OBSAI Interoperability in Multi-Vendor WiMAX Base Station Architecture Environment

SUMANTA SAHA

Master of Science Thesis
Stockholm, Sweden 2009

TRITA-ICT-EX-2009:62

HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information and Natural Sciences
Department of Computer Science and Engineering

Sumanta Saha

OBSAI Interoperability in Multi-Vendor WiMAX Base Station Architecture Environment

Master's Thesis
Espoo, June 19, 2009

Supervisors: Professor Antti Ylä-Jääski,
              Helsinki University of Technology (TKK), Finland
              Professor Gerald Q Maguire Jr.,
              Royal Institute of Technology (KTH), Sweden
Instructor: Petri Mikkonen, Nokia Siemens Networks, Finland

HELSINKI UNIVERSITY OF
TECHNOLOGY
Faculty of Information and Natural Sciences
Degree Programme of Security and Mobile Computing

ABSTRACT OF
MASTER'S THESIS

Author:	Sumanta Saha
Title of thesis:	
OBSAI Interoperability in Multi-Vendor WiMAX Base Station Architecture	

Date:	June 19, 2009		**Pages:** 14 + 95
Professorship:	Data Communications Software		**Code:** T-110
Supervisors:	Professor Antti Ylä-Jääski		
	Professor Gerald Q Maguire Jr.		
Instructor:	Petri Mikkonen		

Wireless networks have become a necessity with the increased mobility in human life. From cellular telephony to the Internet, all types of communication are now provided over wireless networks. However, to offer wireless network coverage over an area requires a potentially expensive infrastructure deployment. Such deployment requires base stations which until now have been completely proprietary to the equipment vendors. Moreover, proprietary equipment is almost always costly and offer less flexibility than standardized modular solutions. This situation results in a high cost for network up-gradation and hinders network development. A remedy is available via modularization, hence the Open Base Station Architecture Initiative (OBSAI) is trying to modularize and standardize one of the most expensive elements of the wireless infrastructure, the base station. OBSAI standards aim to modularize the base station architecture and enable true interoperability among the various modules. However, the goal has not yet been achieved due to some features of the standard. This thesis project has studied the standards and pointed out some areas that must be concentrated upon when performing interoperability tests. It also proposes several standards amendments to foster greater interoperability among the modules of a base station. This study focuses on the RP3 interface of the OBSAI specification with the goal of making truly inter-operable baseband and RF modules, thus commoditizing the modules. The result is expected to be lower cost, greater interoperability, faster time-to-market, and more cooperative research.

Keywords:	OBSAI, RP3, RP3-01, WiMAX, LTE, Base Station
Language:	English

Tekijä:	Sumanta Saha
Diplomityön Otsikko:	
OBSAI Interoperability in Multi-Vendor WiMAX Base Station Architecture	

Langattomat laajakaistaverkot ovat tulleet välttämättömäksi osaksi liikkuvien ihmisten elämää. Lähes kaikki kommunikaatiotarpeet äänipuheluista internettiin pystytään toteuttamaan langattomien verkkojen avulla. Kuitenkin jotta langattomilla verkoilla pystytään tarjoamaan täysi peittävyys yli maan, se vaatii varsin kalliita investointeja verkkoinfrastruktuuriin. Langattomien verkkojen investoinnit koostuvat suurelta osin tukiasemista, jotka tähän asti ovat olleet kullakin verkkotoimittajalla täysin omanlaisensa toteutus. Kun jokainen verkkotoimittaja toteuttaa kaikki tukiaseman osat erilailla, se tarkoittaa että kutakin tukiaseman osia valmistetaan suhteellisesti pienempiä määriä ja sitä myötä niistä tulee mahdollisesti kalliimpia verrattuna standardoituhin modulaarisiin tukiasemaratkaisuihin. Nykyinen tilanne siis osaltaan johtaa siihen että verkkojen rakentaminen ja päivittäminen on kallista. Eräs ratkaisu tähän ongelmaan on tarjolla modulaarisessa tukiasemaratkaisussa ja siksi OBSAI, Open Base Station Initiative, pyrkii modulaarisoimaan ja standardoimaan yhden kalliimmista verkkoinfrastruktuurin osista, tukiaseman. OBSAI standardi pyrkii modularisoimaan tukiasema-arkkitehtuurin ja mahdollistamaan todellisen yhteensopivuuden tukiaseman eri osien välillä. Tätä todellista yhteensopivuutta ei ole vielä täysin pystytty toteuttamaan, johtuen tietyistä standardin epätarkkuuksista. Tässä lopputyössä on analysoitu OBSAI standardia ja identifioitu alueet, joihin pitää keskittyä, kun modulien välistä yhteensopivuutta testataan. Työn lopputulemana myös ehdotetaan useita parannuksia ja muutoksia standardiin, jotta todellinen yhteensopivuus modulien välillä saavutetaan. Painopiste lopputyössä on OBSAI standardin RP3 rajapinta, joka määrittelee kantataajuusosan (BB) ja radiotaajuusosan (RF) välisen rajapinnan. Kun OBSAI standardia saadaan parannettua työssä ehdotetuin toimenpitein, lopputuloksena on oletettavasti alhaisempi tukiaseman kokonaiskustannus, mahdollisuus käyttää yhteensopivia moduleita eri valmistajilta, nopeampi tuotteiden markkinoille vienti sekä parantunut tutkimusyhteistyö eri yritysten välillä.

Kieli:	Englanti

Författare: Sumanta Saha
Titeln på Avhandlingen:
OBSAI Interoperability in Multi-Vendor WiMAX Base Station Architecture

Trådlösa nät har blivit en nödvändighet i vår allt mer mobila livsstil. Från mobiltelefoni till Internet, trådlösa nät erbjuder många typer av kommunikation. Men att erbjuda trådlös täckning i ett område kan kräva installation av en mängd dyrbar telekomutrustning. En sådan utbyggnad kräver basstationer som fram till nu har varit patentskyddade av respektive leverantör. Och patentskyddad utrustning är oftast både dyrare och mindre flexibel jämfört med standardiserade modulära lösningar. Resultatet är höga kostnader för att uppgradera näten och att utvecklingen försvåras. Ett botemedel är användningen av standardiserade moduler. Därför försöker Open Base Station Architecture Initiative (OBSAI) att standardisera moduler i ett av de dyraste nätelementen i trådlösa nät, basstationen. OBSAI har som mål att dela upp basstationen i definierade moduler och möjliggöra fullständig interaktion mellan olika moduler. Men på grund av vissa egenskaper hos standarden har detta inte lyckats. Denna studie har undersökt standarden och pekar på områden som man måste fokusera på när man utför tester mellan moduler. Dessutom föreslås flera tillägg till standarden för att möjliggöra bättre interaktion mellan basstationens moduler. Studien fokuserar på RP3-gränssnittet med målet att möjliggöra standardiserad interaktion mellan basbands- och radio-moduler, så att dessa moduler kan kommerisialiseras. Det förväntade resultatet är lägre kostnader, bättre interaktion mellan moduler, snabbare marknadsintroduktion och mer samarbete inom forskning och utveckling.

Språk: Engelska

Acknowledgments

I would like to show my gratitude to all of those who supported my work: Professor Antti Ylä-Jääski of Helsinki University of Technology and Professor Gerald Q Maguire Jr. of Royal Institute of Technology for their excellent supervision and valuable feedback. Sanna Suoranta of TML for the LaTeX template that I have used for this thesis.

I would also like to thank my friends in Otaniemi, who kept me alive during this solitary period. And of course my family, who inspired me in all the phases of my strive to become a great human being.

Finally, my gratitude to Petri Mikkonen and Asko Räsänen of Nokia Siemens Networks for providing me with the opportunity to work in this exciting area and for their continuous support. Many thanks to my other friends and colleagues at the same company, including Mika Nortes; Tommi Huhtala; Maarit Makkonen; Sami Makinen; Seppo Porspakka; Timo Viero; Kai Lummi; Bo Axerud; and many others, without their support this work would not have finished. Further gratitude to Anders Bak, Christian Lanzani, and Søren Asmussen for their valuable input and cooperation.

Espoo, June 19th, 2009

Sumanta Saha

Contents

List of Tables

List of Figures

Abbreviations and Acronyms

3GPP   3rd Generation Partnership Project

AMC   Adaptive Modulation and Coding
ARQ   Automatic Repeat Request
ASN   Access Service Network

BB   Baseband Block
BBU   Baseband Unit
BE   Best-effort Service
BER   Bit Error Rate
BPSK   Binary Phase Shift Keying
BS   Base Station
BTS   Base Transceiver System
BWA   Broadband Wireless Access

CCB   Control and Clock Block
CN   Core Network
CP   Cyclic Prefix
CPRI   Common Public Radio Interface
CSN   Connectivity Service Network

DFT   Discrete Fourier Transform

eNodeB   Evolved NodeB
EPC   Evolved Packet Core
ertPS   Extended Real-time Polling Service

FDD   Frequency Division Duplex
FEC   Forward Error Correction
FFT   Fast Fourier Transform

GPB   General Purpose Block

HSPA   High Speed Packet Access

IEEE	Institute of Electrical and Electronics Engineers
IOT	InterOperability Testing
ISI	Inter-Symbol Inference
LCV	Line Code Violation
LOS	Line of Sight
LTE	Long Term Evolution
MIB	Management Information Base
MME	Mobility Management Entity
nrtPS	Non-real-time Polling Service
NSP	Network Service Provider
OAM&P	Operation, Administration, Maintenance and Provisioning
OBSAI	Open Base Station Architecture Initiative
OEM	Original Equipment Manufacturer
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OID	Object Identifier
PHY	Physical Layer
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RAB	Radio Access Bearer
RAN	Radio Access Network
REST	Representational State Transfer
RFB	Radio Frequency Block
RP	Reference Point
RRH	Remote Radio Head
RRU	Remote Radio Unit
rtPS	Real-time Polling Service
SAE	System Architecture Evolution
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SS	Subscriber Station
TB	Transport Block
TDD	Time Division Duplex
TDM	Time Division Multiplex

UDPCP	UDP based Communication Protocol
UE	User Equipment
UGS	Unsolicited Grant Service
UL/DL	Uplink/Downlink
UPE	User Plane Entity
WCDMA	Wideband Code Division Multiple Access

Chapter 1

Introduction

1.1 Motivation

With the increase of mobility and dynamics in the current world, Broadband Wireless Access (BWA) is becoming the choice for last-mile access when connecting to the Internet [1]. Simultaneously, the introduction of smart phones with powerful processors brought a whole new era of rich client side computing to mobile phones. Responding to the increasing need for bandwidth and capacity, standards bodies have standardized various wireless access protocols such as IEEE 802.16 (WiMAX) [2], 3GPP Long Term Evolution (LTE) [3], WCDMA [4], and many others. Continuous research efforts are ongoing to develop these standards further; in order to improve the capacity, bandwidth, and quality of service. However, actually deploying wireless mobile networks is a gigantic effort, requiring large amounts of capital, expertise, and addressing many regulatory issues. These factors combine such that it requires a large amount of research and development work from the equipment manufacturers and a large amount of capital from the mobile operators to upgrade their infrastructure.

It is a common practice that standardization normally specifies the protocols and interfaces needed to communicate among the modules and nodes of a network; however, sometimes these standards are intentionally abstract to enable vendors to offer *their own* flexible solutions. As a result, different implementations of a particular standard are not necessarily inter-operable. This is currently the case for WiMAX. Today one of the most distributed and costly elements of the infrastructure for a WiMAX network is the base station (BS). BSs need to be deployed over the whole coverage area of an operator. If an update to the standard occurs and this needs to be applied to

BSs that are already out in the field, then either all of these BSs need to be changed and redeployed, or the BSs need to be remotely updated. The first of these alternatives requires investing a huge amount of time and money for the operators, as the BS architecture is normally proprietary to a specific equipment vendor, thus only that vendor can make the necessary changes to the already deployed BSs. At the same time in order to meet the constant pressure from their customers (i.e., the network operators), the equipment vendors need to develop and release new technologies as early as possible. However, developing and properly testing a complex system such as a BS requires a lot of time and resources[1]. This often results in increased time to market, a compromise in quality, or both.

One possible way to reduce the time and cost of developing and testing base stations is to modularize the BS system based upon standardized interfaces among a standardized set of modules [6]. The approach will ensure interoperability among the modules, thus promoting diversity in vendor equipment selection. The possibility of using multi-vendor modules in the same BS system will allow small companies to concentrate on a specific part of the BS and thus gain expertise with it. Larger Original Equipment Manufacturer (OEM) companies will gain flexibility in choosing different parts of the BS from different suppliers, while providing a complete solution to their customers in a much shorter time frame. This approach will also allow the research and development effort to be distributed among the vendors, thus reducing both the development cost and time-to-market. Based on these ideas, several companies started an initiative called the Open Base Station Architecture Initiative (OBSAI) [7], with the goal of modularizing a BS based upon establishing standardized interfaces among these modules. The initiative started its work in 2004 and has already released specifications for an OBSAI compatible BS architecture. Among the modules of OBSAI, the baseband subsystem (BB) and Radio Frequency subsystem (RF) are the most costly and technically sophisticated parts. The interface that connects these two modules is called Reference Point 3 (RP3) [8, 9]. This approach has already fostered several companies including Runcom (`http://www.runcom.com`), Axis (`http://www.axisnt.com`), Radiocomp (`http://www.radiocomp.com`), and Alvarion (`http://www.alvarion.com`); specializing in specific modules of the system.

More often than not, OEM companies are expected by the operators to provide RF modules with different configurations, perhaps differing in size,

---

[1]Regarding testing of WCDMA cellular base stations see [5].

mounting point, operating frequency band, maximum output power, etc. It would be much more convenient for the OEM vendor, if they could simply outsource the production of RF modules from specialized RF companies rather than producing all the variations of RF modules by themselves. However, to make this a reality, the OBSAI RP3 interfaces of both the baseband and RF modules need to work correctly. Currently, the OBSAI standards have too much flexibility built into them, thus even these standard interfaces may differ from implementation to implementation, leading to the end result of no interoperability. Two approaches can be taken to prevent the lack of interoperability between the RF and BB modules. Either the flexibility of the OBSAI standards must be removed, making it sufficiently rigid as to ensure that all the modules will be inter-operable, or to generate profiles of parameters which correctly indicate inter-operable implementations. The profiling feature can be further utilized by implementing soft interfaces that can adapt themselves to the interface of the other module–hence the effort to implement the interfaces in FPGAs.

1.2   Research Challenges

Modularity in telecommunication infrastructure, interoperability among the modules, and outsourcing of BS components are relatively recent subjects of discussion and sometimes are opposed and challenged by many. However, proprietary RF and BB implementations along with the high cost of telecommunication infrastructure has prevented academic researchers from research in this area. As a result, there are few sources of information regarding specific vendors RF and baseband modules. Most of the information that does exist is only available under a non-disclosure agreement.

The RP3 [8] specification standardized by OBSAI specifies the protocols for communicating between the BB and the RF modules. These protocols mainly address the physical, data link, network, transport, and application layer of the OSI protocol stack. Moreover, there exists a sub-protocol named RP3-01, which is specified along with the RP3 specification. RP3-01 is used for transporting RP1 messages over RP3 frames for Remote Radio Heads (RRH), i.e. allowing the radio module to be remote from the BB module. For more discussion about the specifications and the components of a BS please refer to sections 4.3.2 and 4.3.3. The mapping of the logical OBSAI base station architecture to hardware is such that the RP3/RP3-01 interface

is the only external interface while others are merely internal interfaces of one hardware module. This situation makes the RP3 interface, along with the RP3-01 interface, the most important interface for any interoperability test. This situation is described further with necessary figures in Chapter 4.

Different vendors will use different implementations of the RP3 interface in their products. Whether the RP3 interface of vendor X will be inter-operable with vendor Y depends upon their choice of parameters when implementing the interface. Without a clear definition of the flexible parts of the standard, it will require rigorous testing from the integrating vendor's side to ensure interoperability. An effort to ensure interoperability should study:

- Possible flexible areas of the standards document(s)

- Possible parameterized features of the standards

- Unspecified areas left to the implementer

- Integration test results of different vendors' equipment

- Test results of changing the parameters of a given interface

However, the unavailability of detailed implementation documentation from different vendors, scarcity of proper testing tools, and the lack of many original scientific documents researching the standards have made this effort more challenging than it appears.

Although proposing improvements to the standards to ensure better interoperability is within the scope of this thesis, overall improvement of the standard is not. Features of the standards which do not hinder interoperability between different implementations are not studied.

1.3 Contributions

The primary contribution of this work is to suggest steps to be taken to ensure truly inter-operable base station modules according to the OBSAI architecture. In the course of doing so, this work has identified potential areas of the OBSAI standards document that foster non-interoperability. Additionally, through practical experiments this thesis project demonstrated various situations that can arise while performing an integration effort between different vendors' modules. To complete this research, this thesis proposes various amendments and improvements to the standards document for ensuring better interoperability and performance.

1.4   Thesis Outline

The thesis is logically structured to provide the reader with suitable background knowledge before diving deep into the details of the standards. After introducing the work in Chapter 1, an overview of broadband wireless technologies is presented in Chapter 2. To provide a better understanding of the function of base stations in a network, a brief overview of modern wireless network infrastructures is provided in Chapter 3. This overview is followed, in Chapter 4, by an introduction to the architecture of an OBSAI base station according to the OBSAI standards. This concludes the background work. The original work is presented in Chapters 5 and 6, where an analysis of interoperability is presented, and practical integration tools and experiments are described. This is followed by a set of proposals to improve the standards in Chapter 7. The thesis ends with a very short overview of competing base station architectures in Chapter 8, and a suggestion of possible future work in Chapter 9. Appendices A and B provides some additional material concerning the test setup and test output, as well as a proposal for a SNMP based management plane.

Chapter 2

Broadband Wireless Technology

Broadband wireless access has enjoyed one of the most successful growth stories in the history of the telecommunication industry. While wireless mobile services grew from 11 million subscribers worldwide in 1990 to more than 4 billion in 2008 [10], the Internet grew from mainly an academic tool for university researchers to an everyday commodity for more than a billion users. This high growth of the Internet is leading to a parallel growth in the demand for high data rate broadband access. Users of the Internet represent around 21.9% of the population as of June 30, 2008. This is a staggering 305.5% growth from the year 2000 [11].

To cater to the growing need for broadband access, as well as for mobility, broadband wireless technology was developed. This technology uses high frequency wireless signals in a clever way to deliver data rates comparable to wired access networks. In this chapter, WiMAX (based upon IEEE 802.16) is discussed as a representative of wireless broadband access technologies.

2.1 WiMAX (IEEE 802.16) Overview

The IEEE 802.16 working group was formed to develop an air-interface standard for wireless broadband in 1998 and released the first 802.16 standard in 2001. The released standard was initially limited to Line Of Sight (LOS) based operation and operated in the 10GHz-66GHz millimeter wave band. It had a single-carrier based physical (PHY) layer with a burst time division multiplexed (TDM) MAC layer. From this initial standard,

the 802.16 standard has evolved over time resulting in two versions named IEEE802.16-2004 [12] and IEEE802.16e-2005 [13]. These standards are also called Fixed-WiMAX and Mobile-WiMAX respectively. A few of the features of these evolved versions of IEEE 802.16 (referred to jointly as WiMAX) are:

OFDM-based physical layer: The physical layer (PHY) of WiMAX is based on Orthogonal Frequency Division Multiplexing (OFDM) [14]. For a more detailed overview of OFDM please refer to Section 2.2.

High peak data rates: The peak PHY data rate of WiMAX can be 74 Mbps when operating in a 20 MHz wide band. This data rate includes both uplink/downlink throughput. However more typically, these systems operate using a 10 MHz band operating using a Time Division Duplexing (TDD) [15] scheme with a 3:1 downlink to uplink ratio, resulting in a peak PHY data rate of about 25 Mbps for the downlink and and 6.7 Mbps for the uplink.

Scalable bandwidth: WiMAX supports a scalable bandwidth which is achieved using dynamic allocation of subchannels based upon the use of a Fast Fourier Transform (FFT) over the available total channel bandwidth.

Adaptive modulation and coding: Adaptive modulation and coding [16] allows WiMAX to dynamically choose the modulation and forward error correction (FEC) coding scheme based on the available channel quality and bandwidth.

Support for TDD and FDD: Both WiMAX specifications support the use of Time Division Duplexing (TDD) [15] and Frequency Division Duplexing (FDD) [17, 18].

IP-based architecture: The WiMAX standard is based on an all-IP architecture.

OFDMA: WiMAX can utilize an orthogonal frequency division multiple access (OFDMA) technique [19], whereby different users can be allocated different subsets of the OFDM codes.

Link layer retransmission: WiMAX allows link layer retransmission (also known as automatic repeat request (ARQ) [20]), where lost frames are retransmitted by the link layer rather than waiting for the network layer to make a retransmission decision.

