO), it would invoke activate(FillNum, SeedNum) on the TRANSEC to change channels.

For other waveforms, which use the same TRANSEC Variable and vary the WOD, the Service User can accomplish this invoking the activateChannel command with the PresetNum held constant while varying the SeedNum to select different WODs.

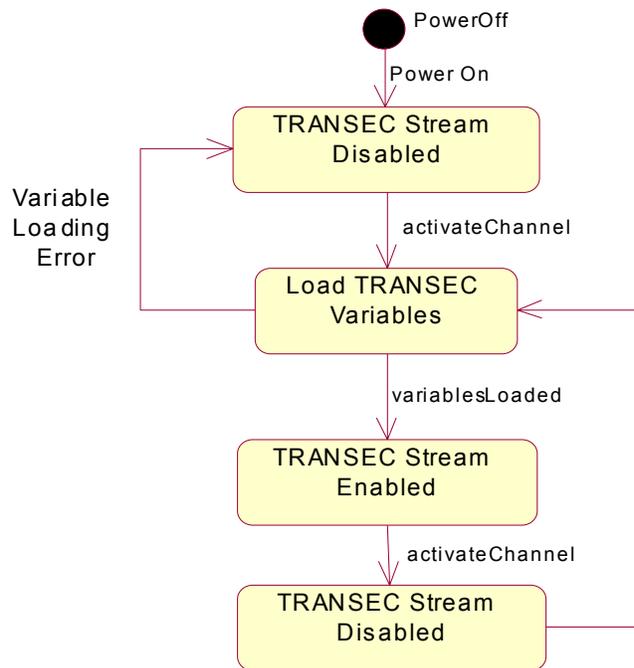


Figure 10. State Diagram, TRANSEC Stream

F.3.3 Channel Error Control Building Block.

The Channel Error Control Building Block attempts to maintain the integrity of the message over the physical channel. Functions commonly performed by this BB are:

- forward error correction,
- interleave or scramble,
- bit tracking including fade bridging, and
- transmit RF power level control.

MAC Layer Service Users also interface with this BB via Uses relationships with the MAC Common Utility and Rx Termination BBs.

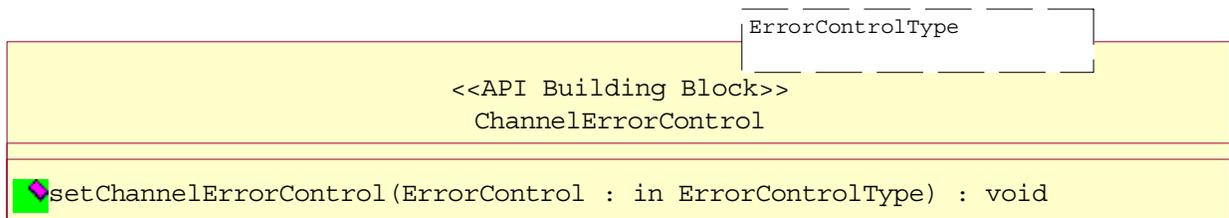


Figure 11. Channel Error Control Building Block

ErrorControl allows the Service User to enable/disable channel error control. As a minimum, this supports requirements in MIL-STD-188-182.

F.3.3.1 Set Channel Error Control Service.

The Set Channel Error Control Service provides Service Users with a method to enable/disable the channel error control function in non-real-time (i.e., not a hop-by-hop basis, which would be implemented in pushpacket).

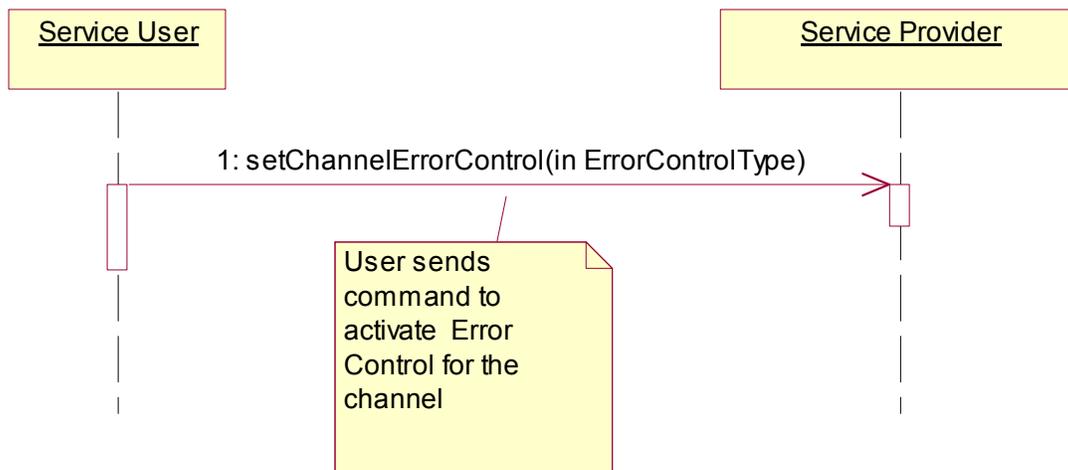


Figure 12. Sequence Diagram, setChannelErrorControl

F.3.3.2 Set Channel Error Control Service States.

Set Channel Error Control Service states and state transitions that occur as a result of the Service User activating or deactivating channel error control are depicted in the following state diagram. The initial channel error Control State is defined by the instantiating API Service Definition.

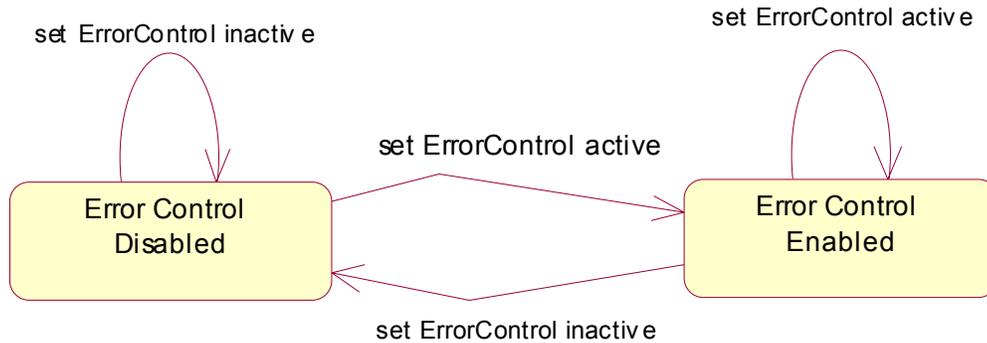


Figure 13. State Diagram, Set Channel Error Control

F.3.4 Channel Access Building Block.

The Channel Access Service Group detects incoming messages and executes the channel access protocol. These functions typically include:

- sync detection
- sync search
- end-of-message detection
- channel access algorithm including TDMA, CSMA, DASA/DAMA, WDW, etc.

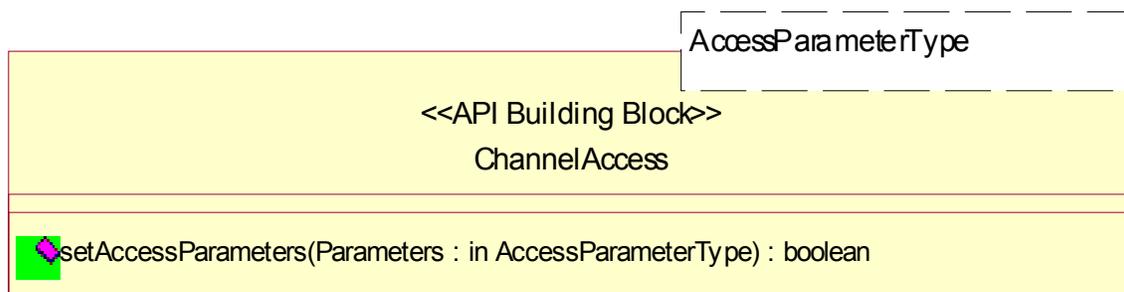


Figure 14. Channel Access Building Block

F.3.4.1 Set Channel Access Parameters Service.

The Set Channel Access Parameters Service provides Service Users with the ability to initialize/configure a Net Access Algorithm with information required to establish slot width,

slot frequency, slot time, number of slots, slot priorities (Flash, Routine, etc.), inter-transmit delay, etc. A Boolean is returned to provide the status of each parameter transfer (i.e., successful or failed).

