



Reference Point 1 Specification

Version 2.1

Table of Contents

1	Summary of changes	6
2	Scope	8
3	Control / Management Plane Topology	9
4	Control / Management Protocol Architecture	9
4.1	Basic Principles	9
4.2	Physical Layer (Layer 1)	9
4.3	Data Link Layer (Layer 2)	9
4.4	Network Layer (Layer 3)	9
4.5	Transport Layer (Layer 4)	10
4.6	Upper Layers	10
5	Control Plane	11
5.1	BTS External Protocols	11
5.2	BTS Internal Protocols	14
5.2.1	Design Principles	14
6	Management Plane	18
6.1	BTS External Protocols	18
6.2	BTS Internal Protocols	19
7	OAM&P Interfaces- General	21
7.1	Modeling XML Schema in OBSAI	22
7.1.1	Element Definition	22
7.1.2	Attribute Definitions	23
8	RP1 - Synchronization Plane	25
8.1	Synchronization Hierarchy	25
8.2	Time and Frequency Reference	25
8.2.1	Transport Module	25
8.2.2	Proprietary Module (PM)	26
8.3	System Clock	26
8.4	Time Synchronization	27
8.4.1	Synchronization Burst Format	27
8.5	Synchronization Network- Physical Layer	30
8.6	Synchronization Network- Topology	31

Appendix A	: UDPCP Specification and Example of SOAP Binding.....	32
Appendix B	Fault Management.....	50
Appendix C	Performance Management.....	58
Appendix D	Configuration Management.....	84
Appendix E	Software Management	106
Glossary	115
	Abbreviations	115
	Definition of Terms.....	121
References	123

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

FOREWORD

OBSAI description and specification documents are developed within the Technical Working Group of the Open Base Station Architecture Initiative Special Interest Group (OBSAI SIG). Members of the OBSAI TWG serve voluntarily and without compensation. The description and specifications developed within OBSAI represent a consensus of the broad expertise on the subject within the OBSAI SIG.

The OBSAI SIG uses the following terminology in the specifications:

- "shall" expresses a provision that is binding
- "should" and "may" expresses non-mandatory provisions
- "will" expresses a declaration of purpose on the part of the OBSAI SIG. It may be necessary to use "will" in cases where the simple future tense is required

Use of an OBSAI reference or specification document is wholly voluntary. The existence of an OBSAI Specification does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the OBSAI Specification. Furthermore, the viewpoint expressed at the time a specification is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Every OBSAI Specification is subjected to review in accordance with the Open Base Station Architecture Initiative Rules And Procedures.

Implementation of all or part of an OBSAI Specification may require licenses under third party intellectual property rights, including without limitation, patent rights (such a third party may or may not be an OBSAI Member). The Promoters of the OBSAI Specification are not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

The information in this document is subject to change without notice and describes only the product defined in the introduction of this documentation. This document is intended for the use of OBSAI Member's customers only for the purposes of the agreement under which the document is submitted, and no part of it may be reproduced or transmitted in any form or means without the prior written permission of OBSAI Management Board. The document has been prepared for use by professional and properly trained personnel, and the customer assumes full responsibility when using it. OBSAI Management Board, Marketing Working Group and Technical Working Group welcome

1 customer comments as part of the process of continuous development
2 and improvement of the documentation.

3 The information or statements given in this document concerning the
4 suitability, capacity, or performance of the mentioned hardware or
5 software products cannot be considered binding but shall be defined in
6 the agreement made between OBSAI members. However, the OBSAI
7 Management Board, Marketing Working Group or Technical Working
8 Group have made all reasonable efforts to ensure that the instructions
9 contained in the document are adequate and free of material errors and
10 omissions.

11 OBSAI liability for any errors in the document is limited to the
12 documentary correction of errors. OBSAI WILL NOT BE
13 RESPONSIBLE IN ANY EVENT FOR ERRORS IN THIS DOCUMENT
14 OR FOR ANY DAMAGES, INCIDENTAL OR CONSEQUENTIAL
15 (INCLUDING MONETARY LOSSES), that might arise from the use of
16 this document or the information in it.

17 This document and the product it describes are considered protected by
18 copyright according to the applicable laws. OBSAI logo is a registered
19 trademark of Open Base Station Architecture Initiative Special Interest
20 Group. Other product names mentioned in this document may be
21 trademarks of their respective companies, and they are mentioned for
22 identification purposes only. Copyright © Open Base Station
23 Architecture Initiative Special Interest Group. All rights reserved. Users
24 are cautioned to check to determine that they have the latest edition of
25 any OBSAI Specification.

26 Interpretations: Occasionally questions may arise regarding the
27 meaning of portions of standards as they relate to specific applications.
28 When the need for interpretations is brought to the attention of OBSAI,
29 the OBSAI TWG will initiate action to prepare appropriate responses.
30 Since OBSAI Specifications represent a consensus of OBSAI Member's
31 interests, it is important to ensure that any interpretation has also
32 received the concurrence of a balance of interests. For this reason
33 OBSAI and the members of its Technical Working Groups are not able
34 to provide an instant response to interpretation requests except in those
35 cases where the matter has previously received formal consideration.

36 Comments on specifications and requests for interpretations should be
37 addressed to:

38 Peter Kenington
39 Chairman, OBSAI Technical Working Group
40 Linear Communications Consultants Ltd.
41 Email: pbk@linearcomms.com

1 Summary of changes

Version	Approved by	Date	Comment
1.0	OBSAI Management Board	10/17/2003	Initial Release
1.05	Draft	08/19/2004	<p>Updated BTS Reference Architecture, .</p> <p>Updated System Clock parameters in sections 8.3 and 8.4.</p> <p>Defined the CRC generator polynomial for Synchronization Burst messages in sections 8.4.1.4</p> <p>Updated UDPCP Specification in Appendix A. See section A.6 for detailed UDPCP revision history.</p> <p>Updated OAM&P – Interfaces General section.</p> <p>Appended Fault Management Specification in Appendix B.</p> <p>Appended Performance Management Specification in Appendix C.</p> <p>Appended Configuration Management Specification in Appendix D.</p> <p>Appended Software Management Specification in Appendix E.</p>
1.1	OBSAI Management Board	08/19/2004	Updated release
1.1.06	Draft	02/24/2005	<p>Added reference to Test Messages Specification (section 7 and References)</p> <p>Definition of “change” operation of <modifyParameterReq> corrected in D.4.5.1</p> <p>Modified UDPCP protocol. Version number incremented to 002.</p>
1.1.07	Proposal approval	06/15/2005	Agreed SW updates included

2.0	OBSAI Management Board	04/27/2006	CRC initialization change WiMAX synchronization burst inclusion Editorial improvements for synchronization New architecture diagram incorporated
2.01	Draft	05/24/07	Minor editorial corrections and minor additions to incorporate LTE
2.02	Draft	03/03/08	Minor updates to pictures
2.1	OBSAI Management Board	07/14/2008	LTE Update release

2 Scope

This document specifies Reference Point 1 characteristics for the Control Plane (C-Plane, also known as "signaling") and Management Plane (M-Plane, also known as "OAM&P").

The User Plane (U-Plane, also known as "payload") belongs to RP2 and RP3.

Nevertheless, RP1 and RP2 shall use a common BTS internal transport infrastructure. That means that the same protocol specifications (layer 1 to 3) apply for both Reference Points. In this document, references are made to respective chapters of the "Reference Point 2 Specification".

Chapter 3 and Chapter 4 of this specification describe the Reference Point 1 Control and Management Plane Topology and Architecture. Chapter 5 and chapter 6 of this specification describe the Reference Point 1 characteristics for the Control Plane and Management Plane. Chapter 7 describes general characteristics of OAM&P interfaces. Chapter 8 describes the Reference Point 1 characteristics for the Synchronization Plane.

1 **3 Control / Management Plane Topology**

2 This applies to both RP1 and RP2. Refer to “Reference Point 2
3 Specification”.

4 **4 Control / Management Protocol Architecture**

5 **4.1 Basic Principles**

6 This applies to both RP1 and RP2. Refer to “Reference Point 2
7 Specification”.

8 OBSAI RP1 uses the IP family of protocols, for example bit presentation
9 uses network bit order (which is generally used in the IP family of
10 protocols). Principles from the IP family of protocols shall also be used
11 within OBSAI.

12 **4.2 Physical Layer (Layer 1)**

13 This applies to both RP1 and RP2. Refer to “Reference Point 2
14 Specification”.

15 **4.3 Data Link Layer (Layer 2)**

16 This applies to both RP1 and RP2. Refer to “Reference Point 2
17 Specification”.

18 **4.4 Network Layer (Layer 3)**

19 This applies to both RP1 and RP2. Refer to “Reference Point 2
20 Specification”.

1

2 4.5 Transport Layer (Layer 4)

3 Interfaces associated with RP1 shall support the following Transport
4 Layer protocols for the Control & Management Plane, appropriate to the
5 Upper Layer (see section 4.6) protocol in use:

6

7

- TCP [RFC793]

8

- SCTP [RFC2960]

9

- UDPCP [Appendix A]

10 4.6 Upper Layers

11 RP1 related Upper Layer protocols fall into following categories:

12

1. BTS external Control Plane protocols (see chapter 5.1)

13

14

2. BTS internal Control Plane protocols (see chapter 5.2)

15

16

3. BTS external Management Plane protocols (see chapter 6.1)

17

18

4. BTS internal Management Plane protocols (see chapter 6.2)

19

20

5 Control Plane

5.1 BTS External Protocols

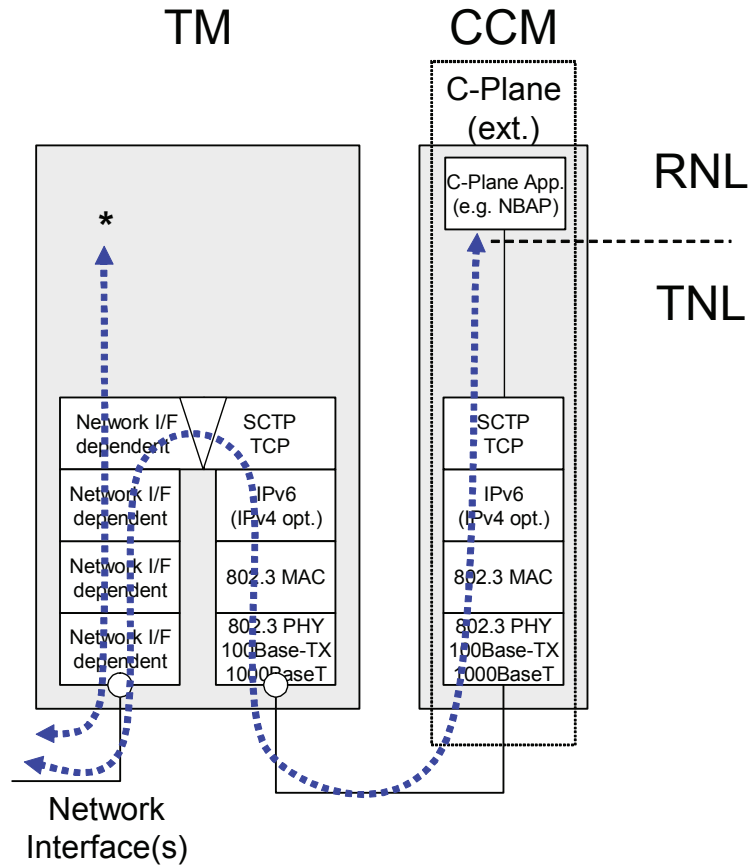
External Radio Network Layer (RNL) protocols have been defined by standardization bodies dedicated to the specification of RAN/BSS functions. They are beyond the scope of OBSAI. 3G examples are

- 3GPP Iub NBAP [3GPP TS 25.433]
- 3GPP2 Abis/A3

External Control Plane protocols are terminated at the CCM. The TM shall pass through respective PDU's transparently and shall not perform any protocol termination or conversion of RNL protocols.

The corresponding protocol architecture may be different depending upon the characteristics of the Network Interface. The TM shall perform an interworking function at the transport layer to adapt the external protocol to either SCTP/IP or TCP/IP if the transport protocol at the network interface is different from that. The TM shall further encapsulate the data into the network, data-link, and physical layers as defined in the following figure.

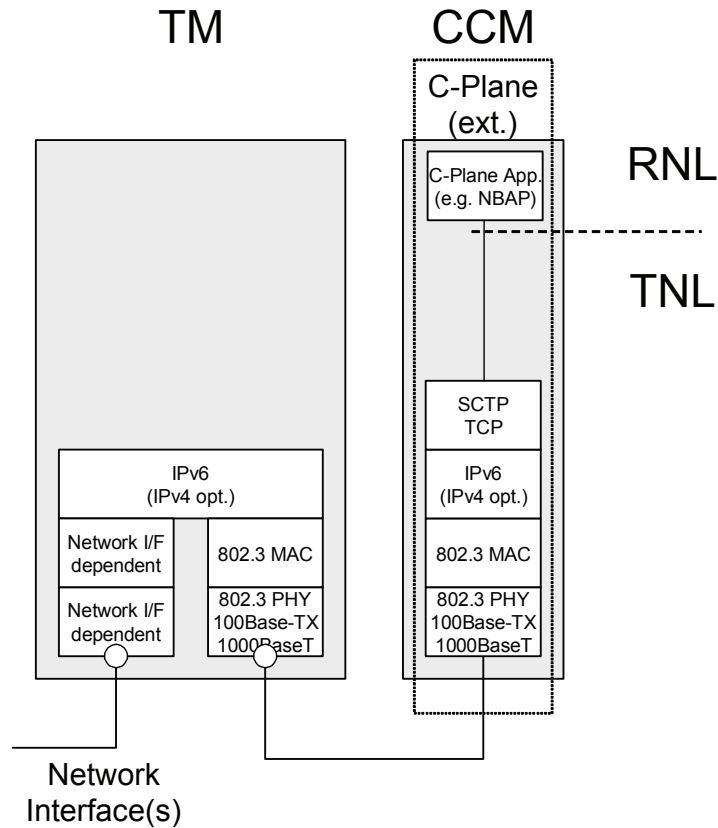
It is permissible to extend SCTP/IP or TCP/IP to the baseband module where required. This can be beneficial in, for example, 3GPP LTE systems.



1
2
3
4
5
6
7
8

Figure 1 Protocol Architecture- TM Provides Transport Layer IWF

* Additional termination points may be located in TM, depending on the Network Interface type.



1
2
3
4
5
6
7
8
9

Figure 2 Protocol Architecture- TM Does Not Provide Transport Layer IWF

For example, if the Network Interface is IP based (e.g. with 3GPP Rel.5), the Transport Module acts as an IP router. Transport Layer PDU's pass through transparently. If the Network Interface is ATM based, the TM shall adapt the external Control Plane protocols to BTS internal IP transport.

1 **5.2 BTS Internal Protocols**

2 Internal call processing signaling between the CCM and the other
3 modules occurs over the RP1 Control plane. This signaling covers
4 management of radio and other system resources of the OBSAI BTS.

5 **5.2.1 Design Principles**

6 The OBSAI architecture for supporting communication between the
7 CCM and the other modules in the BTS shall support a direct interface,
8 an indirect interface or a combination of both. Regardless of the
9 interface chosen, the communication shall be socket based.

10 For all internal communications, TCP or UDPCP shall be used at the
11 transport layer.

12 Module resource management application (MRMA) software
13 responsible for module level resource management shall reside on the
14 local hardware (for example, the BBM MRMA shall reside on the BBM).
15 The overall management of the BTS level radio and shared system
16 resources shall be performed by the BTS resource manager (BRM)
17 resident on the CCM. The MRMAs shall communicate with the BRM
18 via a direct interface or an indirect interface as described in sections
19 5.2.1.1 and 5.2.1.2 respectively.

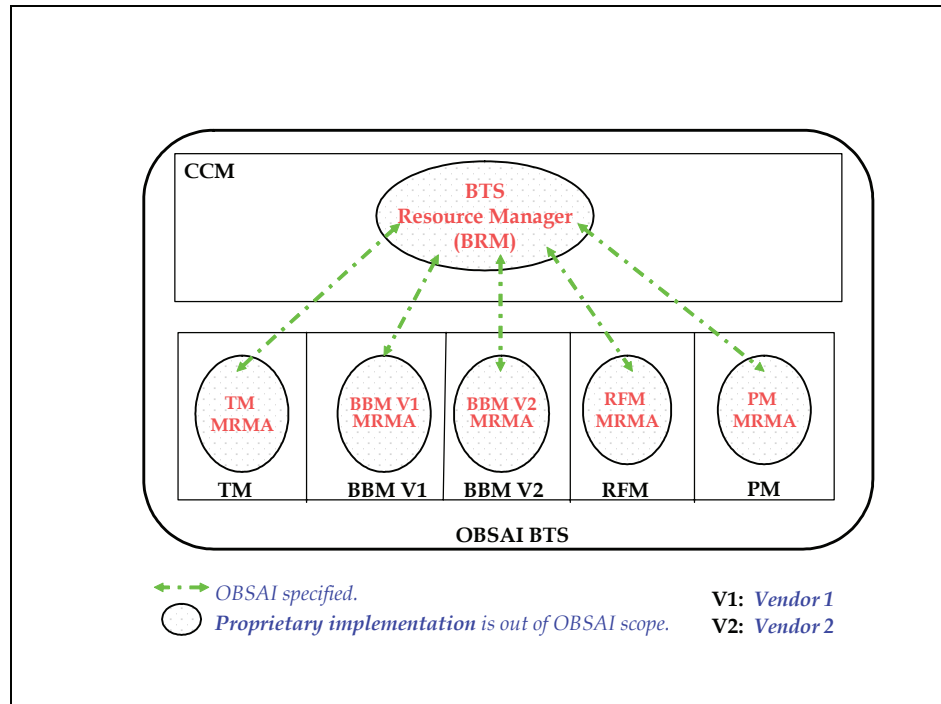
20 All module vendors shall be responsible for providing the MRMA
21 resident on their modules. However, the BRM shall be the
22 responsibility of the BTS integrator. The implementation details of the
23 MRMA and BRM are out of scope of this specification.

24

1 **5.2.1.1 Direct Interface**

2 The direct interface as shown in Figure 3 implies direct communication
3 between the BRM and the MRMAs. This interface is undefined above
4 the transport layer in this release of the RP1 specification.

5



6

7

Figure 3 Direct Interface

8

9

5.2.1.2 Indirect Interface

The indirect interface shown in Figure 4 implies that the communication between the BRM and the MRMAs occurs indirectly via proxies. The module vendor shall provide the MRMA Proxy software resident on the CCM in addition to the MRMA (i.e., the BBM vendor shall provide the BBM MRMA as well as the BBM MRMA Proxy).

A single MRMA Proxy shall be loaded on the CCM for each module type. For example, in a BTS with multiple CDMA BBMs, a single instantiation of the BBM MRMA Proxy shall reside on the CCM or the two TMs of a redundant BTS shall be served by a single TM MRMA Proxy. The interface between the BRM and the MRMA proxies may be OBSAI specified or BTS integrator proprietary.

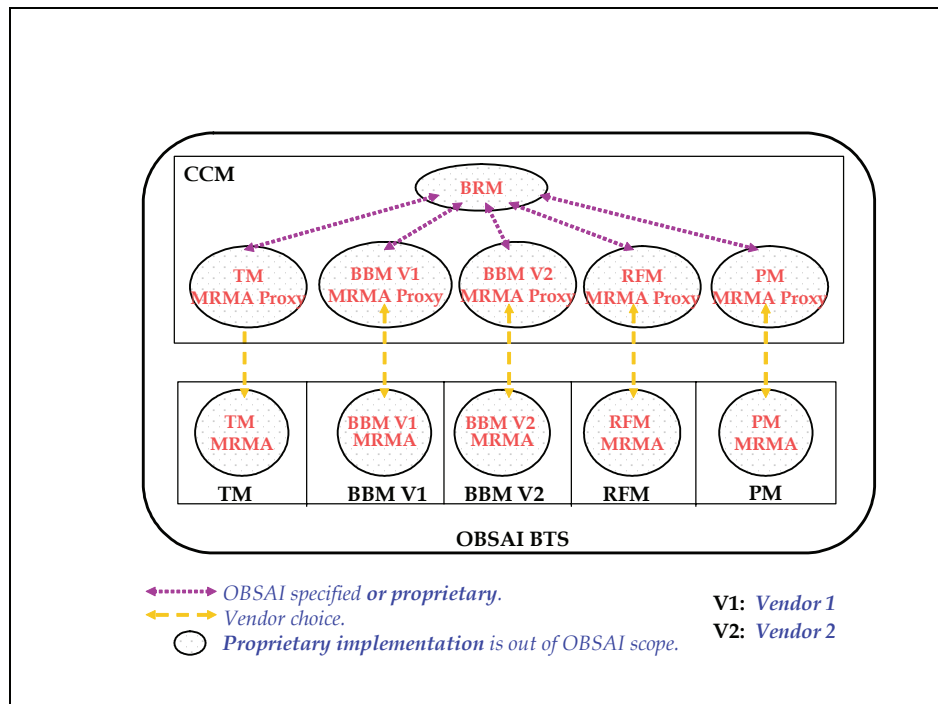
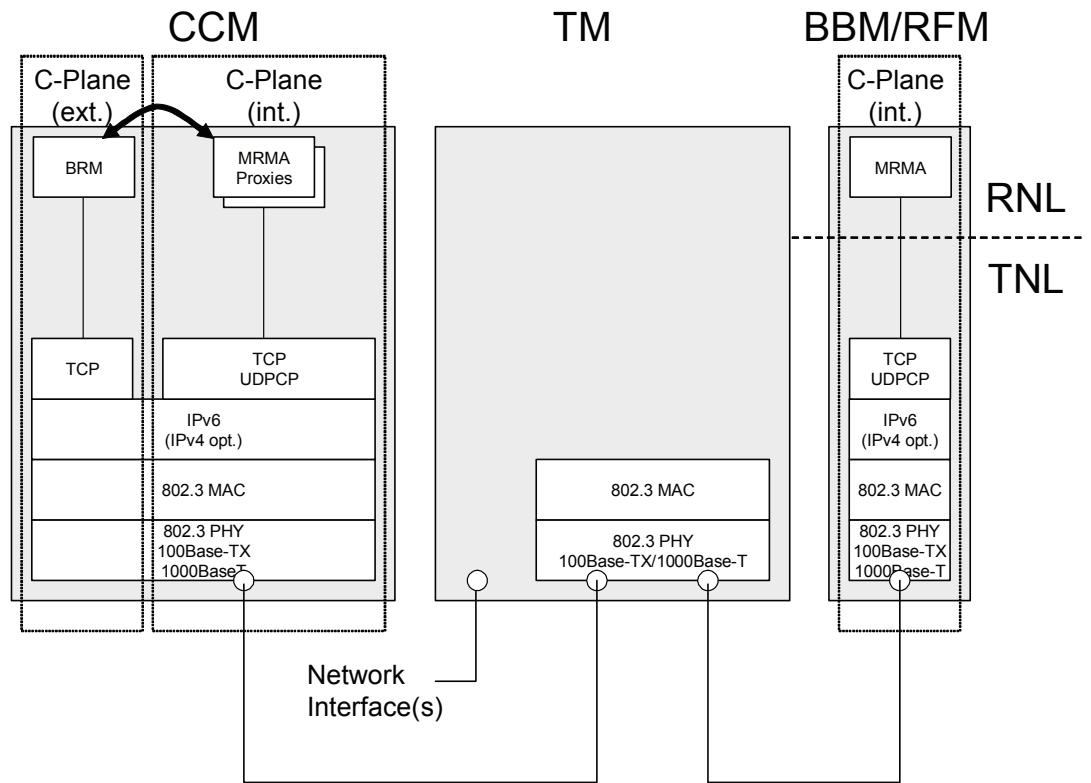


Figure 4 Indirect Interface

With this revision of the RP1 specification, it is expected that all modules of the same function/mode (e.g., TM, CCM) will be manufactured by the same vendor. This will enable effective resource management of N instances of a particular module type without specifying protocols and messages required for inter-module communication across different vendors' modules. Concurrent BTS operation with more than one access technology can be accommodated by loading appropriate MRMA proxies.