2.2 WiMAX Physical Layer

Technologies used in WiMAX are not novel; rather, WiMAX represents a unique combination of technologies selected to provide maximum throughput at a maximum distance with maximum reliability. For this purpose, WiMAX uses all the proven technologies for PHY such as: OFDMA, Adaptive Modulation and Coding (AMC), TDD, FDD, Quadrature Phase Shift Keying (QPSK), and Quadrature Amplitude Modulation (QAM). This section provides a brief overview of the prominent features of WiMAX's PHY.

2.2.1 OFDM Basics

OFDM is a family of transmission schemes called multi-carrier modulation, which are capable of supporting high speed services whilst still being bandwidth efficient. This transmission scheme allows lower inter symbol inference (ISI) [21] by making the symbol long enough to avoid delay spread [22]. OFDM mandates that the sub-carriers be orthogonal to each other, thus eliminating the need to have non-overlapping sub-carrier channels to avoid inter-carrier interference.

OFDM enjoys several advantages over other techniques for high-speed data transmission. Some of the more important features of OFDM are:

- OFDM is extremely easy to implement in hardware using FFT and IFFT, and its computational complexity can be shown to be $O(B \log BT_m)$, where B is the bandwidth and T_m is the delay spread.

- OFDM allows adaptive coding and modulation.

- OFDM facilitates the exploitation of frequency diversity while supporting interleaving across sub-carriers in the frequency domain.

2.2.1.1 Fast Fourier Transform

In OFDM systems, the transmitting radio can be considered as a combination of hundreds of radios each tuned in parallel to a separate orthogonal frequency or sub-carrier. However, the use of hundreds of physically separate transmitters and receivers is not a realistic approach. Rather, it is desirable to use modern digital signal processing techniques, such as the Fast Fourier Transform(FFT) to achieve the same effect using a

wide band receiver. As discussed earlier, OFDM signals consist of a large number of narrowband signals closely spaced in the frequency domain. Mathematically, each carrier can be described using the following formula:

$$S_c(t) = A_c(t)e^{j[\omega_c t + \Phi_c(t)]} \tag{2.1}$$

The real part of $S_c(t)$ is the real signal and, both amplitude(A_c) and phase(Φ_c) of the signal can change on a symbol by symbol basis. Since the OFDM signal consists of many carriers, the complex signal becomes:

$$S_s(t) = \frac{1}{N} \sum_{n=0}^{N-1} A_n(t)e^{j[\omega_n t + \Phi_n(t)]} \tag{2.2}$$

where $\omega_n = \omega_0 + n\Delta\omega$. This is a continuous signal and if we sample the frequency using a sampling frequency of $\frac{1}{T}$, then the equation becomes:

$$S_s(kT) = \frac{1}{N} \sum_{n=0}^{N-1} A_n e^{j[(\omega_0 + n\Delta\omega)kT + \Phi_n]} \tag{2.3}$$

At this point, the time required to analyze the signal has been restricted to N symbols. It would be convenient to analyze the signal during one symbol time, thus obtaining the relation: $t = NT$. If the equation 2.3 is simplified by letting $\omega_0 = 0$, then without loss of generality, it becomes:

$$S_s(kT) = \frac{1}{N} \sum_{n=0}^{N-1} A_n e^{j\phi_n} e^{jn\Delta\omega)kT} \tag{2.4}$$

Equation 2.4 can be compared to the general form of an inverse Fourier transform:

$$g(kT) = \frac{1}{N} \sum_{n=0}^{N-1} G(\frac{n}{NT})e^{\frac{j2\pi nk}{N}} \tag{2.5}$$

Equation 2.4 and 2.5 are same if:

$$\Delta f = \frac{\delta\omega}{2\pi} = \frac{1}{NT} = \frac{1}{\tau} \tag{2.6}$$

This condition is the requirement for orthogonality and thus, a consequence of maintaining orthogonality is that OFDM signals can be defined by using Fourier transform procedures[1].

To avoid aliasing and storage problem the signal needs to be time limited. This is done by using Discrete Fourier Transform (DFT) [23, 24]. Details about using DFT for sampling signals can be found in [25]. FFT is an algorithm for the computation of DFT which allows the implementation of the procedure in an integrated circuit.

---

[1]http://wireless.per.nl/reference/chaptr05/ofdm/ofdmmath.htm

2.2.1.2 Cyclic Prefix

In OFDM-based systems, a cyclic prefix is used to counter the multipath problem by preventing ISI using circular convolution. A cyclic prefix can be defined as a guard period which is inserted just before the user data and is copied from the very last portion of the period. For a detailed mathematical explanation of the working of cyclic prefix, please refer to [26].

2.2.1.3 Sub-Channelization

Sub-Channelization refers to a method where the available subcarriers are divided into several groups and a group is assigned to a subscriber station (SS). Uplink channelization allows a WiMAX SS to transmit only for a fraction of the bandwidth allocated to it by the base station, thus providing a link budget can be used to enhance range or performance.

2.2.2 Orthogonal Frequency Division Multiple Access

Orthogonal frequency division multiple access (OFDMA) can be seen as a high level subchannelization. Whereas OFDM sub-channelization is semi-static due to the allocation of a fixed number of subcarriers to each SS, OFDMA is more dynamic. OFDMA will allocate a number of sub-carriers for different symbol periods based upon network conditions and user requirements. OFDMA allows dynamic allocation of channels, thus minimizes bit error rate (BER) and can better utilize the channel's current condition.

Adaptive modulation and coding (AMC) is used to compensate for the instability of radio channels. AMC improves the Bit Error Rate (BER) of the channels suffering from shadowing and fading [27, 28, 29]. Selected AMC schemes for IEEE 802.16 are binary phase shift keying (BPSK), Quadrature PSK (QPSK), 16-Quarter Amplitude Modulation (QAM), and 64-QAM.

2.3 WiMAX MAC Layer

The MAC layer in WiMAX provides an interface between the PHY layer and the higher transport layers. The IEEE 802.16 MAC design includes a convergence sublayer that can interface with a variety of higher-layer

protocols, such as ATM, IP, TDM Voice, and Ethernet & other IEEE 802.2 compatible protocols.

2.3.1 Quality of Service

The WiMAX MAC layer is a fundamental part of its QoS design. It provides QoS by supporting connection-based access rather than contention based access. WiMAX defines a service flow, which is a unidirectional flow of packets associated with a particular set of QoS parameters over a path. Service flows can be mapped to DiffServ code points or MPLS flow labels to enable end-to-end IP-based QoS. In order to standardize the level of QoS, WiMAX has defined five MAC layer scheduling services: Unsolicited Grant Service (UGS), Real-time Polling Service (rtPS), Non-real-time Polling Service (nrtPS), Best-effort Service (BE), and Extended Real-time Polling Service (ertPS).

2.3.2 Channel-Access Mechanisms

The MAC layer at the base station (BS) is responsible for allocating bandwidth to each of the currently served SSs. On the downlink, the bandwidth is allocated by the BS based on the amount of incoming traffic, and on the uplink the bandwidth is allocated based on the requirements stated by the SS (subject to limits imposed by the BS).

2.3.3 Power-Saving Mechanisms

To conserve power for a mobile SS, the MAC layer utilizes several different power-saving features. Power-saving is achieved by effectively turn off portions of the transceiver of the SS while it is not in active use. WiMAX defines signaling methods to allow the SS to transition to SLEEP or IDLE mode when inactive.

2.4 Discussion

This chapter provided some insight of the ideas and techniques used in a WiMAX network to illustrate a typical broadband wireless access network. This information is necessary to appreciate the mechanisms of different modules in a base station.

Chapter 3

Telecommunication Network Infrastructure

It is conceivable that to perform the variety of tasks stated in chapter 2, a number of infrastructure nodes will be necessary. Additionally, due to the fading and multipath effect wireless signals are attenuated with range even faster then $1/r^2$. This limitation in range requires the installation of wireless signal transceiver within some bounded distance of all the places where an operator wishes to provide coverage. Along with the transceivers, several processing nodes for post and pre-processing of the signal also need to be installed. To organize this complex architecture and the many interactions among nodes standardized, the standards body for each broadband technology publish a reference architecture for their network. This chapter reviews two of the contemporary wireless technology architectures: WiMAX and LTE. Further discussion of other standardization efforts such as OBSAI and CPRI to make the architecture even more modular are also discussed.

3.1 WiMAX Architecture

A WiMAX network is based on end-to-end IP connectivity and makes extensive use of IETF's protocols. The clean-slate design of WiMAX resulted in a very simple architecture with very few elements.

A logical representation of the network reference model (NRM) of WiMAX is illustrated in Figure 3.1. The NRM identifies the functional entities and the reference points between the functional entities over which

interoperability is to be achieved. The architecture divides the system into three main logical parts: (1) Mobile stations (MSs); (2) Access Service Network (ASN), which comprises of several base statiosn and ASN gateways; and (3) the Connectivity Service Network (CSN), which provides IP connectivity and IP core network functionalities. The home network service provider (NSP) is where the subscriber belongs and the visited NSP is where the user is being served.
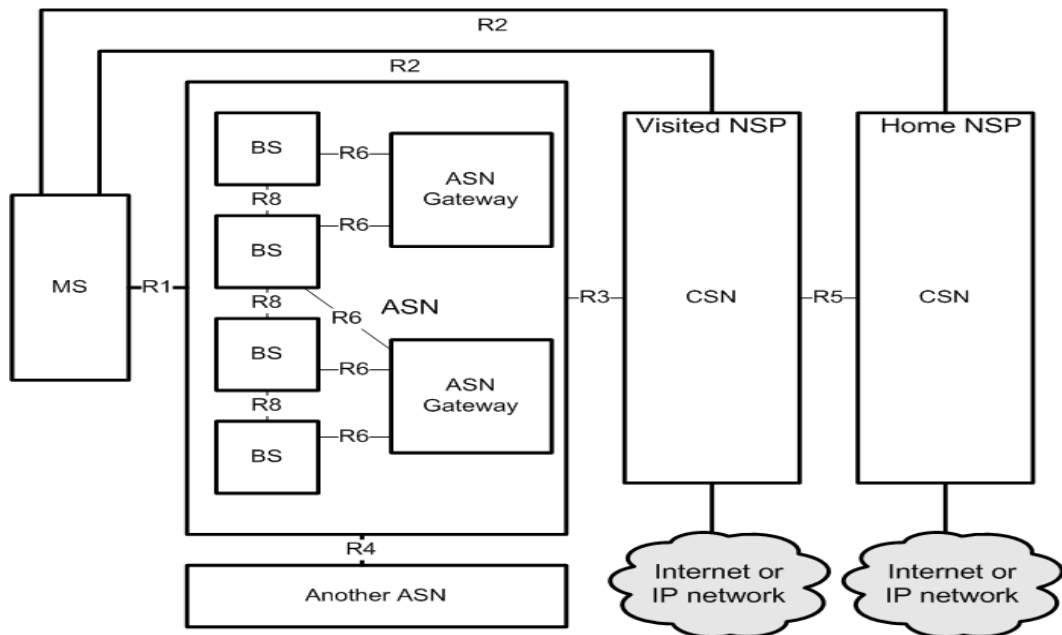
Figure 3.1: WiMAX Network Reference Model

The ASN can be decomposed into one or more base stations (BS) and one or more ASN gateways (ASN-GW). Each BS represents one sector with one frequency assignment implementing the IEEE802.16e interface to the MS. The BS also handles additional functions such as scheduling, traffic classification, service flow management, providing DHCP functionality, supporting a tunneling protocol, relaying authentication message, and serving as an RSVP proxy. These functions of BS need to be replicated in all the BSs in a deployment because each of these have BS specific features and configurations. On the other hand, the ASN-GW provides authorization, authentication & accounting (AAA) functionalities; location management and paging; admission control; mobility tunnel creation; and routing. Compared to the ASN, the CSN has far less replication as one set of CSN elements can serve multiple ASNs. The functionality of a typical CSN includes IP address range allocation for each BS, AAA proxy, QoS

management, billing and policy management, inter-CSN tunneling, and access to other services connected to the IP network. A logical representation of WiMAX's network architecture as shown in Figure 3.2 is useful to visualize various usage scenarios. In this figure "ASP" stands for Application Service Provider. Such a provider might provide value added services to users via their MS.
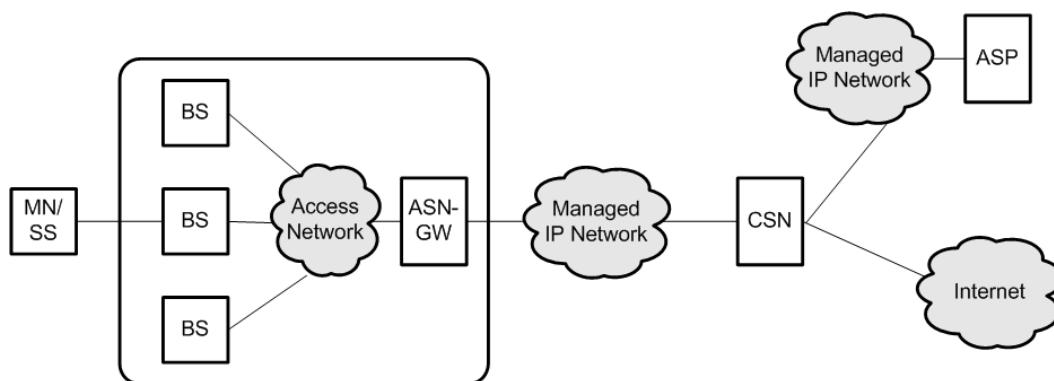
Figure 3.2: Logical representation of end-to-end WiMAX Network

3.2 3GPP LTE Architecture

While WiMAX body started with a clean slate to build the standard from the scratch, 3GPP's Long Term Evolution (LTE) initiative was conceived as a evolution of existing GSM and WCDMA telecommunication standards. The 3GPP standards body started two parallel efforts: (1) To develop High Speed Packet Access (HSPA) with backward compatibility to allow a smooth migration by the operators and (2) LTE with less stress on backwards compatibility because by the time when LTE would be available, operators were already expected to have a HSPA network. The desire for compatibility with the existing standards and installed networks resulted in a more complex network architecture with more elements. However, it allowed the operators to gradually deploy upgrades and lead to capital cost-saving. While the radio access part of the network is called LTE, the flat IP based core network architecture is called the System Architecture Evolution (SAE).

A simplified diagram of the LTE/SAE network architecture is depicted in Figure 3.3. The architecture is divided into two main parts: a Radio Access Network (RAN) and a Core Network (CN). The same two parts exist in 2.5G

GSM and 3G WCDMA networks. The RAN consists of an evolved-NodeB (eNodeB) which is the Base Station. This eNodeB is connected with the core network using the S1 interface and any two eNodeBs can be connected to each other using the X2 interface. Each eNodeB is in charge of its own radio resource allocation, handover, scheduling decisions, and the classical physical layer functionalities.
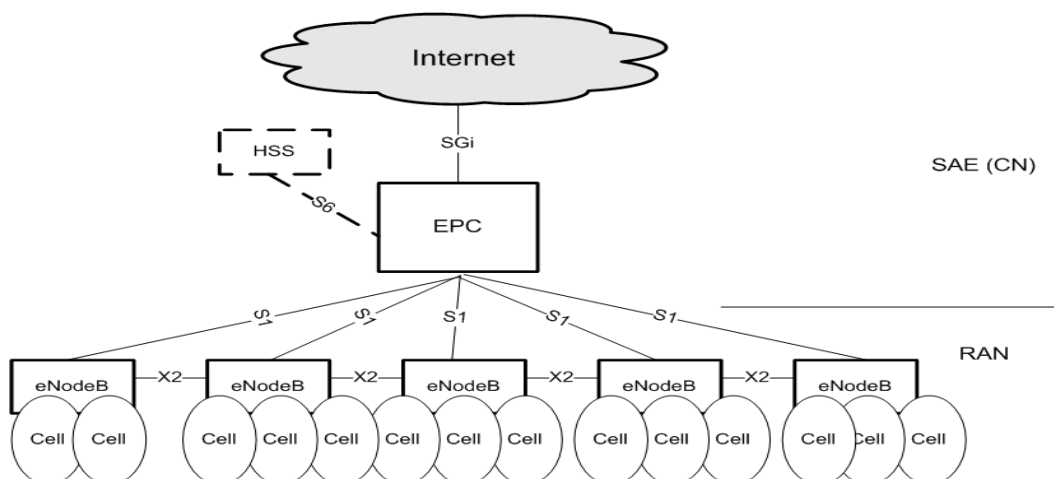
Figure 3.3: Simplified diagram of LTE/SAE network architecture

The CN architecture or SAE was motivated by a decision to make a simple design and to reduce the number of different types of nodes. This decision resulted in a much simpler architecture for SAE's CN. While HSPA CN had many elements such as Serving GPRS Support Node, Gateway GPRS Support Node, Policy and Charging Rules Function, and Mobility Management Entity; SAE has only two elements: Evolved Packet Core (EPC), and Home Subscriber Server (HSS). The simplified EPC is a radical change from the previous 3GPP architectures and it is a completely packet-switched domain. The EPC consists of the following types of nodes: Mobility Management Entity (MME), User Plane Entity (UPE), 3GPP Anchor, and SAE Anchor. Core network functionalities such as routing, roaming, and charging are taken care of by EPC, while HSS acts as a subscriber base and provides AAA services; it corresponds to a home location register in a GSM core network.

3.3 BS Standardization Efforts

The previous sections of this chapter showed that the RAN is composed of a large number of elements in a typical network deployment. This occurs because several BSs can be attached to a single CN and each eNodeB needs to be connected to the CN, thus the overall cost of a RAN exceeds the cost of the CN. One possible means of reducing this cost is to modularize the BS by standardizing its elements and interfaces. As noted earlier, today BSs are completely proprietary modules & systems, as are the protocols used for interfacing these modules. However, recently several initiatives to standardize BSs have been started by OEM companies. This has resulted in two well known standards: (1) OBSAI and (2) CPRI. Each of them will be briefly introduced below. A detailed treatment of OBSAI is provided in Chapter 4 and CPRI in Chapter 8.

3.3.1 Open Base Station Initiative Architecture

Open Base Station Initiative Architecture (OBSAI) was started by equipment vendors, this included Nokia, NEC, RadioComp, ZTE, Texas Instruments, Alcatel, and many others. It defines the elements of the BS covering the areas of transport, clock, radio, and baseband, together with a hardware interconnection specification. OBSAI has defined a complete protocol stack ranging from the application to the physical layer of the OSI model. OBSAI allows a fully standardized BS architecture that can leverage interoperability and modularity.

3.3.2 Common Public Radio Interface

Common Public Radio Interface (CPRI), another standardization effort for modular BSs, has a much narrower scope than OBSAI. CPRI's target is to specify the key internal interface between the Radio Equipment Control and the Radio Equipment. Companies involved in the CPRI specification are Ericsson, Huawei, NEC, Nortel, Nokia Siemens Networks, and Alcatel-Lucent. CPRI identifies the radio equipment as the most diverse element of a BS and standardizes the interface to the radio equipment, thus allowing companies to use any inter-operable radio equipment with a CPRI interface. CPRI defines only the PHY and MAC layers, leaving the upper layers to be implementation specific. Thus CPRI is a much more flexible and easy-to-implement standard compared to OBSAI as it standardizes less. On the

other hand, CPRI leaves too many areas to the implementor to make it a truly plug-and-play standard.

3.4   Summary

This chapter briefly discussed two contemporary wireless technology architectures along with two standardization efforts for modularizing BSs. The discussion should provide with necessary insight to typical network architectures which is necessary to understand the importance of BSs in a deployment.   Additionally, the brief introduction to the OBSAI and the CPRI standards should familiarize the readers with these standards' contribution to the network architecture.

Chapter 4

OBSAI Base Station Architecture

The Base Station (BS) is the portion of mobile network infrastructure that is responsible for signaling and data transferring between the subscriber station (SS) and the core network. A BS handles modulating and transcoding of data channels, allocation of radio channels, paging, quality of service, reception and transmitting of data over air interface, and various other radio network specific functions. In addition to this, a BS also consists of mechanical devices; user, control, and management plane interfaces; power supply interfaces; and a clock. The BS is one of the most complex and costly element of a network. On the other hand, quantitatively, BS is the most numerous node on the network as it needs to be placed in all the regions an operator desires to provide coverage[1]. To leverage the cost efficiency and availability of BS elements, OBSAI has introduced a standardized architecture for the BS.