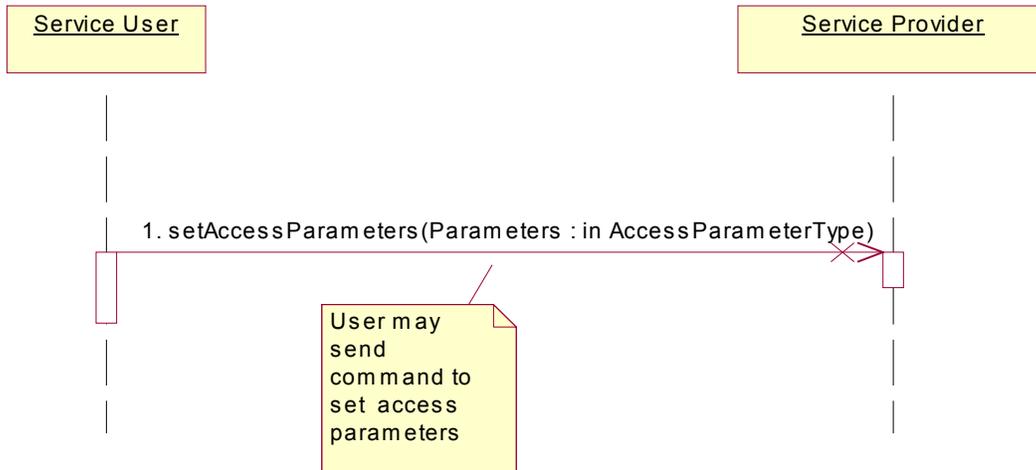


Figure 15. Sequence Diagrams, setChannelAccessParameters

F.3.4.2 Channel Access States.

The Channel Access Service provides Service Users with controlled access to a RF channel. To achieve controlled access to a particular slot, the access control algorithm must be properly configured (as defined by the instantiating API Service Description) and the Service User must specify a specific access slot/time, using the setSlot service. Access control algorithm configuration is achieved via the setSlotParameters service or alternatively at instantiation using concrete values specified by the implementation. Otherwise, channel access is immediate. As an alternative to the setSlot service, SlotNum can be specified on a per-packet basis based on control data contained within a packet. The Start State is defined by the instantiating API Service Description.

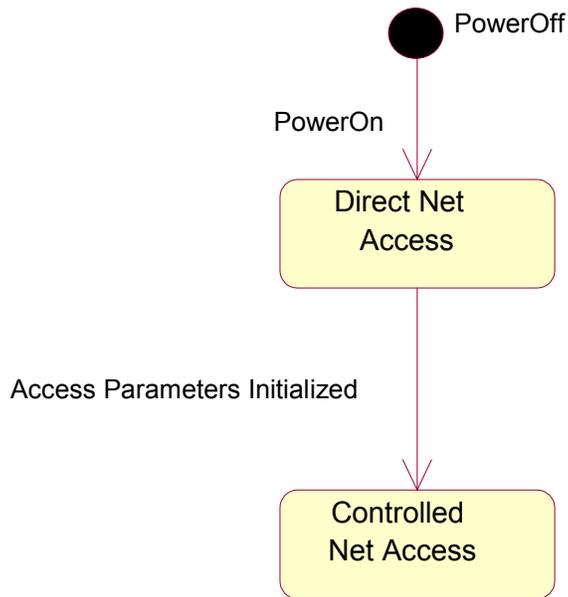


Figure 16. State Diagram, Channel Access

F.3.5 MAC Address Building Block.

The MAC Address Building Block provides a generalized interface to the MAC layer, which allows the Service User to define the MAC layer’s own address. The service returns a Boolean to indicate whether or not the address was bound.

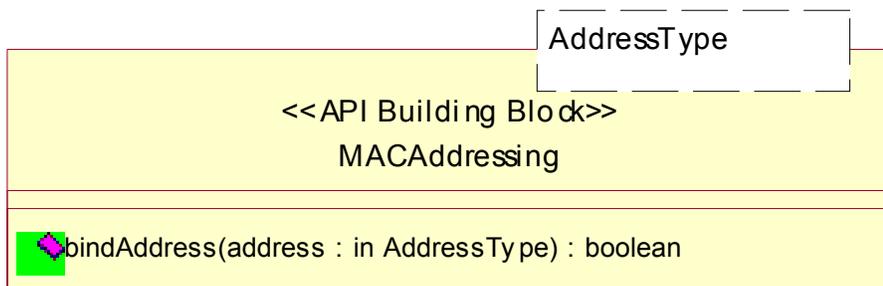


Figure 17. MAC Address Building Block

F.3.5.1 Bind Address Service.

The Bind Address Service provides an interface for binding the MAC's address. This address is used by other Service Users to obtain packet services provided by the MAC layer.

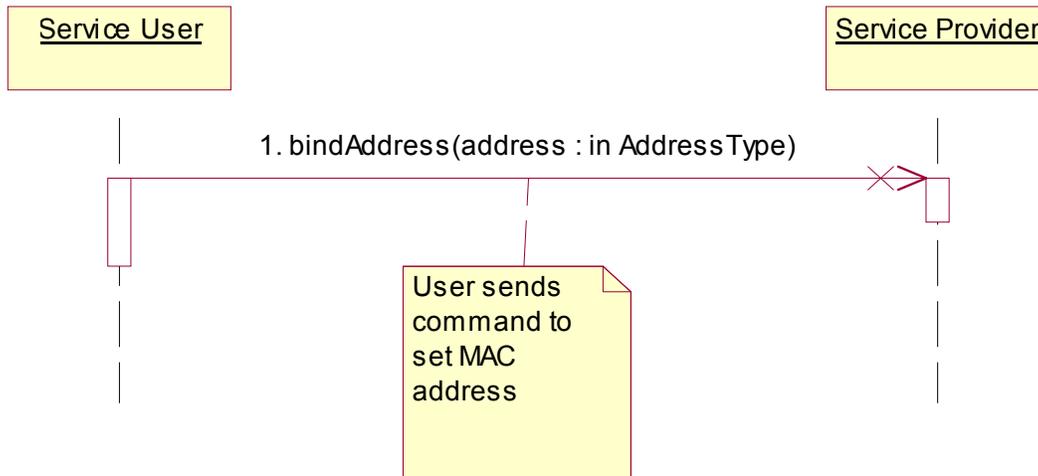


Figure 18. Sequence Diagram, bindAddress

F.3.5.2 MAC Bind Address Service States.

For the MAC layer to function as a Service Provider of real time services or to respond to real-time control data, it must be assigned a unique packet address by the Service User. When the MAC layer recognizes its own address is contained within a packet, it will respond by attempting to provide the requested service. Real-time services provided are specific to the MAC Layer implementation, which is a function of the type of waveform API being instantiated.

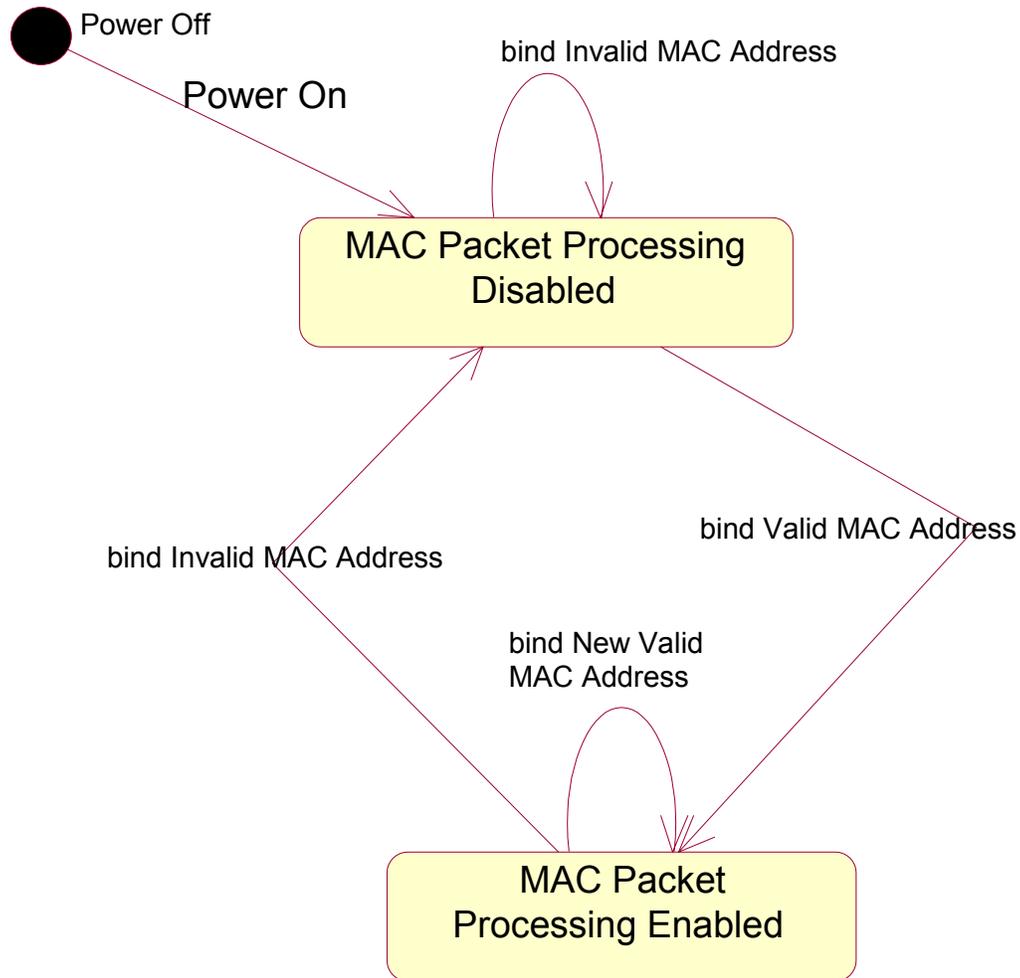


Figure 19. State Diagram, MAC Address BB

F.3.6 Rx Termination Building Block.