1
2
3
4
5

Figure 5 Indirect Interface Examples: BBM/RFM Signaling

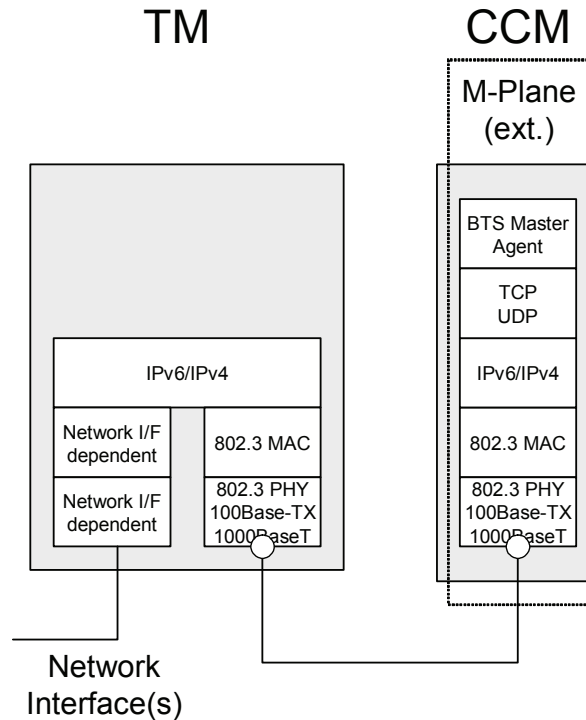
1 **6 Management Plane**

2 **6.1 BTS External Protocols**

3 Although external OAM&P protocols (e.g. SNMP, TELNET/CLI) are not
4 specified by OBSAI, they are assumed to be IP based. The protocols
5 are terminated at the CCM.

6 There are no conceptual differences with regard to the characteristics of
7 the Network Interface (ATM or IP). The Transport Module always acts
8 as an IP router for external OAM&P protocols. In case of an ATM based
9 Network Interface, OAM&P related IP datagrams are adapted to ATM
10 cells (AAL5) for transport over the RAN.

11 The figure below illustrates the protocol architecture.
12



1
2

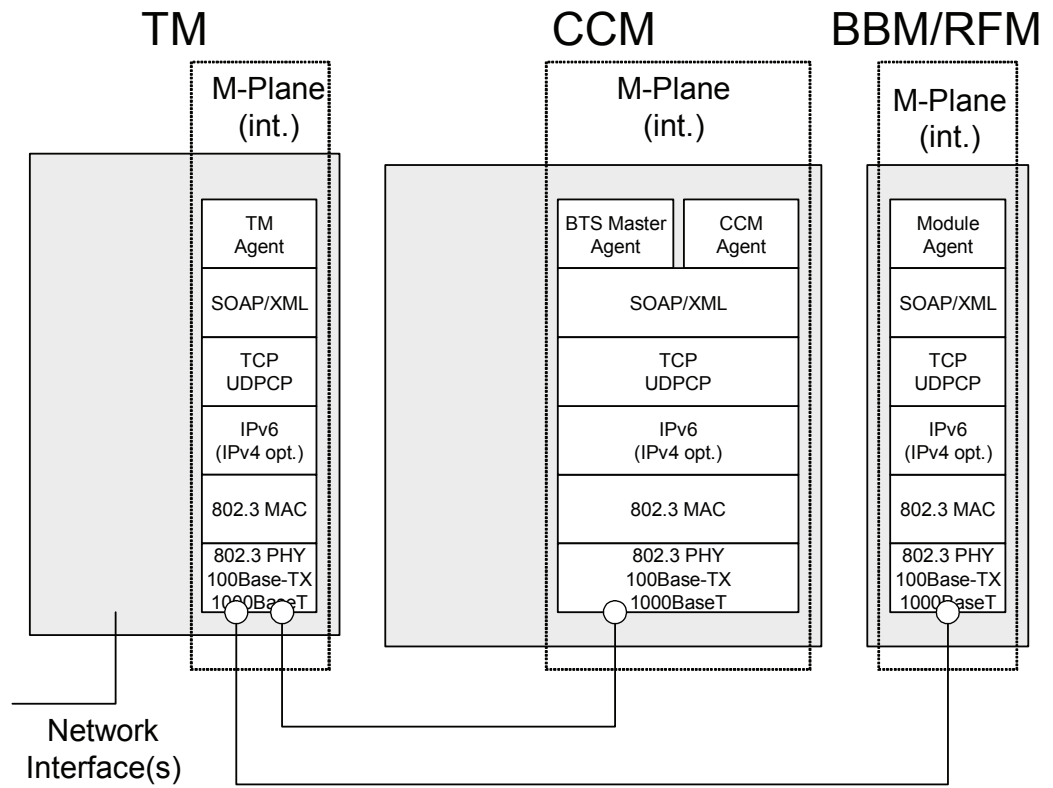
Figure 6 External Management Plane

3 **6.2 BTS Internal Protocols**

4 Control, status and alarm data sent between a BTS Master Agent and
 5 the Module Agents shall be coded in Simple Object Access Protocol
 6 [SOAP] messages. A corresponding parser does not need to be a
 7 validating one.

8 SOAP can potentially be used in combination with a variety of transport
 9 protocols. The transport protocol options specified by OBSAI are TCP
 10 and UDPCP.

11 The figure below illustrates the protocol architecture that supports
 12 communication between the BTS Master Agent and Module Agents.
 13



1
2
3

Figure 7 Internal Management Plane

7 OAM&P Interfaces- General

This section specifies aspects common to all OAM&P functions at the RP1 interface. Definition of terms associated with each aspect of OAM&P is in accordance with those provided by ITU, 3GPP, 3GPP2 and other industry management standards. These concepts and the corresponding SOAP message structures are defined.

It is assumed that the reader is familiar with the following:

- OAM&P principles described in the OBSAI System Reference Document.
- General concepts of XML and SOAP defined by W3C (www.w3c.org).

The messages are based on usage scenarios specified by the W3C.

All SOAP header elements must be namespace qualified. For example, for RP1 the namespace is “http://www.obsai.org/oamp/2004/09/rp1”.

Header

The `Header` element contains the following elements:

Element Name	Usage
to	Specifies the address of the message receiver.
from	Specifies the address of the message sender.
Id	The id is used by the sender to uniquely identify the request or notification. The value is assigned by the sender during creation of the request or notification. The sender has to ensure that the value is unique in the scope of the sender.
RelatesTo	Reference identifier to correlate request/response message pairs. If the receiver receives the Request message and the id is non-null and non-empty, it copies it to the Response or Exception message (i.e. receiver shall reply with same string by copying the received value to the response or error message).

Action	<p>Defines the intent/type of the message as follows:</p> <p>Fault Management: "OBSAI_FM"</p> <p>Performance Management: "OBSAI_PM"</p> <p>Configuration Management: "OBSAI_CM"</p> <p>Software Management: "OBSAI_SwM"</p> <p>Test: "OBSAI_Test". Test messages are specified in [OBSAI_Test].</p> <p>Other vendor specific values may also be specified.</p>
Version	<p>Specifies the version of the OAM&P fragment that this message belongs to.</p> <p>For example, if the action is "OBSAI_FM" and version is 1.1, then this message is an FM message from FM fragment version 1.1.</p>
responseRequested	<p>If included in the header, this indicates that the receiver is required to respond to the message.</p>

1

2

3

Body

4

The `Body` element contains the payload of the message (i.e. where the main end-to-end information is conveyed). The tag of the first element in the body in a message shall contain the message name.

5

6

7

All timestamp information in the body of OBSAI OAM&P messages shall be formatted in Coordinated Universal Time (UTC) according to XML datatype "dateTime" (e.g. YYYY-MM-DDThh:mm:ssZ).

8

9

10 **7.1**

Modeling XML Schema in OBSAI

11

The following section provides guidelines for modeling XML Schema for the OBSAI OAM&P specification. This guideline provides recommended practices on how to model the XML schema that is used to validate the XML file.

12

13

14

15

The following design guidelines for XML elements are applicable in the scope of OAM&P.

16

17 **7.1.1**

Element Definition

18

Element Definitions specify a type definition for an element by either reference or explicit Datatype. The element type can be a built-in data type defined in the XML Schema language or a user-defined Datatype

19

20

1 (using the `simpleType` or `Complex Type` element definitions). Below
2 is an example of an element definition:

```
3  
4 <element name="<elementName>" type = "<Datatype>"  
5 default="<Value>" nillable="true" minOccurs=0  
6 maxOccurs=1/>
```

7
8 The first letter of the `<elementName>` shall be lower case and any
9 subsequent word that is embedded in the name shall begin with an
10 upper case letter followed by a lower case letter.

11 In case the element contains no valid data (e.g. "null" value in a
12 database table) it shall be marked by using the `nillable` attribute in
13 XML.

```
14  
15 <element name="code" nillable="true"/>
```

16
17 If the element can be empty (i.e. no value) then the `minOccurs`
18 attribute shall be used.

19 7.1.2 Attribute Definitions

20 The first letter of the `<attributeName>` name is to be lowercase.
21

22 7.1.2.1 Complex Type/Simple Type Definitions

23
24 Type definitions in XML schema are defined using the `Complex Type`
25 `element` and `simpleType` element. The first letter of the `<Complex`
26 `TypeName>` and `<SimpleTypeName>` name is to be an upper case
27 letter followed by a lower case letter, and any subsequent word that is
28 embedded in the name shall begin with an upper case letter followed by
29 a lower case letter.
30

31 7.1.2.2 Comments in XML schema

32
33 Comments in the XML schema are represented using the annotation
34 tag:

```
35  
36 <annotation>  
37 <documentation>Documentation Text</documentation>  
38 </annotation>
```

- 1 **7.1.2.3 Modeling Date types in XML**
- 2 XML built-in datatypes as defined by W3C shall be used.
- 3 **7.1.2.4 Modeling Arrays in XML**
- 4 XML built-in datatypes as defined by W3C shall be used.
- 5 **7.1.2.5 Modeling Enumerations in XML**
- 6 XML built-in datatypes as defined by W3C shall be used.
- 7

8 RP1 - Synchronization Plane

8.1 Synchronization Hierarchy

The synchronization hierarchy and basic principles are described in the "OBSAI System Reference Document"

8.2 Time and Frequency Reference

8.2.1 Transport Module

The TM shall provide synchronization information, such as clock and Synchronization Status Messages (SSM) extracted from the Network Interface. The TM shall provide the derived clock and SSM to the CCM for further processing to generate the BTS system clock and time reference signals.

In addition, the TM may be able to synchronize the Network Interfaces based on the following:

- System clock provided by CCM
- A clock reference recovered from a selected ingress Network Interface
- A free running clock

The TM and the CCM shall negotiate Clock and Frequency Reference selection to select proper reference clock input where there is more than one source available.

1 8.2.2 Proprietary Module (PM)

2 A PM may take many forms, although one likely form is that of a module
3 to provide system time and a frequency reference, based on Global
4 Positioning System (GPS) time. If used for this purpose, the PM shall
5 provide the following signals to the CCM:

- 6 • A system time reference synchronous to UTC time (except for
7 leap seconds) with the same time origin as GPS time. This
8 information is transported in a System Time Message (STM)
9 over the C-Plane infrastructure.
- 10 • A one pulse per second (PPS) reference clock, synchronized to
11 GPS, over a dedicated link. This signal serves to validate the
12 STM (mentioned above) providing the means of delivering highly
13 accurate system time information to the CCM. Additionally, PPS
14 serves as a frequency reference from which the CCM can
15 generate the system clock.

16 8.3 System Clock

17 The CCM shall provide a common system clock to the BBM, RFM, TM
18 and PMs. This common system clock will be used as a frequency
19 reference for all modules in the system and shall support concurrent
20 operation of WCDMA, CDMA2000, GSM/EDGE, WiMAX and 3GPP
21 LTE. The system clock is delivered through the use of a single clock
22 network.

23 The CCM shall provide the system clock to all modules contained within
24 its shelf. This shall be accomplished over dedicated, point-to-point
25 links. Additionally, the CCM shall provide one extra system clock output
26 for the purposes of delivering a frequency reference to an Extension
27 Module that may be located in an additional shelf.

28 All modules receiving the system clock shall contain two clock inputs-
29 one from each of the CCMs that comprise a redundant pair.

30 See Figure 11 for a description of the synchronization network topology.

31 The system clock shall be a digital signal operating with the parameters
32 described in Table 1. Additionally, the System Clock shall conform to
33 requirements for wander generation, tolerance, and transfer according
34 to G.812, Type II.

35

1

Table 1 System Clock Operating Parameters

Parameter	Symbol	Values			
		Min	Typical	Max	Unit
Frequency	F_{CLK}	-0.016 ppm	30.72	+0.016 ppm	MHz
Duty Cycle	T_{DUTY_CYCLE}	40	50	60	%
PP Jitter	$T_{P-P\ JITTER}$	--	400	600	ps

2

8.4 Time Synchronization

3

4

5

6

7

8

9

10

Over a common synchronization network, the CCM shall provide frame timing and time stamping, for each of the air interface systems, independently. The time synchronization shall be provided via periodic synchronization bursts containing air-interface-specific frame numbers and time references. The CCM shall periodically cycle through each type of burst necessary for synchronization of the BTS. The CCM shall provide bursts with a minimum of 30 milliseconds between any two bursts.

11

8.4.1 Synchronization Burst Format

12

13

14

15

The synchronization bursts shall follow the message format in Figure 8, and conform to the frame timing in Figure 9. The LSB of each field shall be transmitted first. The length of each synchronization burst bit shall be eight periods of the System Clock.

Start	Type	Type Specific Information	CRC	End
1	8	64	16	1

16

17

Figure 8 Synchronization Burst Format

18

19

20

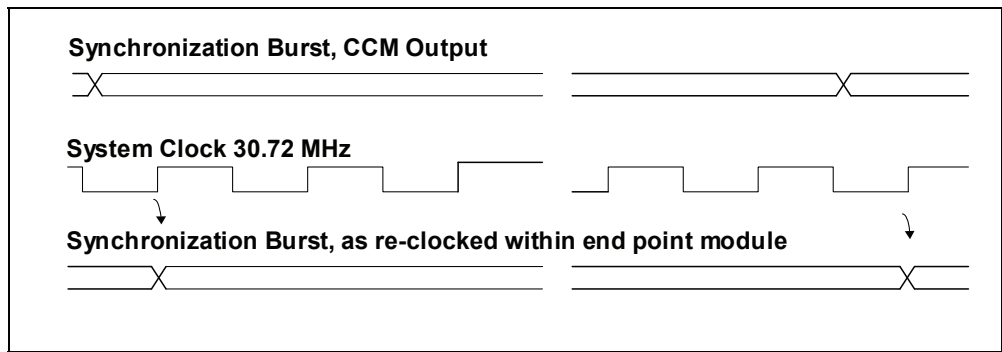


Figure 9 Synchronization Burst Frame Timing

1
2
3
4
5
6
7

8.4.1.1 Start and End Definition

The Start and End bits shall be one bit in width and shall both be set to a “1”. The link shall be set to “0” between messages.

8.4.1.2 Type Field Definition

Type	Bits
Not Used	00h
RP3 Bus (FDD) Frame Number	01h
WCDMA/FDD Frame Number	02h
GSM/Edge1 Frame Number	03h
GSM/Edge2 Frame Number	04h
GSM/Edge3 Frame Number	05h
WCDMA/TDD Frame Number	06h
CDMA2000 Frame Number	07h
Time of Day	08h
Reserved	09h – 7Fh
802.16 Frame Number, 2 ms Frame Duration	80h
802.16 Frame Number, 2.5 ms Frame Duration	81h
802.16 Frame Number, 4 ms Frame Duration	82h
802.16 Frame Number, 5 ms Frame Duration (WiMAX Mandatory)	83h
802.16 Frame Number, 8 ms Frame Duration	84h

802.16 Frame Number, 10 ms Frame Duration	85h
802.16 Frame Number, 12.5 ms Frame Duration	86h
802.16 Frame Number, 20 ms Frame Duration	87h
3GPP LTE Frame Number	88h
Spare	89h - FFh

1
2
3
4
5
6
7
8
9

Figure 10 Synchronization Burst Format- Type Field

Notes:

1. Any module receiving a synchronization burst of type “Spare” shall ignore the burst.
2. Any module receiving a synchronization burst of type “Reserved” shall send an appropriate alarm to the CCM.

10 **8.4.1.3 Type Specific Information Field Definition**

11
12

Frame Numbering

Type	Type Specific Information
01h-07h 80h-88h	Frame Number

13
14
15
16
17
18
19

Type Specific Information:

- Frame Number: Frame Number with reference to the corresponding air interface (64 bit, unsigned integer)

Time of Day Stamping

Type	Type Specific Information			
08h	Spare	TODSrc	LeapSecs	GPSTime

20
21
22

Type Specific Information:

- Spare: Unused bits, shall be set to zero by the CCM. (16 bit)

- 1 • TODSrc: Time Of Day Source (8 bit, unsigned integer) shall be
2 encoded as
- 3 ○ 01h: GPS
- 4 ○ 02h: NTP (Network Time Protocol over Network Interface)
- 5 ○ 03h: Real Time Clock
- 6 ○ 04h: Time Since Power-Up (CCM keeps track of the
7 elapsed time since the last power cycle)
- 8 • LeapSecs: Time, in seconds, between GPS and UTC time (8 bit,
9 signed integer)
- 10 • GPSTime: GPS Time, seconds since January 6, 1980 (32 bit,
11 unsigned integer)
- 12

13 **8.4.1.4 CRC Field Definition**

14

15 The CRC field shall be a 16 bit sequence calculated over the *Type* and
16 *Type Specific Information* fields of the message, starting with the left-
17 most bit of the Type field (Figure 8). The following generator polynomial
18 is used:

19
$$x^{16} + x^{12} + x^5 + 1$$

20 The initial value of the CRC shall be 0x0000.

21

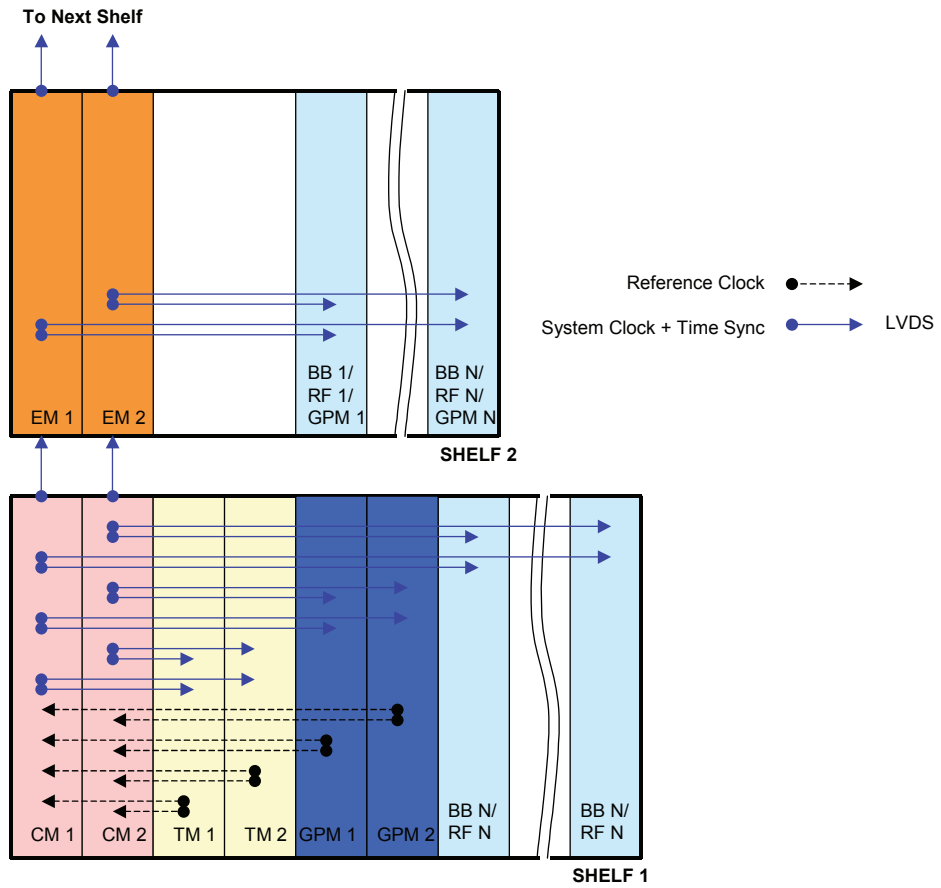
22 **8.5 Synchronization Network- Physical Layer**

23 The physical layer for the synchronization network, both system clock
24 and time synchronization (synchronization bursts) shall be LVDS.

1 **8.6 Synchronization Network- Topology**

2 The following figure details the synchronization network topology.

3



4

5 **Figure 11 Synchronization Network Topology**

6

7

Appendix A : UDPCP Specification and Example of SOAP Binding

A.1 Scope

This appendix describes the UDP-based Communication Protocol (UDPCP) that is designed to meet the needs of Base Station internal communications. UDPCP provides reliable and unreliable transport services over UDP/IP so that it can be applied as a general purpose transport solution in Base Stations.

In addition to the UDPCP specification this document also provides an example of XMLP (SOAP) layer binding to the UDPCP that is compliant with the W3C "XML Protocol Abstract Model" specification.

The layered architecture of the UDPCP-based communication stack is illustrated below:

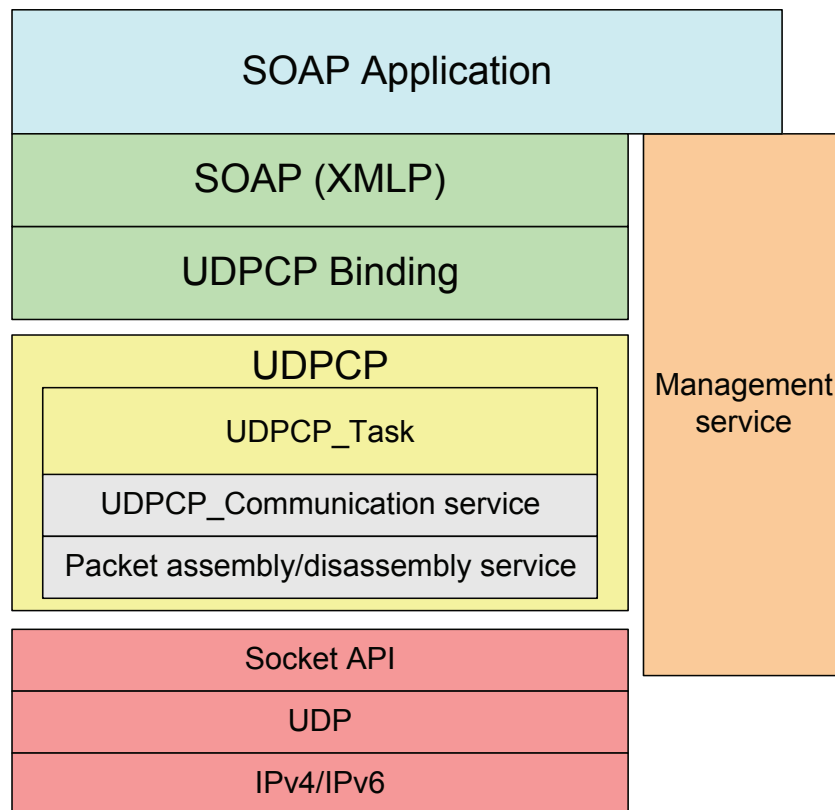


Figure 12 Layered architecture of the UDPCP-based communication stack.

1

2

A.2 UDPCP Features

3

The UDPCP communication service supports the following features:

4

- Connectionless communication for serial mode data transfer

5

- Acknowledged and unacknowledged transfer modes

6

- Retransmissions Algorithm

7

- Checksum Algorithm

8

- Fragmentation of long messages (dis-assembly/re-assembly) to match to the MTU during transport:

9

10

- Broadcasting and multicasting messages to multiple peers in unacknowledged transfer mode

11

12

A.3 UDPCP Transport Protocol

13

UDPCP supports application level messages up to 64 KBytes (limited by 16-bit packet data length field). Messages that are longer than the MTU shall be fragmented. The protocol places the following restrictions upon the fragment that is submitted to the underlying UDP layer:

14

15

16

17

- The size of a valid fragment shall be 12 octets minimum

18

- The maximum size of a valid fragment shall conform to the MTU.

19

UDPCP provides a reliable transport service that will perform message retransmissions in case transport failures occur

20

21

A.3.1 UDPCP Packet Types

22

There are two types of UDPCP packets:

23

- UDPCP Data Packet

24

- UDPCP Acknowledge Packet

25

A.3.2 Data Packet Transfer Modes

26

UDPCP supports the following:

27

- Acknowledged data packet transfer

28

- Unacknowledged data packet transfer

A.3.2.1 UDPCP Data Packet Format

A UDPCP Data Packet includes header and payload. The packet format is as shown below:

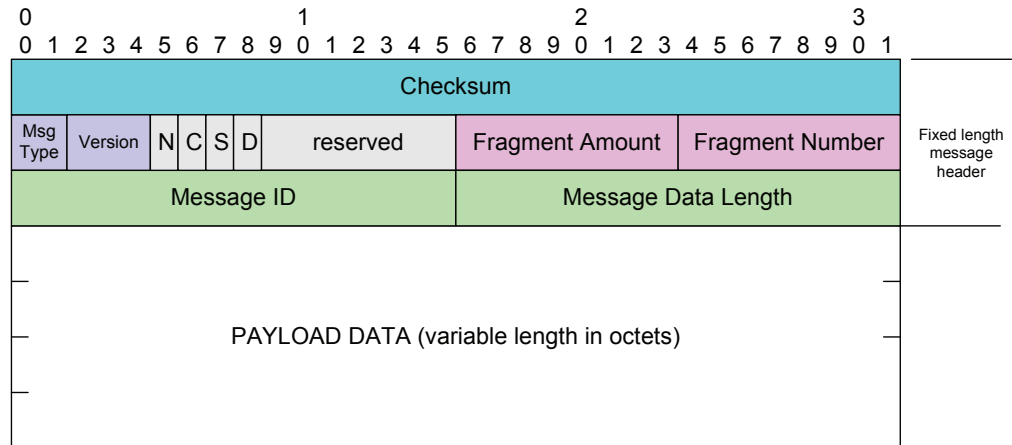


Figure 13 UDPCP Data Packet

Checksum: 32-bits (unsigned integer)

This field contains the Adler32 checksum value of the UDPCP Data packet. If the C-bit is set to 0 (i.e., checksum not in use), this field shall contain a value of 0.