4.1 Motivation

BSs have thus far mainly been researched, designed, and developed by Original Equipment Manufacturers (OEM). This is due to the fact that the

---

[1]Although it is likely to be far more subscriber stations than BSs; and these devices also perform almost all the functions of a BS such as modulation, coding, transmission & reception, etc.; so while they have nearly the same complexity as a BS then generally have a much lower duty cycle than a BS. However, while SSs has been manufactured in high volumes keeping their manufacturing cost down, BSs have not traditionally been built in large enough volumes to justify the design efforts to reduce their manufacturing costs.

network infrastructure business has been dominated by manufacturers with a vertical integration model. OEMs have argued that the architecture is complex enough that it must have many proprietary interfaces, which results in complete vertical integration[2]. The advantage of this type of integration is that the technology is completely concentrated within one company and is well understood by all the stakeholders of a system. Thus, the parts of the system work very well together and have very well managed interfaces and interoperability with other components from this same vendor. On the other hand, due to the concentration of knowledge and research activities, the development time of new technology is high. The long time-frame leads to a long time-to-market and/or lack of quality. However, from the user and operator perspectives, there is a continuous need to increase bandwidth and to address increased subscriber station (SS) mobility. This has attracted many researchers to this field, resulting in lots of research and standardization. Emergence of new standards promising higher pick data rates and better mobility features would attract many wireless and mobile operators. However, when operators try to expand or upgrade their network, they must turn to the OEM vendors of equipment for state-of-the-art equipment as well as to the original manufacturer of their existing equipment for network upgrades. This expectation from the operators puts a huge amount of pressure on the research and development activities of the OEM vendors because the demand from the operators are driven by their market demand and not by the complexity of the research. However, vendors need to quickly release products to market can result in poor quality and missed deadlines.

To overcome these problems, OBSAI has specified a modularized architecture for the Base Transceiver System (BTS) based upon standardized interfaces among the modules [7]. The goal is to enable modularity in the BTS, thus allowing module manufacturers to enter this business. The architecture allows a vendor to concentrate on one specific module, thus allowing a high degree of specialization. This specialization enables a company to maintain their research activity on par with the current trends in community and standards bodies. In addition, research collaboration will foster a healthy environment based on inter-working and diversity. On the other hand, standards-based interfaces between the modules ensure correct interoperability and complete plug-n-play. This will allow OEM vendors to outsource various modules and integrate them

---

[2]This situation is true even in the case of standards that were supposed to promote interoperability, such as GSM, where one vendor's base station transceivers would not work well with another vendor's radio network controller.

without difficulty, thus reducing time-to-market by leveraging horizontal integration. The architecture also enables formation of a supplier ecosystem and allows innovation to focus on customer needs.

Although VLSI fosters single chip solutions and there already is evidence of low-powered, single-sector BSs developed in a single chip [30], modular design always allows a simpler approach to the development and is equally applicable to single chip designs [31]. Potentially the various modules can be integrated as intellectual property from multiple designers.

4.2   Challenges

The OEM companies were not very eager to join in an effort to commoditize the BTS, partly due to the fact that it can directly affect their profit margin, and partly because of the argument that a BTS is a sufficiently complex system that it should be proprietary and not built of standardized modules. However, the ability to bring products to the market faster, to reduce the cost of the system, and to reduce the testing & maintenance costs have became more important for these companies, thus OBSAI came into being. The main challenge for OBSAI was to modularize and specify the components of a BTS in such a way that it matches most of the existing vendor's implementations. Moreover, the specification should leave enough flexibility to foster better implementations. Simplicity was one of the properties desired of the standardization, otherwise, it might fail to attract anyone to follow it.

4.3   OBSAI Architecture

OBSAI has defined a complete architecture for base stations to ensure modularity and interoperability. The primary objectives of the standardization work are [32]:

- An open, modular architecture of a wireless BTS.

- Specification of a set of standard BTS modules so that BTS vendors can acquire BTS modules and integrate them as an integrator.

- Specification of standardized interfaces among the modules based on a logical model of an entity-controlled through these interfaces to ensure interoperability and compatibility among the BTS modules.

- Specification of standards based distribution of a clock and synchronization messages to ensure timing, frequency stability, phase noise, and jitter constraints are met as required by the air interface standards.

- Specification of common operations, maintenance, and performance (OAM&P) principles for integrating multi-vendor modules to create a fully functioning BTS.

- Specification of an internal BTS structure to allow scalability as well as support multiple air-interface standards

4.3.1 Architectural Overview

The OBSAI architecture of a BTS consists of several modules and the interconnections among them. In addition, it also describes the external network connections that allows the BTS to connect to other networks. The modules are described here as functional blocks from a WiMAX point of view, according to the OBSAI BTS system reference document [32]. The architectural discussion of the OBSAI BTS system in this chapter will be limited to the following elements:

- **Functional Blocks** that represent a logical grouping of a set of functions and attributes. Each block is a logical separation with regard to protocol processing.

- **Internal interfaces** among BTS functional blocks allowing interchanging of control, performance, status, alarm, payload, air-interface user data, and provisioning data. These interfaces are specified as reference points (RPs). Three RPs are specified: RP1, RP2, and RP3.

- **External network interfaces** allow communication with the core network from a BTS. Examples are: in WiMAX: R6 to an ASN gateway or R3 to a CSN; in 3GPP systems: (lub) to the radio network controller (RNC); in 3GPP2 systems: Abis to the base station controller (BSC).

- **External radio interfaces** that allow communication with the SS. Examples are: R1 for WiMAX, Uu or Um to the user equipment (UE) for 3GPP systems, and Um for 3GPP2 systems.

A logical structure of the functional architecture of an OBSAI BTS is shown in Figure 4.1 and the correspondence of the logical architecture with the physical equipment is depicted in Figure 4.2. The mapping of the logical architecture to the hardware reveals that the RP3/RP3-01 interfaces are the most important interface in an OBSAI base station architecture–as they are the key to the interoperability of the radio heads to the rest of the base station. The other interfaces such as the RP2, RP3, and RP4 resides practically within the same hardware module (System module in Figure 4.2) rendering their importance to a much lower level than that of the RP3. According to the system specification [32], the configuration of a BTS is constrained by several parameters which are specified in Table 4.1.

4.3.2 Functional Specification of Blocks

The OBSAI architecture has been divided into five main functional blocks. These blocks are:

- Transport Block (TB)

- Control and Clock Block (CCB)

- Baseband Block (BB)

- RF Block (RFB)

- General Purpose Block (Optional)

The functionality and purpose of each type of block is described briefly in the following paragraphs.

4.3.2.1 Transport Block (TB)

The transport block of a BTS transports user data to/from the core network. It should provide for concurrent operation of two or more air interface standards. The prominent functions of the TB are [32]:

Internal networking functions: The transport block will perform routing of user plane data (U-plane), control plane data (C-plane), and management plane data (M-plane) between the network interface, and the RP1 [33] & RP2 [34] interfaces. The TB should support communication between other BTS blocks via RP1 and for the U-plane data using RP2.
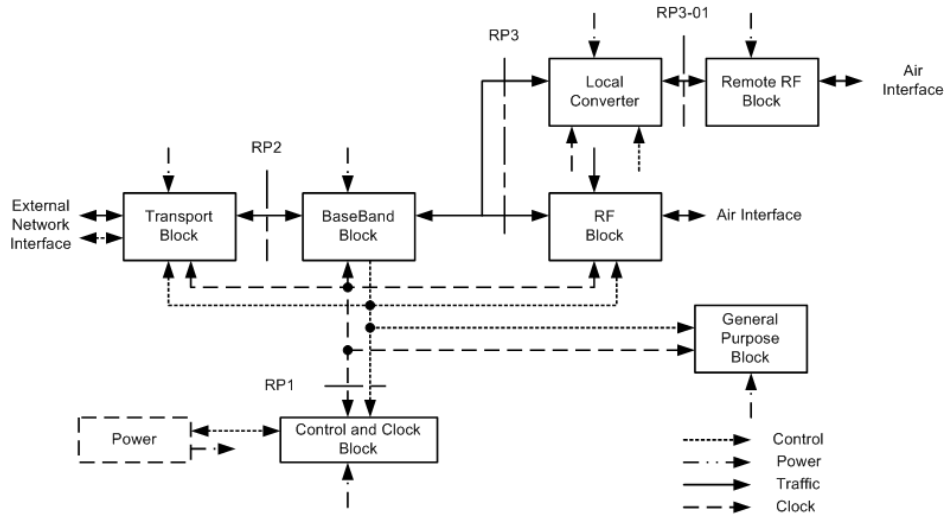
Figure 4.1: OBSAI BTS Reference Architecture [32]
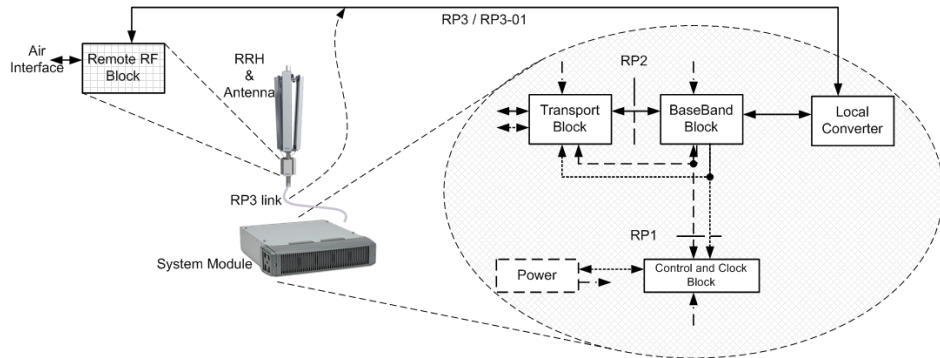
Figure 4.2: Mapping between Reference Architecture and Equipment Blocks

External network interface functions: The TB provides interconnection between the external systems, such as RNC/BSC/ASN-GW or CSN, and the BTS's internal systems via the network interface and reference points. For a complete list of protocols to be supported by the TB please refer to [32] and [35]. The TB uses features from [36, 37, 38, 39, 40, 41].

QoS functions: The TB maintains QoS requirements specified by the Radio Network Layer (RNL) in the Transport Network Layer (TNL). To ensure this requirement is met, the TB implements established throughput, delay, delay variation, and drop rate limits. It also implements packet fragmentation and reassembly.

Table 4.1: BTS Configuration Options

		GSM	WCDMA	CDMA	WiMAX
Frequency bands		800, 900, 1800, 1900	1800, 1900, 2100	800, 1900, 2100	2.5, 3.5, 5.8 GHz
Number of carriers/sec		1...16	1...4	1...15	1...15
Number of sectors		1–6	1–6	1–6	1–6
Transport modules		1–2	1–2	1–2	1–2
Control and clock modules		1–2	1–2	1–2	1–2
Baseband modules		1–12	1–6	1–12	1–12
RF modules		1–9	1–9	1–9	1–9
Antennas/sector	Regular	2–4	2–4	2–4	2–4
	Smart	4–8	4–8	4–8	4–8

Synchronization functions: The TB is responsible for distributing synchronization elements across the BTS. These elements include a clock signal and Synchronization Status Messages (SSM) derived from the ingress network interfaces. Usually, three sources of synchronization are used: (1) a clock signal recovered from the ingress network interface, (2) a clock signal from CCB, and (3) a free running clock.

OAM&P functions: TB will conform to system level OAM&P for the BTS. The OAM&P messages are usually encoded in XML, allowing legacy network management system (NMS) to be able to control the BTS using a web interface.

Security functions: The TB can also provide security for U-plane, M-plane, and OAM&P messages. If this security feature is enabled, then the TB terminates a secure tunnel using IPSec which ensures access control, integrity, authentication, and confidentiality.

4.3.2.2 Control and Clock Block (CCB)

A CCB is the heart of the BTS in terms of processing power. It includes the primary control processor which controls resources, supervises BTS activities, monitors status, and reports BTS status & performance data. The CCB should also provide the BTS with the ability to concurrently utilize two or

more air interface standards. The main functions of the CCB are briefly described below. More detail can be found at [42].

Congestion Control: The congestion control function of the CCM determines the TB, BB, or RFB resource overload conditions and report them to the application layer. In addition, the CCM congestion control function also determines the corrective measures that should be taken in order to restore the system to a normal state.

Admission Control: The admission control function of the CCB controls the admission of new users, new radio access bearers (RAB), and new radio links according to the current load and performance measurements.

BTS level OAM&P Functions: The CCB should conform to the OAM&P functions as specified in [32].

Radio Resource Management: The CCB is responsible for allocation and deallocation of radio resources of the BTS. This management includes functionalities including: call admission control, transmit power allocation, channel element allocation/deallocation, code allocation, data call admission control, and radio channel rate setting.

Multi-vendor configuration: The CCB should provide the high-level interface in a multi-vendor scenario. It performs any interoperation functions needed for allowing another vendor's RFB, BB, or TB integrate with the CCB.

RF scheduling: The CCB should perform the RF scheduling.

System clock generation and distribution: This function generates the clock signals necessary for synchronizing all the BTS modules. The CCB is responsible for clock monitoring, clock selection, and clock switchover.

Network interface signaling termination: This function terminates the backhaul signaling protocols.

4.3.2.3 Baseband Block (BB)

The BB in a BTS provides all the baseband related functions for different air-interface standards. The OBSAI BB should support baseband operation of various air interface standards, such as GSM, WiMAX, CDMA, and LTE.

The functionality of the BB is dependent upon the air interface standard it supports. The prominent functions it provides are:

MAC: BB provides the necessary MAC layer functionality for the air interface standards it supports. For example, in IEEE 802.16, it provides a connection-oriented service to the upper layers of the protocol stack where connections have QoS features that are granted and maintained by the MAC. QoS service in the IEEE 802.16 MAC service can be of four types: (1) constant bit rate grant, (2) real time polling, (3) nonreal-time polling, and (4) best effort.

Randomization/De-randomization: Data randomization is performed on each burst of data according to the IEEE 802.16 standard.This function is performed on each data block, which means, the randomizer is used independently of the subchannel in the frequency domain or the OFDM symbol in the time domain.

FEC: According to the IEEE 802.16 standard, convolutional or turbo-coding is performed for the downlink, and decoding of convolutional and turbo-coded data is done for uplink.

Interleaving/Deinterleaving: The BB performs interleaving of encoded data bits according to the IEEE 802.16 standard. Interleaving protects long runs of low reliablity bits.

Modulation/Demodulation: The BB converts the data into I&Q baseband signals which can be transmitted by OFDM or OFDMA carriers.

Beamforming algorithm: This is an optional feature. If multiple adaptive antennas are used, then beamforming algorithms to improve performance are executed in the BB.

MIMO: This is an optional feature. The IEEE 802.16 BB module can provide spatial signal processing that employs an array of antennas such as MIMO, space division multiple access (SDMA), etc.

Frame construction/Deconstruction: The BB performs on-air framing according to the IEEE 802.16 standard. In time division duplex (TDD), each frame is further divided in to uplink and downlink subframes where the division point can vary per frame, allowing asymmetric allocation of on-air time between uplink and downlink if required.

FFT/IFFT: The IFFT and FFT algorithms are used to convert between time and frequency domains. This processing is required when decoding the multiple carriers in an OFDM system. This feature is the responsibility of the BB.

Cyclic prefix addition/removal: To provide synchronization capability for a WiMAX system, a cyclic prefix is added to the frame by the BB.

TDD switch control: The BB controls the RFB using this control message to indicate when it should transmit and when it should receive.

Baseband OAM&P functions: The BB performs OAM&P functions in the BTS system level. Further details about these functions are specified in [43].

4.3.2.4 RF Block (RFB)

The RFB provides the RF related services for the BTS. It provides the capability to perform concurrent operations of two or more air-interface standards. The functions of the RFB are specified below. These functions are based on WiMAX standards.

D/A and A/D conversion: The RFB converts the digital signal received from the BB to its analog form for transmitting and vice versa.

Carrier selection: RFB provides the ability to select or change the carrier frequency for any module in the RFB.

Linearized power amplification: The RFB amplifies the signal to be transmitted while preserving the original wave form of the input signal.

Antenna interface: The RFB provides an interface to attach antennas to the block. The standards only specified a 50-ohm antenna interface.

Diversity transmit/receive: The RFB should provide the ability to transmit/receive signals through a polarization antenna or multiple antennas.

RFB OAM&P function: The RFB should conform to the OAM&P architecture as defined in [44].

4.3.2.5   General Purpose Block (GPB)

An OBSAI BTS may need additional capabilities that are not provided by the previously described BB, RFB, CCB, or TB. For such capabilities, an optional GPB can be added. Although all the interfaces of such an unstandardized block cannot be specified, it should support the requirements specified by RP1. Examples of a GPB are:

- Network interface module

- External I/O interface module

- GPS receiver module

4.3.3   Functional Specification of Internal Interfaces

The OBSAI base station architecture specifies standardized interfaces for communication among the different modules. These interfaces are named Reference Points (RP) and are numbered according to their functionality. The interoperability goal of OBSAI is highly dependent upon the interfaces as these specifications will govern how different modules will communicate. OBSAI has specified four interfaces for connecting different modules.

- Reference Point 1 (RP1)

- Reference Point 2 (RP2)

- Reference Point 3 (RP3)

- Reference Point 4 (RP4)

Along with these reference points, OBSAI also specified an additional RP3-01 interface which allows RP3 to carry RP1 messages to remote radio heads (RRH). Before delving deep into the problem areas of these interfaces, a brief introduction to these interfaces is given.

4.3.3.1   Reference Point 1

RP1 defines the characteristics of an OBSAI BTS's control and management planes. It interchanges control, performance, status, alarm, and provisioning data between the CCB and other BTS blocks according to

the protocol defined in the reference point 1 specification [33]. It also defines an open, standardized interface for distributing clock and synchronization signals. To support communication between the CCB and other BTS blocks, RP1 allows a direct or indirect interface (Figure 4.3). In both cases, the interface from the CCB to the designated module is completely specified by OBSAI standards.

The management-plane signaling can be either external, where the management commands come from an external user or going to external equipment, or it can be internal for managing OBSAI specified modules such as BB or RFB. External management-plane signaling usually comes through the TB and is translated to the OBSAI protocol architecture from the network interface specific protocol, while internal management data usually flows between the BTS master agent and other module agents encoded in Simple Object Access Protocol (SOAP) and are carried over IP. The transport protocol used is TCP or UDPCP [45]. Figure 4.4 shows the protocol stack for both internal and external M-plane communication. The definition of specific SOAP messages for fault, performance, configuration, and software management can be found at appendix B [46], C [47], D [48], and E [49] of the RP1 specification respectively.
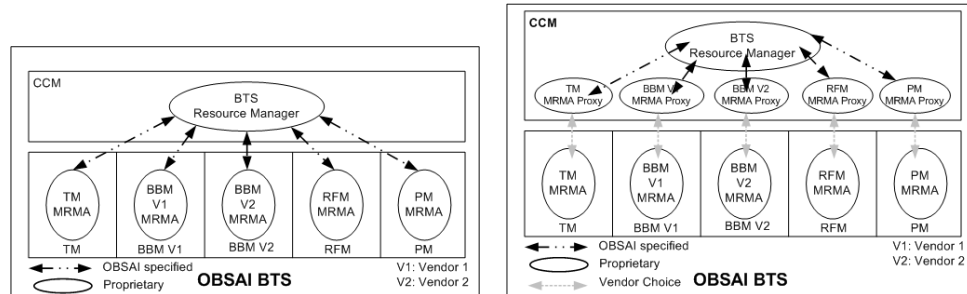
Figure 4.3: OBSAI RP1 (A) direct, and (B) indirect interface (Adapted from [33], Page 15-16)

4.3.3.2 Reference Point 2

Reference point 2 exchanges user plane data between the TB and the BB according to the protocols stated in the RP2 specification document [34]. RP2 utilizes well-known LAN technologies (such as Ethernet) and IP. This reference point is agnostic to the specific radio interface protocol and passes frames transparently according to the associated QoS attributes. RP2 uses
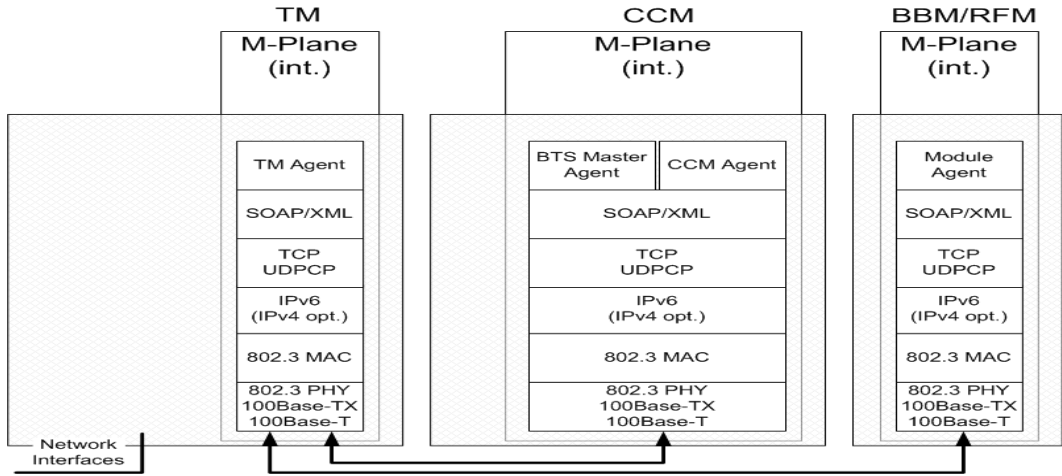
Figure 4.4: RP1 Management plane (a) Internal, (B) External (Adapted from [33], Page 20)

the Ethernet MAC layer defined in IEEE 802.3 [50], IPv6 [51] as the network layer, and UDP [52] or GRE [53] at the transport layer. To convert the specific network interface protocol to the RP2 protocol, the TB performs the necessary interworking. RP2 places some constraints on the communication done through it. It mandates that data link layer latency, i.e. Ethernet switching delay, should not exceed $100\mu s$ and the network layer latency should not exceed 1ms. For QoS purposes RP2 allows the use of DiffServ [37] code points at the endpoints. The supported DiffServ code points are shown in Table 4.2.