The Terminate Reception Service provides a generalized interface to the MAC layer that the Service User can use to terminate an active reception and return to the signal search state. The service returns a Boolean to indicate reception was terminated (True) or the operation failed (False).

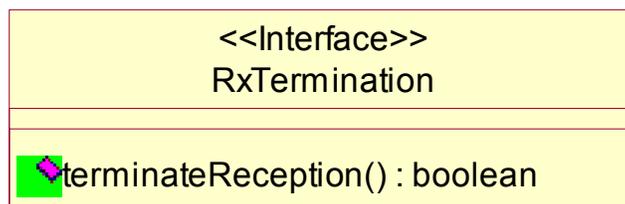


Figure 20. Rx Termination Building Block

F.3.6.1 Terminate Reception Service.

The Terminate Reception Service provides MAC layer Service Users with a capability to perform the following:

- terminate the current reception,
- flush the message,
- stop bit tracking,
- clear all variables associated with the current reception including forward-error-correcting code and interleavers and
- return to the signal search state.

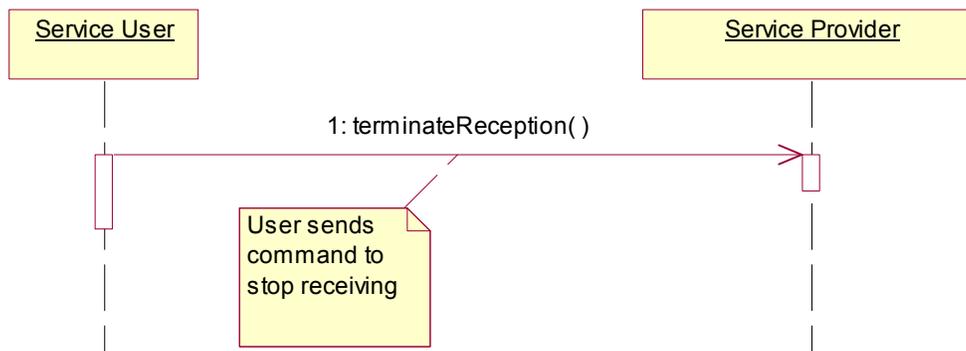


Figure 21. Sequence Diagram, Terminate Reception

F.3.6.2 Terminate ReceptionService State Influence.

In the Active Receive State, the Terminate ReceptionService causes a transition to the Terminate Reception State, as depicted in Figure 22. When a reception is terminated, the state automatically transitions to the Passive Receive (or signal search) state.

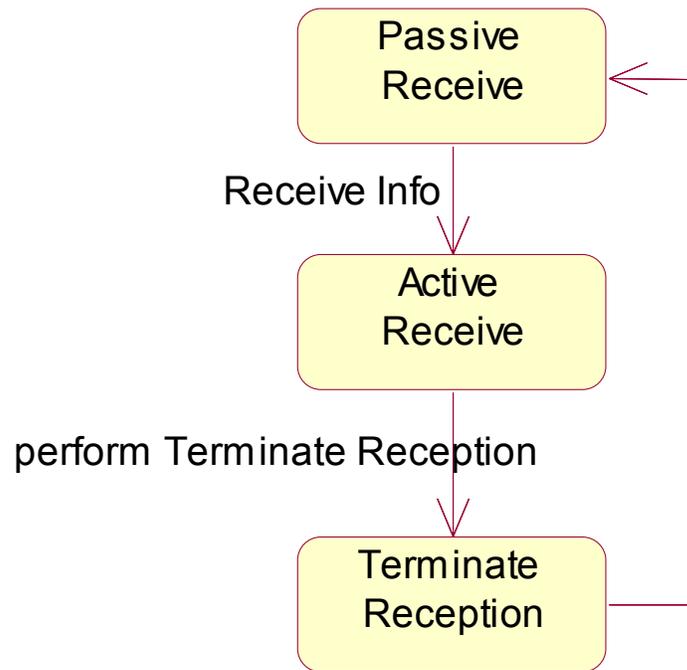


Figure 22. State Diagram, Terminate Reception BB

F.3.7 Quality of Service (QOS) Building Block.

The QOS Service provides Service Users with a standard method to obtain channel quality information, when these are calculated within the MAC Layer. Examples of these parameters are:

- residual bit-error-rate
- block error rate or correctable block status
- erasure count and
- raw bit-error-rate.

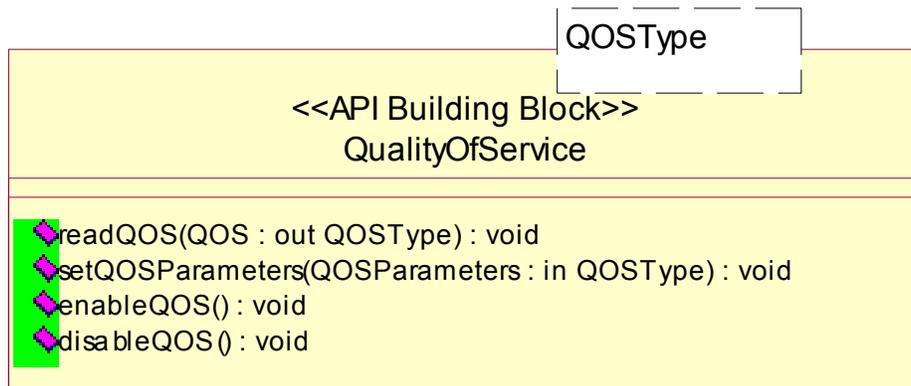


Figure 23. Quality of Service Building Block

F.3.7.1 Read QOS Service.

The Read QOS Service provides Service Users with an ability to read quality of service information. The type of information read is declared by the inheriting API.

F.3.7.2 Set QOS Parameters Service.

The Set QOS Parameters Service provides the Service User with a standard interface that can be used to configure the QOS Service to provide the type of error rate data desired.

F.3.7.3 Enable QOS Service.

Enable QOS Service provides the Service User with the ability to enable the QOS function implemented by the inheriting API.

F.3.7.4 Disable QOS Service.

Disable QOS Service provides the Service User with the ability to disable the QOS function implemented by the inheriting API.

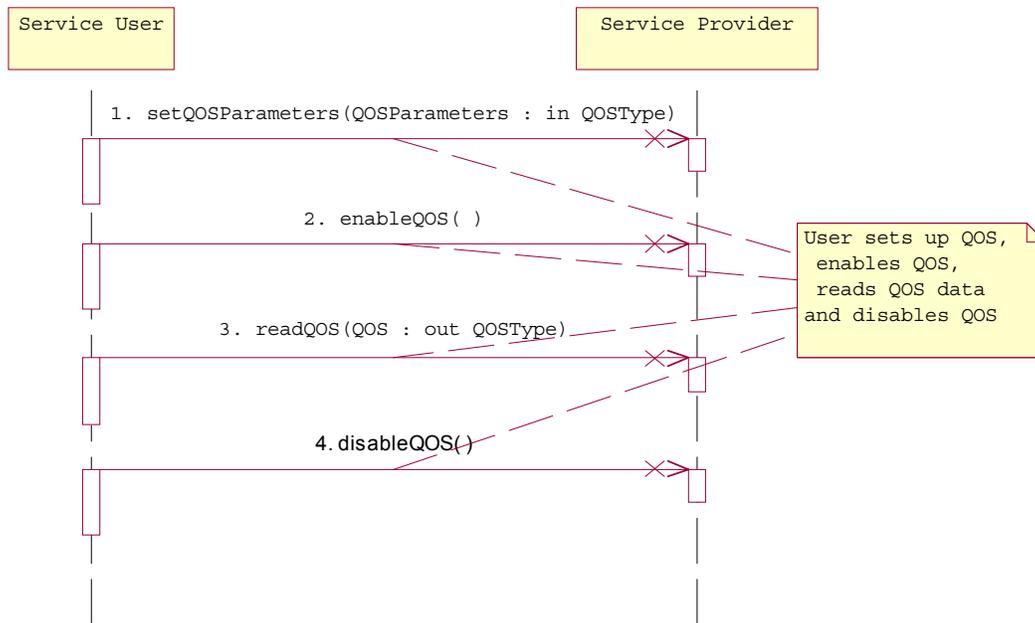


Figure 24. Sequence Diagram, QOS BB

F.3.7.5 Quality of Service States.

