

Generic Design Framework for an Efficient Wireless MAC Data Path

1.0 EXECUTIVE SUMMARY

Recent years have seen a phenomenal growth in demand for residential broadband. With the advent of small form-factor high performance integrated SoCs (see, for example, the TI) and short area broadband wireless technologies (Wi-Fi and the upcoming WiMAX/LTE), the broadband access point or Base Station Router (BSR) is becoming a mass-market item. Coupled with the ongoing interest in 'femto-cells', it is likely that over the next five years, residential "access point" base-stations will to a significant percentage of the communication requirements. The cBSR can be defined as a base station which has limited capabilities in terms of user data throughput, number of subscribers and is typically based on off-the-shelf hardware. It is more suitable for small scale deployments like wall mount or SOHO networks. There are varied differences between the multiple wireless technologies that currently exist or are developing in terms of protocol procedures, radio resource management the frequencies they operate and the underlying radio access technologies they use. However, all of them are focused on one primary goal of providing high speed broadband access to users, using standardized, low-cost hardware platforms and off-the-shelf software. In all of these products, the engineering challenge is to marry packet processing capabilities of standard desktop class operating systems, to the special requirements of the SoC servicing the wireless interface. If the data path of MAC for all these technologies is analyzed, they have many functions that are in common and the design concepts to implement these functions can be used across any wireless technology like WiMAX or LTE or even older WiFi. Consequently, if we consider the BSR as a generic product (with specialized interfaces for wireless access), then the common core functions can be formulated into a design framework which is suitable multiple wireless technologies.

In the rest of this document, we discuss the generic framework of the BSR, with special reference to the high-speed data-path and discuss architecture for realization of the same.

CONTENTS	
<u>SECTION</u>	<u>PAGE</u>
1.0 EXECUTIVE SUMMARY.....	1
2.0 OBJECTIVE	1
3.0 DESIGN OVERVIEW.....	2
3.1.1 Connection Database.....	2
3.1.2 Resource Allocator.....	3
3.1.3 Network Interface.....	3
3.1.4 Queue Management	3
3.1.5 Composition.....	4
3.1.6 Unification/Phy Handling	5
3.1.7 Triggering Criteria	5
4.0 WIMAX CASE STUDY	5
4.1.1 How we did it for WiMAX	2
4.1.2 Extension to LTE	3

2.0 OBJECTIVE

Generally any data link layer performs the following functions

- Medium Access Control
- Logical Link Control
- Flow Control
- Error Control
- Packetizing
- Addressing

The logical link control is the function performed by the data link layer to provide services to multiple kinds of network layers. In the wireless context they are termed as a Radio Link Control and typically perform identical operations independent of the MAC below them.

Similarly, the data part of any wireless layer 2 MAC performs the following generic function in addition to the many technology specific functions. Needless to say this makes each technology special in its own way. The wireless MAC functions mainly include

- Providing a service interface to higher network layer typically the native IP stack, Directly or via an RLC layer. The services are described below; the interface has to comply with the standard form expected by the IP stack.
- Queue Management for the IP packets to ensure Flow Control and QoS. The IP packets on delivery by the native IP packets are stored in the Layer 2 queues by the data link layer. The queues are arranged per connection which can be identified on basis per MSID or RAB ID. The queue depth is kept as a configurable item based on the QoS that has to be offered on a particular connection e.g. if a particular connection has to be offered a constant bit rate than the queue depth will be higher than for a connection which has to be offered a best effort bit rate. By means of this design the flow control information is given to the IP stack. Whenever the queue depth of a particular connection crosses its threshold an Alarm is raised to the IP stack to take a reactive action. The IP stack can use this alarm to stop pumping traffic on the wireless interface by putting a back pressure on the ingress side or buffering the packets within itself for some time or until the Alarm is cleared.
- Conversion of received IP packets (SDUs) into packets suitable for transmission over the wireless link (PDUs) and vice versa. In order to do this conversion efficiently, the segmentation, re-assembly and packing functions have to be implemented, based on agreed QoS and radio resources available for data to be transmitted for a particular context.
- PDU formation based on the kind of technology used for transmission as the organization of fragmentation/re-assembly sub headers may vary across technologies
- Provide a Generic API to plug-in Radio Resource Allocation