Table 4.2: DiffServ Code Points for RP2 [34]

Plane	Service	Per-Hop Behavior	Code Point
User-plane	Voice	Expedited Forwarding	101110
	QoS-guranteed data	Assured Forwarding 4, medium drop precedence	100100
	QoS non-guranteed data	Assured Forwarding 3, high drop precedence	011100
Control-plane		Assured Forwarding 3, low drop precedence	011010
Management-plane		Assured Forwarding 2, high drop precedence	010110

4.3.3.3   Reference Point 3

Reference Point 3 interchanges formatted air-interface data and fast control data between the BB and the RFB using protocols specified in the RP3 specification [8]. The RP3 interface has received the most attention by the community, such as [54, 55], due to the fact that it allows communication with more than one RFB which demands specialized capabilities and protocols to meet stringent air-interface standard requirements. RP3 can carry RP1 control messages inside RP3 frames towards RRH. Further detail about RP3 will be presented in Chapter 5. Being the sole connection between the system module and the radio head, RP3 interface has complete responsibility for controlling the RFB. As this interface carries all the data from the RFB to the BB, its the data rate is very high and its requirements are stringent. The RP3 interface utilizes a line rate of $i * 768$ Mbit/s, where i is $1, 2, 4, or 8$. This line rate is achieved using an optical link between the RFB and BB. This optical link can be single mode or multimode fiber, where a single mode optical fiber link offers greater length.

4.3.3.4   Reference Point 4

Reference Point 4 provides the interface between the Power Module and other BTS modules. Although RP4 does not cover the internal specification of the power module, RP4 provides DC power to the BTS modules and a communication channel to the CCB. The communication methods of the channel follow the RP1 specification. OBSAI defines some of the power characteristics of the RP4 interface as information; however, it does not mandate the use of them in the OBSAI architecture nor does it specifies any length limit for the power line nor any minimum or maximum loss rate for the line. The power interface specification can be found in [56].

4.4   OBSAI Protocol Structure

OBSAI has defined a set of protocols to be used for communication between modules. Various well defined and widely-used standards from the Internet community were used in these specifications. In a few cases, a small modification to the original protocols has been made to comply with the stringent timing requirements of telecommunication networks.

RP1 and RP2 interfaces use a PHY layer according to the IEEE 802.3 specification of 100Base-TX or 1000Base-TX [50]. For longer distance,

1000Base-T [50] can be used. The data link layer uses the Ethernet MAC protocol according to IEEE 802.3, however VLAN (IEEE802.3p/q) techniques are not used. As a network layer, RP1 and RP2 use IPv6 [51] and optionally, IPv4. IP based scheduling and priorazation is allowed using DiffServ code points [36, 37, 38, 39, 41]. The transport layer supports UDP [52] and GRE [53]. In case of different transport layer protocols coming from the external networks, the transport layer performs interworking. The overall protocol stack for RP1 and RP2 are depicted in Figure 4.5.

The application layer of RP1 is an important part of the specification as this layer maintains the operation and control of the whole system. According to OBSAI RP1 specification [33], there exists two application layer planes: a control plane and a management plane. Simple Object Access Protocol (SOAP) [57] is used for communication with the module agents. SOAP can be used in conjunction with any transport protocol and OBSAI specified the use of TCP and UDPCP [45]. An example SOAP message is depicted in Figure 4.6; while the layered architecture of UDPCP-based communication stack for RP1 is depicted in Figure 4.7. UDPCP is a lightweight transport protocol based on UDP, which integrates several features including acknowledgment with UDP while preserving the low-overhead of original UDP protocol.
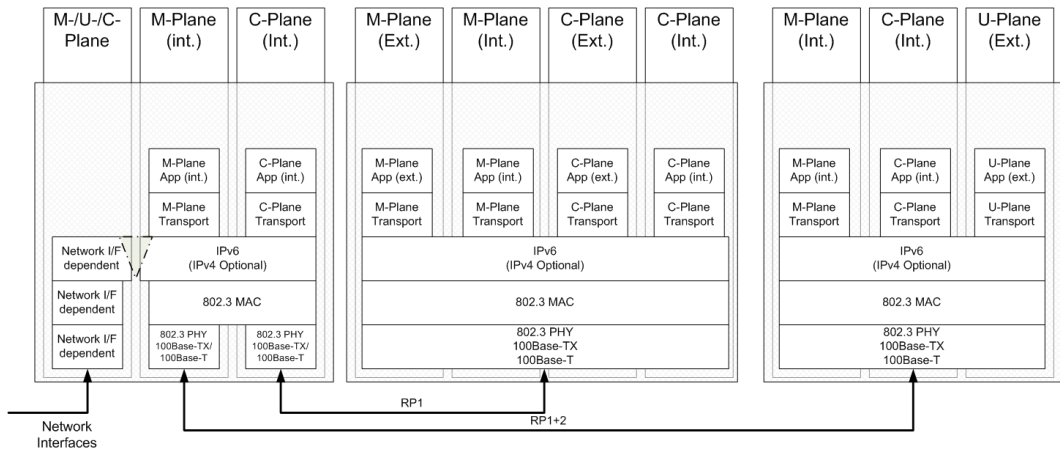
Figure 4.5: RP1 and RP2 protocol overview (Adapted from [34], Page 13)

The RP3 interface utilizes a completely different set of protocols and rules due to its direct communication with the air interface. The RP3 interface defines specialized rule-sets and protocols to meet the requirements of synchronization and timing of TDD systems as well as FDD systems.

RP3-01 is a special interface defined by OBSAI to facilitate the transmission of RP1 messages over RP3 frames. This specific requirement emerges from the fact that in the case of RRH there is only one RP3 connection to the remote site and no RP1 connection; thus to carry RP1 messages to control the RRH, they need to be carried via RP3 frames. In RP3-01, only the application layer of RP1 is used over the lower layers of RP3.

4.4.1 RP3 PHY

The electrical part of RP3 PHY is guided by the existing protocols such as the Ten (Roman X) Attachment Unit Interface (XAUI) defined in Clause 47 of IEEE 802.3ae-2002 [58] for upto 3072 MBaud, OIF-CEI-02.0 [59] Interoperability Agreement with its section 7 and related clauses, and the Serial RapidIO v2 PHY specifications [60] for 6144Mbaud line rates. A 8b10b transmission code is used to encode all the data transmitted over the RP3 bus. This allows for serialization and clock recovery. At the receiver module phase synchronization is done separately as a initialization procedure. The PHY layer data flow structure is shown in Figure 4.8.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP-ENV:Header>
                <to>BTS_OM</to>
                <from></from>
                <id>1001</id>
                <action>OBSAI_CM</action>
                <version>2.0</version>
        </SOAP-ENV:Header>
        <SOAP-ENV:Body>
                <moduleReadyInd>
                        <moduleType>RFM</moduleType>
                        <vendor>SOME_VENDOR</vendor>
                        <productCode>ABC.1234</productCode>
                        <serialNum>M2345678</serialNum>
                        <subRack>11</subRack>
                        <slot>1</slot>
                        <bldVersion>0.10</bldVersion>
                        <restartCause>power_on</restartCause>
                        <hwSwId>2</hwSwId >
                </moduleReadyInd>
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 4.6: SOAP message structure

4.4.2   RP3 Data Link

The RP3 data link uses fixed length frames with most-significant-bit-first. Before performing the 8b10b coding as indicated for the PHY, scrambling is done in case of 6144 MBaud. The RP3 data link layer uses master frames, IDLE bytes, and DATA bytes to synchronize the interfaces. More details about the operational mode of the RP3 link layer can be found in chapter 4 of [8].
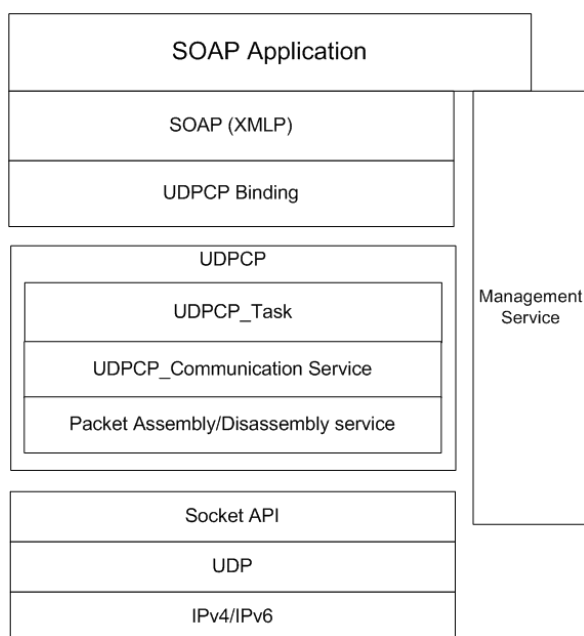
Figure 4.7: Layered architecture of UDPCP communication stack (Adapted from [33], Page 32)

4.4.3   RP3 Transport

The transport layer is responsible for message routing, multiplexing, demultiplexing, and summing (Section 5.1.3). Based on the destination address of the message, routing of messages are done at transport layer. The transport layer only inspects the address field of the message, which occupies the first 13 bits. The remaining part of the message is passed transparently by the transport layer. The summing unit of transport, which is one of the new features, combines together payloads of messages that are to be transmitted simultaneously to the same output port. The address in

use is hierarchical, and is divided into two parts: node (8 bits) and subnode (5 bits) address. This situation is shown in Figure 4.9.
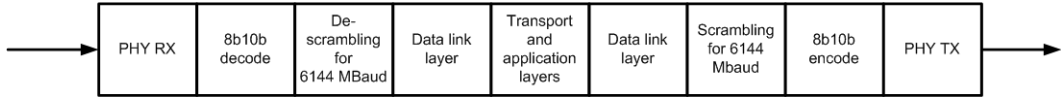
Figure 4.8: RP3: PHY layer data flow approach (Adapted from [8], Page 26)

4.4.4 RP3 Application

The application layer is responsible for mapping different types of data and control messages to the lower layer protocols. Two types of application layer protocols are in use: an air-interface specific application layer and a bus interface application layer. The air-interface specific application layer generates data messages compatible with the specific air-interface in use and the bus interface maps these data messages to the bus ready to transmit using the lower layer protocols. From the OBSAI point of view, the bus interface application layer has drawn the most attention as the air-interface application layer is chosen by the operator; and the RP3 protocol should be agnostic to the air-interface application layer protocol in use.
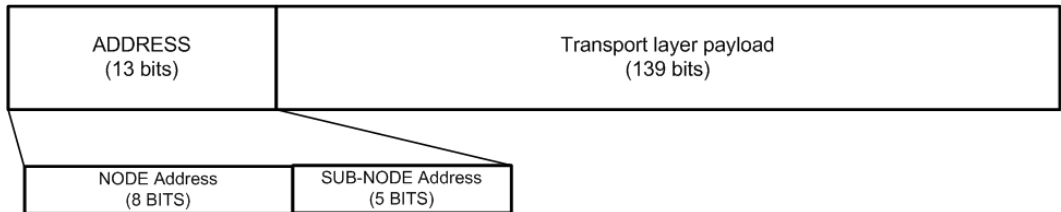
Figure 4.9: RP3: Transport layer addressing scheme (Adapted from [8], Page 48)

Two layers of message transmission rules exist: modulo computation and bit maps. Modulo computation, which utilizes a lower layer counter of message slots, is mandatory whereas bit maps are optional. In modulo transmission rule, message slots for each RP3 virtual link are specified by the pair of numbers $(I(index), M(modulo))$, such that the equation $MessageSlotCounter$ modulo $M = I$ holds. Dual bitmap concept is used when modulo rule alone cannot fillup the total capacity of the RP3 link. The application layer is also responsible for defining the type of the

message using a Type field and relates the payload data to a specific time instant using a Timestamp field.

RP3-01 enables a completely different set of protocols to be carried over RP3 message frames. As discussed earlier in Section 4.4, RP1 messages are carried over RP3 frames via the RP3-01 interface. Thus, RP3-01 uses the same SOAP message structure over UDPCP as RP1 and the resulting data unit is carried over RP3 data frames using RP3's message transmission rules. Transmission rules are defined for each air interface protocol according to its sample rate and channel bandwidth. Example transmission rules for LTE can be found in Table 55 of [8].

4.5   Summary

This chapter summarized the OBSAI base station architecture as defined in the OBSAI standards. After stating the motivation behind standardizing BS architecture for wireless networks, this chapter discussed the challenges faced during the standardization effort. This discussion is followed by an architectural overview of an OBSAI BS including the functional specification of logical blocks as well as internal interfaces. In the end, the chapter summarized the protocols used in one of the most important interfaces of an OBSAI BS: the RP3 interface.

Chapter 5

OBSAI RP3/RP3-01 Interoperability Status Analysis

The OBSAI specification aimed to diminish the proprietary nature of a modern BS architecture. The extensive specification documents have tried to establish a concrete standard for BS modules which should be interoperable using these standardized interfaces. In spite of this, modules from different vendors are still not completely interoperable. An extensive study of the specification documents revealed many areas of the specification that allow flexibility in implementations, thus creating opportunities for non-interoperability. This chapter will concentrate on reviewing the status of interoperability according to the current specification of OBSAI. In the course of this review, this chapter will list the expectations for interoperability in different layers of the protocol stack and comment on whether the expectations are met.

5.1 Reference Point 3

As integration of RF modules with the BB is the primary focus of equipment vendors, the RP3 interface is of utmost importance–since the interface connects the RFB and BB, and exchanges protocol data. This exchange should be fully harmonized between the modules if the standardization is robust and correctly implemented. This section concentrates on the interoperability status of the RP3 on a layer by layer basis.

5.1.1 RP3 PHY layer

The RP3 PHY layer specifies the physical properties of the link. These specifications include possible conversion of link layer frames into corresponding medium signal, data format and line coding, and bus clock. The following points briefly describe the main functionalities of this PHY:

Line coding and data format: All data should be encoded in 8b10b transmission coding to provide a mechanism for serialization and clock recovery.

Synchronization of PHY receiver: In PHY receivers phase adjustments are automatically done while initiating the connection.

Line Code Violation (LCV) detection: PHY detects all the line code violations or erroneous codes at the 8b10b decoder and reports these to the link layer via a K30.7 character [61].

Bus clock: The PHY layers of the bus phase and frequency are locked to a BTS system clock.

If an implementation is correct, these functions should be performed without any errors and two ends of a physical link should interpret each others' signaling correctly. However, this may not be the case currently. Incompatibility in optical interfaces or type of fiber can block the physical link from operating. Synchronization of PHY receiver can also be disrupted. A detailed study of this layer identifies the problem areas listed in Table 5.1.

5.1.2 RP3 link layer

The link layer of RP3 protocol stack performs various important tasks, such as message framing, bit level scrambling, maintaining counters, empty message insertion, synchronization, and measurements. The link layer comprises of a lot of functionality, therefore there are multiple areas of concern. Major features and parameters of the link layer are:

Message overview and frame structure: The RP3 link layer uses a fixed length frame of 19 bytes. The fields of the message are illustrated in figure 5.1. This message is packed in a frame. The rules for packing are: a block of data consisting of M_MG messages

Table 5.1: OBSAI RP3 PHY interoperability concentration areas

Feature	Standard	Comment
Physical layer loop-back	Shall be supported (See Figure 10 of [8])	While implementation of loopback interfaces in different internal paths of the Remote radio head is recommended in the standard, it is possible for the implementors not to implement one or more of these to simplify the design.
PHY indicates LCV to MAC	PHY transmits K30.7 to MAC in case of LCV	PHY should indicate with this error code an line code violation. But, it is possible for implementations to lack this feature for simplification.
Equalization	At 6144Mbaud line rate, electric equalization is recommended to avoid ISI	While ISI is a problem for electrical paths, it is not a problem for an optical path. Thus implementations may or may not support ISI.
Line Rate	Can be upto 6144Mbps	Although current specification supports line rate up to 6144Mbps, most of the implementations support only upto 3072Mbps.
Internal delay measurement	The measurement of internal delay should be stored as parameters $\Delta_{1,2}; \Delta_{2,3}$ and so on	The method of measuring this delay are not specified, making it difficult for two implementations to match in the way they support this feature.
Topology	Can be daisy chained, star etc.	All the radio heads may not support all the topologies specified in the standard.

$(0 < M\_MG < 65536)$, and K_MG IDLE bytes $(0 < K\_MG < 20)$ is called a message group (MG). A master frame is of 10ms with a fixed length and consists of $i * N\_MG$ consecutive message groups, where $0 < N\_MG < 65536$ and $i\epsilon\{1, 2, 4, 8\}$, for line rates of $i * 768Mbps$. The parameters N_MG and M_MG are selected such that $i * M\_MG * N\_MG < 2^{32}$. The size of a message group is $M\_MG * 19 + K\_MG$ and size of a master frame is $i * N\_MG * (M\_MG * 19 + K\_MG)$. Details of the K28.5 and K28.7 are given in [61].

ADDRESS (13 BITS)	TYPE (5 BITS)	TIMESTAMP (6 BITS)	PAYLOAD (128 BITS)
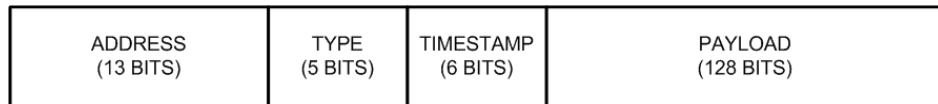
Figure 5.1: OBSAI RP3 link layer message overview (Adapted from [8], page 28)

Slots in a message group: The M_MG message slots are divided into control and data slots. For a line rate of $i * 768$ Mbps, the i last

message slots of every i^{th} message group are control slots while all others are data slots.

IDLE bytes: K_MG K28.5 IDLE codes exist at the end of each message group with one exception: when K28.7 is needed to mark the end of a master frame, which is used to facilitate reception.

Bit level scrambling: For 8x line rates, bit level scrambling is done to reduce cross talk and inter-symbol interference (ISI) between links. For lower speed links this is not required as cross talk and ISI is not a problem there.

Counters: The link layer provides higher layers with data and control message counters to facilitate scheduling of message transmissions.

Transmission of a frame: Transmission of a master frame is synchronized with the RP3 bus frame clock. A parameter value Δ is used to offset the internal delays of a node in a BS. The value of Δ is computed using the equation

$$\Delta = \Pi + D + B + P$$

Where D is the processing delay of the receiver module, B indicates the maximum amount of buffering available, while P stands for the latency across the node and Π is an offset defined below.

Reception of a frame: A offset parameter Π is calculated for each receiver to indicate the earliest possible time instant when a master frame can be received. The value of Π is set such that the master frame is received within the reference time, which is (bus frame clock + Π), or at most MAX_OFFSET ns after the reference time. An example of master frame timing with Δ and Π is depicted in Figure 5.2.

Link layer synchronization The receiver and transmitter of both ends of the fiber need to be synchronized before any real transmission can occur. Well defined state machines has been specified for receiving and transmitting functions. Behavior of state machines and precise timing of transferring from one state is required for the correct link layer synchronization. The state machines of transmitter and receiver are illustrated in Figures 5.3 and 5.4 respectively. A detailed description of these state diagrams can be found in Section 4.2.8 of [8].