Quality of Service BB states and state transitions that occur as a result of the Service User invoking the `setQOSParameters` method are depicted in the following state diagram.

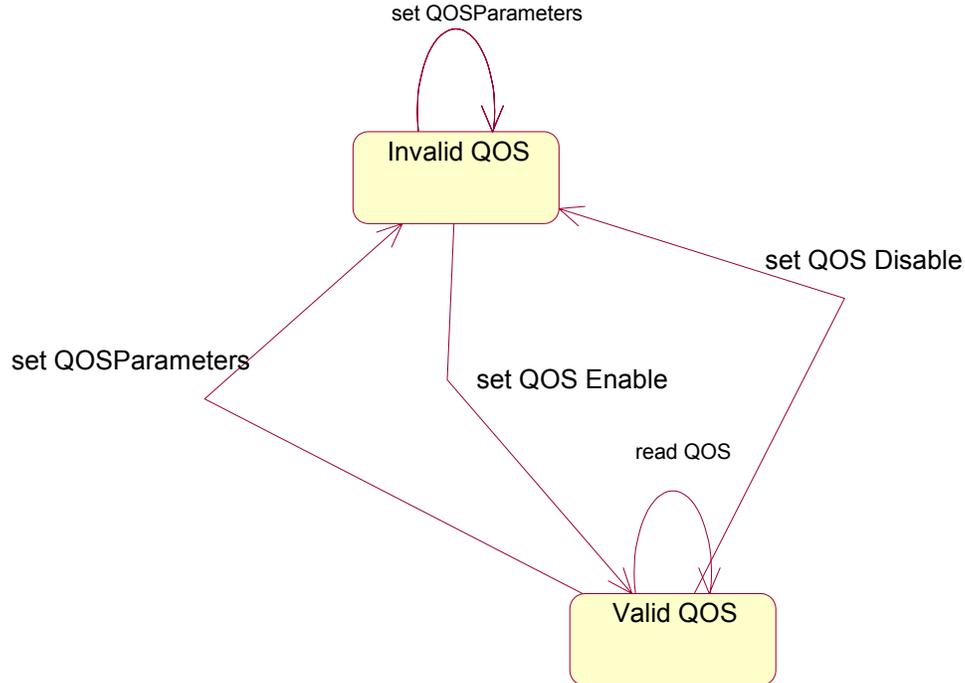


Figure 25. State Diagram, Quality of Service BB

F.3.8 Simple Packet BB (Inherited).

Real-time control and data will be pushed up-stream and down-stream using control and data packets. Like the non-real-time control information, these service parameters are waveform-specific due to the significant differences in waveform characteristics. For more detail, refer to the Packet Building Block Service Definition.

F.3.8.1 Parameters Filled in by Inheriting Class.

The following parameters are waveform specific and will be defined by the MAC API Service Definition.

F.3.8.1.1 *PayloadType*.

This can be defined to be a single Presentation Protocol Data Unit (PPDU) or an array of PPDUs. The *PayloadType* is a sequence of Traffic Units. The Traffic Unit is user defined for each Service User/provider pair. Typical examples of a traffic unit are Octets, Audio Samples, digital waveform data-words, etc.

F.3.8.1.2 *ControlType*.

This is the control information associated with *ControlType*. Examples of Control Type information are: Beginning of Stream, End of Stream, Next Frequency, Time of Transmission, Quality of Service (QoS), etc.

F.3.8.2 Attributes.

All attributes are read only. If an attribute is to be changed directly via the instantiating API, an operation to do so is provided separately to ensure that the change takes effect immediately.

F.3.8.2.1 MaxPayloadSize.

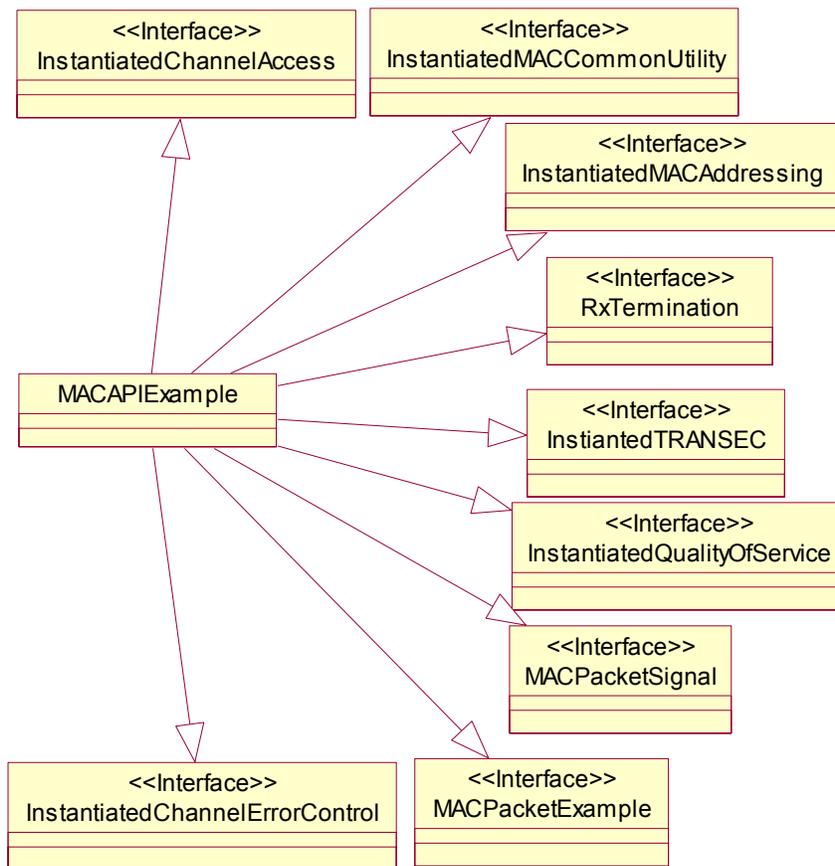
MaxPayloadSize indicates the maximum packet size in traffic units.

F.3.8.2.2 MinPayloadSize.

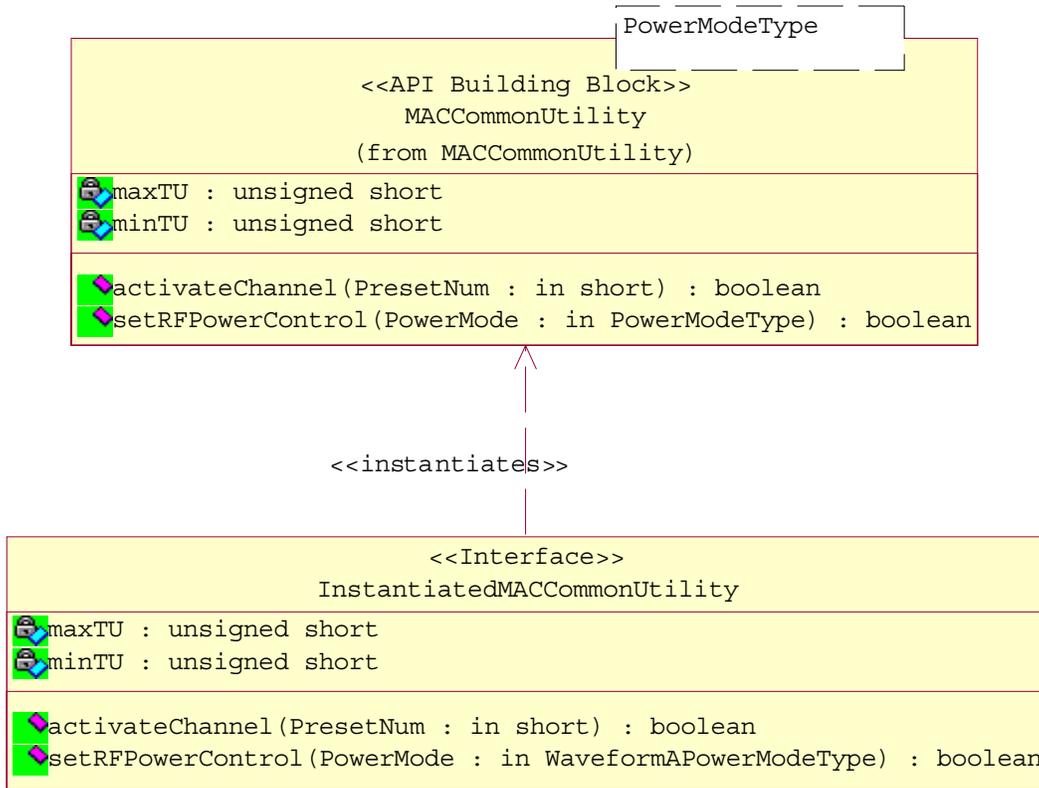
MinPayloadSize indicates the minimum packet size in traffic units.