Note: A sample C code for computing Adler-32 checksum can be found in RFC 1950 and RFC 2960

Msg Type: 2-bits

These bits indicate the message type. For a data packet this field shall be binary 01.

Version: 3-bits

These bits indicate the protocol version. Packets conforming to this version of this UDPCP specification shall have this field set to 002 binary.

Packet Transfer Options (N, C, S, D and Reserved): 11-bits

The individual bits of this field shall be used as follows:

- If the N-bit is 0, and the S-bit is 0, every packet shall be acknowledged.
- If the N-bit is 0 and the S-bit is 1 only the last fragment that completes the message shall be acknowledged.
- If the N-bit is 1 and S bit is "don't care", the packet or message shall not be acknowledged.

- 1 • If the C-bit is 1 the Checksum field contains the checksum
- 2 of the UDPCP packet. If C-bit is 0, the Checksum field
- 3 shall contain a value of 0.
- 4 • In a data packet the D-bit shall be set to 0.
- 5 • The Reserved-bits shall be set to 0.

6 Fragment Amount: 8-bits (unsigned integer)
 7 This field indicates the total number of fragments of a message. If
 8 a message is unfragmented this field may be 0 or 1.

9 Fragment Number: 8-bits (unsigned integer)
 10 This field shall be used to indicate the fragment number of the
 11 current packet in transport. The first fragment of the message shall
 12 have a value of 0.

13 Message ID: 16-bits (unsigned integer)
 14 This field contains a unique 16-bit message identifier. Generation
 15 of the message identifier is specified in A.3.2.3. Same Message ID
 16 value is used for all fragments belonging to the same message.

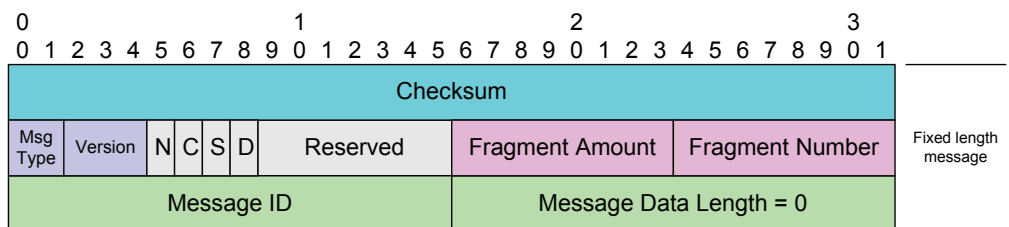
17 Message Data Length: 16-bits (unsigned integer)
 18 This field indicates the length of the payload data of the message in
 19 octets.

20 Payload Data: variable length in octets
 21 This field contains user data.

22 MSB = bit 0

23 A.3.2.2 UDPCP Acknowledge Packet Format

24 A receiver shall acknowledge reception of an error free packet or
 25 message in Acknowledged Transfer mode (N = 0). This packet has the
 26 following format:



27
 28 **Figure 14 UDPCP Acknowledge Packet**

29
 30 Checksum: 32-bits (unsigned integer)

- 1 This field contains the Adler32 checksum value of the UDPCP
2 Acknowledge packet. If the C-bit is set to 0 (i.e., checksum not in
3 use), this field shall contain a value of 0.
- 4 Msg Type: 2-bits
- 5 These bits indicate the message type. For an Acknowledge packet,
6 this field shall be binary 10.
- 7 Version: 3-bits
- 8 These bits indicate the protocol version. Packets conforming to this
9 version of this UDPCP specification shall have this field set to 002
10 binary.
- 11 Packet Transfer Options (N, C, S, D and Reserved): 11-bits
- 12 The N and S bits of this field shall be set. The other bits shall be
13 used as follows:
- 14 • If the C-bit is 1 the Checksum field contains the checksum
15 of the UDPCP Acknowledge packet. If C-bit is 0, the
16 Checksum field shall contain a value of 0.
 - 17 • In an acknowledge packet the D-bit shall be set to 0 the
18 first time a packet is acknowledged. The D bit may be set
19 to 1 to indicate a duplicate data packet.
 - 20 • The Reserved-bits shall be set to 0.
- 21 Fragment Amount: 8-bits (unsigned integer)
- 22 This field contains a copy of the total number of fragments of the
23 data packet being acknowledged.
- 24 Fragment Number: 8-bits (unsigned integer)
- 25 This field contains a copy of the fragment number of the data
26 packet being acknowledged.
- 27 Message ID: 16-bits (unsigned integer)
- 28 This field contains a copy of the unique 16-bit message identifier of
29 the packet being acknowledged.
- 30 Message Data Length: 16-bits (unsigned integer)
- 31 The message data length field shall be set to 0 in the Acknowledge
32 packet i.e. no payload bytes will be transported.

33 **A.3.2.3 Message ID Generation and Synchronization**

- 34 The transmitter shall guarantee that the Message ID in each message
35 sent to a receiver is unique. During normal operation, the Message ID
36 shall be non-zero.

1 However, to enable transmitter and receiver synchronization after a
 2 UDPCP reset, the first transmitted message shall have Message ID = 0
 3 and shall be sent using acknowledged mode. This message is used
 4 only for synchronization and is hereafter referred to as Sync Msg. The
 5 Sync Msg shall have the following:

- 6 • Type = Data Packet
- 7 • Message Data Length = 0
- 8 • N-bit = 0
- 9 • S-bit = 0

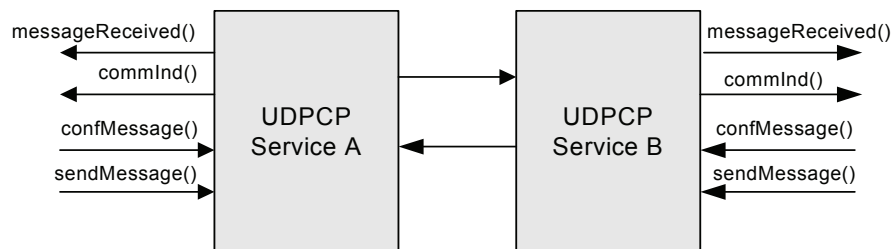
11 A.3.2.4 Duplicate Message Detection

12 Duplicate Messages are detected by the receiver if all of the following
 13 are the same as those of the recent fragment(s):

- 14 • Fragment Amount
- 15 • Fragment Number
- 16 • Message ID
- 17 • Source IP Address
- 18 • Source Port

19 The receiver should maintain the latest received message from each
 20 transmitter. The receiver shall acknowledge (with D-bit set to 1) and
 21 discard a duplicate fragment in acknowledge transfer mode.

22 A.4 UDPCP API



23
 24 **Figure 15 UDPCP API Usage**
 25

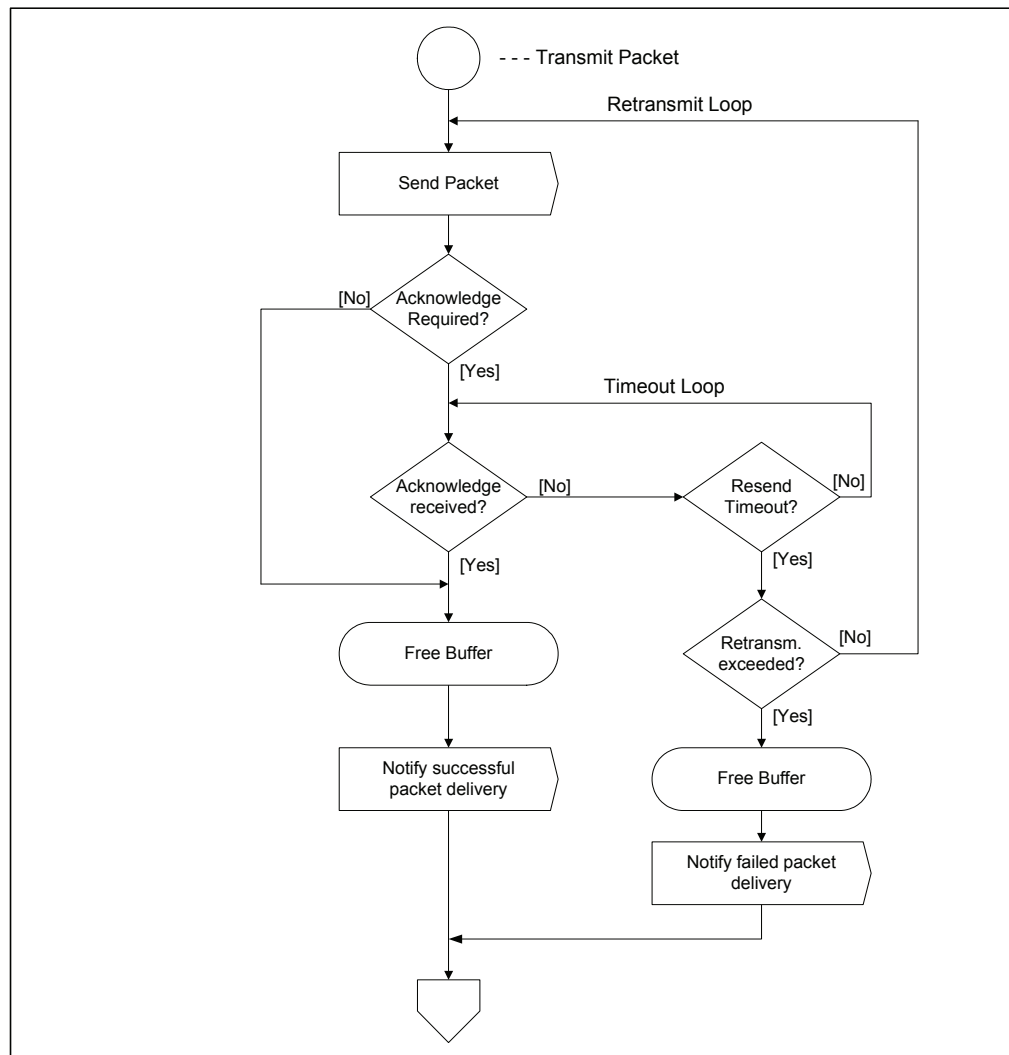
- 1 The UDPCP provides an API to the upper layer applications that use
2 the UDPCP Services. The UDPCP API shall have the following
3 methods:
- 4 • sendMessage (to, opt, message): This method is used for
5 sending a message with specified options (opt).
 - 6 • messageReceived (from, message): This method is used for
7 handling a received message.
 - 8 • commInd (): This method is used to indicate communication
9 status to the upper layers.
 - 10 • confMessage (): UDPCP service configuration

11 **A.4.1 Transmitter Functions**

12 If acknowledge transfer mode is used, the transmitter shall wait for an
13 acknowledge response for a configurable timeout period. If the
14 transmitter does not receive the acknowledge message during this
15 timeout period it shall resend the message with the same Message ID
16 for a configurable retry count. Once the maximum number of
17 retransmissions is exceeded, the transmitter should notify the upper
18 layers of the failed transmission.

1
2

The following flowchart shows a simplified example of the transmitter software.



3
4

Figure 16 Simplified Transmitter Example Flowchart

5

A.4.2 Receiver Functions

6
7
8
9

Since the receiver function in the UDPCP Service may get corrupted or receive unknown fragments (not valid UDPCP packets) from the lower layers, it is more complex and must be capable of responding to many adverse situations. Some examples are listed here:

10
11
12
13
14

- The message cannot be delivered to the upper layers e.g. due to full buffers
- The message integrity check (checksum) fails i.e. it is corrupted
- The message can be too long or short
- The message is in the wrong format (no valid UDPCP header)

- Any combination of the above can occur
- The Acknowledge packet can be lost, causing the transmitter to send a duplicate copy of a message that has already passed to the upper layers

The receiver function shall silently discard all other types of packets from other peer(s) except valid UDPCP Data Packets or Acknowledge packets. If a received Data Packet is valid, the receiver sends an Acknowledge packet (acknowledge transfer mode) to the transmitter.

When a complete message is received (all fragments of a long message or the single packet containing a non-fragmented message), the message and information about its sender are delivered to the upper layers.

The following figure shows a simplified example of the receiver software.

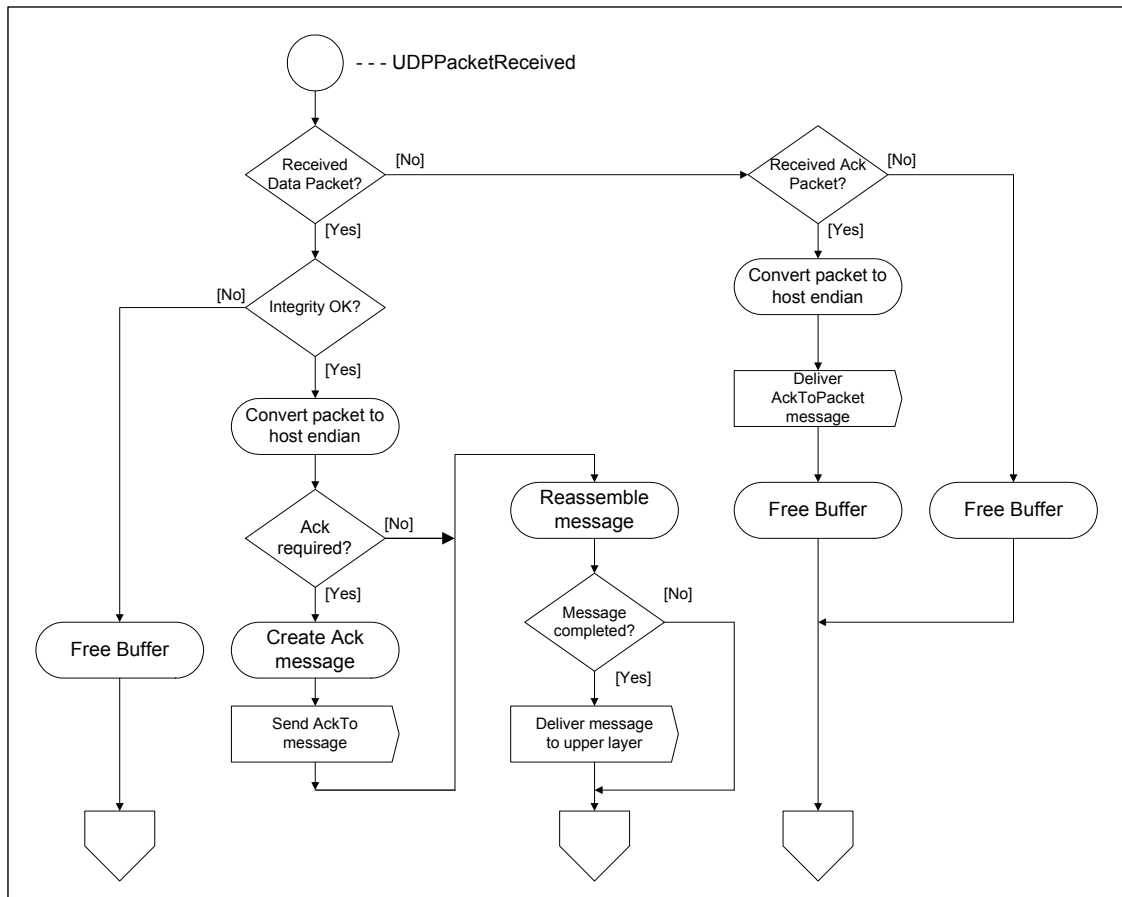


Figure 17 Simplified Receiver Example flowchart

A.4.3 Example Protocol Diagrams

The following figure shows an acknowledged message transfer. In this example it is assumed that the message length does not exceed the used maximum UDP packet size i.e. messages need not be fragmented at the UDPCP.

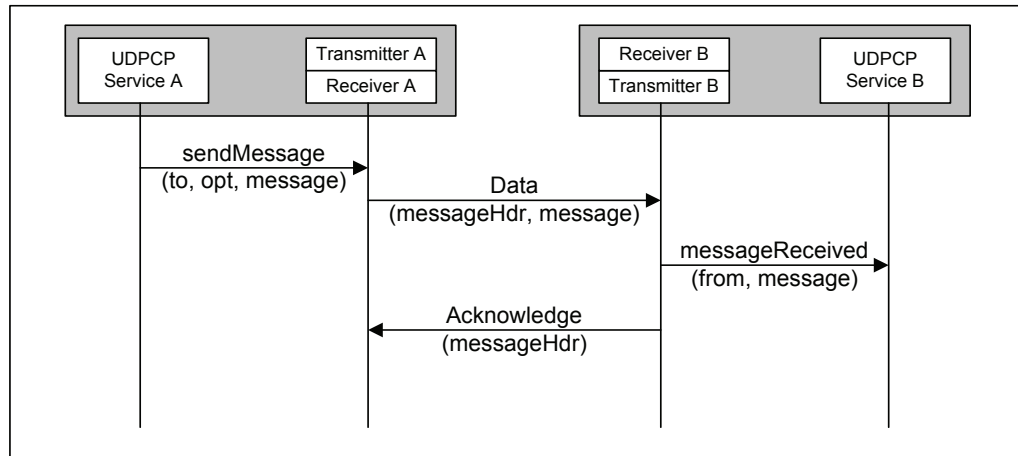


Figure 18 Message transfer (no error) in Acknowledge mode (Non-Fragmented)

The following figure shows a message transfer with errors, in Acknowledge transfer mode. Again it is assumed that the message is not longer than the used maximum UDP packet size.

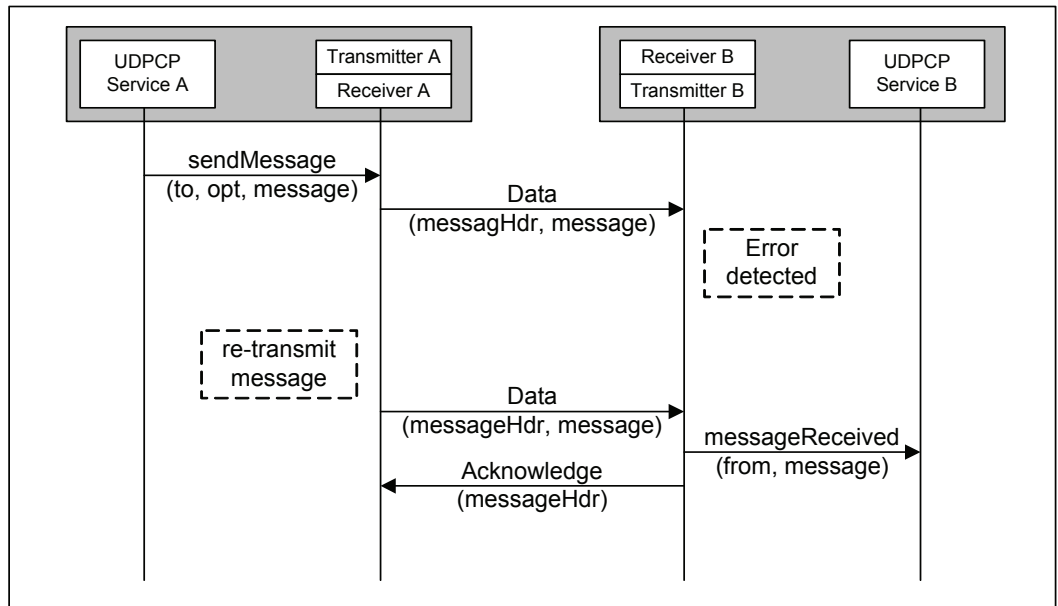
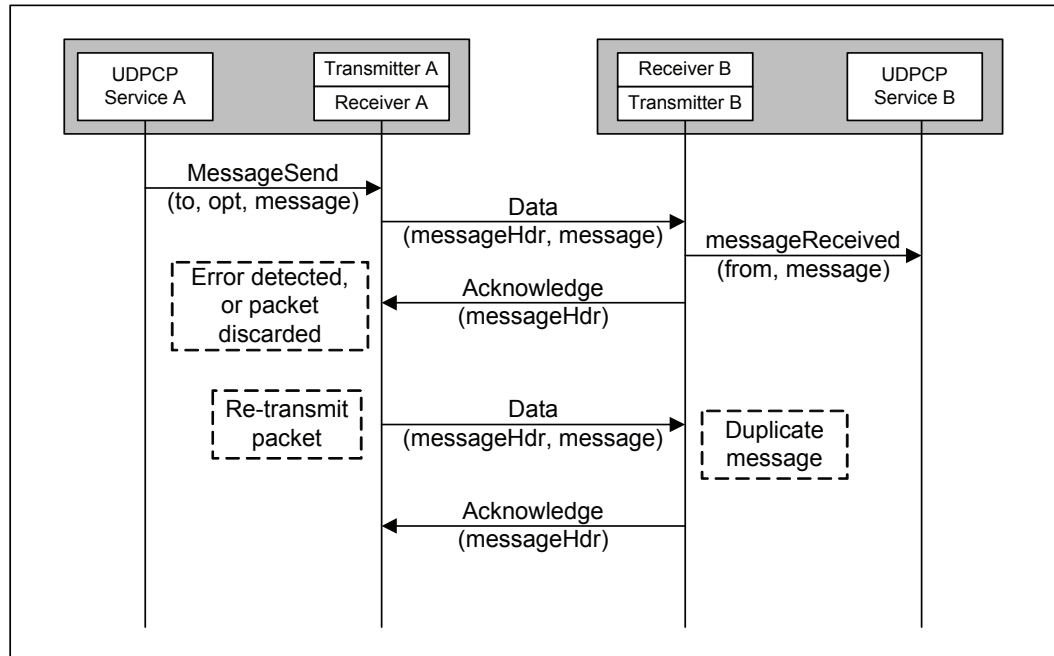


Figure 19: Message transfer with error in Acknowledge mode (Non-fragmented)

1
2
3
4

The following figure shows a message transfer with errors, or timeout and re-transmit, in Acknowledge Transfer mode. Again it is assumed that the message is not longer than the used maximum UDP packet size.