Table 5.2 summarizes the result of my careful inspection of the standard specification.
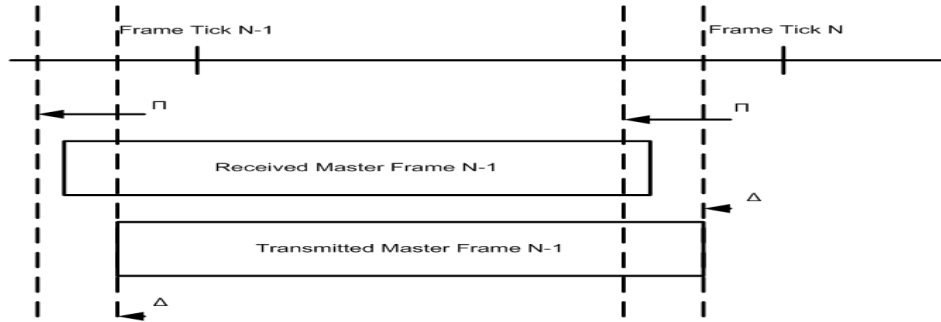
Figure 5.2: OBSAI RP3 master frame timing overview (Adapted from [8], page 38)
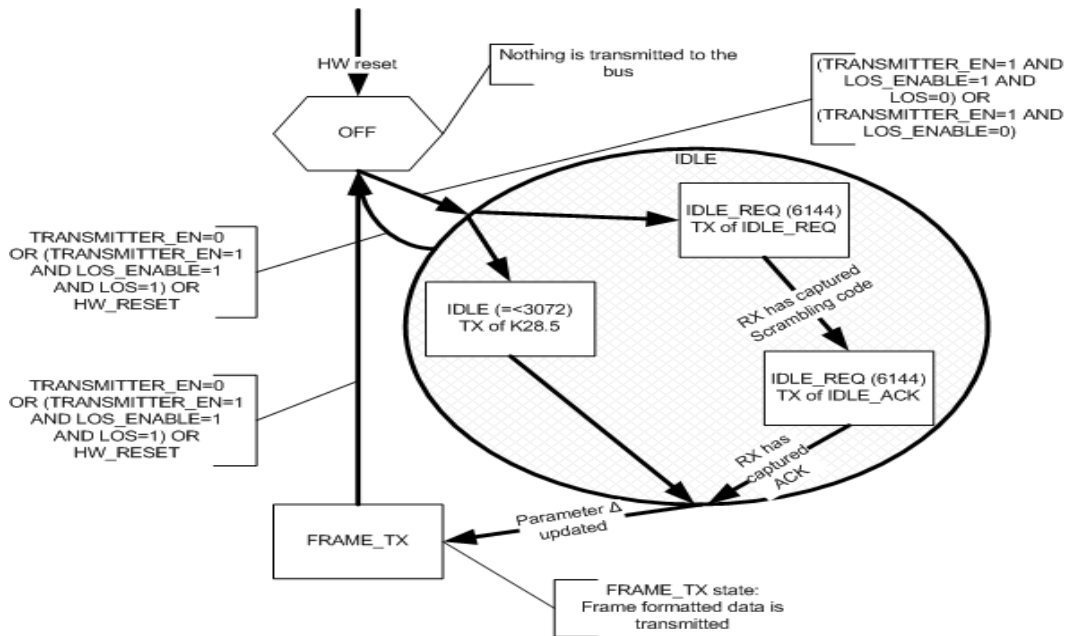
Figure 5.3: State diagram of RP3 transmitter (Adapted from [8], page 41)

5.1.3 RP3 Transport Layer

The transport layer of the RP3 stack functions as a message router, multiplexer, demultiplexer, and summing unit (described below). The routing techniques used in the transport stack are local, i.e., nodes are not visible to the whole message path.
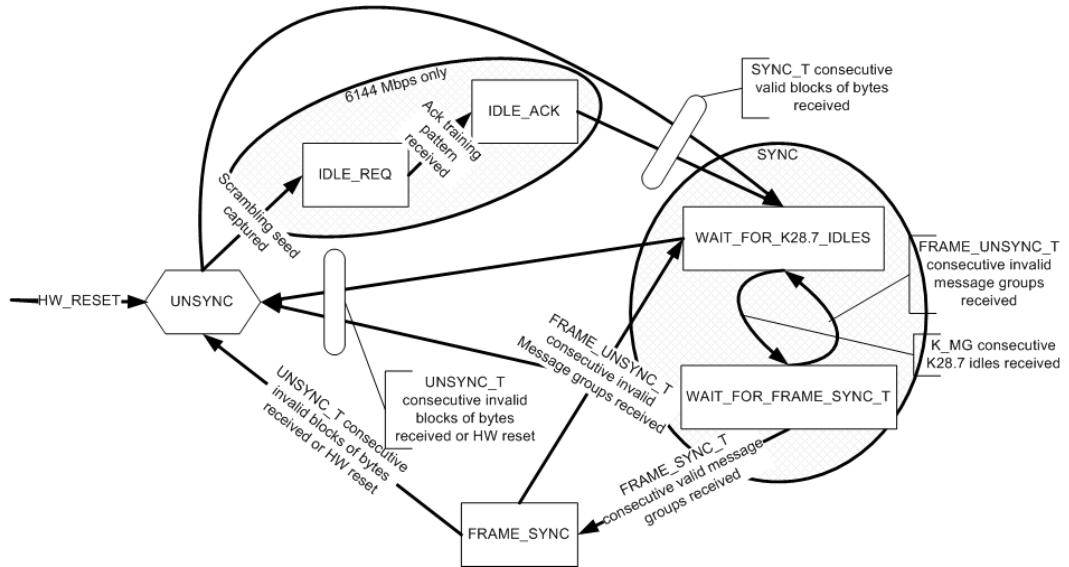
Figure 5.4: State diagram of RP3 receiver (Adapted from [8], page 43)

Table 5.2: OBSAI RP3 Link layer interoperability concentration areas

Feature	Standard	Comments
6144Mbaud line rate needs to be scrambled	6144Mbaud data is scrambled before 8b10b encoding	Implementations may decide not to scramble the data.
Parameters N_MG, M_MG, K_MG, i	$0 < K\_MG < 20$, $0 < M\_MG < 65536$, $0 < N\_MG < 65536$, $i\epsilon\{1, 2, 4, 8\}$, $i*N\_MG*M\_MG < 2^{32}$	Parameter values need to be matched and should be configurable.
Control Slots	The i last message slots of every i^{th} message group are control slots, while all others are data slots.	Proprietary implementations may follow internal rules to insert control messages in the message frame.
IDLE codes	K_MG K28.5 idle codes at the end of each message group except the last one of the master frame, where K28.7 is used to indicate the end.	Correct number and type of Idle codes need to be present in each message frame for correct functioning
Bit level scrambling for 8x line rate	The scrambling code generator generates seed value and communicates them to the receiver. Tx communicates this value to Rx via a training sequence.	For 8x line rate, the scrambling is done by codes which are generated from seeds and communicated to the receiver using training sequence. The number of seeds used needs to match.

Continued. . .

Table 5.2: (Continued)

Feature	Standard	Comments
Offset value Δ	Value of Δ should be received at start up by each bus node. Their values are specified in clock ticks with two byte accuracy. Two's complement numbers are used.	The specification of Δ can be in one of two ways, which needs to be agreed upon.
Value of MAX_OFFSET	Exact value of Π is provided so that the master frame end is received by the reference time or at maximum MAX_OFFSET ns after it. MAX_OFFSET can be 52.08ns or 104.17ns.	There are two possibilities for MAX_OFFSET value. This value should match for a bus transmitter and receiver.
Error in case of frames outside MAX_OFFSET	Error should be indicated to upper layers. But its optional for RRH. The message transmission can go on even if the master frame is outside the window.	This feature is optional. Agreement on whether normal operation will continue or not needs to be agreed upon.
The update of Π	The Rx state machine needs to be resynchronized in case of an update of Π. Advanced RP3 implementations can allow this without resync.	A real time update of Π is not always supported by the implementations.
Minimum time at IDLE state for Tx state machine	A Tx state machine will stay at IDLE state for at least t seconds, where t is implementation specific.	The number of seconds that the Tx state machine stays at IDLE state needs to be synchronized with the Rx.
Extra states in Tx FSM for 6144Mbaud	After the Rx macro has received IDLE_ACK from the Tx, the Tx starts waiting for Δ updating. It can either stay at the same state, or can go to an extra state.	
Measurements done at link layer	Timing offsets of received master frames with respect to a BB clock plus Π is measured.	Measurement procedures and accuracy should be monitored

End of Table

Addressing: Addressing in the transport layer is done using two level addresses: the node address and a sub node address. Total length of the address field in a packet is 13 bits, of which 8 bits is for the node address and the rest (5 bits) for sub-node.

Multiplexer and demultiplexer: The multiplexer performs the task of time-interleaving of messages from several incoming interfaces to a single outgoing interface. Messages from input links $k, 0 \leq k < N$, with message slot indices $M_k * i + j_k$ are forwarded to messages slots $M_{out} * i + j_{out}$ at the output link, where line rate of input link k is $M_k * 768Mbps$ and the line rate of output link is $M_{out} * 768Mbps$, $M_k \epsilon \{1, 2, 4, 8\}$, $j_k \epsilon \{0, ..., M_k - 1\}$, $M_{out} \epsilon \{1, 2, 4, 8\}$, and $j_{out} \epsilon \{1, ..., M_{out} - 1\}$.

Summing: When the header of two messages are exactly same, then the summing unit generates a single message from multiple messages by

copying the header from one of the input messages and performing a point-wise sum [8] of payload of the messages, thus saving bandwidth.

5.1.4   RP3 Application Layer

The application layer is mainly divided into two parts: air interface applications, and bus functions. The air interface applications are specified by the corresponding air interface standardization body. OBSAI specification defines the bus function standards. The main features of the application layer are:

Message Transmission rules: Two layers of message transmission rules can be used. At the lower layer, message slots for each RP3 virtual link are specified by the pair of numbers (I, M) where I specifies an Index and M specifies a Modulo. In the higher layer, dual bit map is used where two bit maps are used to determine the position for a message in the slots. For the algorithm of dual bit map, please refer to Section 4.4.3 of [8] .

Buffering requirement: As the bus serializes data, the application layer needs to have buffering capabilities. For example, a buffer of size six frames is required for each IEEE 802.16 signal in order to compensate for jitter.

Message types: The application layer defines the type of the messages by inserting a type in a 5 bit field.

Timestamp: This field of each data frame relates the payload data to a specific time instant. For different air interface standards, this field carries different significance.

Generic Control messages: The application layer defines a structure for generic control messages which can be used by an implementation to send custom control messages to the receiver.

To get a fair understanding of areas of concern in this layer, Table 5.3 lists the potential interoperability issues.

Table 5.3: OBSAI RP3 Application layer interoperability concentration areas

Feature	Standard	Comment
Message transmission rules	Application layer utilizes data and control message counters provided by link layer. Two levels of rules. Lower layer uses modulo computation over message slot counters. The higher layer defines a path utilizing bit map.	Two implementation can differ on which level of rules they support. In case of modulo rules, the implementations need to agree on the rules for transmitting and receiving.
Buffer size	Depends on the air interface standard used. For WiMAX a buffer of six messages is required for compensating the jitter caused by message transmission.	Implementations needs to be sure of the buffer size used on the other side of the communication to be sure of achieving synchronization.
Generic Packets	Generic packets allow multiple RP3 messages to represent a larger packet.	This feature is optional, which may cause problem in different implementations
Control messages	Structure for generic control message has been specified	Proprietary control messages need to be communicated
Status messages	Structure for generic status messages has been specified	Proprietary status messages need to be agreed upon.

5.2   Reference Point 3-01

Remote RF units tend to be installed in mast heads or walls away from the BB installation to achieve greater LOS as well as for the RF signals experience less attenuation between the transmitter's and receiver's antenna. However, it is not practical to install separate physical connection for both an RP3 and RP1 connection between the BB and RFB. That is the reason why the protocol RP3-01 has been developed to carry RP1 messages over RP3 frames over the same link. While RP3-01 uses the RP3 rules for transmission, it defines a number of extra features required to efficiently transmit management plane messages.

Transfer over RP3 data: Different types of RP1 messages are transfered over RP3; they include Ethernet, RP1 frame clock burst, RTT measurement, virtual hardware (HW) reset, loop-back, and RP1 data. These frames can be transmitted in any RP3 data slot.

RP1 frame clock burst: RP1 frame clock bursts generated by the CCU are transmitted to the RFB using RP3-01 frames. Two runtime parameters named C1 and C2 as well as RTT time is used for synchronizing the timing of transmission according to the fiber's

length. A detailed description of the algorithm can be found in Section 6.2.3 of [8].

Ethernet transmission: The ethernet messages are transferred using point-to-point links over RP3 media.

Line rate auto negotiation: The line rate for RP3-01 is auto negotiated using the algorithm stated in Section 6.2.5 of [8].

RTT measurement: To configure and synchronize a distant RRH from the BS, an RTT measurement is necessary. The details of the RTT measurement algorithm are stated in Section 6.2.6 of [8]; along with internal delay definitions.

Virtual Hardware reset: To reset the whole BS for any reason, a virtual HW reset command is transmitted via a RP3-01 message.

An analysis on the areas of RP3-01 protocol that needs further attention while testing for interoperability are stated in Table 5.4.

Table 5.4: OBSAI RP3-01 interoperability concentration areas

Feature	Standard	Comments
Message transmission rules	Seperate rules can be defined for RP1 Ethernet, RP1 frame clock bursts, RTT measurement, virtual HW reset, loop back, RP3 data and RP3 control messages.	Transmission rules should be agreed upon for both sides.
Frame Clock burst buffering time at RRH	Algorithm for computing this time is specified in section 6.2.3 of [8].	Depending on the implementation and need for synchronizing the DSPs, the parameter $B_r ru$ may or may not be used.
MAC address for RRH	If RRH does not have any globally unique MAC address, it shall construct one using the ID which is communicated to RRH from BTS using RP3-01 messages via broadcasting.	MAC address construction mechanism may be different from implementation to implementation.
RTT measurement message slot	The procedure for RTT measurement has been specified.	Agreement of which slot is used for transmission is necessary.
Virtual HW reset	RRH can optionally support virtual HW reset. Moreover, it can be received in either a data or control message slot.	Agreement of which slot is used for transmission is necessary.
Customized control messages	The standards specifies a generic control message structure for RP3.	However, the implementation can use customized generic control messages for performing proprietary operations.

Table 5.5: OBSAI RP1 interoperability concentration areas

Message Name	Standard Elements	Standard sub-elements & values	Vendor A Elements	Vendor A sub-elements & values	Vendor B Elements	Vendor B sub-elements & values	Compa-tability
alarmNotif	faultId		supported	yes	supported	yes	yes
	faultSource			yes		yes	yes
	faultSeverity	critical		yes		yes	yes
		major		yes		yes	yes
		minor		yes		yes	yes
		warning		yes		yes	yes
	faultstate	active		yes		yes	yes
		cleared		yes		yes	yes
	faultText			yes		yes	yes
	eventTime			yes			yes
			affected-Objects	name	will be done		no
notification-Ack	relatesTo	information about the notification being acknowledged	supported	no sub-element	supported	yes	no

5.3   Reference Point 1

While the messages and protocol stack of RP3 and RP3-01 are the main focus of this thesis, RP1 messages constitute a large part of the interoperability testing as well. Moreover, as RP3-01 is specifically used to carry RP1 messages over RP3, no discussion of RP3-01 can be complete without a thorough analysis of RP1 messages. In the OBSAI standards, RP1 is also called the management plane and consists of SOAP/XML messages used for different configurations and management operations. Although the structure of RP1 messages has been explicitly defined in [33], implementations can differ in their treatment of optional elements of the message as well as in the extent of support for specific messages. There is also the possibility of custom messages that have to be taken into consideration while integrating two different implementations. RP1 messages support TCP, SCTP, or UDPCP as a transport layer. Four types of management plane messages are defined in the specification. They are:

- Fault Management,

- Performance Management,

- Configuration Management,

- Software Management.

A full analysis of all the messages would be extensive and out of scope of this document. However, in Table 5.5 a compatibility matrix is presented that provides the necessary structure to carry out a through analysis of the RP1 plane between two implementations. This table illustrates two example SOAP messages (i.e. alarmNotif and notificationAck) and can be extended for all the messages defined in [33].

5.4 Summary

One of the most important aspect of the thesis is to analyze the current situation of interoperability in OBSAI BSs. This chapter analized the specification of RP1, RP3, and RP3-01 interfaces and discussed about various areas of possible non-interoperability. The discussion was carried out on the protocol stack in a layer-by-layer basis.

Chapter 6

OBSAI RP3/RP3-01 Integration Tools and Experiments

When connecting a RFB and BB having different implementations of the RP3 protocol set, careful integration is required for every aspect of the protocol ranging from the PHY to the Application layer. Unfortunately, the flexibility in the OBSAI standard as discussed in the previous chapter forces an extensive interoperability test procedure–whenever an integration between two different implementations is performed. However, with the correct tools and testing guidelines this effort can be minimized. This chapter summarizes the findings from an extensive array of integration tests and lists necessary tools to accelerate the process. Integration experiments and observations are also a part of this chapter. This part of the thesis assumes that the implementations being tested against the following guidelines are module tested according to the recommendations of OBSAI [62]. Non-standard implementations are not taken into consideration.

This work divides the task of interoperability testing (IOT) of OBSAI RP3/RP3-01 interface into two phases: (1) Theoretical IOT phase, and (2) Practical IOT phase. In the first phase, the specification of two implementations are studied extensively to theoretically pre-determine the areas of non-interoperability. After correcting the identified areas in the first phase, second phase starts. In this phase practical integration task is carried out with necessary equipments.

6.1   Theoretical IOT Phase

The theoretical study is primarily dependent on the extensive analysis done in Chapter 5. Based on the identified areas, analysis of the implementation details and checking for match is possible. A compatibility matrix, as developed in this work, is a very useful tool for such work. An example matrix, which has been used in the theoretical experimentation for this work, is presented in Table 6.1. This matrix can be used to map the features of two implementations side by side and thus quickly figure out any possible differences between them.

Before starting practical experiments with the equipment and RP3 interfaces, a theoretical analysis using the features indicated in Chapter 5 and Table 6.1 should be performed. This pre-experimentation should help researchers to save a large amount of effort from the practical experiments. An example table with some of the features taken from the previous chapter is presented in Table 6.2.

Table 6.1: Construct of an example compatibility matrix

Feature	Standard	Vendor X	Vendor Y	Compatibility	Comments
...	...	...	...	...	...
...	...	...	...	...	...

6.2   Practical IOT Phase

Following the theoretical phase of the IOT test, practical experiments with equipment needs to be done. In this work, this phase has been divided into two primary approaches depending on the availability of the necessary equipment. First approach is suitable for preliminary R&D activities when only one part of the setup is available, e.g. RRH or BBM, and the validity of the RP3 implementation can be proved without practically attaching the interface to other parts of the network. And the second approach is based on a more practical scenario when the experiment is the part of a real integration project and equipment from two different vendors are available. Equipment used in the experiments include Radio module from vendor X and Y, Baseband module from vendor X, Elektrobit RP3 tester and necessary connectors. The Elektrobit RP3 tester is developed by Elektrobit Oy and more information about the tester can be found in their website (http://www.elektrobit.com/index.php).

Table 6.2: Construct of a theoretical analysis of interoperability

Feature	Standard	Vendor X	Vendor Y	Compatibility	Comments
Frame Clock burst buffering time B_{rru} at RRH	Algorithm for computing this time is specified in section 6.2.3	B_{rru} is not used	B_{rru} is used	Not compatible	Need clear specification on the need of B_{rru}
Message transmission rules	Message transmission rules should be specified for various types of messages using either modulo or dual bit map	Uses module arithmetic for specifying transmission rules.	Supports both modulo and bitmap rules	Compatible	Clearer indication of the need of each type of rules is necessary
MAC address for RRH	If RRH does not have any globally unique MAC address, it shall construct one using the ID which is communicated to RRH from BTS using RP3-01 messages via broadcasting	MAC address is computed using a custom control message from BBU	MAC address can be set using command line interface	Not compatible	Standard way of setting MAC address of the units need to specified
FCB header address		FCB is sent to the subnode address 0x20	FCB is broadcasted to 0x0	Not compatible	Exact specification of these addresses are required

6.2.1 Approach 1: RP3 validity check using RP3-tester

One of the primary goals of OBSAI was to facilitate small-scale companies to specialize on one part of the network solution and to perform R&D work on that. However, this type of business requires specialized test solutions to verify the equipment produced by the company. An RP3 tester was developed by Elektrobit Oy to perform such tests on the RP3 and RP3-01 interfaces and thus prove their correctness. This thesis considers this scenario as a very important one and therefore experiments were performed to module test the equipment in question.

6.2.1.1 Experimental Setup

Two separate setups has been prepared to experiment with the RP3 interface. First setup comprises of the RP3 tester along with a RRH while the second setup is with BBM and RP3 tester. The setups are shown in Figure 6.1. The

parameters of RP3 interface in EB tester and the RRH are shown in Table 6.3.
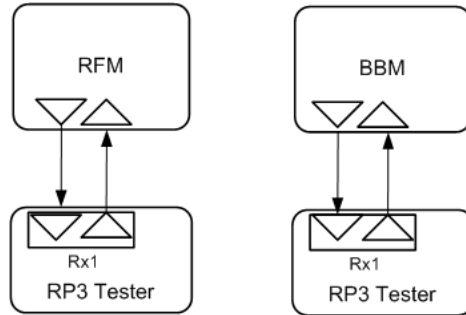
Figure 6.1: Experimental setup for Approach 1

Table 6.3: Parameters used in experimental setup 1 with RRH

Parameter Name	EB Tester	RRH
Mode	Master	Slave
Clock Source	Internal	RP3-01 Burst
CRC	Default	Default
RP1 Burst Type	WiMAX	N/A
Interface Tx1	WiMAX (5ms)	N/A
Interface Rx1	WiMAX (5ms)	N/A
RP1 Burst Reception	WiMAX	N/A
Receive Samples	1/Sec	N/A
Line Rate	4x	Auto Negotiation
Message in a group	21	21
Group in a frame	7680	7680
Idle codes	1	1
Tx FSM	IDLE	Auto
Rx FSM	Auto	Auto

The experiment was performed by conducting the following steps[1]:

- The RRH and the EB tester was connected with each other using compatible SFPs with a compatible fiber.