F.3.9 Examples.

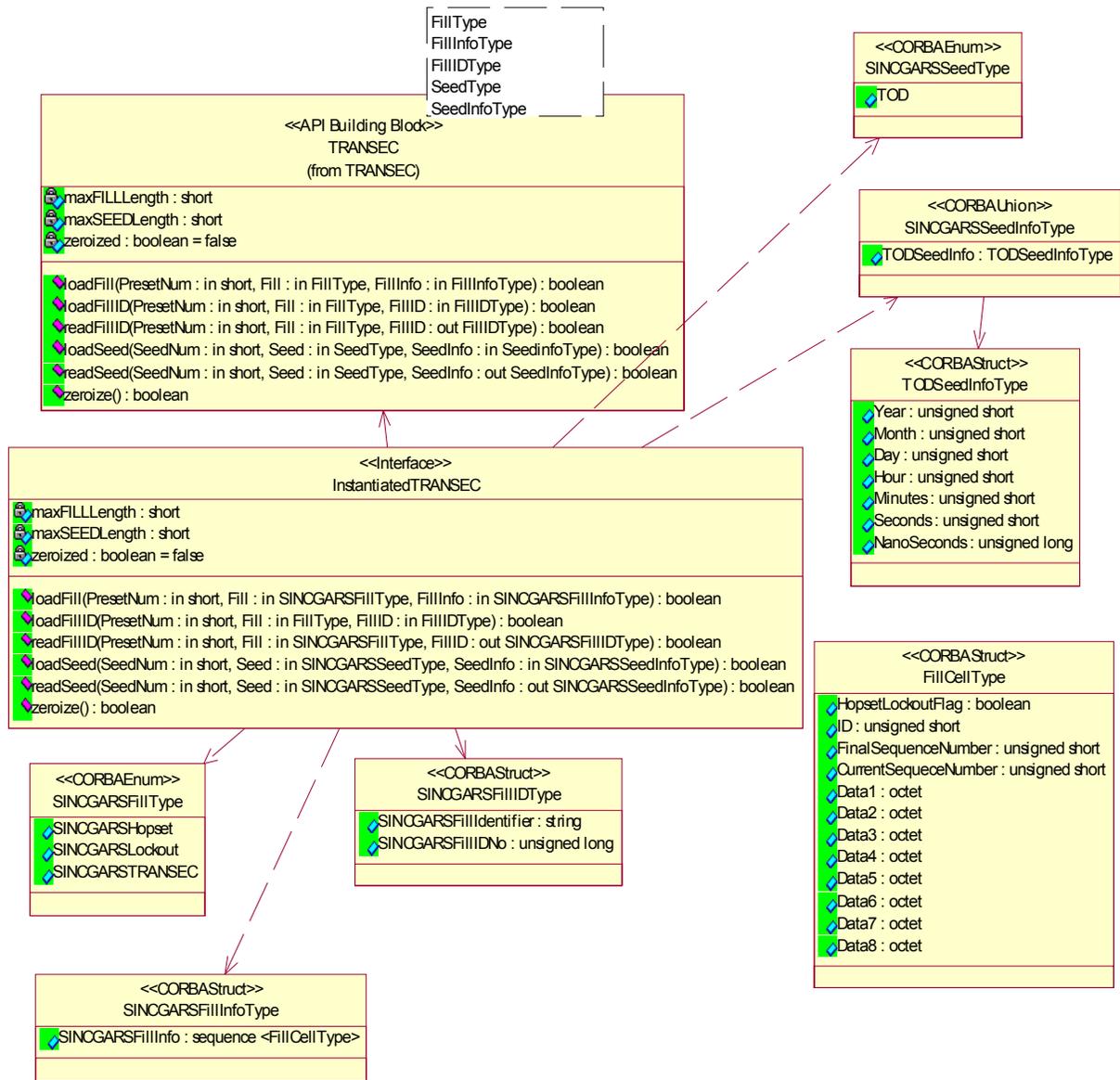
The following MAC API example and BB examples are instantiated using a specific waveform type.



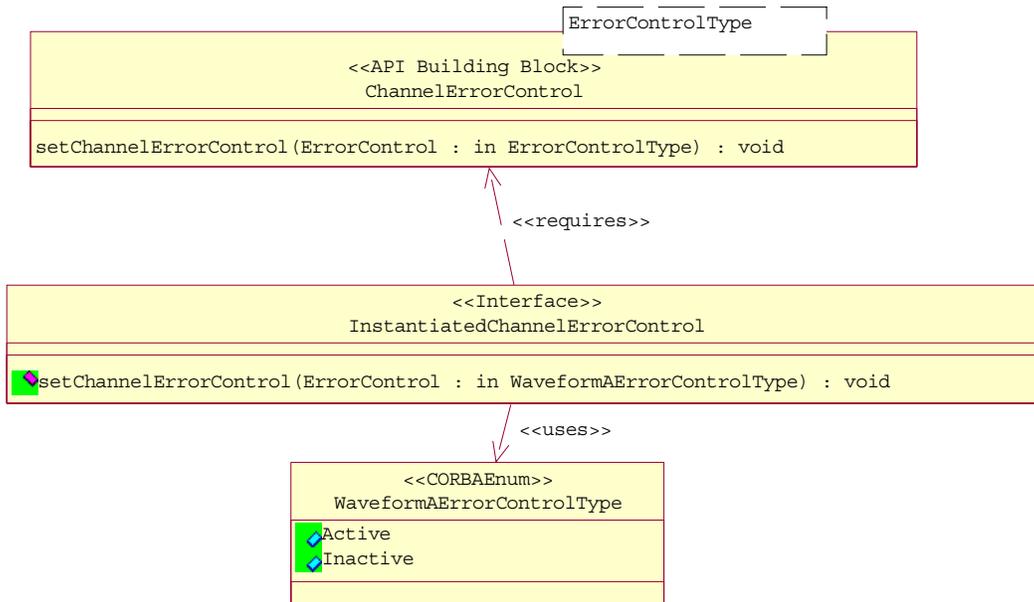
F.3.9.1 MAC Common Utility BB.