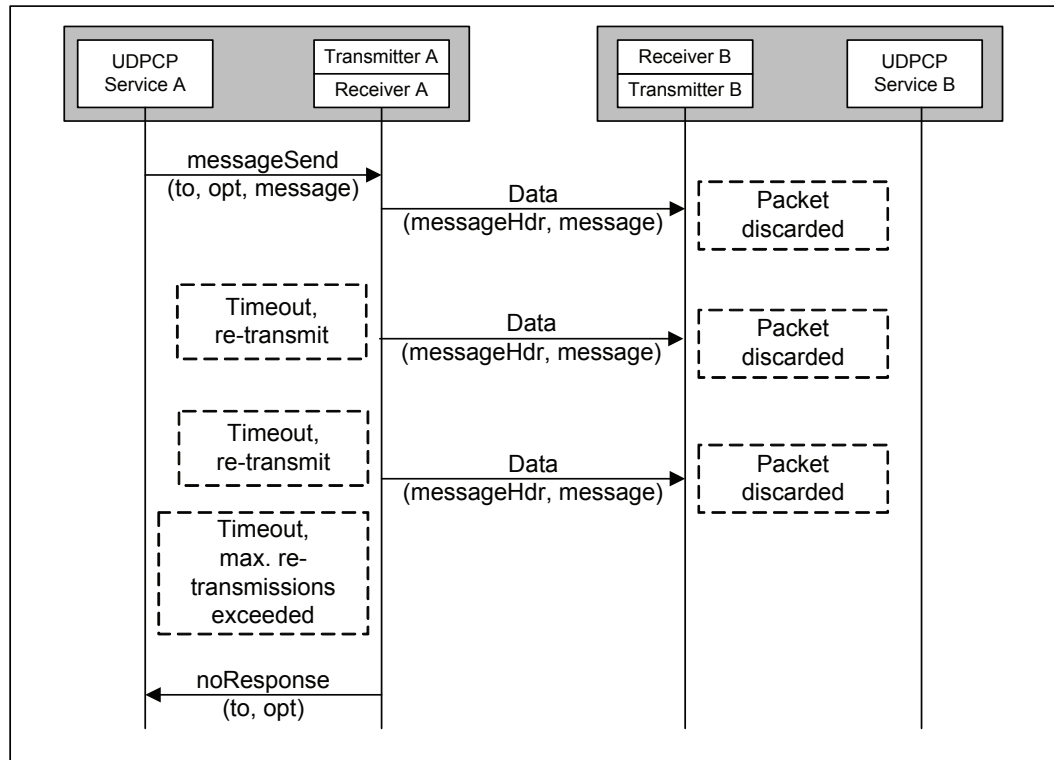


5
6
7

Figure 20 Message transfer with error or timeout and re-transmit in Acknowledge Transfer mode (Non-Fragmented)

8
9
10
11
12

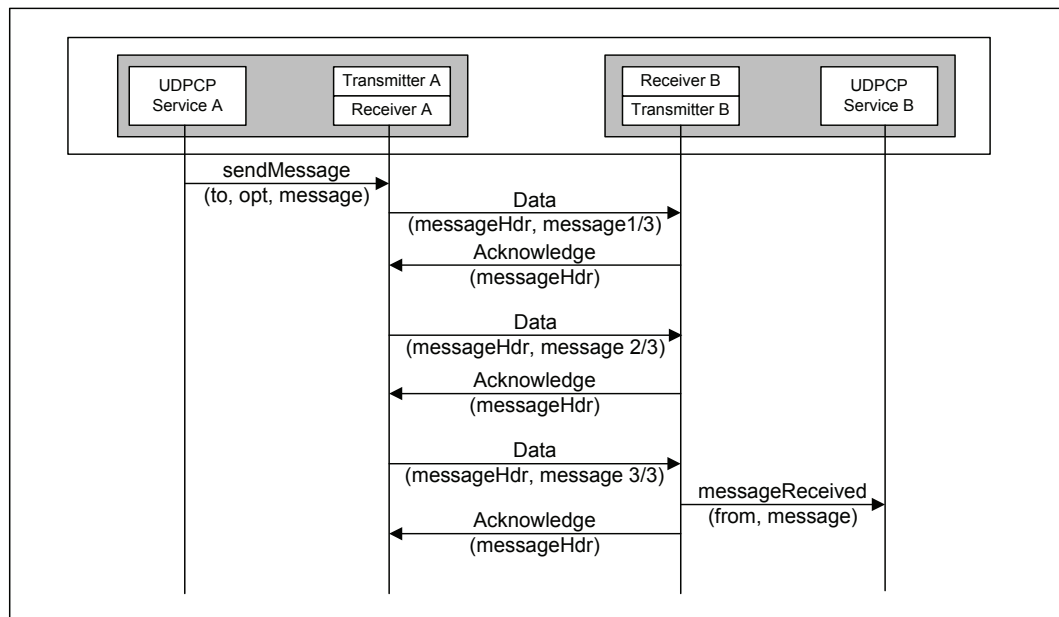
The following figure shows a failed message transfer with multiple retransmissions in Acknowledge Transfer mode. Again it is assumed that the message is not longer than the used maximum UDP packet size.



1
2
3
4
5
6
7

Figure 21 Message transfer with multiple retransmissions in Acknowledge Transfer mode (Retry exceeded and Non-Fragmented)

The following figure shows a message transfer, for a message that is longer than the MTU. The case shown is for Acknowledge Transfer mode (for each fragment).

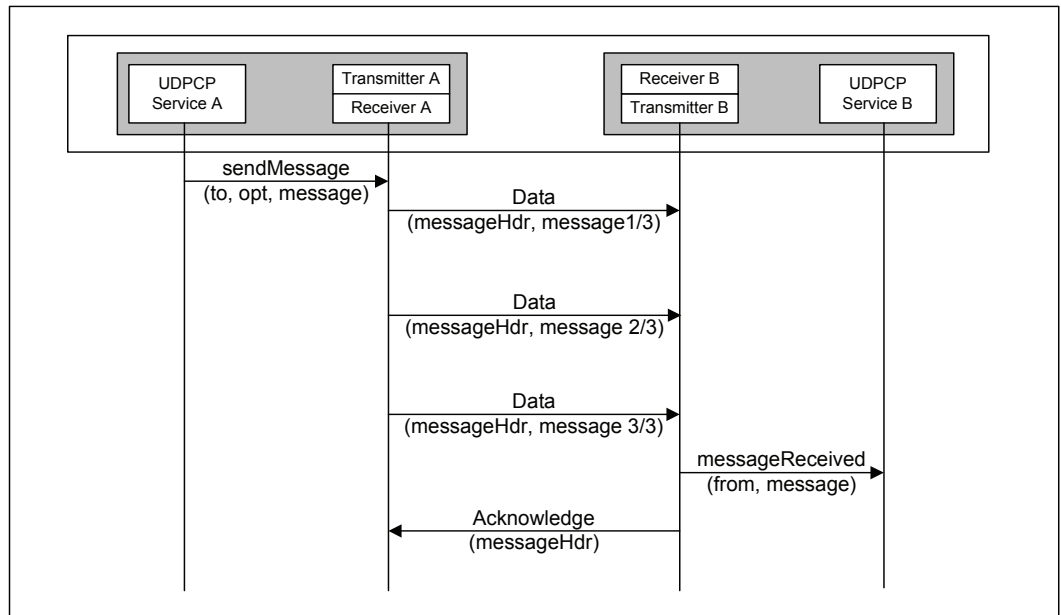


8
9
10

Figure 22 Successful Message transfer in Acknowledge (per fragment) Transfer mode

1
2

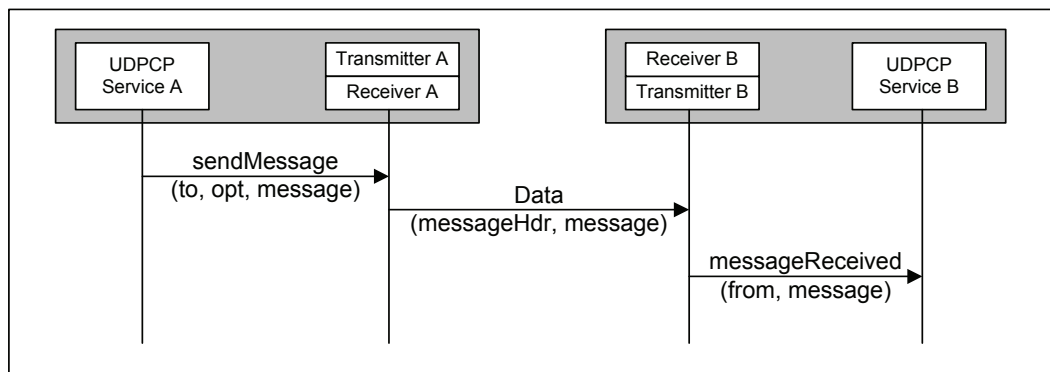
The following figure shows message transfer for a message that is longer than the MTU in Acknowledge Transfer mode (per message).



3
4
5
6
7
8

Figure 23 Successful Message transfer in Acknowledge Transfer mode (per message and Fragmented)

The following figure shows a message transfer in non-acknowledged mode.



9
10

Figure 24 Message transfer in non-acknowledged mode

11 **A.4.4 Interface to UDP layer**

12 The UDPCP Communication Service assumes that standard BSD
13 Socket API shall be applied toward the IP Stack.

14

A.5 Example of UDPCP Interface to the XMLP layer

The UDPCP communication service can provide transport services for various application protocols. One example is to use UDPCP for transporting XMLP messages (SOAP). This requires that an application layer API should be specified in conformance with the underlying protocol binding recommendations, that are specified in the W3C's XML Protocol Abstract Model specification [<http://www.w3c.org/TR/2003/WD-xmlp-am-20030220/>].

This chapter provides an example binding of the XMLP Processor and UDPCP layer.

The following figure shows a logical layered view of the binding model with the XML protocol processor (SOAP) being bound to underlying transports:

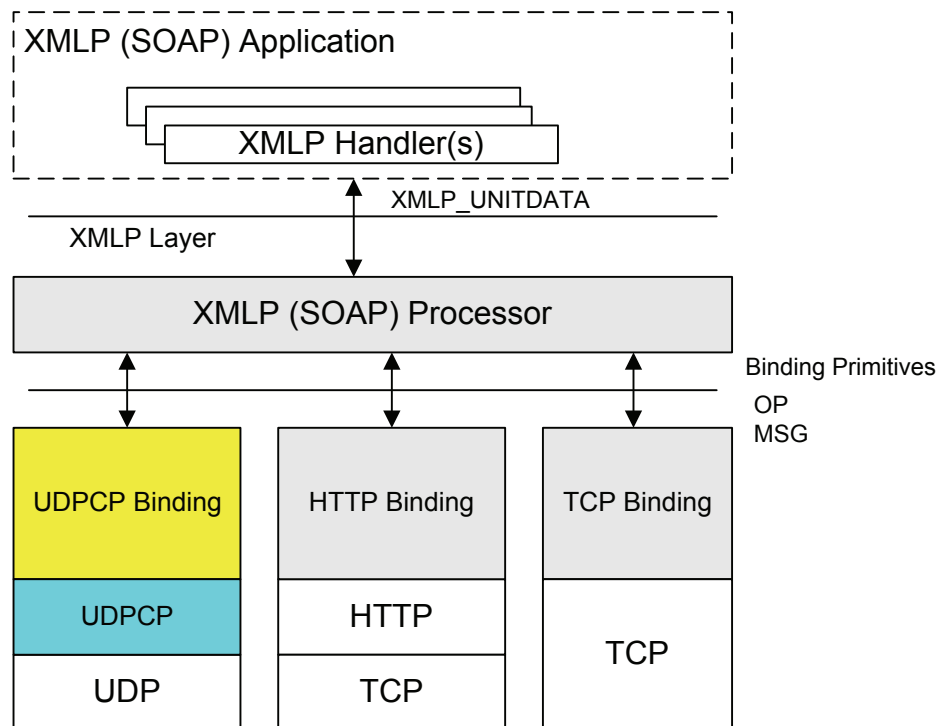


Figure 25 XMLP (SOAP) binding model

A.5.1 UDPCP Binding

The following UDPCP binding specification is based on the XML Protocol Abstract Model specification.

1 **A.5.1.1 Message Correlation**

2 The UDPCP supports only one-way message exchange patterns, so
3 e.g. two-way message correlation must be implemented at the upper
4 layers. The RP1 specification defines the required information elements
5 “Id” and “relatesTo” for message correlation to be used in the header
6 part of a SOAP message.

7 In order to keep UDPCP Binding layer compatible with the XMLP
8 Abstract Model, there shall be applied four pairs of primitives associated
9 with operation delineation and hence message correlation:

10 `OP.start-req` and `OP.start-conf` A `OP.start-req` primitive is sent from
11 the XML protocol processor to the binding to request initialisation of a
12 new correlated message exchange. The binding responds with a
13 `OP.start-conf` primitive.

14 `OP.start-ind` and `OP.start-resp` A `OP.start-ind` primitive is sent from
15 the binding to the XMLP layer to indicate that a new correlated
16 message exchange is being requested. The XML protocol processor
17 responds with a `OP.start-resp` primitive.

18 `OP.end-req` and `OP.end-conf` An `OP.end-req` primitive is sent from the
19 XML protocol processor to the binding to terminate a correlated
20 message exchange. The binding responds with an `OP.end-conf`
21 primitive

22 `OP.end-ind` and `OP.end-resp` An `OP.end-ind` primitive is sent from the
23 binding to the XML protocol processor to indicate that a correlated
24 message exchange is to be terminated. The XML protocol processor
25 concludes the operation with the invocation of an `OP.end-resp` primitive.

26 The final primitive `ERR.ind` is sent from the binding to the XML protocol
27 processor when an error occurs, e.g. if a `MSG.req` cannot be honoured
28 then an `ERR.ind` is generated. Errors are correlated to a particular
29 message exchange using the mechanism described above.

30 **A.5.1.2 Message Exchange Patterns**

31 There are only two simple one-way message exchange patterns in
32 UDPCP Binding: sending or receiving only one message.

33 In order to maintain compatibility with the XMLP Abstract Model, a
34 session is opened for receiving/sending a message. A new session is
35 opened for each subsequent message.

36 The Initiator peer should initialize UDPCP receiver functions i.e.
37 listening socket(s) that use a well-know port(s) during its start-up in
38 order to receive replies from the Responder peers.

39 The following tables show how the XMLP binding primitives are mapped
40 onto the UDPCP protocol actions, on the initiator and receiver, for a

1 simple one-way message exchange; time increases moving down the
2 tables.

3

Initiator	
Binding Primitive	Binding Action
OP.start-req(To, From, BindingContext)	
	Open UDPCP session
OP.start-conf(SessionID)	
MSG.req(SessionID, Message)	
	Send message
ERR.ind(SessionID)	If sending fails
OP.end-req(SessionID)	
	Close UDPCP session
OP.end-conf	

4

Responder	
Binding Primitive	Binding Action
	Receive message
OP.start-ind(SessionID, To, From, BindingContext)	
	Open UDPCP session
OP.start-resp	
MSG.ind(SessionID, Message)	
OP.end-ind(SessionID)	
	Close UDPCP session
OP.end-resp	

5 The layer that started the session is responsible for closing it as soon
6 as possible.

7 UDPCP does not have any means to close sessions remotely. Both
8 responder and sender peer have to close sessions independently.

9 **A.5.1.3 To and From**

10 Because UDPCP is an IP based protocol both “To” and “From” contain
11 two fields: IP address and port.

1 A.5.1.4 SessionID

2 Because UDPCP is meant to be simple, only one unique number that
3 identifies the session is required. Identification numbers are
4 independent from other peers and may be different for the 'sending' and
5 'reponder' sides.

6 A.5.1.5 Message

7 'Message' is a character array type variable containing the full
8 message.

9 A.5.1.6 BindingContext

10 BindingContext is to provide UDPCP customization for upper layers. Its
11 structure is flexible because new UDPCP versions may appear in the
12 future, having more features.

13 If any of these values are missing, they shall be replaced with default
14 values.

15 BindingContext structure example in abstract language:

16 BindingContext("UDPCP").property("Ack") = true/false

17 BindingContext("UDPCP").property("ChecksumInUse") = true/false

18 BindingContext("UDPCP").property("Version") = number

19 A.6 Appendix A Revision History**20 A.6.1 RP1 Version 2.0**

21 The following changes and improvements are made to this version of
22 the UDPCP specification:

- 23 • Homogenized terms "portion", "fragment" and datagram. Now
24 referred to as "Fragment".
- 25 • Updated UDPCP version number to 002.
- 26 • Changed section A.3.2.3 to clarify usage of Message ID during
27 synchronization.

28

A.6.2 RP1 Version 1.1

The following changes and improvements are made to the UDPCP specification:

- Added references to the Adler-32 checksum calculation algorithm specifications (RFC 1950 and RFC 2960)
- SAPI fields are removed from the UDPCP header. This functionality can be implemented easily at the upper layers, and now the UDPCP header will be fixed length (12-octets)
- Added explanation how the checksum will be calculated from UDPCP messages
- S-bit for former SAPI option is changed to fragment/message acknowledge option.
- Added D-bit to enable duplicate message detection.
- Added chapter for Message ID generation.
- Added a mechanism how UDPCP is able to detect peer reset is explained in order to make UDPCP more reliable.
- Implementation specific issues are removed from the figure “simplified receiver flowchart”.
- Added Protocol Example Diagrams
- “Upper Layers” Chapter has been removed, as the presented runtime structure of the UDPCP and SOAP is implementation specific.
- UDPCP Implementation example is removed as this is not intended to be a SW architecture specification.
- OAM&P related chapters are removed, as these are implementation specific issues.
- UDPCP Binding is simplified to contain support only for one-way message exchange patterns (“virtual connection” is removed) as two-way message correlation will be handled in the upper layers.
- Restructuring and removal of implementation specific and redundant information.

A.6.3 RP1 Version 1.0

- The original version of the UDPCP

Appendix B Fault Management

This appendix specifies the Fault Management (FM) part of the OAM&P interface at RP1.

In the realm of OBSAI, Fault Management covers the following aspects

- Reporting faults and alarms
- Managing faults and alarms

The concepts for these are presented and the corresponding SOAP message structures are defined. However, the document does not contain a complete set of attributes that must be implemented.

It is assumed that the reader is familiar with the general concepts of XML and SOAP as defined by W3C® (<http://www.w3c.org>).

The fault management functionality in the Module Agent shall fulfill the specified requirements in order to be compliant with this specification.

B.1 Version and history

The following table shall contain the version information associated with the FM aspect of OBSAI OAM&P specifications. This version information shall be used in the <version> element in the header of all FM messages.

Version number	Release information
1.0	First release of FM

B.2 Specific FM Definitions

Alarm: An alarm represents an abnormal condition of a Module, whose severity is characterized by one of the following values: “*minor*”, “*major*” or “*critical*”.

Note: An alarm is different from a warning (see below).

Active Alarm: An alarm that has not been cleared. An alarm is active until the fault that caused the alarm is corrected and a clear alarm is generated.

1 **Alarm List:** The alarm list contains a list of alarms that are currently
2 active.

3 **Alarm Notification:** A notification is used to inform the recipient about
4 the occurrence of an alarm or, to indicate that the alarm situation is
5 cleared.

6
7 **Warning:** A warning is a special case of an alarm that does not require
8 clearing.

9 **B.3 Fault Management Concepts**

10 **B.3.1 Faults and Alarms**

11 The Module Agent shall detect all faults immediately. After the
12 recognition of a fault, the Module Agent shall store the alarm
13 information locally and send an alarm notification to the BTS Master
14 Agent. All detected faults shall cause an alarm on the module. The
15 module shall have the capability to report correlated or raw alarms to
16 the BTS Master Agent.

17 The Module Agent shall clear an alarm when a detected fault is no
18 longer present in the module and send a notification to the BTS Master
19 Agent (see also “B.3.3 Clearing of Alarms”).

20 **B.3.2 Active Alarms List**

21 The Module Agent shall maintain a list that contains information
22 pertaining to all active alarms.

23 **B.3.3 Clearing of Alarms**

24 All alarms caused by a fault with severities “critical”, “major” and “minor”
25 shall be cleared. An alarm is cleared if the fault situation in the module
26 ceases.

27 Alarms caused by faults with severity level “warning” are considered
28 informative and they shall not need clearing.

29 Each time an alarm is cleared; the Module Agent will remove it from the
30 active alarms list and send an appropriate clear alarm notification to the
31 BTS Master Agent.

1 **B.3.4 Enabling/Disabling Notifications**

2 The BTS Master Agent shall have the capability to disable or enable all
3 or selected alarm notifications sent by the module agents. The module
4 shall continue to perform all alarm related functions like alarm
5 generation and clearing, even if notifications to the BTS Master Agent
6 are disabled. After reset, all notifications shall be disabled by default.
7 When all notifications are enabled, a notification for every active alarm
8 shall be sent. When a specific alarm notification is enabled, the
9 notification is sent if the alarm is active.

10 When a connection to a module is lost (e.g. due to module restart or
11 removal) the BTS master agent shall clear all alarms from its alarm list
12 for this module.

13 **B.3.5 Alarm History Information**

14 The module agents are not required to maintain the alarm history. It is
15 expected that the BTS Master agent will implement this functionality.

16
17
18

1

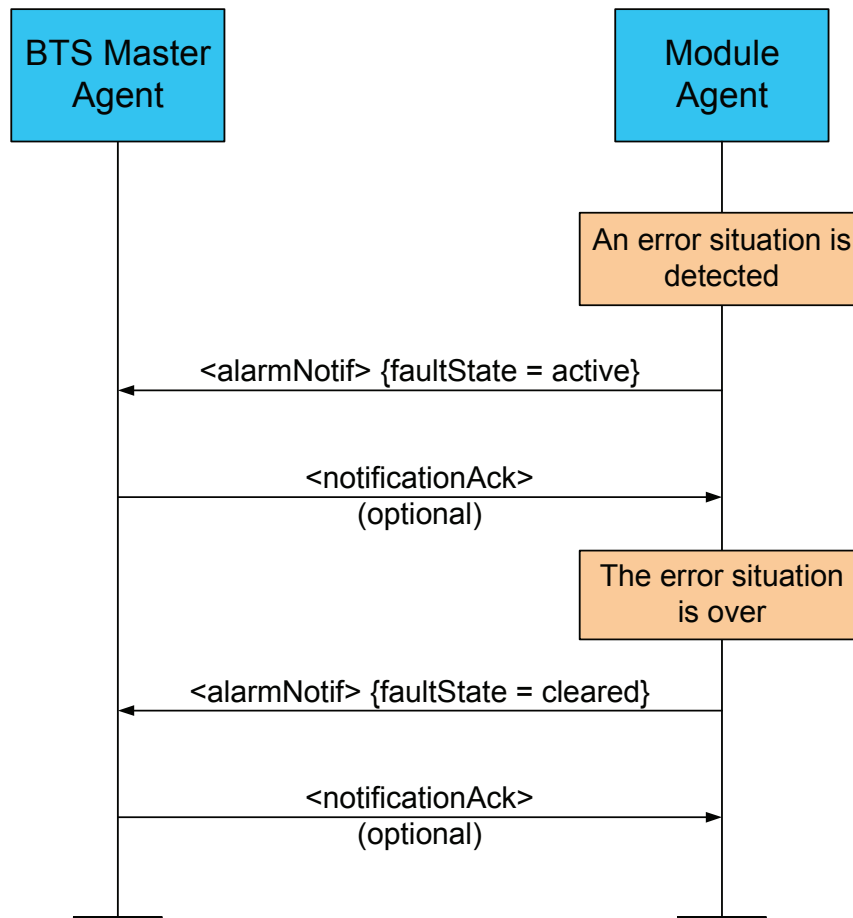
2 **B.4 FM Messages**

3 This section describes the various FM operations specified by OBSAI.

4 **B.4.1 Alarm Notification**

5 The <alarmNotif> message shall be used for communicating alarm
6 notifications to the BTS Master Agent. The notification contains
7 information that identifies the alarm, its source and state etc.

8 It shall be possible to configure Alarm Notifications to be acknowledged.
9 In that case the Module Agent shall include “responseRequested”
10 element in the SOAP Header part of the <alarmNotif> message (see
11 B.4.2.1). The BTS Master Agent shall then send <notificationAck>
12 message to the Module Agent.



13 **Figure 26 Alarm notification scenario**

14

1 **B.4.1.1 Alarm Notification Message Elements**

2 OBSAI specified SOAP Body elements of the <alarmNotif>
3 notification message are shown in the table below:

Element Name	Usage
faultId	This element is used to classify a fault situation in a module. It is a fault-specific integer value that shall be unique within the context of a Module Agent.
faultSource	This element identifies the object or source of a fault. The Module Agent is responsible for ensuring a unique identification of the fault source.
faultSeverity	Fault severity conveys the severity level of the fault as follows: "critical": denotes a critical alarm that requires a clearing event. "major": denotes a major alarm that requires a clearing event "minor": denotes a minor alarm that requires a clearing event. "warning": denotes a warning which does not require a clearing event.
faultState	This element conveys information on the state of a fault "active" : indicates the occurrence of a new fault in the Module Agent "cleared" : indicates the fault has been cleared.
faultText	Textual description of the fault so that it can be understood by humans (optional)
eventTime	Timestamp to indicate the time when the notification was generated.

4

5 **B.4.1.2 Optional Notification Acknowledge message**
6 **elements**

7 The <notificationAck> message is a generic and optional
8 message that may be used to acknowledge the receipt of any
9 notification. The <relatesTo> field in the header provides information
10 about the notification being acknowledged.

B.4.2 Manage Alarms Requests

The <manageAlarmsReq> message shall be used for controlling alarm reporting from the Module Agent to the BTS Master Agent.

B.4.2.1 Enabling Alarm Notifications

The BTS Master Agent shall be able to enable all or selected alarm notifications from a module by using the <manageAlarmsReq> message.

The Module Agent shall respond with the <manageAlarmsRsp> message to indicate that alarm notifications are enabled. Additionally, the Module Agent shall send alarm notifications for all “active” alarms for which notifications were enabled.