This paper proposes an efficient generic design framework for the medium access layer which can be used across wireless data technologies with minimal customizations to ensure high speeds and through

put. The framework proposed here is just a reference design for the cBSR class of products and may not be suitable for large multi sector carrier grade base stations with sophisticated product requirements like high availability

A sample snapshot of the cBSR system architecture is given in Figure 1 which gives the placement of the MAC data path framework in the overall system. It has been shown in yellow color

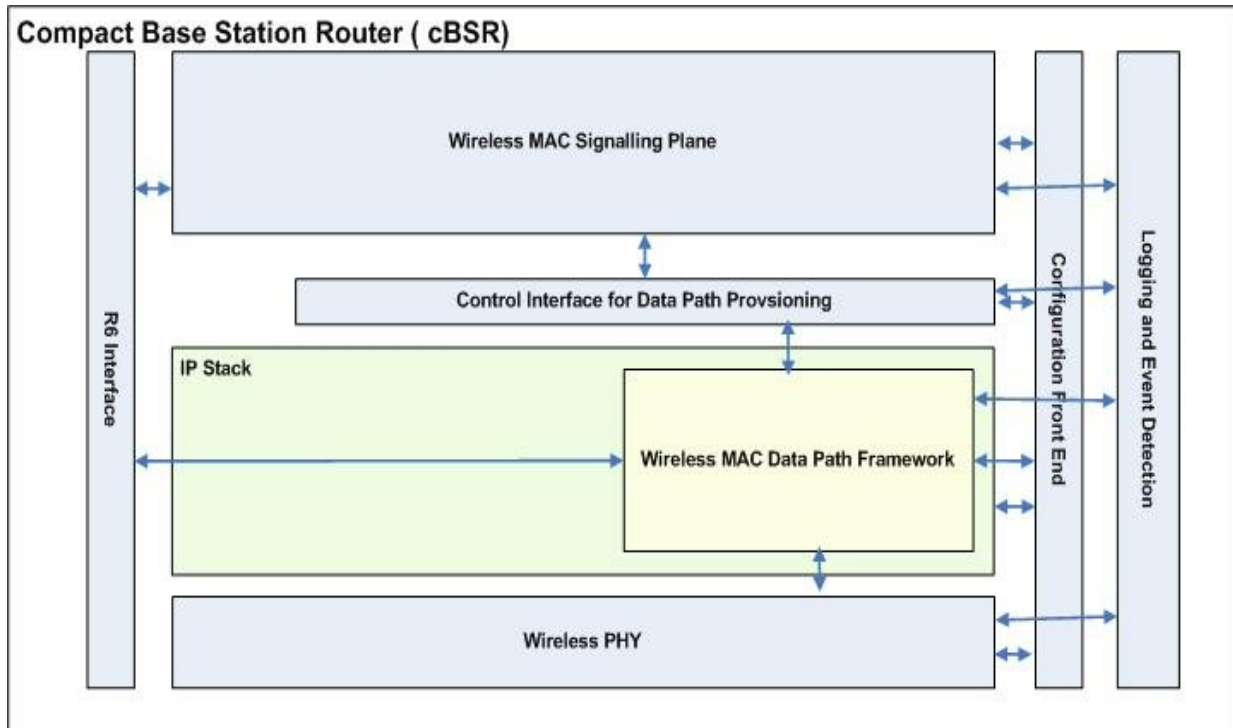


Figure 1 cBSR System Architecture

3.0 DESIGN OVERVIEW

The Figure 2 shows the architecture of a typical wireless data MAC

The design proposes a de-coupled control and data plane which makes it all the more wireless technology agnostic. This means that it is possible to plug in most of the wireless control stacks with this data path design framework with minimal customizations. The framework enables this by means of a clearly defined interface which can be used by any control stack implementation to populate the database. The parameters and the values to be populated are derived as a result of various control messages exchanges that take place for establishing the connection.