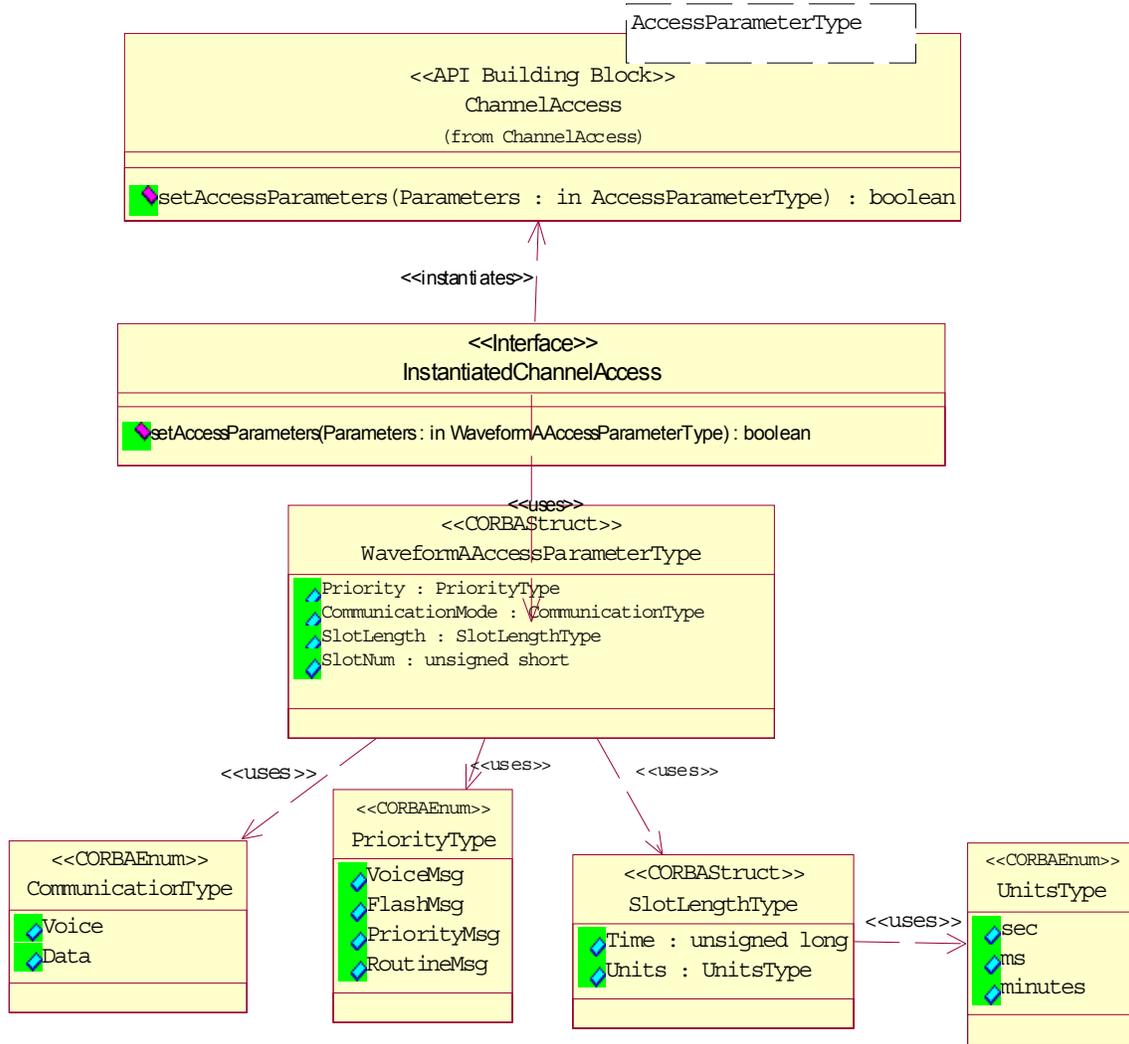
F.3.9.2 TRANSEC BB.



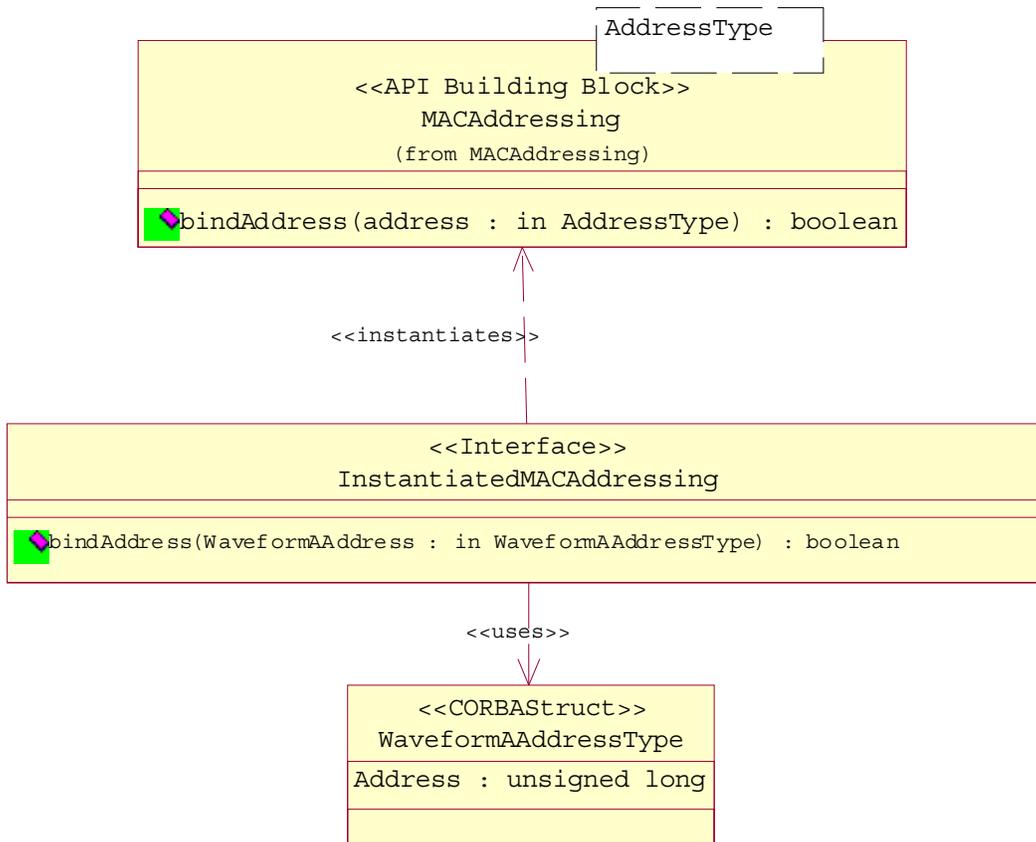
F.3.9.3 Channel Error Control BB.



F.3.9.4 Channel Access BB.

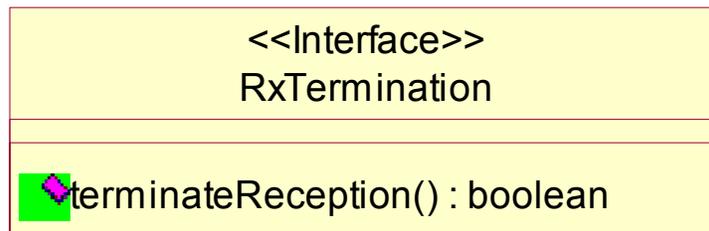


F.3.9.5 MAC Addressing BB.

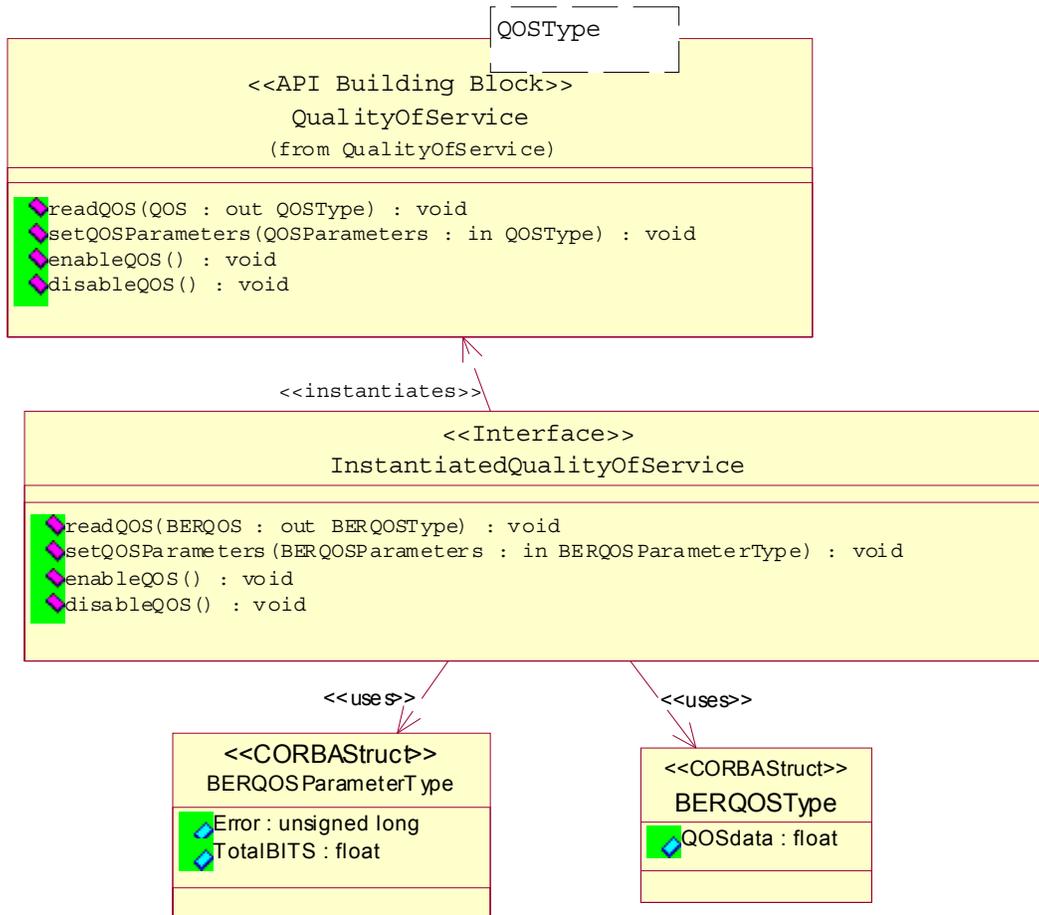


F.3.9.6 Rx Termination.