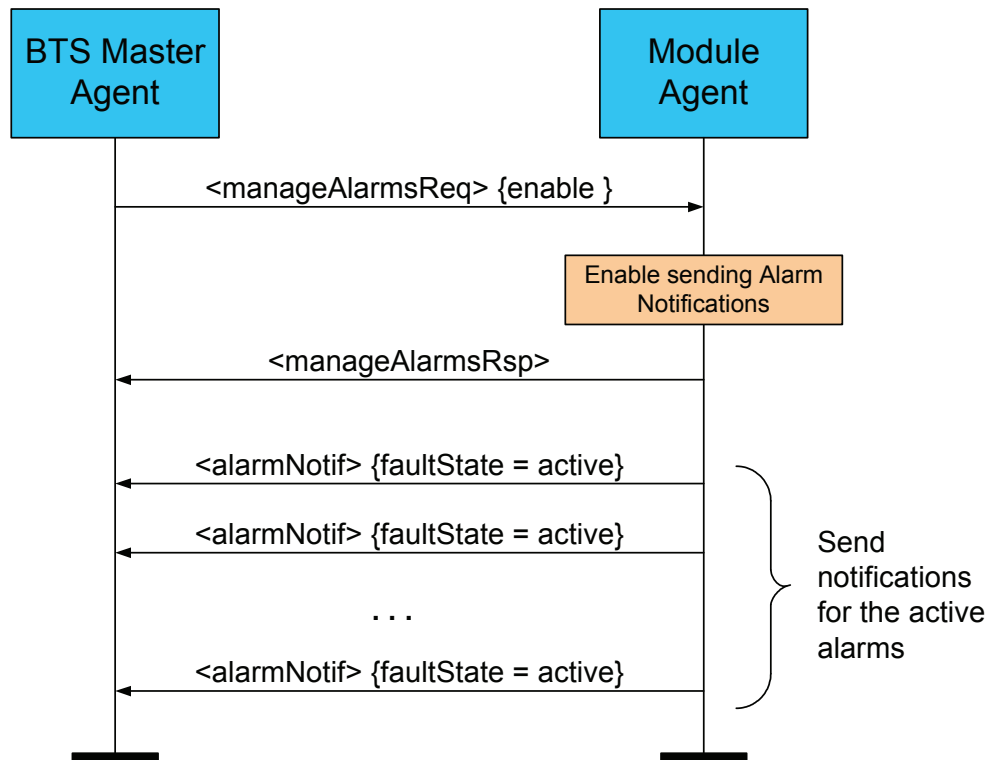


Figure 27 Scenario for enabling alarm notifications

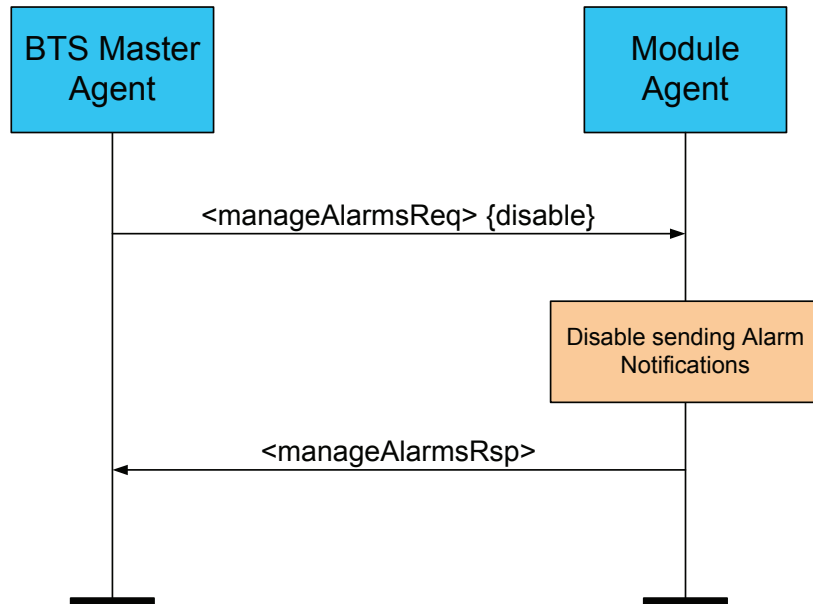
B.4.2.2 Disabling Alarm Notifications

The BTS Master Agent shall disable all, or selected, alarm notifications from a module, by using the <manageAlarmsReq> message.

The Module Agent shall respond with the <manageAlarmsRsp> message to indicate that alarm notifications are disabled. Subsequently,

1 the module agent shall not send notifications for such alarms until these
2 are enabled again.

3



4

5

Figure 28 Scenario for disabling alarm notifications

6

7

B.4.2.3 Manage Alarms Request message elements

8

The <manageAlarmsReq> message shall contain the following SOAP
9 Body elements:

9

Element Name	Usage
scope	Optional element that determines the scope of this request. If this element is set to "all" it indicates that this request pertains to all managed objects in the module.
managedObject	Structure that contains Managed Object, operation and parameter value(s). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor (optional) One or more managedObject elements may be specified in this message. If the scope is set to 'all' then this element(s) may be omitted.

operation	Element that indicates the requested operation to be performed on the managed object(s) containing the attribute: "enable" : indicates that the alarm shall be enabled "disable": indicates that the alarm shall be disabled
parameter	This element contains a user-friendly name of the alarm.

1

2

3

The corresponding reply message `<manageAlarmsRsp>` shall contain the following SOAP Body elements:

Element Name	Usage
status	Status of the requested operation: "OK" : indicates that the operation was successful "fail" : indicates that the operation failed
failureReason	If <code><status></code> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.

4

5

Appendix C Performance Management

This appendix specifies the Performance Management (PM) part of the OAM&P interface at RP1.

In the realm of OBSAI, 'Performance Management' covers the following PM functions:

- Measurement Result Exchange
- Measurement Administration

The concepts for these are presented and the corresponding SOAP message structures are defined. However, the specifications do not contain a complete set of attributes that must be implemented.

It is assumed that the reader is familiar with the general concepts of XML and SOAP as defined by W3C® (<http://www.w3.org/>).

The performance management functionality in the Module Agent shall fulfill the specified requirements in order to be compliant with this specification.

C.1 Version and history

The following table shall contain the version information associated with the PM aspect of OBSAI OAM&P specifications. This version information shall be used in the <version> element in the header of all PM messages.

Version number	Release information
1.0	First release of PM
Future release	<ul style="list-style-type: none"> • Changed element filename to fileName. • Changed element newAttValues to newAttributeValues and oldAttValues to oldAttributeValues in C.4.8.2

1 **C.2 Specific PM Definitions**

2 **Measurement Job:** A process that is executed in the modules to
3 collect performance data

4 **Measurement Object:** Entity from which the performance data is
5 monitored

6 **PM Measurements:** Performance data related storage registers

- 7
 - 8 • **Threshold:** PM measurements may have configurable
9 thresholds. If the value of the PM measurement crosses its
 threshold value, a threshold crossing notification is sent.

10 **C.3 Performance Management Concepts**

11 **C.3.1 Performance Data Storage**

12 PM data storage functionality in the BTS modules is an implementation
13 specific issue. However, current measurement value and previous
14 measurement value shall be retained in the module.

15 **C.3.2 Performance Data Reporting**

16 The messages for the following PM reporting shall be provided:

- 17
 - 18 • On demand reporting
 - 19 • Periodic reporting

20 Furthermore, the capability to set a schedule and to specify the reported
21 information in that schedule, (measurement job) may be provided.

22 The results of a measurement request shall be forwarded in one of
23 three different ways:

- 24
 - 25 • The results shall be sent via notifications as soon as they are
26 available
 - 27 • The results shall be retrieved on demand
 - The results shall be stored in files and transferred or retrieved
 when required.

C.3.3 Thresholds And Alerts

A typical implementation for measurement thresholds and alerts, that the Module Agents shall support, is as follows:

- Ability to set and modify threshold values for performance measurements
- Ability to send a notification when a threshold value is modified.
- Ability to enable/disable threshold crossing alerts (TCA)
- Retrieval of TCA reporting status (enabled or disabled)

C.4 PM Messages

This section describes the various PM operations specified by OBSAI.

C.4.1 Initialize PM Measurements

The BTS Master Agent shall be able to initialize storage registers for performance data by using the `<resetPMMeasurementsReq>` command.

The Module Agent shall respond to the BTS Master Agent with a `<resetPMMeasurementsRsp>` message.

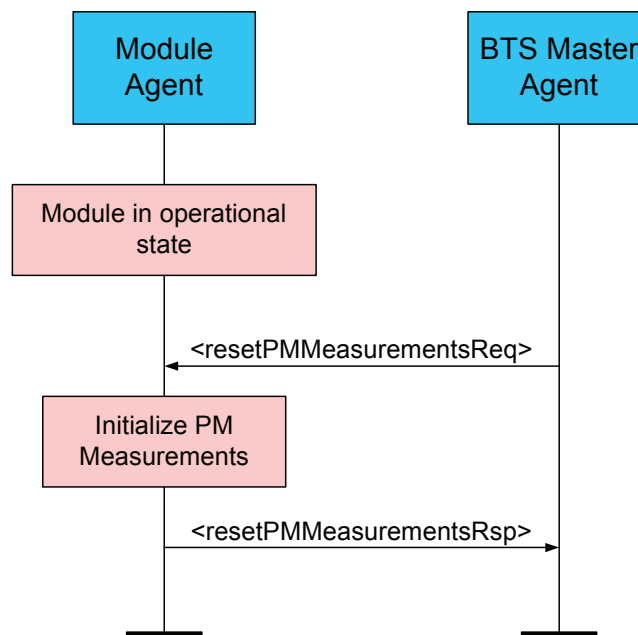


Figure 29 Initialize Measurement Data

1 **C.4.1.1 Reset PM Measurements Request Message**
 2 **Elements**

3 The <resetPMMeasurementsReq> message shall contain the
 4 following SOAP Body elements:

Element Name	Sub-element Name	Usage
scope		Optional element that determines the scope of this request. If this element is set to "all" it indicates that this request pertains to all managed objects in the module.
measurement		Element containing measurement objects to be initialized. There may be more than one instance of this element in this message. If the scope is set to "all" then this element(s) may be omitted.
	measType	Sub-element in <measurement> containing a measurement type e.g., traffic, link, etc. There may be more than one instance of this sub-element in this measurement element.
	managedObject	Sub-element in <measurement> structure that contains Managed Object, operation and parameter value(s). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)

5

6 **C.4.1.2 Reset PM Measurements Response Message**
 7 **Elements**

8 The <resetPMMeasurementsRsp> message shall contain the
 9 following SOAP Body elements:

Element Name	Usage
status	Status of the requested operation:

	<p>"OK": indicates that the operation was successful</p> <p>"fail": indicates that the operation failed</p>
failureReason	<p>If <status> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.</p>

C.4.2 Measurement Data Notification

The Module Agent shall send notification to the BTS Master Agent that performance data is available by using the <resultReadyNtf> message (if notifications are enabled). The 'result ready' notification provides file information that the Module Agent can use to retrieve measurement data.

A <notificationAck> response (refer Appendix B.4.1.2) shall be sent by the BTS Master Agent if an acknowledgement is requested.

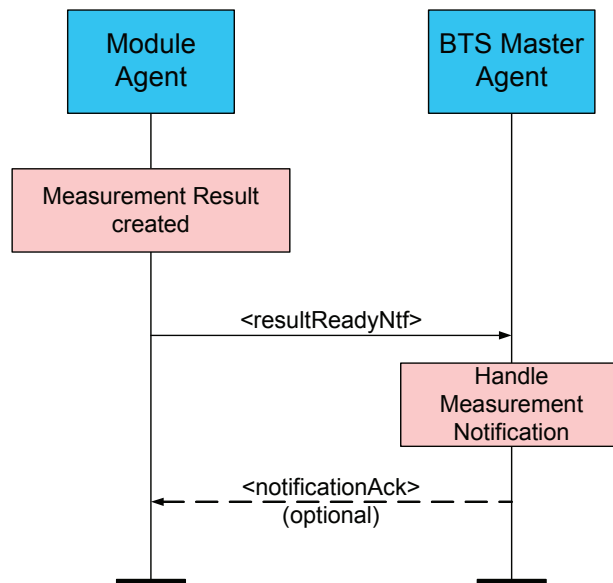


Figure 30 Measurement data notification

C.4.2.1 Result Ready Notification Message Elements

The <resultReadyNtf> message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
measId		Measurement identifier that shall be used by subsequent <measDataReq> as a reference.

measSchedule		<p>An element of Complex Type containing attributes:</p> <ul style="list-style-type: none"> - name: name for the schedule - replaces: optional element - function: optional element <p>This element and its associated sub-elements shall be included when the availability of a scheduled report is being notified.</p>
	scheduleItem	<p>A sub-element of measurement schedule element. Complex Type containing attributes:</p> <ul style="list-style-type: none"> - scheduled: identifier of a schedule item - priority: optional element - startTime: time when the measurement shall be started - stopTime: time when the measurement terminates
fileRetrievalInfo		This element contains file retrieval information.
	protocol	<p>Indicates the preferred file transfer protocol for retrieving the measurement data:</p> <ul style="list-style-type: none"> - "FTP" - "TFTP" - "HTTP" - "SOAP"
	userId	Optional user Id to be used for measurement retrieval
	password	Optional password to be used for measurement retrieval
	resultURL	If the <protocol> is "HTTP", the URL of the measurement result shall be provided. Otherwise, this element shall not be present in this message.
	ipAddress	If the protocol is either "FTP" or "TFTP", the IP Address to open a file transfer session shall be provided. Otherwise, this element shall not be present in this message.

	fileName	If the protocol is either "FTP" or "TFTP", this element identifies an individual filename including path. Otherwise, this element shall not be present in this message.
	format	If the protocol is either "FTP" or "TFTP", this element identifies the file format of the file e.g.3GPP_PM_XML, 3GPP_PM_ASN1...or proprietary. Otherwise, this element shall not be present in this message.

1

2

C.4.3 Request Measurement Data

3

The BTS Master Agent shall be able to request a Module Agent to send performance data, including parameters and accumulation period, by using the <measDataReq> command.

4

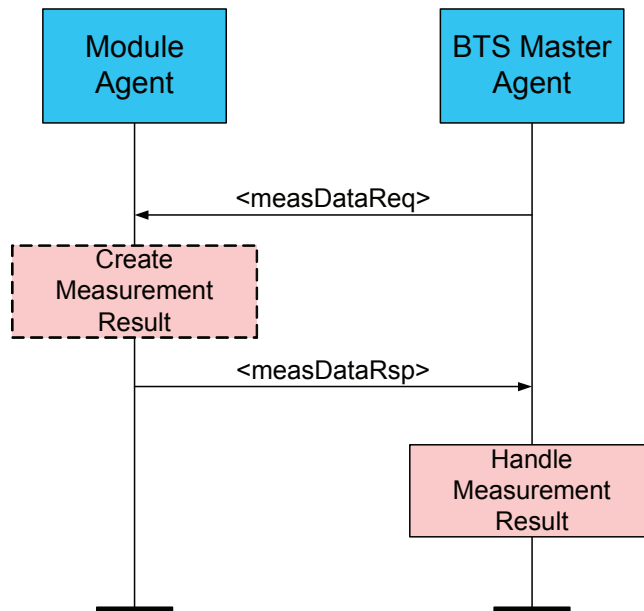
5

6

The Module Agent shall respond with a <measDataRsp> message that contains measurement result data, in XML-format, or the file information necessary for downloading measurement results.

7

8



9

10

Figure 31 Measurement Result exchange, report included in response

11

12

When the <measDataRsp> message contains the file information, the Master Agent can retrieve the file using the specified file transfer procedure.

13

14

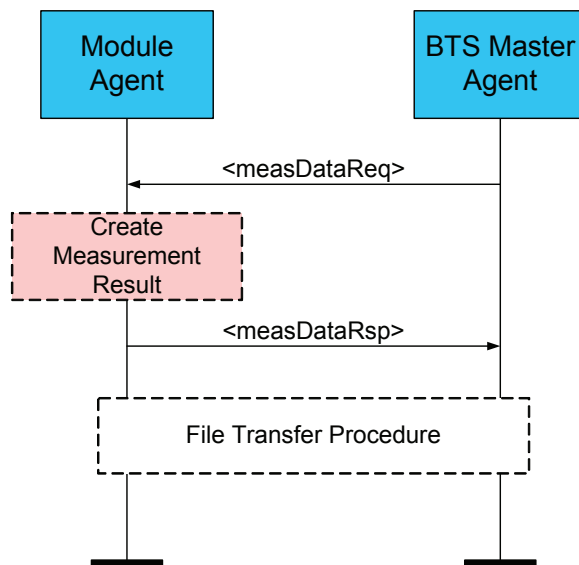


Figure 32 Measurement Result exchange, report file downloaded separately

1
2
3
4
5
6
7
8

C.4.3.1 Measurement Data Request Message Elements

The `<measDataReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
scope		Optional element that determines the scope of this request. If this element is set to "all" it indicates that this request pertains to all managed objects in the module.
measId		If a previous notification message has triggered this message, this element shall contain the measurement Id provided by the module agent. Otherwise this element shall be omitted. This element shall also be omitted if the scope element is present in this message.

measurement		<p>Element containing measurement objects.</p> <p>There may be more than one instance of this element in this message.</p> <p>If the scope is set to "all" then this element(s) may be omitted.</p> <p>This element and its associated sub-elements shall be omitted when the measId element is present in this message.</p>
	measType	<p>Sub-element in <measurement> containing a measurement type e.g., traffic, link, etc.</p> <p>There may be more than one instance of this sub-element in this measurement element.</p>
	measPeriod	<p>Optional sub-element in <measurement> containing duration of measurement information. Complex Type containing attributes:</p> <ul style="list-style-type: none"> - duration: duration of measurement in seconds - endTime: end of measurement time
	managedObject	<p>Sub-element in <measurement> structure that contains Managed Object, operation and parameter value(s). Complex Type containing attributes:</p> <ul style="list-style-type: none"> class: class name (optional) distName: distinguishedName vendor: vendor information (optional)

1 **C.4.3.2 Measurement Data Response Message Elements**

2 The <measDataRsp> message shall contain following SOAP Body
3 elements:

Element Name	Sub-element Name	Usage
measId		This element shall echo the <measId> specified in the associated measDataReq message.
measurement		Element containing measurement objects. There may be more than one instance of this element in this message. This element and its associated sub-elements shall be omitted if this message contains file retrieval information.
	measPeriod	Sub-element in <measurement> containing duration of measurement information. Complex Type containing attributes: - duration: duration of measurement in seconds - endTime: end of measurement time
	measType	Sub-element in <measurement> containing a measurement type e.g., traffic, link, etc. There may be more than one instance of this sub-element in this measurement element.
	managedObject	Sub-element in <measurement> structure that contains Managed Object, operation and parameter value(s). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	measValue	A sub-element in <measurement> containing the measurement result.
fileRetrievalInfo		This element contains file retrieval information. It shall be omitted if <measurement> element is present.

protocol	Indicates the preferred file transfer protocol for retrieving the measurement data: <ul style="list-style-type: none"> - "FTP" - "TFTP" - "HTTP" - "SOAP"
userId	Optional user Id to be used for measurement retrieval
password	Optional password to be used for measurement retrieval
resultURL	If the <protocol> is "HTTP", the URL of the measurement result shall be provided. Otherwise, this element shall not be present in this message.
ipAddress	If the protocol is either "FTP" or "TFTP", the IP Address to open a file transfer session shall be provided. Otherwise, this element shall not be present in this message.
fileName	If the protocol is either "FTP" or "TFTP", this element identifies an individual filename including path. Otherwise, this element shall not be present in this message.
format	If the protocol is either "FTP" or "TFTP", this element identifies the file format of the measurement result file e.g.3GPP_PM_XML, 3GPP_PM_ASN1...or proprietary. Otherwise, this element shall not be present in this message.

1 **C.4.4 Create Measurement Schedule**

2 The BTS Master Agent shall be able to create a measurement schedule
3 for PM data reporting by using the <setScheduleReq> message.

4 This message shall contain the measurement schedule description e.g.
5 the measured parameters (or measured objects), reporting interval,
6 start time of reports and number of reports.

7 The Module Agent shall respond with a <setScheduleRsp>
8 message.

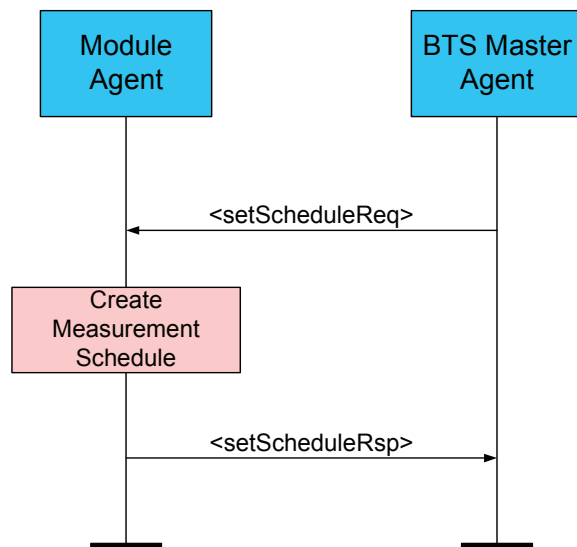


Figure 33 Schedule Measurement Data Reporting

C.4.4.1 Set Schedule Request Message Elements

The `<setScheduleReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
measSchedule		An element of Complex Type containing attributes: - name (name for the schedule) - replaces (optional element) - function (optional element) There can be more than one instance of this element.
	scheduleItem	A sub-element of <code><measSchedule></code> element. Complex Type containing attributes: - scheduleId: identifier of a schedule item) - priority: optional element - startTime: time when the measurement shall be started - stopTime: time when the measurement terminates

measurement		<p>Element containing measurement objects.</p> <p>There may be more than one instance of this element in this message.</p>
	managedObject	<p>Sub-element in <measurement> that contains Managed Object, operation and parameter value(s). Complex Type containing attributes:</p> <p>class: class name (optional)</p> <p>distName: distinguishedName</p> <p>vendor: vendor information (optional)</p>
	measType	<p>Sub-element in <measurement> containing a measurement type e.g., traffic, link, etc.</p> <p>There may be more than one instance of this sub-element in this measurement element.</p>
	measPeriod	<p>Sub-element in <measurement> containing duration of measurement information. Complex Type containing attributes:</p> <ul style="list-style-type: none"> - duration: duration of measurement in seconds - endTime: end of measurement time for the first measurement. This time shall be incremented by <measInterval> for each additional measurement.
	measInterval	<p>Sub-element of <measurement> containing the time interval between measurements in seconds.</p>

1 **C.4.4.2 Set Schedule Response Message Elements**

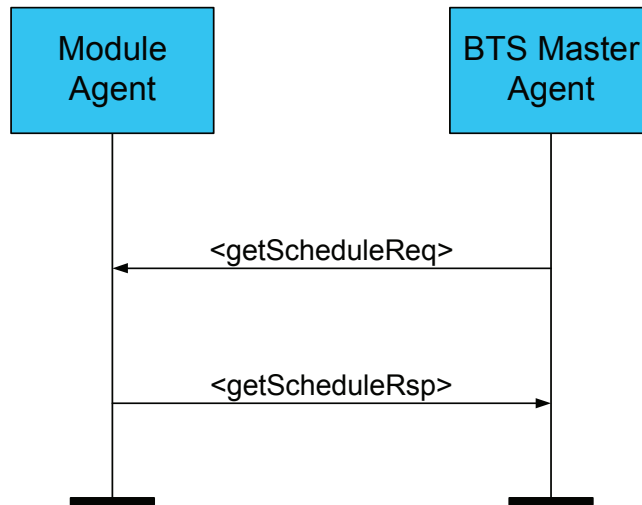
2 The <setScheduleRsp> message shall contain the following SOAP
3 Body elements:

Element Name	Sub-element Name	Usage
measSchedule		An element of Complex Type containing attributes: - name (name for the schedule) - replaces (optional element) - function (optional element) There can be more than one instance of this element.
	scheduleItem	A sub-element of measurement schedule item. Complex Type containing attributes: - scheduleId: identifier of a schedule item) - priority: optional element - start'ime: time when the measurement shall be started - stop'ime: time when the measurement terminates
status		Status of the requested operation: "OK": indicates that the operation was successful "fail": indicates that the operation failed
failureReason		If <status> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.

4 **C.4.5 Request Measurement Schedule**

5 The BTS Master Agent is able to retrieve the current measurement
6 schedule from the Module Agent, by using the <getScheduleReq>
7 message.

1 The Module Agent responds by sending a <getScheduleRsp>
2 message, that contains a measurement schedule description.



3
4 **Figure 34 Retrieve Measurement Data Reporting**
5 **schedule**

6 C.4.5.1 Get Schedule Request Message Elements

7 The <getScheduleReq> message shall contain the following SOAP
8 Body elements:

Element Name	Sub-element Name	Usage
scope		If this element is set to "all", the response shall report information of all schedules in the module. If this is used then the <measSchedule> element shall be omitted
measSchedule		An element of Complex Type containing attributes: - name (name for the schedule) - replaces (optional element) - function (optional element)

	scheduleItem	<p>A sub-element of measurement schedule item. Complex Type containing attributes:</p> <ul style="list-style-type: none"> - scheduleId: identifier of a schedule item) - priority: optional element - startTime: time when the measurement shall be started - stopTime: time when the measurement terminates
--	--------------	--

1 **C.4.5.2 Get Schedule Response Message Elements**

2 The <getScheduleRsp> message shall contain the following SOAP
3 Body elements:

Element Name	Sub-element Name	Usage
measSchedule		<p>An element of Complex Type containing attributes:</p> <ul style="list-style-type: none"> - name (name for the schedule) - replaces (optional element) - function (optional element)
	scheduleItem	<p>A sub-element of measurement schedule element. Complex Type containing attributes:</p> <ul style="list-style-type: none"> - scheduleId (identifier of a schedule item) - priority (optional element) - startTime: time when the measurement shall be started - stopTime: time when the measurement terminates
measurement		<p>Element containing measurement objects.</p> <p>There may be more than one instance of this element in this message.</p>

	managedObject	Sub-element in <measurement> that contains Managed Object, operation and parameter value(s). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	measType	Sub-element in <measurement> containing a measurement type e.g., traffic, link, etc. There may be more than one instance of this sub-element in this measurement element.
	measPeriod	Sub-element in <measurement> containing duration of measurement information. Complex Type containing attributes: - duration: duration of measurement in seconds - endTime: end of measurement time for the first measurement.
	measInterval	Sub-element of <measurement> containing the time interval between measurements in seconds.