- The EB tester was powered up and left alone for 10 minutes to warm up.

---

[1]For sample outputs from the equipments, please refer to Appendix B

- The RRH was powered on.

- All the parameters of the tester is set as shown in Table 6.3.

- The Tx FSM of the tester is first kept in IDLE mode so that the tester continues to send K28.5 IDLE codes to the RRH.

- RRH FSM state is inspected from the command line interface (CLI) to check whether its in correct state or not

- If the RRH has already received valid message blocks it's Rx FSM state should be WAIT_FOR_K28.7_IDLES. In that case, Tx FSM of tester is changed to FRAME to start sending frames.

- The RRH Rx FSM state is inspected again using CLI and it should be in the state FRAME_SYNC in case valid frames are received from the tester.

- The Rx FSM of the tester should also change accordingly with the RRH Tx. This change in Rx and Tx FSM states is further explained in the later sections.

6.2.1.2 Observation

For correct progression of Rx and Tx state machines in a module, synchronization between the state machines is required. It is expected that the Tx state machine of the slave node progresses according to the current state of Rx FSM. The expected behavior of the interaction is illustrated in Figure 6.2. However, while performing the experiment with the RRH and the tester, it was observed that the Tx state machine of the RRH is not changing state according to the Rx FSM. While the Rx FSM is in FRAME_SYNC, Tx FSM is not yet in FRAME_TX mode and thus not transmitting any frame formatted data to the tester. Consequently, tester Rx FSM was stuck in WAIT_FOR_K28.7_IDLES state. This was determined to be an error in the RRH.

6.2.2 Approach 2: Integration between two different vendors

Although a full integration effort between two vendors should constitute interoperability of all the layers of protocol stack, unfortunately a full RP1
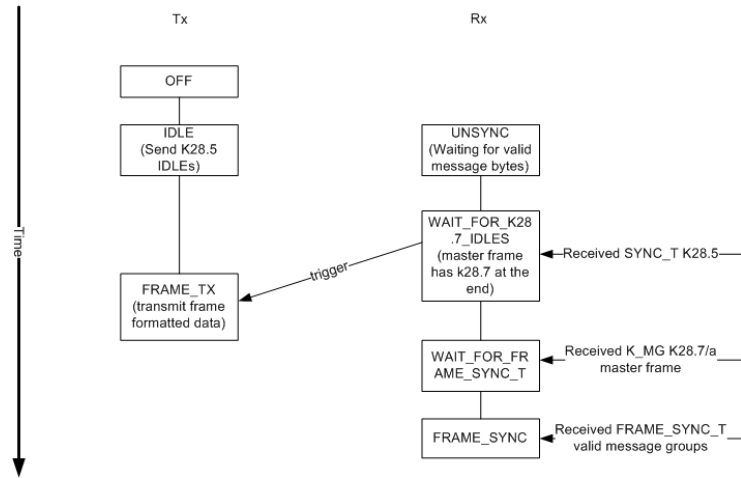
Figure 6.2: Interaction between Rx and Tx FSMs

M-plane implementation was not available from the RRH vendor at the time of this work. Consequently, the full system test was constrained in achieving the link layer synchronization. It was decided that as the RP1 messages are well defined in the specifications, ensuring application layer interoperability will not be a very difficult task in case all the functionalities are already available in the RRH. However, it should be beneficial to analyze the properties of the modules using the construct provided in Table 6.1 before starting interoperability tests.

6.2.2.1   Experimental Setup

The setup comprises of Baseband module from vendor A, Radio head from vendor B, EB RP3 tester, a Logging tool for the BBM, Command Line Interface (CLI) for the RRH, and two optical splitters. The setup is designed in a way that allows the RP3 tester to receive a fraction of signal from both uplink and downlink direction. This setup is depicted in Figure 6.3. The BBM logging tool and the RRH CLI were used to monitor the state of the units at any point of time while the RP3 tester was used to analyze the frames passing through the RP3 link.

The experiment was limited to the RP3 synchronization phase only due to the lack of RP1 management plane in the radio. The software at BBM is such that it requires a RP1 handshake before starting to exchange traffic between the BBM and RRH. However, as the RRH lacks RP1 plane, the experiment could not reach that level. For this setup, both RRH and BBU are kept at
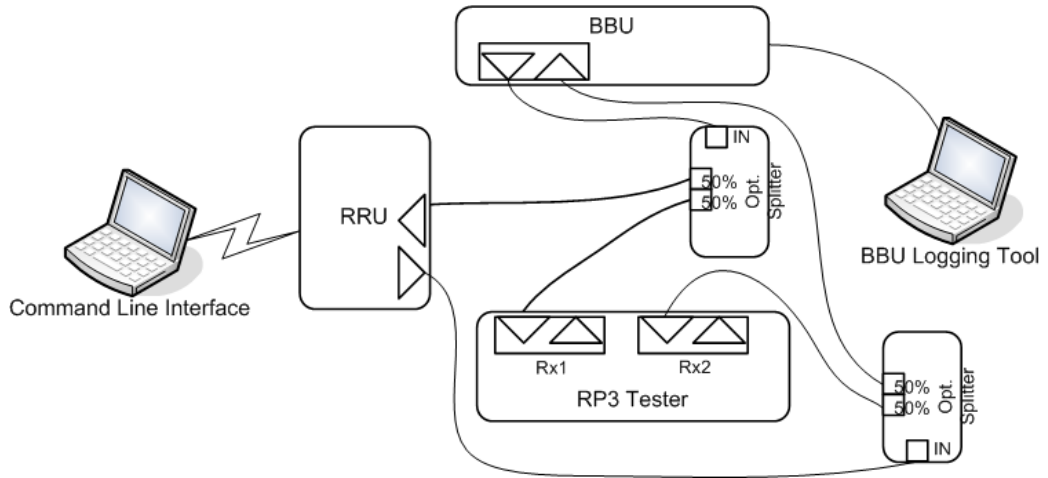
Figure 6.3: Experimental setup for Approach 2

their default setup, as shown in Table 6.4.

Table 6.4: Parameters used in experimental setup 2

Parameter Name	BBU	RRH
Mode	Master	Slave
Clock Source	GPS	RP3-01 Burst
RP1 Burst Type	Default	Default
Line Rate	4x	Auto Negotiation
Message in a group	21	21
Group in a frame	7680	7680
Idle codes	1	1
Tx FSM	Auto	Auto
Rx FSM	Auto	Auto

6.2.2.2 Observation

After the systems were connected as per Figure 6.3 and powered up, both sides were trying to synchronize according to the RP3 FSM. However, due to some strange behavior from the radio, they could not establish a working RP3 link. Preliminary, it was observed that the BBU is receiving some unusual signal from the radio which it did not expect and as a result it did not start to transmit the K28.5 IDLE bytes to the radio. Readings from the BBU log and the RRU CLI can be found in Appendix B. As discussed earlier with reference to 6.2, the state machines are dependent on each other and cannot

progress in case of any misbehavior in any part of it. After a rigorous testing phase using the logging tool; CLI; and the RP3 tester, the root causes of the malfunction were isolated. They are:

1. Status of wire transmission in the OFF state of Tx FSM

2. Destination address of RP3 frames

According to the specification, in the OFF state of Tx link, the wire should be in a Hi-Z state and nothing should be transmitted. However, it was found that the RRU was transmitting some signal even in OFF state. Further analysis showed that the Tx link is shut down before the 8b10b encoder of the RRU and the 8b10b encoder was encoding that absence of signal which resulted in a specific type of signal in the wire. This signal confused the BBU Rx state machine and prevented the Tx FSM of BBU to proceed further.

On the other hand, after the Rx state of RRU proceeds to WAIT_FOR_K28.7_IDLES, BBU should start transmitting Frame formatted data and upon detecting correct frames the RRU FSM transitions to WAIT_FOR_FRAMES status. However, the radio in our setup was not detecting any frames at all, and as a result the state transition was not happening. This situation resulted in further problems in the Tx FSM of the radio. As the radio's Tx FSM is dependent on the Rx FSM in slave mode, it was not transitioning to FRAME_TX mode and the BBU was also not receiving any frames from the radio. Further analysis with the traffic to and from the BBU captured with the logger revealed that the BBU is using a broadcast address in the frames it was transmitting while the RRU is listening for the frames in the address 0x20. As a result, the RRU was not receiving the frames and not changing its Rx status.

6.3 Analysis of Experimental Observations

From the theoretical study performed before starting the experiments, the non-interoperability between the equipment was expected. There were several conflicting features in the RP3 implementations of the BB and the RRH. Major ones are:

1. RRH listening to node address 0x20 for FCB messages

2. Unknown Tx OFF state for RRH

3. Unavailability of RP1 management plane in the RRH

4. No support for dual bit map in the BB

5. Proprietary control messages used in BB to control RRH

6. Mismatch in RP3 data alignment between BB and RRH

While staring the experiments, these areas were taken into consideration. However, because the RRH lacks the RP1 management plane completely, so it was not possible to send any traffic from the BB to the RRH. Therefore, practically testing the non-interoperability of the last three major differences as listed above were not possible. However, the experiments described in Sections 6.2.1 and 6.2.2 proved the incompatibility of the BB and the RRH in the first two points as mentioned above. While the radio should listen to the broadcast address as well as the module address, the experiments revealed that the RRH was listening only to the module address (0x20) and not to the broadcast address. This resulted in missing the FCB frames from the BB and thus not being able to synchronize with the BB. Moreover, the module testing done in approach 1 revealed that the RRH was not following the expected relation between the Tx and Rx FSMs. Which indicates the necessity of describing more precisely the relation among the Tx and Rx FSMs in the specifications (as shown in Figure 6.2).

Although could not be proved experimentally, theoretical study indicated that the integration would face problem in payload communication even if the interface is in full synchronization. This problem would occur from the fact that the BB did not implement dual bit map as specified in [8]. The specification says that whenever the antenna streams are not enough for filling up the RP3 frames completely, dual bit map algorithm is used to fill up the frames with empty data. However, the BB under consideration used a proprietary algorithm rather than the standard one. Moreover, should the modules were able to establish a working management plane among them, the BB would not be able to fully control the radio remotely until all the custom control messages implemented in the BB are also implemented in the RRH.

According to the observations from the experiments, both the BB and the RRH were fixed to operate correctly with each other.

6.3.1 Solution for Approach 1

Modifications in the RRH firmware to make the Tx FSM follow Figure 6.2 made the setup to work properly. With the modified firmware the modules were synchronized perfectly and ready to exchange necessary RP1 and RP3-01 management & payload data. This scenario indicates to the necessity of including clarification of Tx and Rx FSM in the specifications.

6.3.2 Solution for Approach 2

After identifying the problems and their root cause, the solution was fairly straightforward. Firstly, the output of the RRU at the beginning of synchronization state was turned off completely using low level DSP commands from the CLI of the radio. This measure resulted in the expected scenario of BBU Tx transitioning to the expected state and starting to transmit IDLE bytes. Then the Tx and Rx FSM of the RRU was again set to Auto mode manually to observe whether the FSMs work as intended. After confirming that this procedure works, the solution was built into the firmware of the radio to make it automatic rather than a manual process.

Secondly, to solve the frame receiving problem, the listening address for frames in the radio was changed using CLI and the logger tool as well as the CLI of the radio was monitored. The process revealed that the change in listening address of the radio resulted in accurately detecting and receiving the frame formatted data from the BBU.

Sample outputs before and after the solution process are provided in Appendix B.

6.4 Discussion

Through the help of the analysis done in Chapter 5, this chapter explained the experiments done to integrate modules from two different vendors. Experiments from two different approaches has been presented in this chapter along with the observations and solutions to the problems. With the help of practical experiments, points revealed in the theoretical approach regarding the non-interoperability between OBSAI implementations were established.

Chapter 7

Proposal for Ensuring Better Interoperability

As discussed in previous chapters, ensuring interoperability is currently time-consuming and tedious. However, two approaches exist to work around this problem: modifying the specifications making them more concrete or generating a detailed integration plan to be followed by the OEM when integrating multi-vendor modules into a BS. Both solutions come with their own advantages and disadvantages. Modifying the standards to make them more concrete will render the standard inflexible and will diminish adaptability to different scenarios. On the other hand, requiring an extensive integration testing plan following every change (of any module) defeats the plug-n-play interoperability target of OBSAI. An intermediate solution would be to specify profiles of OBSAI-standard modules and categorize different options of implementations in separate profiles. This approach has the advantage of not completely loosing the flexibility that OBSAI offers while still offering some degree of plug-n-play provided that both modules support a specific profile. One approach to increase the probability of multiple modules following the same profile is to accept the market-leading versions of OBSAI implementations as references and create profiles according to them. Thus, the new providers can follow those profiles and ensure compatibility.

7.1 OBSAI Profiles

A profile document is a subset of the RP3 specification as stated in [8]. The goal behind the creation of such document is to accelerate the integration between the baseband block and the RF block or remote RF block. Time-consuming integration and interoperability testing, a consequence of flexibility of the RP3 specification, has received the attention of OBSAI forum members who proposed creating profile documents to alleviate these problems. Although the profile documents are not yet public, effort is going on inside OBSAI community to make them public as soon as possible.

7.1.1 Advantages of profiles

As discussed in earlier chapters, many properties and functions of OBSAI RP3 and RP3-01 are flexible with many options to choose from. Profile documents try to reduce this flexibility by assigning only one value to a particular property. There can be various profiles corresponding to a particular vendor or class of vendors. This type of profiling will allow suppliers to know exactly what to expect from a particular implementation of RP3 interface. This knowledge would accelerate the integration process by reducing the chance of mismatch in parameter values provided there will be compatible profiles in practice.

7.1.2 Impact of profiling

Creation of profiles instead of introducing rigidity in the original specification of RP3 will generate several adverse effects. In case of vendor-specific profiles, disagreement among vendors regarding the values and functions will lead to a creation of different profiles per vendor. This scenario will in turn force sub-module vendors to maintain separate versions of firmware for separate vendors according to their own profile, thus diminishing the cost-savings goal of OBSAI.

7.2 Standards Amendment Proposal

This section discusses about various proposals directed to the standards body for ensuring better interoperability in OBSAI architecture. Should

the proposals be accepted by the body, they will require further detailed experiments and validation to be passed as modifications to the standards.

7.2.1 RP3 Synchronization and messages

As discussed in Chapter 5, the RP3 interface has several areas where the standards specification document can be improved. Some of the most important areas of improvement are discussed in this section.

- **Buffering Requirement:** As the timing requirements of RP3 messages are very precise and the synchronization of the equipments are dependent on the frame bursts which are sent over the RP3 link, variance in fiber lengths as well as delay in internal processing needs to be compensated for. Although the standard proposes algorithms for compensating for internal delays by using δ and π values, it does not discuss how to compensate for fiber length delay variances. Because of the fact that newer wireless standards supports much greater distance between the BBM and the RRU, this variance of delay is becoming much more significant. For example, in NSN's FARADAY antenna interface at a 11.2MSPS sample rate, the air frame tick is $1/11.2 * 10^6 = 89.29ns$. If we have a 70 sample deep buffer at the RRU, the total buffering time will be $70 * 89.29ns = 6.25\mu s$. The speed of light in multimode fiber (with a diameter of 850nm) is $2.02 * 10^8 m/s$, and in single mode fiber (with a diameter of 1310nm) it is $2.04 * 10^8 m/s$. Therefore, on an average the propagation velocity of a signal is about $1/2.0 * 10^8 m/s = 5ns/m$. That means, with a 70 samples deep buffer we can only compensate for $6.25 * 10^{-6}/5.0 * 10^{-9} = 1250$ meters of fiber length difference. To compensate for a 5km long fiber difference, which is expected to be commonplace in future standards, the delay would be $25\mu S$ thus requiring $25 * 10^{-6}/89.29 * 10^{-9} = 280$ samples deep buffer. These requirements are not currently in the standards and it is recommended that the specification include a clear guideline for this.

  Alternatively, the timing synchronization can be done in the RRH, thus avoiding the compensation requirements from BB. This will allow asynchronous communication to the RRH leading to a more flexible solution.

- **IQ data alignment:** There can be scenarios where the amount of data for the antenna stream is not sufficient to fill up the slots in the frame.

In this case, the other side of the RP3 interface needs to know exactly how the samples are populated in the slots. The knowledge of whether the samples have been aligned to the MSB or not is very important to know in order to calculate the power level and other measurements that depend upon the samples. The specification does not state any particular rule for this situation which needs to be clarified.

- **Interaction between Tx and Rx state machines:** The Tx and Rx state machines in a given piece of equipment such as the RRU are dependent on each other. They also depend upon the state of other FSMs (in particular the ones they are communicating with). Although the state changing conditions and scenarios of Tx and Rx FSMs are relatively clear from the standards document, the complete scenario and interaction between the FSMs in the slave RP3 interface or the master RP3 interface is not clear and the specification would be more unambiguous if a brief description would be appended to the standard specifying the interaction between them.

7.2.2   RP1 Management Plane

The RP1 management plane specification is quite well defined. Most of the messages necessary for a BS to be up and functioning are already specified; including all the message dialogs and values. However, a basic problem exists in terms of the choice of management message carrier in the application layer. The OBSAI working group decided to use SOAP/XML for transporting the messages. However, SOAP messages tend to be very large and heavyweight. Consequently, to transport even a small message of 5 or 6 bytes, a large SOAP message structure is required. This adds overhead due to the bulkiness of SOAP messages and has been rather unpopular among developers. Moreover, the SOAP messages are carried over a custom made transport layer protocol named UDPCP, which is a modified version of UDP. This use of custom-made protocols has the inevitable negative effect on increasing the development time of the RP1 management plane. I propose two improvements to the OBSAI RP1 specification to make it more lightweight and easy-to-implement. These are described in the next paragraphs. However, it is to be noted that these improvements are proposed just to improve the performance of the OBSAI standards.

7.2.2.1 SNMP-based M-Plane

Simple Network Management Protocol (SNMP) is the de-facto management protocol for IP networks. SNMP is extremely simple and efficient and has been used for many years in different types of equipments ranging from routers to handheld devices. Presently, SNMP agents are built-in inside consumer devices as well to remotely manage them. An in-depth review of SNMP protocol is out of scope of this thesis, however, the official website [63] of SNMP lists several books that describe the protocol. A thorough analysis of an SNMP-based M-Plane is presented in Appendix A.

7.2.2.2 REST-based M-Plane

SOAP was primarily developed for web services development. However, the community has sought an alternative which is more intuitive and lightweight. Representational State Transfer (REST) [64, 65] allows performing similar tasks as SOAP simply using HTTP methods. In REST, all the objects have an URI and they can be called and executed using HTTP calls. This makes REST an extremely lightweight and intuitive paradigm. For example, if the IP address of the radio is 192.168.1.2, then to send a virtual HW reset command to the radio one could generate a HTTP get command of the format:

```
http://192.168.1.2?command=VHWR
```

Any HTTP server listening to this GET command could easily call the necessary function for performing the virtual HW reset. Therefore, a management plane consisting of RESTful messages, rather than SOAP messages, would be much more efficient and lightweight.

7.2.2.3 UDPCP Timing Constraint

As defined in Appendix A of [33], UDPCP imposes a timing constraint on the communication between two RP3 interfaces. After this timeout period the message is rendered invalid and dropped. According to the specification this timing should be configurable. However, the correct delivery of RP3 and RP3-01 messages depends on this timing and therefore it would be catastrophic if this parameter does not match between the BBU and RRU. If the parameter does not match, then the communication channel will break and the control system will not be able to properly configure the

RRU. For these reasons, it would be better if the specification or the profiles state the values of these parameters for UDPCP. Otherwise, a method of negotiating the parameters for UDPCP should be established between the control module and the other modules.

7.2.2.4 Supported-RP1-Message Negotiation

As per the RP1 specification, many of the RP1 messages are optional. This results in one side of communication generating a lot of RP1 messages which are completely disregarded in the other module, wasting a large amount of bandwidth of the RP3 interface. A better idea would be to have a negotiation phase at the beginning of the communication phase where both ends would send the set of supported RP1 messages to the other, thus they can agree upon a common subset of the RP1 messages for communicating.

7.2.2.5 Reaction to Abnormal Startup

RP1 messages has been defined in [33] to control and configure various modules of the BS and there are also messages to report abnormal situations. However, the specification of what the RRU should do in case of abnormal startup and what should be its state are not very clear. More work needs to be carried out in the area of abnormal situation handling and improper startup of the devices to remove the uncertainty of the current specification.

7.3 Integration of LTE

LTE is already supported in OBSAI, this presents a new opportunity to reuse radios or even allow multi-use of a single radio. Although the RP3 protocol description has been developed with the aim of making it agnostic to the air-interface protocol, there are some areas that need attention while adapting to a new air-interface technology. Among these areas, the following are the most important:

- **Transmission rules:** Message transmission rules for control and data messages are provided by the bus manager. Rules for LTE messages need to be defined, as indicated at Section 4.4.3 in [8].