Rx Termination is a concrete building block that does not require instantiation.



F.3.9.7 Quality of Service BB.



F.4 SERVICE PRIMITIVES.

F.4.1 MAC Common Utility Service Primitives.

F.4.1.1 Activate Channel Service.

“*activateChannel*” provides the ability to communicate to the active radio channel (e.g., 1, 2, Guard, Emergency, etc.) to the MAC Layer.

F.4.1.1.1 Synopsis.

boolean activateChannel (in short PresetNum)

F.4.1.1.2 Parameters.

PresetNum: specifies the channel number or preset number of the active (selected) RF channel.

short: range = -2^{15} to $2^{15}-1$

F.4.1.1.3 State.

Any channel active.

F.4.1.1.4 New State.

New channel active.

F.4.1.1.5 Response.

This method returns True, if the channel corresponding to the PresetNum is activated; otherwise, False is returned.

F.4.1.1.6 Originator.

Service User.

F.4.1.1.7 Errors/Exceptions.

Return = True if the channel is successfully activated; otherwise return = False.

F.4.1.2 Set RF Power Control Service.

“*setRFPowerControl*” provides the ability to communicate the desired RF power output level (e.g., Auto, Lo, Med, Hi, Max, etc.) to the MAC Layer.

F.4.1.2.1 Synopsis.

boolean setRFPowerControl (in PowerModeType PowerMode)

F.4.1.2.2 Parameters.

PowerModeType: *To be defined by instantiating API Service Definition.* Parameter may specify RF power output level, specify algorithm to be used to automatically control the RF power output level, etc.

F.4.1.2.3 State.

Any Power Mode.

F.4.1.2.4 New State.

New Power Mode.

F.4.1.2.5 Response.

Return = True if the power control is successfully set with the given PowerMode, otherwise it Boolean = False.

F.4.1.2.6 Originator.

This primitive is initiated by the Service User.

F.4.1.2.7 Errors/Exceptions.

N/A.

F.4.1.3 Minimum Transmission Unit Length Attribute.

Minimum Transmission Unit Length provides Service Users with the capability to read the minimum Transmission Unit (TU) length the Service Provider will accept as being a valid packet.

F.4.1.3.1 Synopsis.

readonly attribute unsigned short MinTU

F.4.1.3.2 Parameters.

MinTU: inheriting API must declare the minimum length of a valid packet.

F.4.1.3.3 State.

Any state.

F.4.1.3.4 New State.

State remains unchanged.

F.4.1.3.5 Response.

unsigned short: range is 0 to $2^{16}-1$. Value is minimum number of octets in a valid packet.

F.4.1.3.6 Originator.

Service User.

F.4.1.3.7 Errors/Exceptions.

N/A.

F.4.1.4 Maximum Transmission Unit Length Attribute.

Maximum Transmission Unit Length provides Service Users with the capability to read the maximum Transmission Unit (TU) length the Service Provider will accept as being a valid packet.

F.4.1.4.1 Synopsis.

F.4.1.4.2 readonly attribute unsigned short MaxTU Parameters.

MaxTU: inheriting API must declare the maximum length of a valid packet.

F.4.1.4.3 State.

Any state.

F.4.1.4.4 New State.

State remains unchanged.

F.4.1.4.5 Response.

unsigned short: range is 0 to $2^{16}-1$. Value is maximum number of octets in a valid packet.

F.4.1.4.6 Originator.

Service User.

F.4.1.4.7 Errors/Exceptions.

N/A.

F.4.2 TRANSEC Service Primitives.

F.4.2.1 Load Fill Service

“*loadFill*” provides a method to transfer TRANSEC Fill data to the MAC layer.

F.4.2.1.1 Synopsis.

boolean loadFill (in short PresetNum, in FillType FILL, in FillInfoType FillInfo)

F.4.2.1.2 Parameters.

PresetNum short range 0 to $2^{16} - 1$. Typically, associates fill data with a channel number.

FillType: *To be defined by instantiating API Service Definition.* Examples of Fill Types are hopset, lockout, TRANSEC variable, etc.

FillInfo: *To be defined by instantiating API Service Definition.* An example is a Hopset data sequence comprised of a Hopset/Lockout Flag, ID, FinalSequenceNumber, CurrentSequenceNumber, and Data [8].

F.4.2.1.3 State.

This service is available in any state except when a zeroize operation is in progress.

F.4.2.1.4 New State.

FillType loaded (e.g., Fill or Seed)

F.4.2.1.5 Response.

Boolean returns ‘True’ if the operation was completed successfully, otherwise ‘False’ is returned.

F.4.2.1.6 Originator.

Service User.

F.4.2.1.7 Errors/Exceptions.

N/A.

F.4.2.2 Load Fill ID Service

“*loadFillID*” provides a method to transfer a new Fill ID to the MAC layer.

F.4.2.2.1 Synopsis.

boolean loadFillID (in short PresetNum, in FillType FILL, in FillIDType FillID)

F.4.2.2.2 Parameters.

PresetNum short range 0 to $2^{16}-1$. Typically, associates fill data with a channel number.

FillType: *To be defined by instantiating API Service Definition.* Examples of Fill Types are hopset, lockout, TRANSEC variable, etc.

FillIDType: *To be defined by instantiating API Service Definition.* An example is a Hopset ID.

F.4.2.2.3 State.

This service is available in any state except when a zeroize operation is in progress.

F.4.2.2.4 New State.

New FillID loaded (e.g., Hopset ID)

F.4.2.2.5 Response.

Boolean returns ‘True’ if the operation was completed successfully, otherwise ‘False’ is returned.

F.4.2.2.6 Originator.

Service User.

F.4.2.2.7 Errors/Exceptions.

F.4.2.3 Read Fill ID Service.

“*readFillID*” provides the identity (not the actual Fill data) of a fill element.

F.4.2.3.1 Synopsis.

boolean readFillID (in short PresetNum, in FillType Fill, out FillIDType FillID)

F.4.2.3.2 Parameters.

PresetNum: short range 0 to $2^{16}-1$. Typically associates Fill ID with a channel number.

FillIDType: *To be defined by instantiating API Service Definition.* Example Fill Types are Hopset and Lockout.

FillID: Example Fill IDs are F200, L100, L7L8

F.4.2.3.3 State.

Any state.

F.4.2.3.4 New State.

State remains unchanged.

F.4.2.3.5 Response.

Boolean returns ‘True’ if the operation was completed successfully, otherwise ‘False’ is returned.

F.4.2.3.6 Originator.

Service User.

F.4.2.3.7 Errors/Exceptions.

N/A.

F.4.2.4 Load Seed Service.

“*loadSeed*” provides the ability to transfer seeds such as Time of Day or Word of Day to the MAC layer.

F.4.2.4.1 Synopsis.

boolean loadSeed (in short SeedNum, in SeedType Seed, in SeedInfoType SeedInfo)

F.4.2.4.2 Parameters.

SeedNum: short range = 0 to $2^{16}-1$. Typically associates the seed (TOD as an example) with a preset channel number.

SeedType: *To be defined by instantiating API Service Definition*. May be time of day, net time of day, word of day, etc.

SeedInfo: An example is a TOD data sequence comprised of Day, Hours, Minutes and Seconds.

F.4.2.4.3 State.

Any state except zeroize.

F.4.2.4.4 New State.

Seed (PresetNum) loaded.

F.4.2.4.5 Response.

Boolean returns ‘True’ if the operation was completed successfully, otherwise ‘False’ is returned.

F.4.2.4.6 Originator.

Service User.

F.4.2.4.7 Errors/Exceptions.

N/A.

F.4.2.5 Read Seed Service.

“*readSeed*” returns the actual Seed data. E.g., Time of Day (TOD) or Word of Day (WOD).

F.4.2.5.1 Synopsis.

boolean readSeed (in short SeedNum, in SeedType Seed, out SeedInfoType SeedInfo)

F.4.2.5.2 Parameters.

SeedNum: short range = 0 to $2^{16}-1$. Typically, associates a seed (TOD/WOD as an example) with a preset channel number or crypto period.

SeedType: *To be defined by instantiating API Service Definition.* Typically TOD or WOD.

SeedInfo: An example is a TOD data sequence comprised of Day, Hours, Minutes and Seconds.

F.4.2.5.3 State.

Any.