1 **C.4.6 Control Measurement Data reporting**

2 **C.4.6.1 Activate Measurement Data Reporting:**

3 The BTS Master Agent shall be able to start/activate performance data
4 reporting by using the <controlScheduleReq> message.

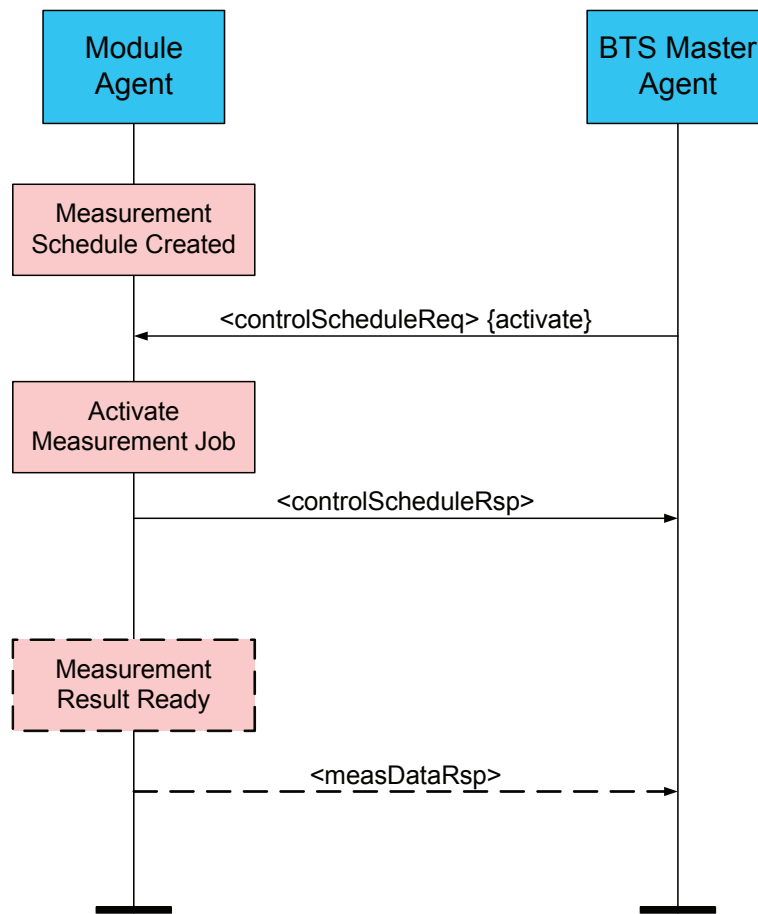


Figure 35 Activate Measurement Data Reporting

The Module Agent responds with `<controlScheduleRsp>` message. When the measurements pertaining to an active schedule are available, the Module Agent shall be able to send:

- a) `<measDataRsp>` message asynchronously to the BTS Master Agent.
- b) Notification that measurement data is available by using the `<resultReadyNtf>` message.

C.4.6.2 Suspend Measurement Data Reporting:

The BTS Master Agent shall be able to stop/suspend the reporting of performance data by using the `<controlScheduleReq>` message with the operation “suspend”.

The Module Agent responds with a `<controlScheduleRsp>` message.

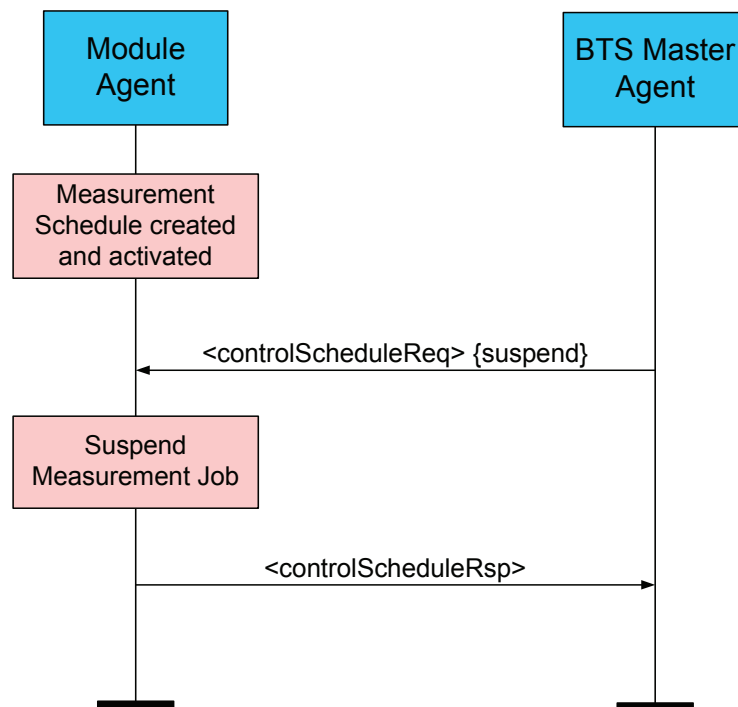


Figure 36 Suspend Measurement Data Reporting

C.4.6.3 Control Schedule Request Message Elements

The `<controlScheduleReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
operation		Operation identifier element for measurement control: "activate": activate the schedule "suspend": suspend the schedule "remove": remove the schedule "stateReq": request the state of a schedule
measSchedule		An element of Complex Type containing attributes: - name (name for the schedule) - replaces (optional element) - function (optional element)

	scheduleItem	A sub-element of measurement schedule element. Complex Type containing attributes: <ul style="list-style-type: none"> - scheduleId (identifier of a schedule item) - priority (optional element) - startTime: time when the measurement shall be started - stopTime: time when the measurement terminates
--	--------------	--

1 **C.4.6.4 Control Schedule Response Message Elements**

2 The <controlScheduleRsp> message shall contain the following
 3 SOAP Body elements:

Element Name	Sub-element Name	Usage
measSchedule		An element of Complex Type containing attributes: <ul style="list-style-type: none"> - name: name for the schedule - replaces: optional element - function: optional element
	scheduleItem	A sub-element of measurement schedule element. Complex Type containing attributes: <ul style="list-style-type: none"> - scheduleId: identifier of a schedule item - priority: optional element - startTime: time when the measurement shall be started - stopTime: time when the measurement terminates
operation		This element indicates the requested operation: <ul style="list-style-type: none"> "activate": activate the schedule "suspend": suspend the schedule "remove": remove the

		<p>schedule</p> <p>"stateReq": request the state of a schedule</p>
operationStatus		<p>If the controlScheduleReq operation was "stateReq", then this element shall have one of the following values:</p> <p>"active": indicates that the schedule was activated</p> <p>"suspended": indicates that the schedule</p> <p>"invalid": indicates that the requested schedule is invalid</p> <p>If the controlScheduleReq operation was "activate", "suspend" or "remove", then this element shall have one of the following values:</p> <p>"OK": indicates that the operation was successful</p> <p>"fail": indicates that the operation failed</p>
failureReason		<p>If <operationStatus> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.</p>

1 **C.4.7 Modify Measurement Attributes**

2 The BTS Master Agent shall be able to modify measurement attributes
3 (e.g. parameters to be monitored, type of threshold, threshold value,
4 data reporting mode, and starting time) by using the
5 <setMeasAttributesReq> message.

6 The Module Agent shall respond with a <setMeasAttributesRsp>
7 message that contains new attribute designations.

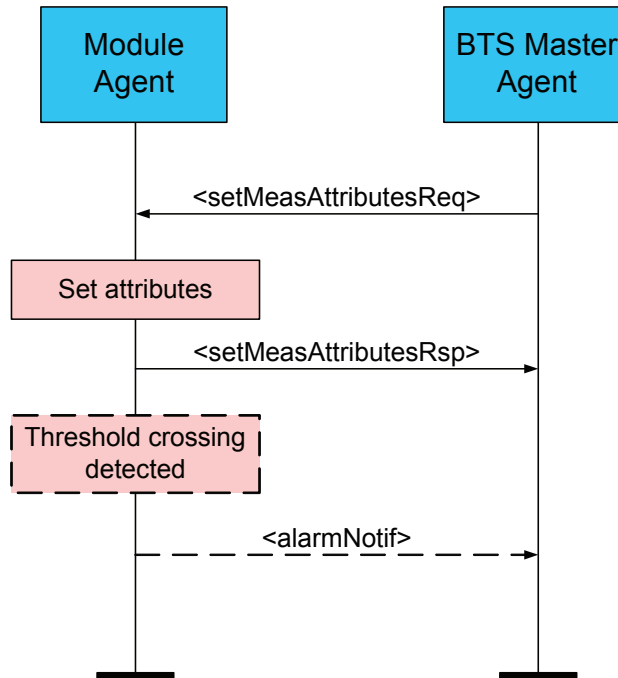


Figure 37 Modify Measurement Attributes and threshold-crossing Notifications

1
2
3
4
5
6
7
8

C.4.7.1 ‘Set Measurement Attributes Request’ Message Elements

The `<setMeasAttributesReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
measurement		Element containing measurement objects. There may be more than one instance of this element in this message.

	measType	Sub-element in <measurement> containing a measurement type e.g., traffic, link, etc. There may be more than one instance of this sub-element in this measurement element.
	managedObject	Sub-element in <measurement> that contains Managed Object, operation and parameter value(s). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	newAttributeValues	Element containing new attribute value(s) for the given parameter of Complex Type.

1

2

C.4.7.2 'Set Measurement Attributes Response' Message Elements

3

4

The <setMeasAttributesRsp> message shall contain the following SOAP Body elements:

5

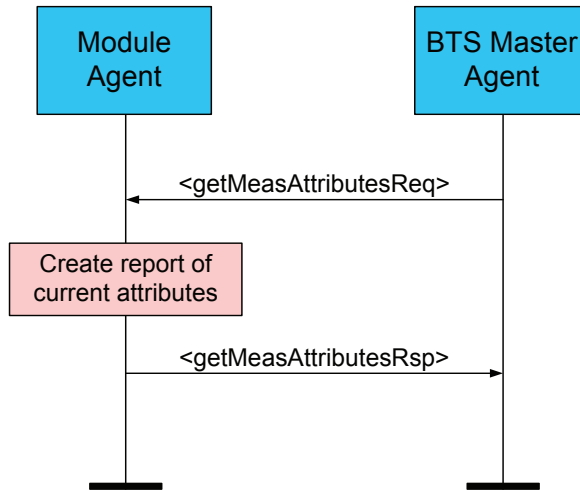
Element Name	Sub-element Name	Usage
measurement		Element containing measurement objects. There may be more than one instance of this element in this message.

	measType	Sub-element in <measurement> containing a measurement type e.g., traffic, link, etc. There may be more than one instance of this sub-element in this measurement element.
	managedObject	Sub-element in <measurement> that contains Managed Object, operation and parameter value(s). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	newAttributeValues	Element containing new attribute value(s) for the given parameter of Complex Type.
	oldAttributeValues	Element containing previous attribute value(s) for the given parameter of Complex Type.
status		Status of the requested operation: "OK": indicates that the operation was successful "fail": indicates that the operation failed
failureReason		If <status> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.

1 **C.4.8 Retrieve Measurement Attributes**

2 The BTS Master Agent shall be able to retrieve the current
3 measurement attributes by using <getMeasAttributesReq>
4 command.

- 1 The Module Agent shall respond with a `<getMeasAttributesRsp>`
 2 message.



3
 4 **Figure 38 Requesting Measurement Attributes**

5 **C.4.8.1 ‘Get Measurement Attributes Request’ Message**
 6 **Elements**

7 The `<getMeasAttributesReq>` message shall contain the following
 8 SOAP Body elements:

Element Name	Sub-element Name	Usage
measurement		Element containing measurement objects. There may be more than one instance of this element in this message.
	measType	Sub-element in <code><measurement></code> containing a measurement type e.g., traffic, link, etc. There may be more than one instance of this sub-element in this measurement element.

	managedObject	Sub-element in <measurement> that contains Managed Object, operation and parameter value(s). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	measAttribute	Element containing requested attribute for a managed object. There may be more than one instance of this element.

1 **C.4.8.2 'Get Measurement Attributes Response' Message**
2 **Elements**

3 The `getMeasAttributesRsp` message shall contain the following
4 SOAP Body elements:

Element Name	Sub-element Name	Usage
measurement		Element containing measurement objects. There may be more than one instance of this element in this message.
	measType	Sub-element in <measurement> containing a measurement type e.g., traffic, link, etc. There may be more than one instance of this sub-element in this measurement element.

	managedObject	Sub-element in <measurement> that contains Managed Object, operation and parameter value(s). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	newAttributeValues	Element containing new attribute value(s) for the requested attributes.
	oldAttributeValues	Element containing previous attribute value(s) for the requested attributes.

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

Appendix D Configuration Management

This appendix specifies the Configuration Management (CM) part of the OAM&P interface at RP1.

In the realm of OBSAI, Configuration Management covers the following CM functions:

- Module Configuration Management
- Resource State Control

The concepts for these are presented and the corresponding SOAP message structures are defined. However, the document does not contain a complete set of attributes that must be implemented.

It is assumed that the reader is familiar with the general concepts of XML and SOAP as defined by W3C® (<http://www.w3c.org>).

The configuration management functionality in the Module Agent shall fulfill the specified requirements in order to be compliant with this specification.

D.1 Version and history

The following table shall contain the version information associated with the CM aspect of OBSAI OAM&P specifications. This version information shall be used in the <version> element in the header of all CM messages.

Version number	Release information
1.0	First release of CM
Future release	<ul style="list-style-type: none"> • Changed the element name filename to fileName in this appendix. • Added operation to create and delete managed object in C.4.5.1

D.2 Specific CM Definitions

Resource: Physical or logical entity that is subject to system management. ITU-T X.730 recommends representing each resource in a system by a managed object.

Module: Physical entity (Hardware) that implements a logical BTS Block.

State: The management state of the managed object represents the instantaneous condition of availability and operability of the associated resource from the management point of view.

D.3 Configuration Management Concepts

D.3.1 Module Configuration Management

The BTS Master Agent is responsible for managing parameters in other modules by using CM messages specified in this document. The Master Agent shall be able to:

- Set parameter(s) on any module
- Retrieve parameter(s) from any module

- 1 • Receive 'parameter value changed' notifications from any
2 module

3 All supported parameters for module configuration shall be described in
4 the Module Property File by the module vendor.

5 **D.3.2 Resource State Control**

6 The Master Agent shall be able to:

- 7 • Change resource state(s) on any module
8 • Retrieve the current state of a resource on any module
9 • Receive 'state changed' notifications from any module

10 The recommended administrative and operational states of the
11 resources are defined in [ITU-T X.731].

12 It shall be possible to issue the resource state control commands to one
13 or more resources at a time.

14 **D.4 CM Messages**

15 This section describes the various CM operations specified by OBSAI.

16

17 **D.4.1 Module Ready Indication and Acknowledge**

18 The BTS Master and Module Agents perform auto-detection by
19 monitoring for the <moduleReadyInd> messages. This enables the
20 master agent and other modules to monitor each other's availability.

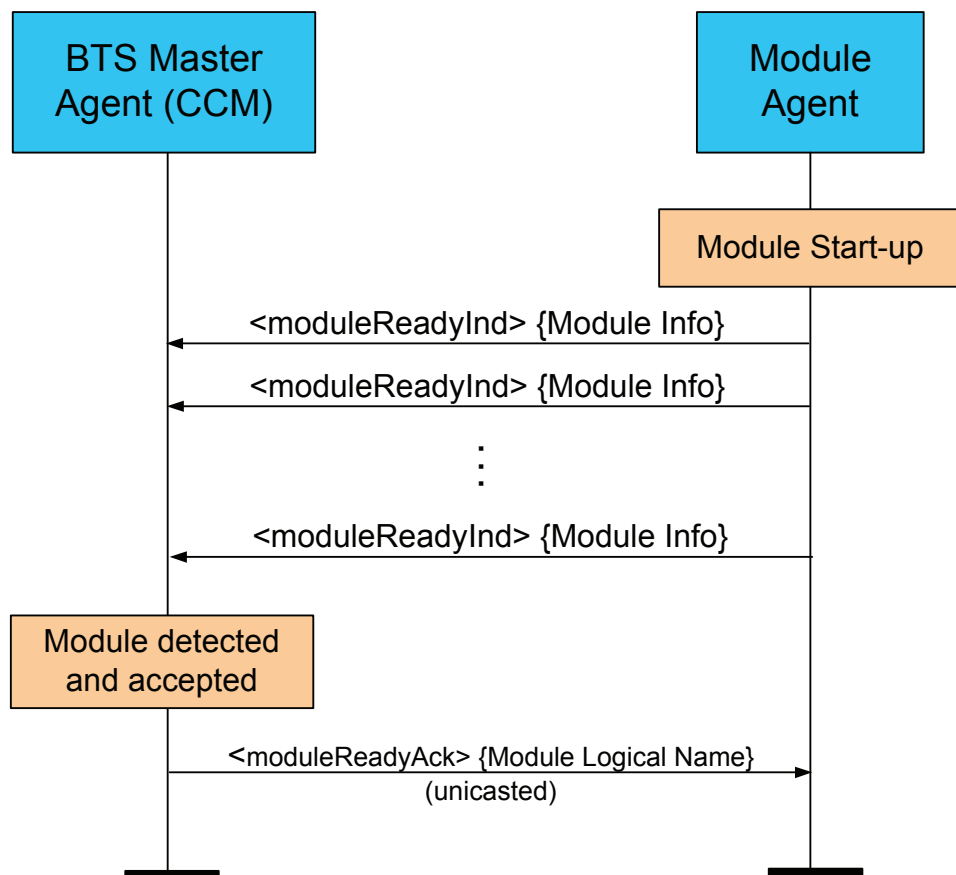


Figure 39 Module Ready Indication in the module start-up

The very first `<moduleReadyInd>` message that is sent to the master agent shall be used for registering the basic module information to the BTS Master Agent. Based on the received information, the BTS Master Agent is able to decide if a module can be integrated into the system.

When the BTS Master Agent accepts a module into the system, it shall reply with a unicast `<moduleReadyAck>` message to a module that contains a Module Identifier (“Logical ID”), for a module in the `<to>` element of the SOAP message header and other parameters in the message body, e.g. keep-alive polling period. A BTS module shall store the received Module Identifier in order to use it in subsequent messages to the master agent and when registering its services.

In case no `<moduleReadyAck>` is received from the BTS Master Agent, a module shall repeat the `<moduleReadyInd>` messages at a configured interval (seconds), until the BTS Master Agent replies.

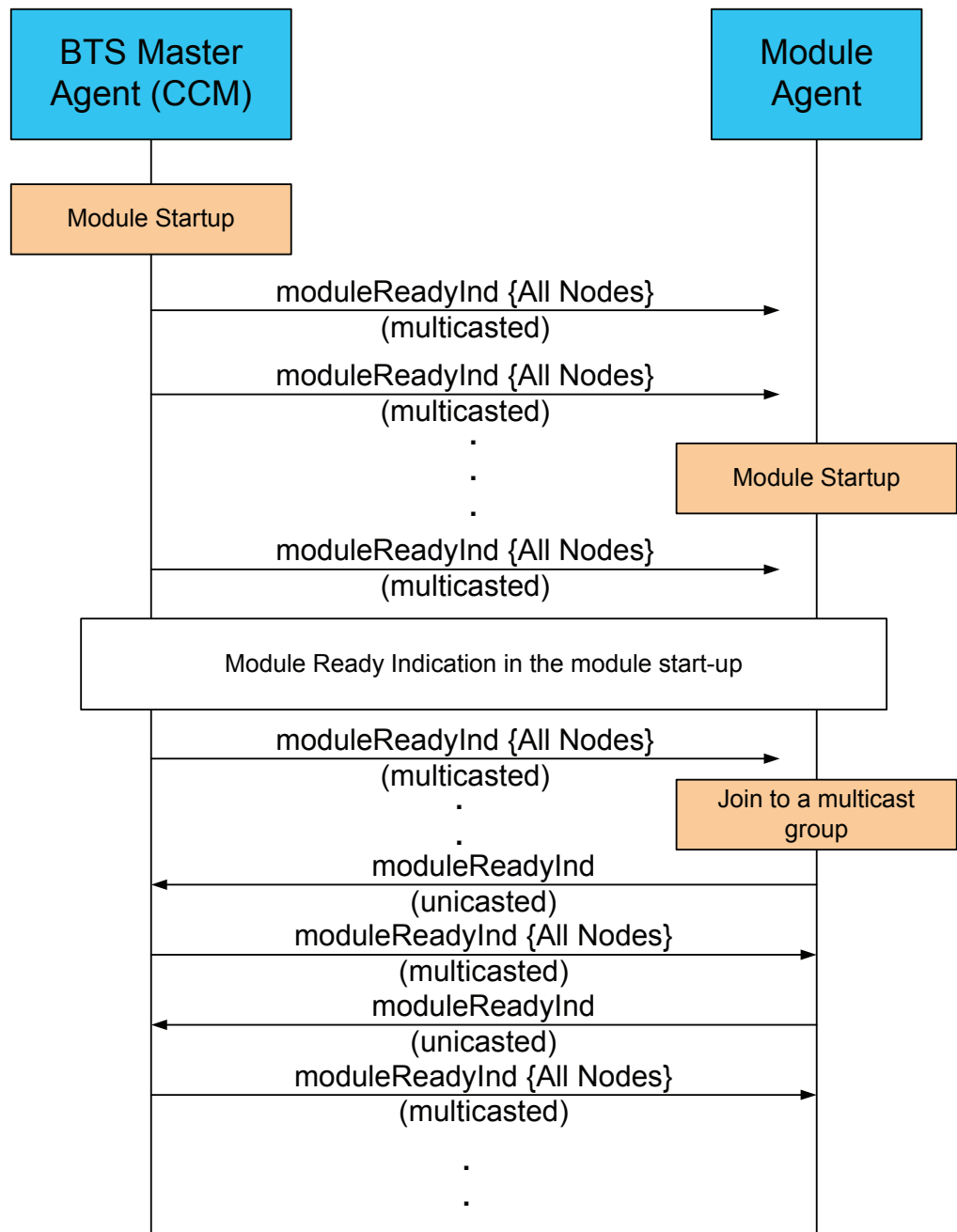


Figure 40 Module Ready Indication keep-alive polling

When the BTS Master Agent has detected and accepted a module (see Figure 39), the Module Agent shall start sending the <moduleReadyInd> messages at configured intervals.

The BTS Master Agent shall send <moduleReadyInd> messages at configured intervals to the modules that have registered their Auto Detection service. This <moduleReadyInd> sent by the master agent may be sent in one of the following two ways:

1
2
3
4
5
6
7
8
9
10

- 1 1. Multicasting to “all-nodes” group (IPv6) and/or all-hosts” group
2 (IPv4), or to a given link-local group. This requires each module
3 to join a given multicast group for monitoring
4 <moduleReadyInd> messages from the master agent.
- 5 2. Synchronously unicasting to all modules. If this method is used,
6 the BTS Master Agent shall omit the multicast group address
7 parameter in the <moduleReadyAck> message that is sent to a
8 newly inserted module.

9 **D.4.1.1 Module Ready Indication and Acknowledge message** 10 **elements**

11 The first <moduleReadyInd> message sent from a module to the
12 master agent shall contain the following SOAP Body elements:

Element Name	Usage
moduleType	This element is used to convey the type of the BTS Module to the master agent in the first <moduleReadyInd> message sent. This element may be omitted in subsequent messages
vendor	Name of the module vendor. May be omitted after the first message is accepted by the master agent.
productCode	Manufacturer specific Product Code of the module. May be omitted after the first message is accepted by the master agent.
serialNum	Serial Number of the module. May be omitted after the first message is accepted by the master agent.
rack	Optional module position information indicating the rack in which the module is installed. May be omitted after the first message is accepted by the master agent.
subRack	Optional module position information indicating the subrack in which the module is installed. May be omitted after the first message is accepted by the master agent.
slot	Optional module position information indicating the slot in which the module is installed. May be omitted after the first message is accepted by the master agent.
bldVersion	Element to convey the active software load version.

restartCause	<p>Indicates cause for the latest module restart:</p> <p>"power_on"</p> <p>"watchdog" – software failure detected by hardware watchdog logic</p> <p>"SW_failure" – software failure detected by software</p> <p>"management" - Restart initiated by external or internal management functions</p> <p><other> - any other failure e.g., "unknown"</p> <p>This element may be omitted after the first successful moduleReadyInd message is accepted by the master agent.</p>
timerValue	<p>This element conveys the keep-alive polling timer value in seconds.</p> <p>When this message originates from the CCM, this element shall convey a new value for this time.</p> <p>When this message originates from a module, this element is used to echo the latest timer value.</p> <p>This element may be omitted after the first successful moduleReadyInd message is accepted by the master agent.</p>
multiCastGroup	<p>This element is present only in messages originating from the master agent. It indicates the Multicast Group address to which the master agent shall send its <moduleReadyInd>.</p> <p>This element shall be omitted when the CCM is configured to send synchronously unicast <moduleReadyInd> messages.</p>

1
2
3
4
5
6
7
8
9
10
11
12

In the first <moduleReadyInd> message sent by the module, the <From> element of the SOAP Header shall be empty indicating that it has not yet received its Module Identifier (as will be the case for a newly inserted module) from the BTS Master Agent.