- **Antenna carrier:** An important aspect to consider while integrating a new air-interface standard is the rate of the antenna carrier. Because in contemporary technologies multiple antenna carriers are used to construct a data stream, it is necessary that the combination of all antenna carriers completely fill up the RP3 line rate in use. In case of LTE, this is the case for all the channel rates except for a 15 MHz wide band. Therefore, in a 15 MHz wide band a dual bit map is used to match the data rate with the line rate of RP3. The detailed rule is provided in Appendix F of [8].

- **Bytes per sample:** LTE antenna-carrier data is transferred over the RP3 link using the sample rate defined in Appendix F of [8]. No oversampling is used and each I&Q sample consists of 4 bytes. Four samples are mapped to the payload.

- **Buffering requirement:** Two levels of buffering is required in the RP3 interface. First level is required due to the application layer operation of storing and processing multiple parallel data streams from multiple antennas. In LTE, a double buffering concept similar to WCDMA is used (section 4.4.5 in [8]). A second level of buffering is required in case of differences in fiber length to the radios connected to the same baseband module. If one radio is located at a maximum distance away and another very close to the BB, then there needs to be buffering in the nearer radio to compensate for the delay of the longer fiber. This is not a standards issue and depends on the category of DSP used in the BB module. However, companies manufacturing radios should keep this phenomenon in mind.

- **Payload mapping:** Payload mapping deals with the way the payload is mapped into the RP3 frame. One important factor is whether the payload is mapped MSB first and whether the payload is shifted towards the MSB in case of a partially-filled frame.

Other than those issues mentioned above, no other changes should be necessary to integrate a new air-interface such as LTE to be carried over RP3. However, if there are any vendor-specific limitations in the radio or BB module, that will require further investigation and cannot be specified or predicted.

7.4 Summary

Based on the theoretical analysis and experiments done, this chapter proposes several ideas to leverage interoperability in OBSAI BSs. Extensions currently in review at the OBSAI standardization work-group such as profiles have been described. Amendments required to ensure interoperability were also proposed and detailed. The chapter ends with a brief description of how a new air interface protocol such as LTE can be integrated to an OBSAI BS.

Chapter 8

Competing Base Station Standards

Alongside OBSAI, which has been the center of discussion in this thesis, some other standardization efforts has also been under way for ensuring interoperability in a Base Station system. This chapter briefly discusses the prominent alternatives and draws conclusions about their relative advantages and disadvantages. The most important standardization effort other than OBSAI in this area is named the Common Public Radio Interface (CPRI) [66]. In addition, an interesting discussion about the trends in the telecommunication industry of developing new standards due to political reasons can be found in [67].

8.1 CPRI

CPRI is an initiative to define a publicly available specification that standardizes the protocol interface between the radio equipment control and the radio equipment. Similar to OBSAI, the goal of CPRI is to allow interoperability of equipment from different vendors. CPRI allows a distributed architecture of base stations where the radio equipment control is connected to remote radio heads via lossless fiber links that carry CPRI data. This reduces the cost for service providers because only the radio equipment needs to be situated in challenging locations, whereas the whole BS with the radio equipment control can be situated in a favorable position with easier access to power and better physical environment.

The CPRI specification is an open serial digital I/Q standard. It encapsulates

information from transport, connectivity and control, including user-plane data, control & management plane transport mechanisms, and means for synchronization. CPRI specifies layer 1 and layer 2 of the OSI protocol stack. Layer 1 supports an electrical interface as in a traditional base station as well as an optical interface for BSs with RRH. All data, such as vendor-specific data, user-plane data, and L1 inband protocols [68] are time multiplexed and transmitted via this interface. CPRI does not specify a mandatory PHY layer protocol, rather it simply imposes a strict BER specification, as well as clock stability, and noise requirements. Thus it allows the realization of the interface as Gigabit Ethernet; 10-Gigabit Ethernet; Fiber channel; or Infiniband, avoiding the need to do its own synchronous communication system design. An overview of CPRI is depicted in Figure 8.1.

By defining only the PHY and link layer of the stack, CPRI maintains a very narrow focus and leaves more areas for flexibility. On the other hand, this narrow focus can conflict with the original focus of an inter-operable standard. Due to the fact that the message structure starting with Layer 3 can be anything and also the application layer message format can be completely proprietary, it may take much more time to properly integrate a third party radio equipment with a OEM radio equipment controller.

Figure 8.1: The CPRI digital interface
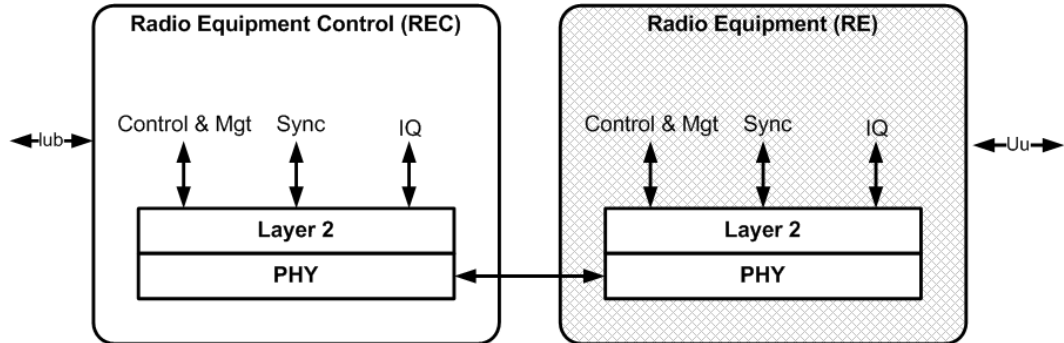
8.2   Proprietary Solutions

Prior to these standardization initiatives, all BS equipment as well as many of the interfaces were proprietary (many base stations used standard interfaces such as PCI to interconnect board level modules). The protocols for communication between the modules were also proprietary. These solutions were often completely company internal and never published.

8.3 Comparative Analysis

Although both CPRI and OBSAI have a similar goal of ensuring interoperability in an environment with multiple-vendors equipment, the stark difference between these two initiatives is their range of focus. While CPRI focuses on a much narrower field and standardizes only the PHY and link layer communication protocol, OBSAI takes a different approach and standardized the whole protocol stack. OBSAI also specifies four standard blocks of the BS while CPRI only specifies radio equipment control and radio equipment. Thus, CPRI is much less complex than OBSAI and requires less time to develop and apply. This property of CPRI allowed a much faster penetration in the market compared to OBSAI. Additionally, by allowing the usage of any established physical link such as the Gigabit ethernet, 10-gigabit ethernet, or fiber channel CPRI avoids the need to do its own synchronous communication system design [69].

However, OBSAI has the advantage of specifying a complete framework for the communication between all the BS modules. Thus, it could allow true interoperability between correctly implemented interfaces. While CPRI got a head start due to its low implementation time, OBSAI may have the upper hand in the long run due to very low integration and interoperability test. However, this is only true if compatible profiles emerge and vendors avoid proprietary extensions/profiles. In the future world of modular BS architecture, where modules from different vendors will be assembled in real time, a framework such as OBSAI would be necessary to make a true plug-and-play environment possible. CPRI, to be more specific, is only comparable to the OBSAI RP3-01 interface. Table 8.1 shows primary comparisons between OBSAI RP3-01 and CPRI.

Proprietary solutions always have the advantage of low-latency and high-performance. As the whole system is developed in a closed environment by a dedicated group of engineers, development of a highly customized and efficient system is possible. On the other hand, the proprietary nature of the interfaces hinders the use of any other vendor's equipment in the network, resulting in a permanent dependency of the service provider on a single OEM. Moreover, the amount of research and development effort that every OEM has to spend individually makes the cost of proprietary systems extremely high.

8.4   Summary

This chapter provides a brief summary of the competing base station architecture technologies.   Additionally, a table indicating the major differences between the standards is also provided.   This chapter should help the reader to decide which standard to follow.

Table 8.1: Comparison between OBSAI RP3-01 and CPRI

Features	OBSAI RP3-01	CPRI
User Data (I&Q)	80%	93.75%
Control Data (O&M)	4%	6.23%
Synchronization (K-Char)	0.25%	0.03%
Fixed Overheads	15.75%	0%
Physical Layer	XAUI upto 6.144 Gbps. 8b10b coding/scrambling.	XAUI upto 3.072 Gbps. 8b10b coding.
Data Link Layer	10ms Frames. RP3 Frame = i, N_MG, M_MG, K_MG.	10ms Frames. CPRI Frame = HF, BF, Word (Z.X.W).
Transport Layer	RP3 Address Filtering. RP3 Type Filtering.	Vendor Specific.
Application Layer	RP1 M-Plane (XML / SOAP). Index-Modulo/Dual BitMap Scheduling. RTT Measurements. RP1 Frame Clock Burst Synchronization.	Vendor Specific.
Carriers Capacity	Fixed IQ sample size.	Programmable IQ sample size.
RRH Network Design	RP3 Header enables easier Network configuration. RP1 Frame Clock Burst & RTT Mechanisms allows easy synchronization.	Network Configuration is Vendor specific. System Synchronization is Vendor specific.
Customization Efforts	RP3 Generic and Sync Control message allows easy customization.	Vendor Specific. Much higher customization effor required.
O&M Integration Effort	RP1 UDPCP/XML/SOAP integration requires around 2-3 man/month.	Vendor Specific. Much higher integration effor required.

Chapter 9

Conclusion

Despite being basis for a for monopoly market and business, proprietary solutions rarely are as economically feasible as modular, inter-operable solutions. Producers tend to manufacture complex solutions as a proprietary system for greater flexibility in their own development and to hold onto their market share. However, the huge research and development effort involved in developing such a complex solution in-house almost always results in very high cost and a long production cycle. The business of Base Station manufacturing for wireless networks followed the proprietary business model due to the complexity of the solution. However, in a competitive yet quality-aware market as today, low cost with high quality is increasingly demanded by customers. The wireless world responded and developed standards (such as OBSAI) based modular solution for base stations so that any supplier can produce any part of the whole system and the system integrator simply integrates other modules bought from different vendors.

Although the standards specification was supposed to create a truly inter-operable environment, the reality is different. Still, rigorous compatibility testing needs to be performed before any two modules from separate vendors can start to work. This requirement defeats the goal of true interoperability and extends the time frame required for building a new system with new features.

9.1   Summary of the Work

This thesis analyzed the OBSAI specification through a real-world integration task of the two most important modules of a base station, namely the BBU and the RRU. Where the OBSAI RP3 implementations were different in two modules. In course of doing this, the thesis identified potential features of the standard that hinder true interoperability and found working solutions for the problematic features. These solutions, such as specifying the link layer FSMs more clearly; indicating the algorithm to decide proper fiber length; specifying the value of Δ to be used; and many more, were proposed as potential amendments to the standards themselves. In course of doing the integration task, it was found that two implementations of the OBSAI RP3 interface are highly unlikely to inter-operate as such in current situation. The experiments revealed several areas that can restrict this interoperability. These, along with the solutions to overcome these problems, are detailed in Chapter 6. The thesis also proposes various tools and methods including a compatibility matrix to make the integration work faster and easier. This tool was developed based on the analysis done on Chapter 5. With the theoretical and practical experimentation methods proposed, integration of two different systems should be possible in a smooth and easy manner. Additionally, if the proposals for amending the standards specification are approved, the interoperability features of OBSAI would be much more rigid making it more likely that devices will be able to inter-operate.

As an extension to ensure interoperability among different modules, the thesis also proposes several improvement to the OBSAI standards document including the use of an SNMP-based management plane, OBSAI profiles, REST-based management plane, and increased buffering requirements. A through analysis of the proposed schemes as opposed to the current schemes is also done in this thesis. The thesis should offer some insight to further improvements of OBSAI standard to ensure better acceptance and smoother functionality.

9.2   Future Work

Due to the unavailability of a management plane from any other vendor, it was not possible to perform a practical interoperability test for the RP1 interface. Therefore, the analysis of the RP1 interface was limited to a theoretical study in this thesis. When a complete RP1 implementation will become available, a rigorous integration experiment on the application layer

of the RP1 interface should be performed to complete this work. If the proposed amendments are accepted as drafts by the standards body, then a full implementation and interoperability test of those features are also necessary.

In addition, an extensive analysis of the approach taken by the standardization body to develop the specifications should also be performed to validate the rigidity and future-proofness of the standards. A thorough study from a more theoretical point of view should establish the necessary groundwork for such a work, which will allow future similar efforts to be more organized and robust.

Development of a formal method that lays the theoretic background of such standardization efforts is considered by the author to be one of the most important future works. This kind of study will help the build a foundation that will ensure the interoperability of future standards and thus reduce the time required for performing such interoperability tests.

Bibliography

[1] Adam Wright. Broadband internet access peaking - wireless connectivity to drive next phase of global internet usage. `http://www.ipsos-na.com/news/pressrelease.cfm?id=3443`, April 2007. Last accessed, June 2, 2009.

[2] IEEE802. IEEE 802.16: Broadband wireless metropolitan area netwrok (wirelessMAN). `http://standards.ieee.org/getieee802/802.16.html`, 2005. Last Accessed, January 25, 2009.

[3] 3rd Generation Partnership Project. Long term evolution. `http://www.3gpp.org/Highlights/LTE/LTE.htm`, 2009. Last accessed, January 20, 2009.

[4] Ericsson. Wideband code division multiple access. `http://www.ericsson.com/technology/tech_articles/WCDMA.shtml`, 2009. Last Accessed, January 20, 2009.

[5] Yike Liu. WCDMA test automation workflow analysis and implementation. Master's thesis, Royal Institute of Technology, 2009. Available at: `http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/090423-Yike_Liu-with-cover-a.pdf`.

[6] B. Tezcan and B. Beane. Modular baseband design-enabling a low cost, reusable wireless infrastructure. *ELECTRONICS WORLD-SUTTON THEN CHEAM-*, 1842:22–27, 2006.

[7] OBSAI. About obsai. `http://www.obsai.com/obsai/obsai/about`, 2009. Last accessed, January 15, 2009.

[8] OBSAI. Reference point 3 specification v4.1. `http://www.obsai.com/obsai/documents/public_documents`, 2009. Last accessed, January 20, 2009.

[9] Altera. OBSAI: Open base station architecture initiative, altera corporation. `http://www.altera.com/technology/high_speed/protocols/obsai/pro-obsai.html`, 2009. Last accessed, June 2, 2009.

[10] ITU. Market information and statistics. `http://www.itu.int/ITU-D/ict/statistics`, 2008. Last accessed, June 2, 2009.

[11] ITU. Internet usage statistics. `http://www.internetworldstats.com/stats.htm`, 2009. Last Accessed, Jan 22, 2009.

[12] IEEE802. Part 16: Air interface for fixed broadband wireless access systems, October 2004.

[13] IEEE802. Part 16: Air interface for fixed and mobile broadband wireless access systems–amendment for physical and medium access control layers for combined fixed and mobile operation in licensed land, December 2005.

[14] J.J. van de Beek, P. dling, S.K. Wilson, and P.O. Brjesson. Orthogonal frequency-division multiplexing (OFDM). `http://www.s3.kth.se/signal/grad/OFDM/URSIOFDM9808.htm`, 2002. Last accessed, February 24, 2009.

[15] Ole Grondalen, GianFranco Vezzani, Silvia Restivo, Michael Schmidt, Isabelle Tardy, Patrizia Testa, and Rune Gronnevik. Time division duplex – flexible and efficient for millimeter broadband access systems. In *Proceedings of International workshop on broadband fixed wireless access*. Information Society Technologies, October 2002.

[16] A. Goldsmith. Adaptive modulation and coding for fading channels. In *Proceedings of the IEEE Information Theory and Communications Workshop*, pages 24–26, 1999.

[17] Argos. GSM frequency division duplex. `http://www.argospress.com/Resources/gsm/gsmfrequedivisiduplex.htm`, 2004. Last accessed, June 2, 2009.

[18] S. Alamouti, E.F. Casas, M. Hirano, E. Hoole, M. Jesse, D.G. Michelson, P. Poon, G.J. Veintimilla, H. Zhang, et al. Method for frequency division duplex communications, November 2008. US Patent 7,450,542.

[19] Hikmet Sari. Orthogonal frequency-division multiple access with frequency hopping and diversity. *Multi-carrier spread-spectrum*, pages 57–68, 1997.

[20] S. Lin, D. Costello, and M. Miller. Automatic-repeat-request error-control schemes. *IEEE Communications Magazine*, 22(12):5–17, 1984.

[21] ITS. Intersymbol interference. `http://www.its.bldrdoc.gov/fs-1037/dir-019/_2849.htm`, 1996. Last accessed, January 25, 2009.

[22] Jean-Paul Linnartz. Delay spread. `http://www.wirelesscommunication.nl/reference/chaptr03/fading/delayspr.htm`, 1995. Last accessed, June 2, 2009.

[23] A. W. M. van den Enden and N. A. M. Verhoeckx. *Discrete-time signal processing: an introduction.* Prentice Hall International, 1989.

[24] A. V. Oppenheim and R. W. Schaffer. *Discrete -time signal processing.* Prentice-Hall International, 1989.

[25] S.B.Weinstein and P.M.Ebert. Data transmission by frequency-division multiplexing using the discrete fourier transform. *IEEE Tansactions of Communication Technology*, com-19(5):628–634, October 1971.

[26] Jeffrey G Andrews, Arunabha Ghosh, and Rias Muhamed. *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*, chapter 4, pages 119–121. Communications Engineering and Emerging Technologies Series. Prentice Hall, 2007.

[27] A J Goldsmith and S G Chua. Variable-rate variable-power MQAM for fading channels. *IEEE Transaction on Communications*, 45:1218–1230, October 1997.

[28] W. T. Webb and R. Steele. Variable rate QAM for mobile radio. *IEEE Transaction on Communications*, 43:2223–2230, July 1995.

[29] H. Matsuoka, S. Sampei, N. Morinaga, and Y. Kamio. Adaptive modulation system with variable coding rate concatenated code for high quality multi-media communication systems. In *IEEE Vehicular technology conference*, pages 487–491, 1996.

[30] Runcom. RNU2000AP - BS, AP & Femto-Cell. `http://www.runcom.com/RNU2000AP`, 2009. Last accessed, June 3, 2009.

[31] T.M. Pinkston and J. Shin. Trends toward on-chip networked microsystems. *International Journal of High Performance Computing and Networking*, 3(1):3–18, 2005.

[32] OBSAI. OBSAI BTS system reference document v2.0. `http://www.obsai.com/obsai/documents/public_documents`, 2006. Last accessed, January 20, 2009.

[33] OBSAI. Reference point 1 specification v2.1. `http://www.obsai.com/obsai/documents/public_documents`, 2008. Last accessed, January 20, 2009.

[34] OBSAI. *Reference Point 2 Specification v2.1*, 2008. Last accessed, January 20, 2009.

[35] OBSAI. Transport module (tm) specification v1.0. `http://www.obsai.com/obsai/documents/public_documents`, 2004. Last accessed, 20 January, 2009.

[36] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474 (Proposed Standard), December 1998. Updated by RFCs 3168, 3260.

[37] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), December 1998. Updated by RFC 3260.

[38] K. Nichols and B. Carpenter. Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification. RFC 3086 (Informational), April 2001.

[39] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597 (Proposed Standard), jun 1999. Updated by RFC 3260.

[40] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC 2598 (Proposed Standard), June 1999. Obsoleted by RFC 3246.

[41] D. Black, S. Brim, B. Carpenter, and F. Le Faucheur. Per Hop Behavior Identification Codes. RFC 3140 (Proposed Standard), June 2001.

[42] OBSAI. Clock and control module (ccm) specification v1.0. `http://www.obsai.com/obsai/documents/public_documents`, 2004. Last accessed, 20 January, 2009.

[43] OBSAI. Baseband module(BBM) specifications v1.0. `http://www.obsai.com/obsai/documents/public_documents`, 2004. Last accessed, 20 January, 2009.

[44] OBSAI. Rf module(RFM) specification v1.01. `http://www.obsai.com/obsai/documents/public_documents`, 2006. Last accessed, 20 January, 2009.

[45] OBSAI. Reference point 1 specification v2.1–appendix a. `http://www.obsai.com/obsai/documents/public_documents`, 2008. Last accessed, January 20, 2009.

[46] OBSAI. Reference point 1 specification v2.1–appendix b. `http://www.obsai.com/obsai/documents/public_documents`, 2008. Last accessed, January 20, 2009.

[47] OBSAI. Reference point 1 specification v2.1–appendix c. `http://www.obsai.com/obsai/documents/public_documents`, 2008. Last accessed, January 20, 2009.

[48] OBSAI. Reference point 1 specification v2.1–appendix d. `http://www.obsai.com/obsai/documents/public_documents`, 2008. Last accessed, January 20, 2009.

[49] OBSAI. Reference point 1 specification v2.1–appendix e. `http://www.obsai.com/obsai/documents/public_documents`, 2008. Last accessed, January 20, 2009.