F.4.2.5.4 New State.

No change

F.4.2.5.5 Response.

Boolean returns 'True' if the operation was completed successfully, otherwise 'False' is returned.

F.4.2.5.6 Originator.

Service User.

F.4.2.5.7 Errors/Exceptions.

N/A.

F.4.2.6 Zeroize Service.

“zeroize” provides a method of removing all Fill and Seed data from memory space owned by the TRANSEC BB.

F.4.2.6.1 Synopsis

boolean zeroize ()

F.4.2.6.2 Parameters.

None.

F.4.2.6.3 State.

Any.

F.4.2.6.4 New State.

Fill (PresetNum = 0 to n) Empty and Seed (PresetNum = 0 to n) Empty

F.4.2.6.5 Response.

Boolean returns True when zeroization is successful, otherwise it returns False.

F.4.2.6.6 Originator.

Service User.

F.4.2.6.7 Errors/Exceptions.

N/A.

F.4.3 Channel error control service Primitives.

F.4.3.1 Set Channel Error Control Service.

“*setChannelErrorControl*” provides for non-real time control of channel error control function(s) in the Mac Layer.

F.4.3.1.1 Synopsis.

void setChannelErrorControl (in ErrorControlType ErrorControl)

F.4.3.1.2 Parameters.

Parameters: *To be defined by the instantiating API Service Definition.* Parameters may be used to configure the error control functionality.

F.4.3.1.3 State.

Any.

F.4.3.1.4 New State.

Channel Error Control function(s) activated or deactivated.

F.4.3.1.5 Response.

None.

F.4.3.1.6 Originator.

Service User.

F.4.3.1.7 Errors/Exceptions.

N/A.

F.4.4 Channel Access Service Primitives.

F.4.4.1 Set Access Parameters Service.

”*setAccessParameters*” provides the ability the ability to initialize/configure a Net Access Algorithm with information required to establish slot width, slot frequency, slot time, number of slots, slot priorities (Flash, Routine, etc.), inter-transmit delay, etc.

F.4.4.1.1 Synopsis.

boolean setAccessParameters (in AccessParameterType Parameters)

F.4.4.1.2 Parameters.

Parameters: *To be defined by the instantiating API Service Definition.* Parameters may be used to initialize CSMA variables.

AccessParameterType: *To be defined by the instantiating API Service Definition.* Access Parameters may be used to specify types of access control (e.g., FDMA, TDMA, CSMA, etc.).

F.4.4.1.3 State.

Any state.

F.4.4.1.4 New State.

Slot Parameters Configured (Controlled Net Access available).

F.4.4.1.5 Response.

Boolean returns True if the AccessParameter is valid (within a range/format defined by the instantiating API Service Definition), otherwise it returns False.

F.4.4.1.6 Originator.

Service User.

F.4.4.1.7 Errors/Exceptions.

N/A.

F.4.5 MAC Address Primitives.

F.4.5.1 Register Address Service.

Provides Service Users with the ability to bind “own address” to the MAC layer.

F.4.5.1.1 Synopsis.

boolean bindAddress (in AddressType address)

F.4.5.1.2 Parameters.

AddressType: *To be defined by the instantiating API Service Definition.*

F.4.5.1.3 State.

MAC Packet Processing Disabled.

F.4.5.1.4 New State.

MAC Packet Processing Enabled.

F.4.5.1.5 Response.

Boolean returns True if the Address is valid (within a range/format defined by the instantiating API Service Definition), otherwise it returns False.

F.4.5.1.6 Originator.

Service User.

F.4.5.1.7 Errors/Exceptions.

N/A.

F.4.6 Terminate Reception Service Primitives.

F.4.6.1 Terminate Reception Service.

Provides the Service User with the ability to terminate an active reception and return to the Search State.

F.4.6.1.1 Synopsis.

boolean terminateReception ()

F.4.6.1.2 Parameters.

None.

F.4.6.1.3 State.

Active Receive.

F.4.6.1.4 New State.

Terminate Receive.

F.4.6.1.5 Response.

Boolean returns True if an active reception is terminated, otherwise it returns False.

F.4.6.1.6 Service Originator

Service User.

F.4.6.1.7 Errors/Exceptions.

N/A.

F.4.7 Quality of Service(QOS) Primitives.

F.4.7.1 Read QOS Service.

Provides the Service User with the ability read the QOS information produced by the MAC layer.

F.4.7.1.1 Synopsis.

void readQOS (out QOSType QOS)

F.4.7.1.2 Parameters.

QOSType: *To be defined by the instantiating API Service Definition.* Quality of service types may include residual error rate, BER, block error rate, etc.

F.4.7.1.3 State.

QOS data valid.

F.4.7.1.4 New State.

State remains unchanged.

F.4.7.1.5 Response.

None.

F.4.7.1.6 Originator.

Service User.

F.4.7.1.7 Errors/Exceptions.

Raises an exception when QOS data is invalid.

F.4.7.2 Set QOS Parameters Service.

Provides the Service User with the ability to configure the QOS function.

F.4.7.2.1 Synopsis.

void setQOSParameters (in QOSType QOSParameters)

F.4.7.2.2 Parameters.

QOSParameters: *To be defined by the instantiating API Service Definition.*

QOSType: *To be defined by the instantiating API Service Definition.*

F.4.7.2.3 State.

Invalid QOS.

F.4.7.2.4 New State.

Valid QOS.

F.4.7.2.5 Response.

None.

F.4.7.2.6 Originator.

Service User.

F.4.7.2.7 Errors/Exceptions.

None.

F.4.7.3 Enable QOS Service.

Provides the Service User with the ability to enable generation of QOS information in the MAC layer.

F.4.7.3.1 Synopsis.

void enableQOS ()

F.4.7.3.2 Parameters.

None.

F.4.7.3.3 State.

Invalid QOS.

F.4.7.3.4 New State.

Valid QOS.

F.4.7.3.5 Response.

None.

F.4.7.3.6 Originator.

Service User.

F.4.7.3.7 Errors/Exceptions.

None.

F.4.7.4 Disable QOS Service.

Provides the Service User with the ability to disable generation of QOS information in the MAC layer.

F.4.7.4.1 Synopsis.

void disableQOS ()

F.4.7.4.2 Parameters.

None.

F.4.7.4.3 State.

Valid QOS.

F.4.7.4.4 New State.

Invalid QOS.

F.4.7.4.5 Response.

None.

F.4.7.4.6 Originator.

Service User.

F.4.7.4.7 Errors/Exceptions.

None.

F.5 ALLOWABLE SEQUENCE OF SERVICE PRIMITIVES.

Intentionally left blank.

F.6 UTILIZATION OF MAC BUILDING BLOCKS.

Services	Parameters	SINGGAR S SC-PT	SINGGARS All Except SC-PT	WDW NB/WB	HQ 1/2	LOS	HF ALE	DAMA	DASA	VRC-99	Total Users
MAC Common Utility	Set Mode Activate Channel Set RF Power Control	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	7
TRANSEC	Fill Zeroize TOD WOD Note 2	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	7
CH Error Control (FEC, interleaver, scrambler, etc)	Data Mode (1 to n) Note 1	No	Yes	Yes	Yes (Error detection only)	No	Yes	Yes	Yes	P Note 3	6
CH Access	Slot No. (ULString) Channel ID	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	6
MAC Addressing	Own Address (ULString)	No	Yes	Yes	No	No	Yes	Yes	Yes	P	5
Terminate Reception	Enable/ Disable	No	Yes	No	No	No	Yes	No	No	Yes	3
QOS	QOS (1 to n) (Read Only) Note 1	No	Yes	Yes	No	No	Yes	Yes	No	P	4

Note 1 – Enumerated list
Note 2 – Sequence of name/value pairs
Note 3 – Real-time interface via packet interface

F.7 PRECEDENCE OF SERVICE PRIMITIVES.

Precedence of primitives is left to service definitions of full APIs.

F.8 SERVICE USER GUIDELINES.

Service User guidelines are left to service definitions of full APIs.

F.9 SERVICE PROVIDER-SPECIFIC INFORMATION.

Not applicable.

F.10 IDL.

The IDL for an API is generated using the parameterized Classes of the building blocks to generate concrete classes with Waveform specific types and attributes. The Building Block documented herein is not intended to be instantiated directly in a UML diagram. The parameterized classes define the attributes and data types that are unique for each waveform.

The parameterized class is a template from which to generate user specific API definitions. To generate valid IDL from this Building Block, a concrete class must be generated that replaces the parameterized items with the user specific types and attributes. The completed UML diagram will not contain any reference to parameterized classes. Only concrete classes that were derived from them.

F.11 UML.

Not applicable.