Once a module insertion has been accepted and acknowledged by the master agent, subsequent <moduleReadyInd> messages sent from that module may have several of the elements omitted as shown in the table above. The <from> element in the header shall have the Module Identifier assigned to it by the master agent.

The <moduleReadyAck> message is unicast by the master agent to the module in response to the module's first successful

1 <moduleReadyInd> message. The elements of <moduleReadyAck>
2 message are as shown below:

Element Name	Usage
moduleType	This element is used to echo the type of the BTS Module that the master agent is acknowledging.
serialNum	This element is used to echo the serial number of the BTS Module that the master agent is acknowledging.
timerValue	This element conveys the keep-alive polling timer value in seconds assigned to the module by the master agent.
moduleId	An element containing a logical Module Identifier assigned to a module by the master agent.
multiCastGroup	This element indicates the Multicast Group address to which the master agent shall send its <moduleReadyInd>. This element shall be omitted when the master agent is configured to send synchronously unicast <moduleReadyInd> messages.

3 **D.4.2 Modify Resource State**

4 The BTS Master Agent shall be able to modify resource states (e.g.
5 activate, block, unblock, shutdown, restart) using the
6 <modifyStateReq> message.

7 When resource state(s) is/are changed, the Module Agent shall send a
8 <stateChangeInd> message (see D.4.3) to all clients that have
9 subscribed to such state change notifications.

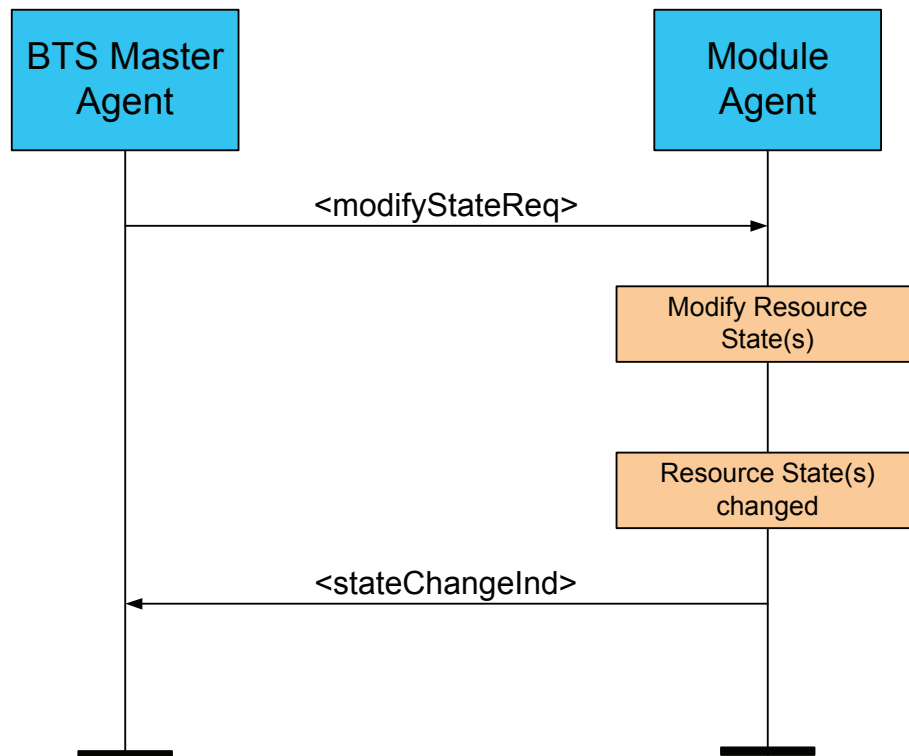


Figure 41 Modify Resource State

1
2
3
4
5
6

D.4.2.1 Modify Resource State Message Elements

The `<modifyStateReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
managedObject		Element that contains Managed Object). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	stateType	The state attribute type of the managed object
	newState	New state of the specified attribute of the managed object.

D.4.3 Resource State Changed Indication

The <stateChangeInd> message shall be used by the Module Agent for communicating resource state changes to the Master Agent and optionally to other clients that have subscribed to state change notifications.

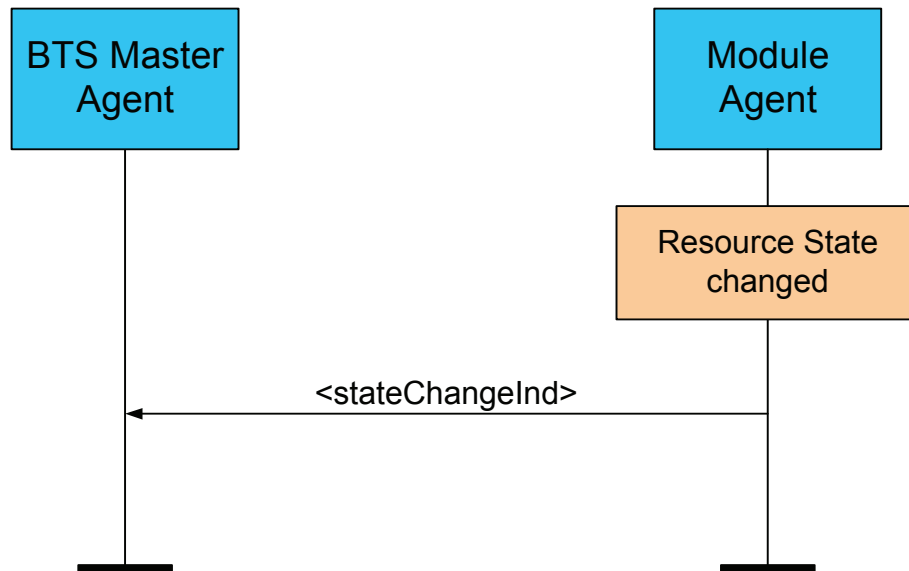


Figure 42 Resource State Changed Indication

D.4.3.1 State Change Indication Message Elements

The <stateChangeInd> message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
managedObject		Element that contains Managed Object. Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	stateType	Indicates the reported state attribute type of the Managed Object
	newState	New state of the Managed Object

	prevState	Old state of the Managed Object
source		Indicates the cause of the operation that led to this notification: "resource": state changed by a resource on the module "management": state changed by management functions <other>: other reasons e.g., "unknown"
status		If the <source> is equal to "management", then this element shall indicate the status of the operation: "OK": indicates that the operation was successful "fail": indicates that the operation failed
failureReason		If <status> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.

1

2 **D.4.4 Retrieve Resource State**

3 The BTS Master Agent shall be able to request the
 4 operational/administrative state of a module, or a resource within a
 5 module, using the <resourceStateReq> message.

6 The Module Agent shall respond with a <resourceStateRsp>
 7 message that contains the state(s) of the requested resource(s).

8

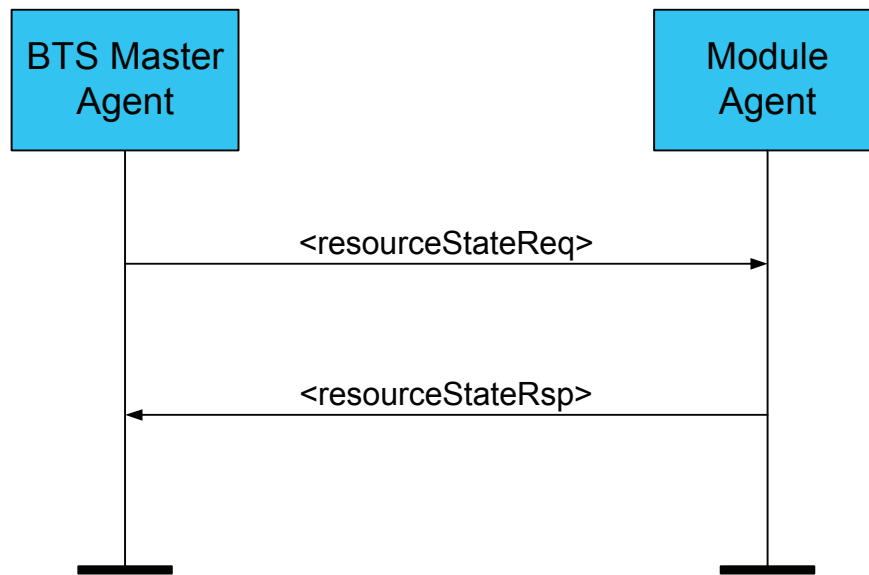


Figure 43 Retrieve Resource State

D.4.4.1 Retrieve Resource State Message Elements

The `<resourceStateReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
managedObject		Element that contains Managed Object). Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	stateType	Indicates the requested state attribute type of the managed object

The corresponding reply message `<resourceStateRsp>` shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
--------------	------------------	-------

managedObject		Element that contains Managed Object. Complex Type containing attributes: class: class name (optional) distName: distinguishedName vendor: vendor information (optional)
	stateType	Indicates the reported state attribute type of the Managed Object
	newState	New state of the Managed Object
	prevState	Old state of the Managed Object
source		Indicates the cause of the operation that led to this notification: "resource": state changed by a resource on the module "management": state changed by management functions <other>: other reasons e.g., "unknown"
status		If the <source> is equal to "management", then this element shall indicate the status of the operation: "OK": indicates that the operation was successful "fail": indicates that the operation failed
failureReason		If <status> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.

D.4.5 Modify Parameters

The BTS Master Agent shall be able to configure module parameters with the `<modifyParameterReq>` message that contains the parameter values to be set into the module.

When parameter(s) is/are changed, the Module Agent shall send a `<parValueChangeInd>` message (see D.4.6) that may be sent to all clients that have subscribed to parameter change notifications.

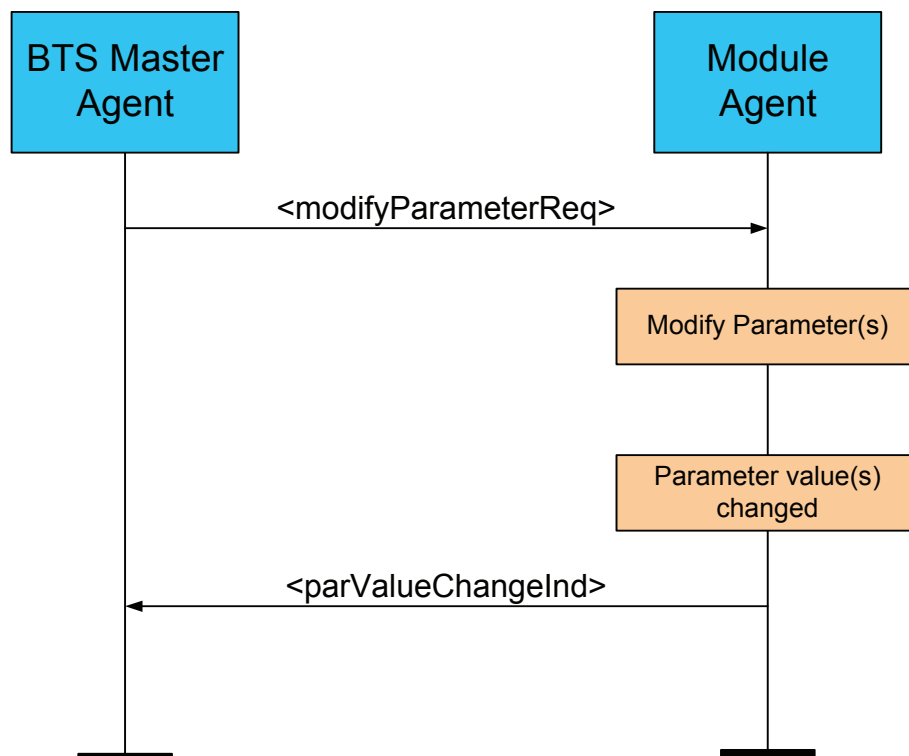


Figure 44 Modify Parameters Request

D.4.5.1 Modify Parameters Request Message Elements

The `<modifyParameterReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage

fileRetrievalInfo		This element contains file retrieval information. If this is specified then the <managedObject> element and its associated sub-elements are omitted.
	protocol	Indicates the preferred file transfer protocol for retrieving the measurement data: <ul style="list-style-type: none"> - "FTP" - "TFTP" - "HTTP" - "SOAP"
	userId	Optional user Id to be used for file retrieval.
	password	Optional password to be used for file retrieval.
	fileURL	If the <protocol> is "HTTP", the URL of the bulk data file shall be provided. Otherwise, this element shall not be present in this message.
	ipAddress	If the protocol is either "FTP" or "TFTP", the IP Address to open a file transfer session shall be provided. Otherwise, this element shall not be present in this message.
	fileName	If the protocol is either "FTP" or "TFTP", this element identifies an individual filename including path. Otherwise, this element shall not be present in this message.

	format	If the protocol is either "FTP" or "TFTP", this element identifies the file format of the bulk data file e.g.3GPP_PM_XML, 3GPP_PM_ASN1...or proprietary. Otherwise, this element shall not be present in this message.
managedObject		If <fileRetrievalInfo> is not specified, this element shall contain the Managed Object element of Complex Type containing attributes: - class (optional) - distName: distinguished name - vendor (optional) There may be more than one instance of this element in this message
	operation	Sub-element of <managedObject> that indicates the requested activity targeted on the Managed Object. "create_mo": instantiate a managed object of the given class with the given distinguished name "create_parm": instantiate the specified parameter "update_parm": update the specified parameter(s) "delete_mo": delete this managed object. "delete_parm": delete the specified parameter.
	parameter	This sub-element of <managedObject> of Complex Type containing sub-elements: - parameter name - newValue - prevValue There may be multiple instances of this sub-element in <managedObject>.

D.4.6 Parameter Value Changed Indication

The Module Agent shall use `<parValueChangeInd>` message for communicating parameter value changes to the Master Agent and to other clients that have subscribed parameter change notifications.

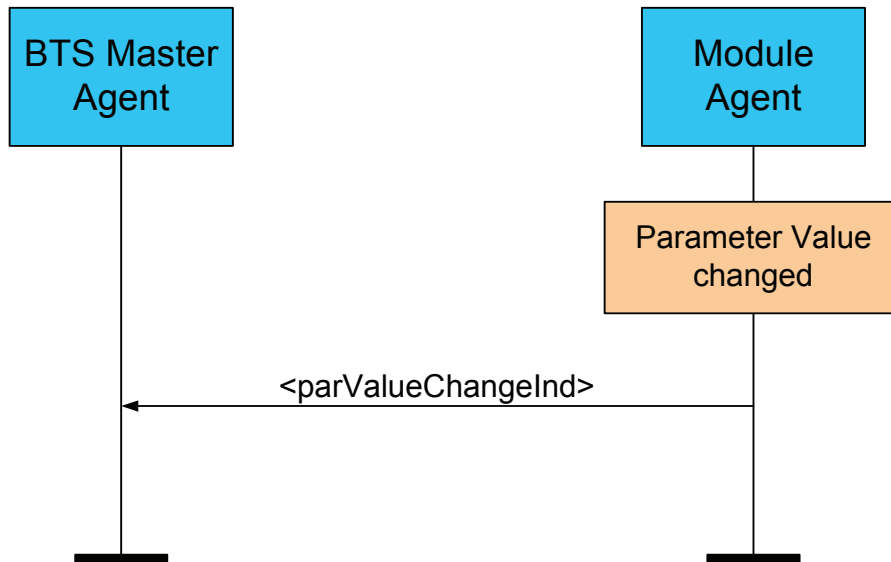


Figure 45 Parameter Value Changed Indication

D.4.6.1 Parameter Value Change Indication Message Elements

The `<parValueChangeInd>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
--------------	------------------	-------

managedObject		<p>This element shall contain the Managed Object element of Complex Type containing attributes:</p> <ul style="list-style-type: none"> - class (optional) - distName: distinguished name - vendor (optional) <p>There may be more than one instance of this element in this message</p>
	parameter	<p>This sub-element of <managedObject> of Complex Type containing sub-elements:</p> <ul style="list-style-type: none"> - parameter name - newValue - prevValue <p>There may be multiple instances of this sub-element in <managedObject>..</p>
source		<p>Indicates the cause of the operation that led to this notification:</p> <p>“resource”: state changed by a resource on the module</p> <p>“management”: state changed by management functions</p> <p><other>: other reasons e.g., “unknown”</p>
status		<p>If the <source> is equal to “management”, then this element shall indicate the status of the operation:</p> <p>“OK”: indicates that the schedule was set successfully</p> <p>“fail”: indicates failure</p>
failureReason		<p>If <status> element is present and is equal to “fail”, this field shall</p>

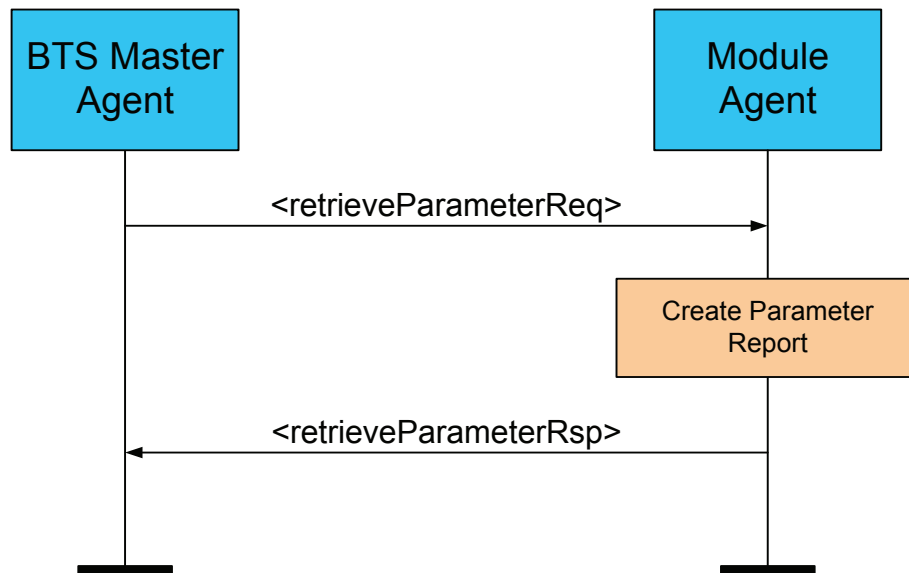
		indicate the reason for the failure. Otherwise, it shall be omitted.
--	--	--

1 **D.4.7 Retrieve Parameters**

2 The BTS Master Agent shall use a <retrieveParameterReq>
3 message to retrieve module parameter(s) or properties.

4 The Module Agent shall respond with a <retrieveParameterRsp>
5 message that contains the requested parameters.

6



7

8

Figure 46 Retrieve Parameters

9

10 **D.4.7.1 Retrieve Parameters Request Message Elements**

11 The <retrieveParameterReq> message shall contain the following
12 SOAP Body elements:

Element Name	Sub-element Name	Usage
scope		If this element is set to "all", the request shall apply to all managed objects and the element <managedObject> and its associated sub-elements shall be omitted.

managedObject		<p>This element contains the Managed Object and requested parameter types. Complex Type containing attributes:</p> <ul style="list-style-type: none"> - class (optional) - distName - vendor (optional) <p>There may be more than one instance of <managedObject> and its associated sub-elements in this message.</p>
	parameter	<p>This sub-element of <managedObject> of Complex Type containing sub-elements:</p> <ul style="list-style-type: none"> - parameter name - newValue - prevValue <p>There may be multiple instances of this sub-element in <managedObject>.</p>

1
2
3

The corresponding reply message <retrieveParameterRsp> shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
fileRetrievalInfo		This element contains file retrieval information. If this is specified, <managedObject> element and its associated sub-elements shall be omitted.

	protocol	Indicates the preferred file transfer protocol for retrieving the measurement data: <ul style="list-style-type: none"> - "FTP" - "TFTP" - "HTTP" - "SOAP"
	userId	Optional user Id to be used for file retrieval.
	password	Optional password to be used for file retrieval.
	fileURL	If the <protocol> is "HTTP", the URL of the bulk data file shall be provided. Otherwise, this element shall not be present in this message.
	ipAddress	If the protocol is either "FTP" or "TFTP", the IP Address to open a file transfer session shall be provided. Otherwise, this element shall not be present in this message.
	fileName	If the protocol is either "FTP" or "TFTP", this element identifies an individual filename including path. Otherwise, this element shall not be present in this message.
	format	If the protocol is either "FTP" or "TFTP", this element identifies the file format of the bulk data file e.g.3GPP_PM_XML, 3GPP_PM_ASN1...or proprietary. Otherwise, this element shall not be present in this message.
	managedObject	If <fileRetrievalInfo> is not specified, this element shall contain the Managed Object element of Complex Type containing attributes: <ul style="list-style-type: none"> - class (optional) - distName: distinguished name - vendor (optional) There may be more than one instance of this element in this message

	parameter	<p>This sub-element of <managedObject> of Complex Type containing sub-elements:</p> <ul style="list-style-type: none"> - parameter name - newValue - prevValue <p>There may be multiple instances of this sub-element in <managedObject>.</p>
--	-----------	--

1

Appendix E Software Management

This appendix specifies the Software Management (SwM) part of the OAM&P interface at RP1.

In the realm of OBSAI, Software Management covers the following SwM functions:

- Downloading new software as well as software updates
- Managing software

The concepts for these are presented and the corresponding SOAP message structures are defined. However, the document does not contain a complete set of attributes that must be implemented.

It is assumed that the reader is familiar with the general concepts of XML and SOAP as defined by W3C® (<http://www.w3c.org>).

The SwM service of the Module Agent shall fulfill the specified requirements in order to be compliant with this specification.

E.1 Version and history

The following table shall contain the version information associated with the SwM aspect of OBSAI OAM&P specifications. This version information shall be used in the `<version>` element in the header of all SwM messages.

Version number	Release information
1.0	First release of SwM
Future release	Changed element name filename to fileName in this appendix.

E.2 SwM Concepts

The OBSAI RP1 interface specification for SwM enables a variety of operations to be performed on the module software. The following is a list of such operations:

- Downloading new software on the various modules

- 1 • Downloading updates to the existing software load on the various
2 modules
- 3 • Managing the software on the various modules
- 4 ○ Request version and inventory information
- 5 ○ Activate a specific software version
- 6 ○ Check software Integrity
- 7 The base station may have the capability to store more than one
8 version of the software load. The number of software versions that can
9 be stored and the management of the required storage space are not
10 specified by OBSAI.

11 **E.3 SwM Interface Functions**

12 This section describes the various SwM operations specified by OBSAI.

13 **E.3.1 Download Software**

14 The BTS Master Agent shall be able to request the module agent to
15 download software file(s) using a file transfer protocol specified in the
16 <swDownloadReq> message. The module agent shall download the
17 file(s) from the specified file server. Files downloaded to the internal
18 BTS modules shall be downloaded from the CCM. Optionally, an
19 alternative location for an external file server may also be specified. On
20 completion of the download, the module agent shall send a response
21 <swDownloadRsp> to the BTS master agent.