[50] IEEE802. IEEE CSMA/CD access method. `http://standards.ieee.org/getieee802/802.16.html`, 2005. Last accessed, January 20, 2009.

[51] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), dec 1998. Updated by RFC 5095.

[52] J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980.

[53] S. Hanks, T. Li, D. Farinacci, and P. Traina. Generic Routing Encapsulation (GRE). RFC 1701 (Informational), October 1994.

[54] Z.M. Wu, S.J. Ren, and X. Liu. Design of 3 G repeater based on OBSAI standard. *Dianxun Jishu/ Telecommunications Engineering*, 47(2):79–82, 2007.

[55] Christian Lanzani. Obsai rp3-01 6.144 gbps interface implementation. In *Proceedings of 4th FPGAworld Conference*, pages 1–6, 2007. `http://fpgaworld.com/conference/`.

[56] OBSAI. *Reference Point 4 Specification v1.1*, 2006. Last accessed, January 20, 2009.

[57] W3C. SOAP version 1.2 part 0: Primer (second edition). `http://www.w3.org/TR/2007/REC-soap12-part0-20070427/`, 2007. Last accessed, June 3, 2009.

[58] IEEE. Standard for information technology–local and metropolitan area networks–part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications–media access control (MAC) parameters, physical layer, and management parameters for 10 Gb/s operation, 2002.

[59] Optical Internetworkin Forum. Common electrical I/O (CEI) – electrical and jitter interoperability agreements for 6G+ bps and 11G+ bps I/O. `http://www.oiforum.com/public/documents/OIF_CEI_02.0.pdf`, 2005. Last accessed, February 9, 2009.

[60] RapidIO. RapidIO part6: Lp-serial physical layer specification rev. 2. `http://www.rapidio.org/specs/current`, 2005. Last accessed, June 4, 2009.

[61] PLD Applications. Gigabit 8b10b core user's guide. `http://www.compumodules.com/pdf/plda-pdf/gigabit8b10b_ug.pdf`, August 2001. Last Accessed, June 3, 2009.

[62] OBSAI. Conformance test specifications. `http://www.obsai.com/obsai/documents/public_documents/download_specifications/conformance_test_specifications`, 2007. Last accessed, June 5, 2009.

[63] SNMP Research International. Snmp print literature. `http://www.snmp.com/protocol/snmpbooks.shtml`, 2009. Last accessed, June 4, 2009.

[64] R.T. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, 2000. Available at: `http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm`.

[65] Roy T. Fielding and Richard N. Taylor. Principled design of the modern web architecture. *ACM Transactions on Internet Technology*, 2(2):115–150, 2002.

[66] CPRI. CPRI specification overview. `http://www.cpri.info/spec.html`, 2008. Last accessed, June 3, 2009.

[67] Lee Pucker. Does the wireless industry really need all these digital if standards? *IEEE Communications Magazine*, 43(3):54–57, 2005.

[68] Ericsson, Huawei, NEC, Nortel, and Siemens. CPRI specification v2.0. `http://www.cpri.info/downloads/CPRI_Specification_V_2_0.pdf`, 2004. Last accessed, June 4, 2009.

[69] Christian Plante and Jason Wong. Opening Base Station Architectures Part 2: An Inside Look at CPRI. `http://www.commsdesign.com/design_corner/showArticle.jhtml?articleID=50500166`, October 2004. Last Accessed, June 10, 2009.

Appendix A

SNMP-based Management Plane

A.1   Motivation

Simple Network Management Protocol (SNMP) is a simple set of operations and information that allows administrators to manage a networked device. SNMP was primarily developed by IETF and has been used in all networked devices such as routers, switches and hubs. With the development of further sophisticated features of SNMP, it is now used in all types of devices including storage devices, and electronic power suppliers. Due to SNMP's long history of proven success; amount of research effort; and extremely efficient nature, this protocol has been the primary choice for any management related solution. On the other hand, the unnecessary amount of information involved in SOAP messages has made it an unattractive choice for sending small management commands. Therefore, it is worth assessing the relative advantages of SNMP if used in management plane.

A.2   Introduction to SNMP

Although a rigorous overview of the protocol is out of the scope of this document, a brief introduction is necessary to understand the following discussion. SNMP is based on request-response architecture and there are two types of entities involved, namely: the manager and the agent. A manager is typically a server running management software and capable of

managing other devices. On the other hand, an agent is a piece of software that is located in the device which needs to be managed. For communication between them, three types of messages are used: (1) Request, (2) Response, and (3) Trap. The manager can request any information from the agent and after analyzing that can send a response to the agent to reconfigure it if necessary. In case of any emergency, the agent can send a trap, which is synonymous to an alarm, to the manager. The situation is shown in Figure A.1.
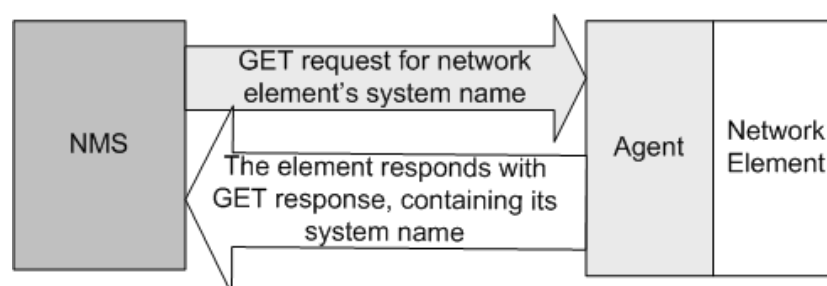

Figure A.1: SNMP Messages between a manager and an agent

To define the managed objects and their behavior, a special type of information called Structure of Management Information (SMI) is used. Additionally, each object keeps a database of all the object that it tracks. This database is called Management Information Base (MIB). There are standard MIBs defined by standards body and it is also possible to define custom MIB for a particular equipment. An MIB is structured in ASN.1 format, which is a tree-like structure, and the objects inside the MIB is expressed using Object Identifiers (OID). For a functioning SNMP environment, a complete MIB and agents supporting that MIB is necessary. Thus, manager asking for any particular information from a branch of the MIB structure will receive a valid response from the agent.

A.3   SNMP Transactions for OBSAI

To be able to manage the OBSAI modules using SNMP, it is necessary to develop a custom branch of MIB for OBSAI and develop an agent to recognize the objects in the MIB as well as respond to the events related to them.

A.3.1   Custom MIB for OBSAI

Because standard MIBs were created for IP networks, they do not contain any required properties of OBSAI environment. This situation necessitates the development of a custom branch of MIB with OBSAI objects. Figure A.2 depicts a sample MIB ASN.1 tree with some of the OBSAI objects as branches. This is just an indication to the way a full tree should be constructed and in no way a complete implementation. The corresponding MIB file is represented in Figure A.3.
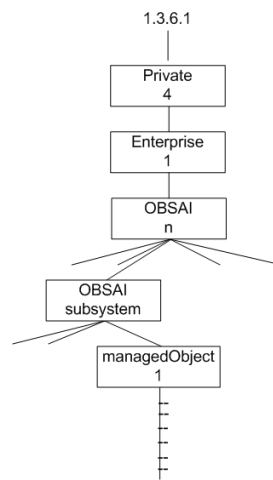
Figure A.2: ASN.1 structure of an example OBSAI MIB

A.3.2   Example M-Plane transaction using SNMP

To replicate the operation of modifying a resource state in an OBSAI subsystem as shown in Section D.4.2 of [33] using SNMP, we assume that there exists an SNMP agent in the subsystem which has the knowledge of OBSAI MIB similar to Figure A.3. Therefore, when the manager decides to change the state (Let us assume, OID: 1.3.6.1.4.1.9.3.2) of a resource in the subsystem, it issues an SNMP request to the agent of the form:

```
snmpset -c private 192.168.1.2 .1.3.6.1.4.1.9.3.2 i 0
```

This command results in an SNMP packet towards the agent. The packet would look like following in wireshark[1]:

---

[1]http://www.wireshark.org/

```
Simple Network Management Protocol
    Version: 1 (0)
    Community: private
    PDU type: SET (3)
    Request Id: 0x1df8e7e6
    Error Status: NO ERROR (0)
    Error Index: 0
    Object identifier 1: 1.3.6.1.4.1.9.3.2
    Value: INTEGER: 0
```

If the operation is successful, the manager will receive an SNMP response from the subsystem indicating that the operation was successful.

```
Simple Network Management Protocol
    Version: 1 (0)
    Community: private
    PDU type: RESPONSE (2)
    Request Id: 0x1df8e7e6
    Error Status: NO ERROR (0)
    Error Index: 0
    Object identifier 1: 1.3.6.1.4.1.9.3.2
    Value: INTEGER: 0
```

Because of the change of the state variable the SNMP agent of the subsystem will call the necessary function which changes the state of the resource according to the implementation of the agent. To replicate the alarm situations and scheduling in M-plane, SNMP traps are enough.

A.4 Comparison between SOAP, REST and SNMP

As already discussed, although more flexible, SOAP messages incur much more overhead on the wire than other approaches. In this section, a comparative analysis among SOAP, REST and SNMP is done from various point of view. Depending on the analysis, it would be possible for the standards body to decide on which approach to take for carrying RP1 messages.

```
managedObjectTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF ManagedObjectEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "This table represents the objects of
                 an OBSAI sub-system"
        ::= { obsaisubsystem 1 }
managedObjectEntry OBJECT-TYPE
        SYNTAX  ManagedObjectEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION     "A row in the managedObject table"
        INDEX   {managedObjectIndex}
        ::= { managedObjectTable 1 }
ManagedObjectEntry ::= SEQUENCE{
        objIndex        Unsigned32,      --table index
        objClass        Unsigned32,      --class of the object
        objDistName     DisplayString,
        objVendor       OBJECT IDENTIFIER,
        objState        INTEGER,         --state of the object
}
objIndex OBJECT-TYPE
        SYNTAX  Unsigned32
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A unique value for each interface."
        ::= { managedObjectEntry 1 }
objClass OBJECT-TYPE
        SYNTAX  Unsigned32
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A coded representation of the type of the object."
        ::= { managedObjectEntry 2 }
```

Figure A.3: Excerpt of example MIB file for OBSAI

A.4.1   Payload Size

From the payload size point of view, SOAP messages have the most bulky
structure of the three. To analyze the issue further, starting with an
example would be clearer. If we take one of the first messages to be
exchanged between the control module and others, the "ModuleReadyInd"

message, which indicates that the module in functional:

SOAP: The SOAP message for ModuleReadyInd message is as follows. As can be seen, It provides the identity of the module and its location along with the vendor name to indicate that its ready. However, the whole message is much more longer than the necessary information because of the extra tags and wrappers. In total, it requires 525 bytes, if we allocate one byte to each character.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.
                org/soap/envelope/">
        <SOAP-ENV:Header>
                <to>BTS_OM</to>
                <from></from>
                <id>1001</id>
                <action>OBSAI_CM</action>
                <version>2.0</version>
        </SOAP-ENV:Header>
        <SOAP-ENV:Body>
                <moduleReadyInd>
                        <moduleType>RFM</moduleType>
                        <vendor>SOME_VENDOR</vendor>
                        <productCode>ABC.1234</productCode>
                        <serialNum>M2345678</serialNum>
                        <subRack>11</subRack>
                        <slot>1</slot>
                        <bldVersion>0.10</bldVersion>
                        <restartCause>power_on</restartCause>
                        <hwSwId>2</hwSwId >
                </moduleReadyInd>
        </SOAP-ENV:Body>
    </SOAP-ENV:Envelope>
```

REST: To communicate the same information using REST over HTTP, a URI for the destination modules would be required. Using the URI, it is possible to send the information over HTTP as shown below.

```
http://someuri.somevendor/ModuleReadyInd?fromId=1001&
action=OBSAI_CM&moduleType=RFM&vendor=SOME_VENDOR&
productCode=ABC.1234&restartCause=power_on
```

As the information would be sent over HTTP GET request, therefore the parameters would be carried over a standard HTTP header and the

recipient with a working HTTP server can easily parse and use the parameter values. As can be seen from the request, RESTful approach takes much less bandwidth on wire than SOAP due to the lack of wrappers, and tags.

SNMP: As the ModuleReadyInd is a message which needs to be sent to the other modules voluntarily, SNMP TRAP or NOTIFICATION message is the best option for it. An example SNMP TRAP message performing similar function would look like the following on wire:

```
Simple Network Management Protocol
    Version: 1 (0)
    Community: public
    PDU type: TRAP-V1 (4)
    Enterprise: 1.3.6.1.4.1.9.3 (SNMPv2-SMI::enterprises.9.3)
    Agent address: 127.0.0.1 (127.0.0.1)
    Trap type: ENTERPRISE SPECIFIC (6)
    Specific trap type: 123456
    Timestamp: 9347423823
    Object identifier 1: 1.3.6.1.4.1.9.3.123
                (SNMPv2-SMI::enterprises.9.3.123)
    Value: STRING: "ModuleReadyInd: Module X"
```

The bandwidth for the SNMP messages is comparable with that of REST, but is far lower than that of SOAP.

A.4.2   Scalability

Although REST and SNMP provides much more lightweight way of communication and management, SOAP is the most flexible of the three. SOAP allows to create complex hierarchy of messages and information. It also allows to arrange the information in an intuitive and readable way. However, the XML processing involved in parsing and manipulating SOAP messages makes it a little less scalable than the others.

A.4.3   Performance

XML processing slows down the functioning of SOAP messages to a large extent. Although, many very optimized XML processors are available, they cannot compete with the efficiency and speed of SNMP agents. Therefore, performance-wise SNMP is superior to SOAP messages.

A.4.4 Security

RP3 interface does not have much security threat as it is now. However, it may be possible that with future development there will be imminent threat on this interface. SOAP does not have built-in security. Although, with the flexibility of SOAP, it is possible to negotiate and establish necessary security parameters for subsequent use. On the other hand, SNMP has built in security in the form of community strings. SNMPv3 has introduces more robust and security measures to ensure security of SNMP traffic.

A.4.5 Protocol Independence

RESTful messages, as it is at present, can only run over HTTP. On the other hand, SOAP messages do not rely on any particular lower layer protocol. This situation provides SOAP a bit more flexibility and independent over REST. In case of SNMP, it does not rely on any application layer protocol as it itself is an application layer protocol which runs on UDP. However, it is possible to adapt SNMP to run over other transport layer protocols such as UDPCP without much alteration.

Appendix B

Sample Test Output

B.1 RRH-BB integration: Before fixing the error

B.1.1 BB Logging tool

Before applying any patches to the RRH, the synchronization phase between the BB and the RRH generated the following log:

```
5546 01.01 02:31:15.158 192.168.255.1 34 WAM_10 <01.01 02:31:15.158> 10052 INF/HWA/OPTS,
Channel RP3-01 0 in UNSYNC state
5552 01.01 02:31:15.159 192.168.255.1 3a WAM_10 <01.01 02:31:15.159> 10052 INF/HWA/OPTS,
Channel RP3-01 0 in WAIT_FOR_K28.7_IDLE state
5553 01.01 02:31:15.159 192.168.255.1 3b WAM_10 <01.01 02:31:15.159> 10052 DBG/HWA/OPTS,
RX digital Reset done for Opt Link 1
5554 01.01 02:31:15.159 192.168.255.1 3c WAM_10 <01.01 02:31:15.159> 10052 DBG/HWA/OPTS,
OPT Link 1 RX changed to Idle state
5562 01.01 02:31:15.162 192.168.255.1 44 WAM_10 <01.01 02:31:15.162> 10052 DBG/HWA/OPTS,
API_OPT_SYNC_REQ_MSG received
5563 01.01 02:31:15.162 192.168.255.1 45 WAM_10 <01.01 02:31:15.162> 10052 DBG/HWA/OPTS,
Channel RP3-01 0 TX set off
5564 01.01 02:31:15.162 192.168.255.1 46 WAM_10 <01.01 02:31:15.162> 10052 DBG/HWA/OPTS,
Opt Link 1 TX changed to Off state
5568 01.01 02:31:15.163 192.168.255.1 4a WAM_10 <01.01 02:31:15.163> 10097 WRN/LGC/BBC,
OptSynchronizer: Wrong synchronization phase in API_OPT_SYNC_IND_MSG for optic link 1 :
EOptSyncPhase_Idle
5585 01.01 02:31:16.250 192.168.255.1 5b WAM_10 <01.01 02:31:16.250> 10052 INF/HWA/OPTS,
Channel RP3-01 0 in UNSYNC state
```

This output was produced repeatedly by the BB module and indicated that it cannot synchronize due to some unwanted signal from the radio.

B.1.2 RRH CLI

The output from the RRH CLI indicated that the RP3 interface is not in sync mode.

```
>> get obsaistatus

OBSAI Status Information
-----------------------

RP3 RX state: Unsync
RP3 TX state (Auto): Off
RP3 RX master frame offset: 0xfff
RP3 RX align delay (before 8b10b decoder): 8

RP3 RX resynchronization: 1
RP3 Virtual HW Reset msg received: 0
RP3 RX frequency alarm: 1
RP3 RX LCV errors: 255

RP3 control CRC errors: 15

RP3 RX buffer resynchronizition
   Carrier 1: 0
   Carrier 2: 0

RP3 mapping FIFO status
   Carrier 1 RX overflow  : 0
   Carrier 1 RX underflow : 1
   Carrier 1 TX overflow  : 0
   Carrier 1 TX underflow : 0
   Carrier 2 RX overflow  : 0
   Carrier 2 RX underflow : 1
   Carrier 2 TX overflow  : 0
   Carrier 2 TX underflow : 0

RP1 FCB
   RP1 system  : 0x0
   RP1 SFN MSB : 0
   RP1 SFN LSB : 0
   C1          : 0
```

B.2 RRH-BB integration: After fixing the errors

B.2.1 BB Logging tool

After the RRH listening address and the RP3 FSM were fixed using a software update, both the BB and the RRH functioned properly and reached FRAME_SYNC state. This can be observed from the BB logging tool output:

```
7538 01.01 02:03:09.497 192.168.255.1 cb WAM_10 <01.01 02:03:09.497> 10052 DBG/HWA/OPTS,
Channel RP3-01 0 TX set to idle
7758 01.01 02:03:10.470 192.168.255.1 a8 WAM_10 <01.01 02:03:10.470> 10052 DBG/HWA/OPTS,
SFP 1 RX OK!
7759 01.01 02:03:10.470 192.168.255.1 a9 WAM_10 <01.01 02:03:10.470> 10052 INF/HWA/OPTS,
SFP RX LOS disabled (channel: RP3-01 0)
7763 01.01 02:03:10.471 192.168.255.1 ad WAM_10 <01.01 02:03:10.471> 10052 INF/HWA/OPTS,
Channel RP3-01 0 in WAIT_FOR_K28.7_IDLE state
7797 01.01 02:03:11.075 192.168.255.1 cf WAM_10 <01.01 02:03:11.075> 10052 DBG/HWA/OPTS,
Channel RP3-01 0 TX set to frame
7798 01.01 02:03:11.075 192.168.255.1 d0 WAM_10 <01.01 02:03:11.075> 10052 DBG/HWA/OPTS,
Opt Link 1 TX changed to Frame state
7837 01.01 02:03:11.585 192.168.255.1 f7 WAM_10 <01.01 02:03:11.585> 10052 INF/HWA/OPTS,
Channel RP3-01 0 in WAIT_FOR_FRAME_SYNC_T state
7842 01.01 02:03:11.587 192.168.255.1 fc WAM_10 <01.01 02:03:11.587> 10052 INF/HWA/OPTS,
Channel RP3-01 0 in FRAME_SYNC state
7843 01.01 02:03:11.587 192.168.255.1 fd WAM_10 <01.01 02:03:11.587> 10052 WRN/HWA/OPTS,
Frame sync lost in RP301#1 cancelled.
```

B.2.2 RRH CLI

Similar results can be observed from the RRH CLI. The CLI also provides us with the indication that it is receiving FCB messages correctly after changing the listeing address for FCB messages from 0x20 to 0x00.

```
OBSAI Status Information
-----------------------

RP3 RX state: Unsync
RP3 TX state (Auto): Off
RP3 RX master frame offset: 0xfff
RP3 RX align delay (before 8b10b decoder): 8

RP3 RX resynchronization: 1
RP3 Virtual HW Reset msg received: 0
RP3 RX frequency alarm: 1
RP3 RX LCV errors: 255

RP3 control CRC errors: 15

RP3 RX buffer resynchronizition
   Carrier 1: 0
   Carrier 2: 0

RP3 mapping FIFO status
   Carrier 1 RX overflow  : 0
   Carrier 1 RX underflow : 1
   Carrier 1 TX overflow  : 0
   Carrier 1 TX underflow : 0
   Carrier 2 RX overflow  : 0
   Carrier 2 RX underflow : 1
   Carrier 2 TX overflow  : 0
   Carrier 2 TX underflow : 0

RP1 FCB
   RP1 system  : 0x0
   RP1 SFN MSB : 0
   RP1 SFN LSB : 0
   C1          : 0
```