22

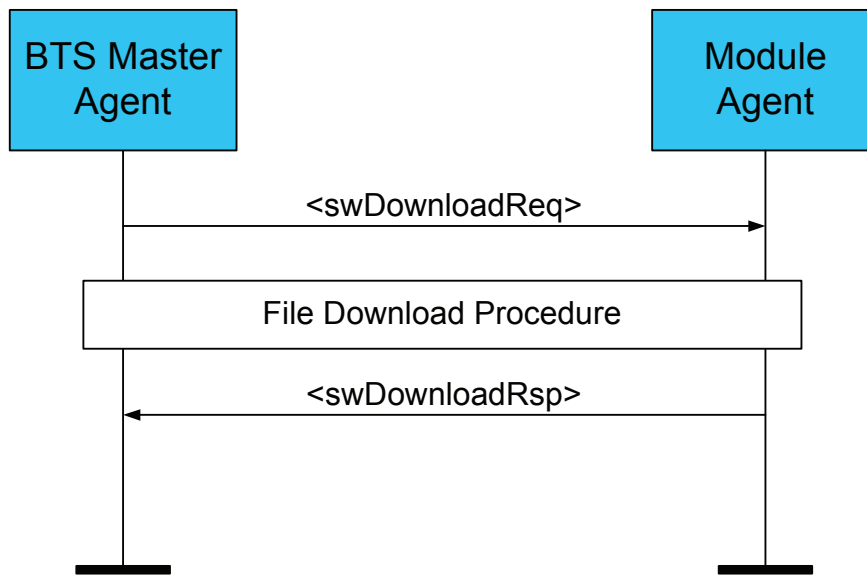


Figure 47 Software Download Request-Response

1
2
3
4
5

The `<swDownloadReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
fileRetrievalInfo		This element contains file retrieval information.

protocol	Indicates the preferred file transfer protocol for retrieving the measurement data: <ul style="list-style-type: none"> - "FTP" - "TFTP" - "HTTP" - "SOAP"
userId	Optional user Id to be used for file retrieval.
password	Optional password to be used for file retrieval.
fileURL	If the <protocol> is "HTTP", the URL of the file shall be provided. Otherwise, this element shall not be present in this message.
ipAddress	If the protocol is either "FTP" or "TFTP", the IP Address to open a file transfer session shall be provided. Otherwise, this element shall not be present in this message.
format	If the protocol is either "FTP" or "TFTP", this element identifies the file format of the file e.g.3GPP_PM_XML, 3GPP_PM_ASN1...or proprietary. Otherwise, this element shall not be present in this message.
sourceFilename	This element identifies a source filename including path.
fileVersion	Version of the SW file
destFilename	This element identifies the destination filename including path.

1
2
3

The corresponding reply message <swDownloadRsp> shall contain the following SOAP Body elements:

Element Name	Usage
destFilename	Name of the downloaded file
status	This element shall indicate the status of the operation: "OK": indicates that the operation was successful

	"fail": indicates failure
failureReason	If <status> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.

E.3.2 Inventory Request

The BTS Master Agent shall be able to perform an inventory of software and active processes on a module, using the `swInventoryReq` command.

The module shall respond with the version information and a list of active processes, using the `<swInventoryRsp>` message.

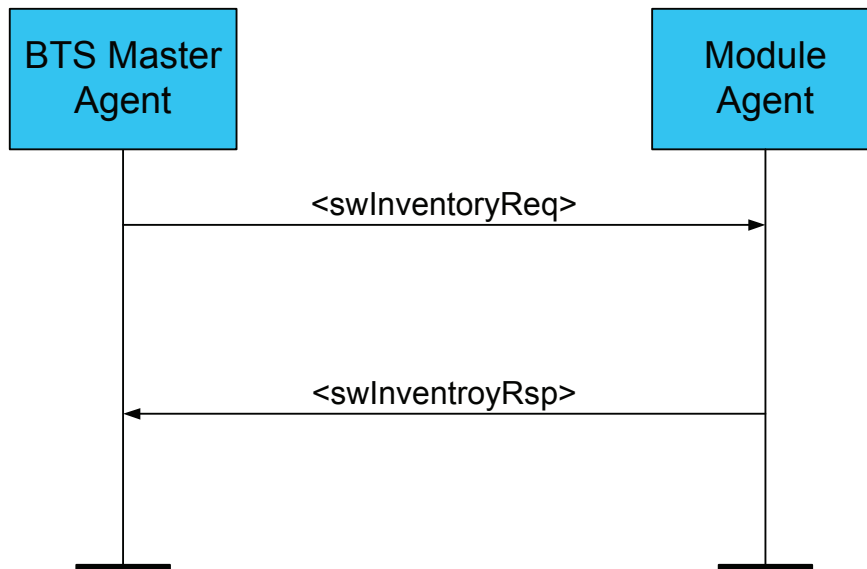


Figure 48 Software Inventory Request-Response

The `<swInventoryReq>` message body has a no elements other than the message name tag.

The corresponding reply message `<swInventoryRsp>` shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
bldData		Element containing software build data
	bldName	Name of an individual module SW build
	bldVersion	Version of an individual module SW build

	fileData	<p>Element of Complex Type containing the following sub-elements:</p> <p>fileName: name of the file including the path</p> <p>fileVersion: Version of the file</p> <p>fileIntegrity: Result of the file integrity check (checksum calculation):</p> <p> "OK" : indicates file integrity check passed</p> <p> "NOK" : indicates file integrity check failed</p> <p>usageStatus: Usage status information of the file:</p> <p> "active": indicates that file is in use</p> <p> "passive": indicates file is not in use</p>
	activeProcesses	Optional element containing names of active processes.

1 **E.3.3 Check SW Integrity**

2 The BTS Master Agent shall be able to request the module to check the
3 integrity of specified software using the <swCheckReq> command.

4 The module shall report the result of the software integrity test using the
5 <swCheckRsp> message. The module agent shall use the algorithm
6 specified in the request message. If no algorithm is specified then the
7 module agent shall use a default algorithm.

8

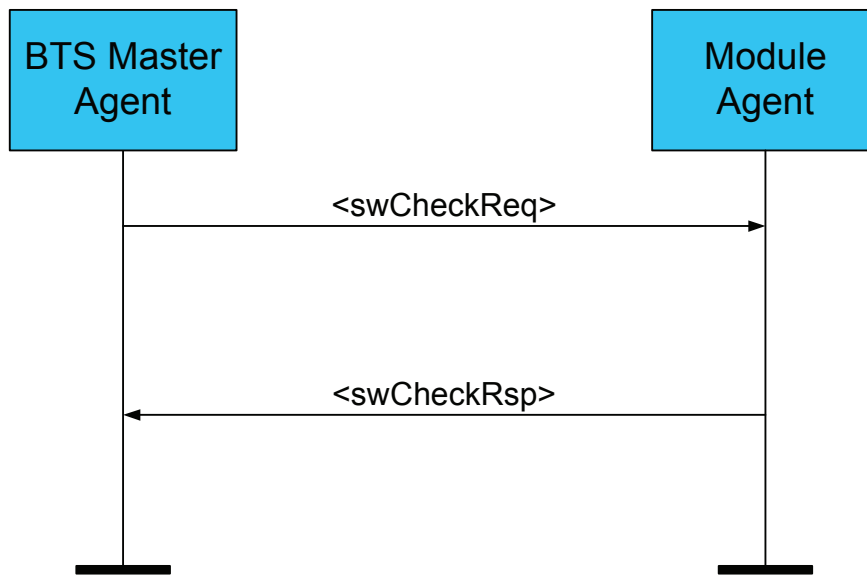


Figure 49 Software Check Request-Response

The `<swCheckReq>` message shall contain the following SOAP Body elements:

Element Name	Usage
fileName	Name of an individual file in the active build on a module.
integrityAlgorithm	Optional element indicating the integrity algorithm to be used.

The corresponding reply message `<swCheckRsp>` shall contain the following SOAP Body elements:

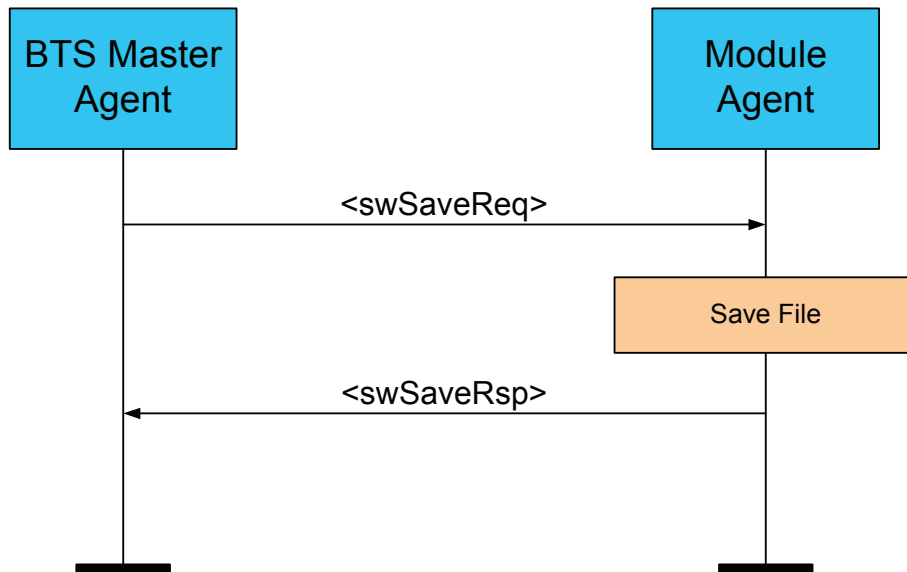
Element Name	Usage
fileName	Name of an individual file in the active build on a module.
fileVersion	Version of the file in the active build on a module.
fileIntegrity	Indicates file status as follows: "OK" "FAIL" or, checksum of the file
integrityAlgorithm	Optional element indicating the integrity algorithm used

E.3.4 Save Software

The BTS Master Agent shall be able to request a module to save a downloaded file by using the `<swSaveReq>`.

1
2

The module shall save the file locally and communicate the result of the operation to the BTS Master Agent, using the `<swSaveRsp>` message.



3
4
5
6
7

Figure 50 Save File Request-Response

The `<swSaveReq>` message shall contain the following SOAP Body elements:

Element Name	Usage
fileName	Name of an individual file to be saved
checksum	Checksum of the file to be saved provided by the master agent.
fileVersion	Version of the file
fileType	Indicates type of a file to be saved e.g., Boot, Runtime, Module Property File etc.

8
9
10

The corresponding reply message `<swSaveRsp>` shall contain the following SOAP Body elements:

Element Name	Usage
fileName	Name of an individual file that was saved
fileVersion	Version of the file
status	Status of the file save operation: "OK": indicates successful file save "fail": indicates failed file save
failureReason	If <code><status></code> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it

	shall be omitted.
--	-------------------

E.3.5 Activate Specified Software

The BTS Master Agent shall be able to request the module agent to activate a specified version of software using the `<swActivateReq>` message. In response to this message, the module agent shall ensure that the requested version of the software is present on the module and respond with the results using the `<swActivateRsp>` message.

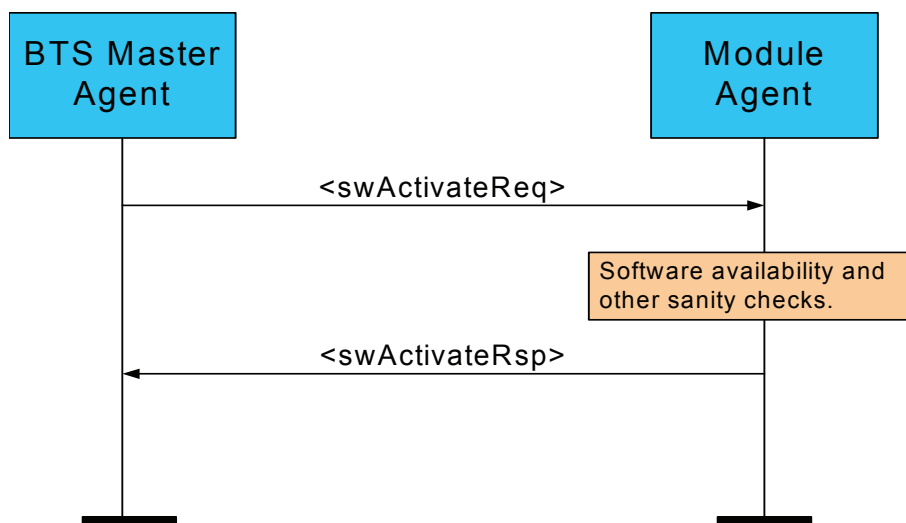


Figure 51 Activate Software Request-Response

The `<swActivateReq>` message shall contain the following SOAP Body elements:

Element Name	Sub-element Name	Usage
ActivationOp		This element indicates the operation to be executed in a module e.g., BuildUpdate, TakeInUse etc.
bldData		Element containing software build data
	bldName	Name of an individual module SW build
	bldVersion	Version of an individual module SW build

	fileData	Element of Complex Type containing the following sub-elements: fileName: name of the file including the path fileVersion: Version of the file
--	----------	---

1

2

3

The corresponding reply message <swActivateRsp> shall contain the following SOAP Body elements:

Element Name	Usage
status	status of the activate SW operation: "OK" indicates SW can be activated "fail" indicates SW activation failure
failureReason	If <status> element is present and is equal to "fail", this field shall indicate the reason for the failure. Otherwise, it shall be omitted.

4

5

Glossary

6

Abbreviations

7

For the purposes of the present document, the following abbreviations apply:

8

AAL	ATM Adaptation Layer
AAL2	ATM Adaptation Layer type 2
AAL5	ATM Adaptation Layer type 5
AC	Authentication Center
ADDS	Application Data Delivery Service
AL	Air Link
ALCAP	Access Link Control Application Part
AMPS	Advanced Mobile Phone System

ANID	Access Network Identifiers
ANSI	American National Standards Institute
ARFCN	Absolute Radio Frequency Channel Number
ATM	Asynchronous Transfer Mode
AUTHR	Authentication Response
AUTHU	Unique Challenge Authentication Response
BM-IWF	Broadcast Multicast Interworking Function
BMC	Broadcast/Multicast Control
BS	Base Station
BSAP	Base Station Application Part
BSC	Base Station Controller
BSMAP	Base Station Management Application Part
BSS	Base Station Subsystem
BTS	Base Transceiver System
CANID	Current Access Network Identifiers
CBC	Cell Broadcast Centre
CBS	Cell Broadcast Service
CC	Call Control
CDG	CDMA Development Group
CDMA	Code Division Multiple Access
CID	Connection Identifier (used with reference to AAL2)
CIE	Content Information Element
SwM	Connection Management
CN	Core Network
CNIP	Calling Number Identification Presentation
CNIR	Calling Number Identification Restriction
COUNT	Call History Count
CPCH	Common Packet Channel
CRNC	Controlling Radio Network Controller
DCCH	Dedicated Control Channel
DCH	Dedicated Channel
DL	Downlink
DLCI	Data Link Connection Identifier
DLR	Destination Local Reference
DPC	Destination Point Code
DRNS	Drift RNS
DRS	Data Ready to Send
DS	Direct Spread

DS0	Digital Signal Level 0
DS1	Digital Signal Level 1
DS-41	An operational mode in which the BS and MS operate with the direct spread (DS) radio layers of the UMTS system defined by 3GPP, and the upper layers defined in IS-2000 that conform to and interoperate with ANSI-41 based networks.
DTAP	Direct Transfer Application Part
DTX	Discontinuous Transmission
EDGE	Enhanced Data rates for Global Evolution
EIA	Electronics Industry Association
EIB	Erasure Indicator Bit
ESN	Electronic Serial Number
EVRC	Enhanced Variable Rate Codec
FACH	Forward Access Channel
FCH	Fundamental Channel
FER	Frame Error Rate
FFS	For Further Study
FPC	Forward Power Control
FQI	Frame Quality Indicator
FSN	Frame Sequence Number
GERAN	GSM EDGE Radio Access Network
GR	Gain Ratio
GRE	Generic Routing Extension
GSM	Global System for Mobile Communications
GTP	GPRS Tunnelling Protocol
IEI	Information Element Identifier
IETF	Internet Engineering Task Force
IMSI	International Mobile Subscriber Identity
IOS	Interoperability Specification
IP	Internet Protocol
IPv4	Internet Protocol, version 4
IPv6	Internet Protocol, version 6
ISDN	Integrated Services Digital Network
ISLP	Intersystem Link Protocol
ITU	International Telecommunications Union
IWF	Interworking Function
kb	kilo bits
LA	Location Area
LAC	Location Area Code

LI	Length Indicator
LSB	Least Significant Bit
MAC	Medium Access Control
MIN	Mobile Identification Number
MIP	Mobile IP
MM	Mobility Management
MS	Mobile Station
MSB	Most Significant Bit
MSC	Mobile Switching Center
MTP	Message Transfer Part
MTP1	Message Transfer Part Layer 1
MTP2	Message Transfer Part Layer 2
MTP3	Message Transfer Part Layer 3
MWI	Message Waiting Indication
N-AMPS	Narrow band AMPS
NAS	Non Access Stratum
NBAP	Node B Application Part
NID	Network Identification
NNSF	NAS Node Selection Function
NSAP	Network Service Access Point
OAM&P	Operations, Administration, Maintenance, and Provisioning
OLT	Outer Loop Threshold
OTAF	Over The Air Function
OTAPA	Over The Air Parameter Administration
OTASP	Over The Air Service Provisioning
OTD	Orthogonal Transmit Diversity
PACA	Priority Access and Channel Assignment
PANID	Previous Access Network Identifiers
PATE	Packet Arrival Time Error
PCF	Packet Control Function
PCH	Paging Channel
PDE	Position Determining Entity
PDSN	Packet Data Serving Node
PIN	Personal Identification Number
PLD	Position Location Data
PLMN	Public Land Mobile Network
PMC	Packet Mode Channel
PN	Pilot Number

PPP	Point to Point Protocol
PSTN	Public Switched Telephone Network
PZID	Packet Zone Identifier
QCELP	Q Code Excited Linear Prediction
QOF	Quasi Orthogonal Function
QoS	Quality of Service
QPCH	Quick Paging Channel
RAB	Radio Access Bearer
RACH	Random Access Channel
RANAP	Radio Access Network Application Part
RAND	Random Variable
RANDC	Random Confirmation
RANDSSD	Random SSD
RANDU	Random Variable - Unique Challenge
RC	Radio Configuration, Radio Class
RF	Radio Frequency
RFC	Remote Feature Control
RLC	Release Complete (SCCP)
RLP	Radio Link Protocol
RLSD	Release (SCCP)
RNC	Radio Network Controller
RNL	Radio Network Layer
RNS	Radio Network Subsystem
RNSAP	Radio Network Subsystem Application Part
RNTI	Radio Network Temporary Identity
RPC	Reverse Power Control
SAB	Service Area Broadcast
SAS	Standalone A-GPS SMLC
SAT	Supervisory Audio Tone
SCCP	Signaling Connection Control Part
SCH	Supplemental Channel
SDB	Short Data Burst
SDU	Selection/Distribution Unit
SID	System Identification
SLR	Source Local Reference
SLTM	Signaling Link Test Message
SME	Signaling Message Encryption
SMS	Short Message Service

SMS-MO	SMS Mobile Originated
SMS-MT	SMS Mobile Terminated
SNA	Shared Network Area
SOAP	Simple Object Access Protocol
SOCI	Service Option Connection Identifier
SPI	Security Parameter Index
SRNC	Serving Radio Network Controller
SRNC-ID	Source Radio Network Controller Identifier
SRNS	Serving RNS
S-RNTI	Source Radio Access Network Temporary Identifier
SSCF	Service Specific Convergence Function
SSCOP	Service Specific Connection Oriented Protocol
SSD	Shared Secret Data
STP	Signal Transfer Point
TCP	Transmission Control Protocol
TEID	Tunnel Endpoint Identifier
TFA	Transfer-Allowed Signal
TFO	Tandem Free Operation
TFP	Transfer-Prohibited Signal
TFR	Transfer-Restricted Signal
TIA	Telecommunications Industry Association
TMSI	Temporary Mobile Station Identity
TNL	Transport Network Layer
TSB	Telecommunications Systems Bulletin
TTI	Transmission Time Interval
UDI	Unrestricted Digital Information
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunication System
USIM	UMTS Subscriber Identity Module
UTRAN	Universal Terrestrial Radio Access Network
VLR	Visitor Location Register
VP	Voice Privacy

1 Definition of Terms

2 For the purposes of the present document, the following terms and
3 definitions apply:

4 **Abis:** designation of the interface between the Base Transceiver
5 System (BTS) and the Base Station Controller (BSC) for IS-2000.
6 TIA/EIA-828-A provides the specifications for this interface.

7 **ALCAP:** generic name in 3GPP systems for the transport signaling
8 protocols used to set-up and tear-down transport bearers

9 **Cell:** Radio Network object that can be uniquely identified by a User
10 Equipment from a (cell) identification that is broadcasted over a
11 geographical area from one radio node.

12 **Iub:** interface between the RNC and the Node B for 3GPP.

13 **Logical Model:** Logical Model defines an abstract view of a network or
14 network element by means of information objects representing network
15 element, aggregations of network elements, the topological relationship
16 between the elements, endpoints of connections (termination points),
17 and transport entities (such as connections) that transport information
18 between two or more termination points

19 The information objects defined in the Logical Model are used, among
20 others, by connection management functions. In this way, a physical
21 implementation independent management is achieved.

22 **Node B:** logical node in the OBSAI System responsible for radio
23 transmission / reception in one or more cells to/from the UE.
24 The logical node terminates the Iub interface towards the RNC.

25 **Packet:** Used interchangeably with the term fragment in the context of
26 UDPCP.

27 **Radio Resources:** resources that constitute the radio interface in
28 OBSAI, e.g. frequencies, scrambling codes, spreading factors, power
29 for common and dedicated channels

30 **Radio Node B Application Part:** Radio Network Signaling over the Iub

31 **Radio Network Controller:** logical node in the RNS in charge of
32 controlling the use and the integrity of the radio resources

33 **Controlling RNC:** role an RNC can take with respect to a specific set of
34 Node B's. There is only one Controlling RNC for any Node B. The
35 Controlling RNC has the overall control of the logical resources of its
36 node B's.

37 **Radio Network Subsystem:** RNS can be either a full UTRAN or only a
38 part of a UTRAN. An RNS offers the allocation and release of specific
39 radio resources to establish means of connection in between an UE and
40 the UTRAN. A Radio Network Subsystem contains one RNC and is

- 1 responsible for the resources and transmission/reception in a set of
2 cells.
- 3 **Serving RNS:** RNS can take this role with respect to a specific
4 connection between a UE and UTRAN. There is one Serving RNS for
5 each UE that has a connection to UTRAN. The Serving RNS is in
6 charge of the radio connection between a UE and the UTRAN. The
7 Serving RNS terminates the lu for this UE.
- 8 **Radio Access Network Application Part:** Radio Network Signaling
9 over the lu
- 10 **RRC Connection:** point-to-point bi-directional connection between
11 RRC peer entities on the UE and the UTRAN sides, respectively
12 An UE has either zero or one RRC connection.
- 13 **User Equipment:** Mobile Equipment with one or several UMTS
14 Subscriber Identity Module(s). A device allowing a user access to
15 network services via the Uu interface.
- 16 **Universal Terrestrial Radio Access Network:** UTRAN is a conceptual
17 term identifying that part of the network which consists of RNCs and
18 Node Bs between lu an Uu.
- 19 **UTRAN Access Point:** conceptual point within the UTRAN performing
20 radio transmission and reception
21 A UTRAN access point is associated with one specific *cell*, i.e. there
22 exists one UTRAN access point for each cell. It is the UTRAN-side end
23 point of a *radio link*.
- 24 **Radio Link:** "radio link" is a logical association between a single User
25 Equipment and a single UTRAN access point
26 Its physical realisation comprises one or more radio bearer
27 transmissions.
- 28 **Radio Link Set:** set of one or more Radio Links that has a common
29 generation of Transmit Power Control (TPC) commands in the DL
- 30 **Uu:** Radio interface between UTRAN and the User Equipment
31

1 References

- 2 [3GSwM] 3G TS 32.600, Version 5.0.1
3 3rd Generation Partnership Project;
4 Technical Specification Group Services and System
5 Aspects;
6 Telecommunication Management; Configuration
7 Management (SwM); Concept and high-level requirements
8 (Release 5)
9 Internet: <http://www.3gpp.org/>
- 10 [3GOD] 3G TS 25.401, Version 6.1.0
11 3rd Generation Partnership Project;
12 Technical Specification Group Radio Access Network;
13 UTRAN Overall Description
14 (Release 6)
15 Internet: <http://www.3gpp.org/>
- 16 [NBAP] 3G TS 25.433, Version 5.6.0
17 3rd Generation Partnership Project;
18 Technical Specification Group Radio Access Network;
19 UTRAN Iub interface NBAP Signaling
20 (Release 5)
21 Internet: <http://www.3gpp.org/>
- 22 [ITU-T X.731] ITU-T Recommendation X.731: "Information Technology –
23 Open Systems Interconnection – Systems Management:
24 State management function
25 Internet: <http://www.3gpp.org/>
- 26 [OBSAI_Test] Open Base Station Architecture Initiative:
27 RP1 Test Messages,
28 Internet: www.obsai.org
- 29 [SOAP] Simple Object Access Protocol (SOAP)
30 Version: 1.1
31 <http://www.w3c.org/>
- 32 [SOAPUS] SOAP Version 1.2 Usage Scenarios
33 <http://www.w3c.org/>

- 1 [XMLSCHE] XML Schema Specification of World Wide Web Consortium
- 2 XML Schema Part 1: Structures
- 3 W3C Working Draft 24 September 1999
- 4 XML Schema Part 2: Datatypes
- 5 W3C Working Draft 24 September 1999
- 6 <http://www.w3c.org/>
- 7