Based on the generic functions which have to be provided by the wireless MAC as mentioned in the Section 2.0 the following key elements have been identified

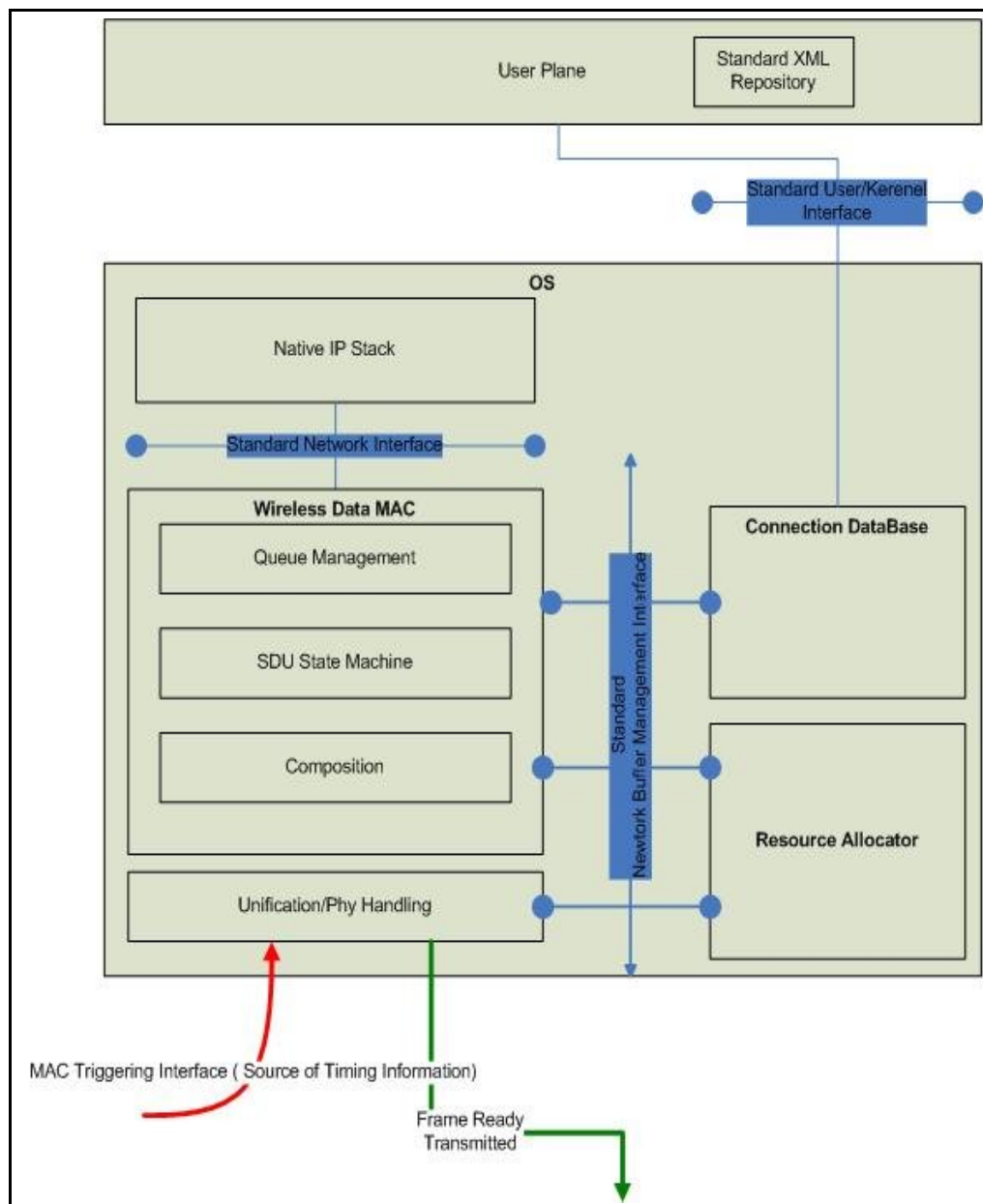


Figure 2 Design Framework

3.1.1 Connection Database

This database contains the connection setup information. This information is organized in a such a manner that it

- Is sufficient for the MAC to appropriately compose PDUs to be transmitted over a physical link based on the connection identifiers and the agreed QoS parameters for those connections. The connection identifiers could be anything like CID in case of WiMAX or RABID in case of 3G/LTE
- Can be populated by system control applications from user land by means of a well defined and standard interface for e.g. Netlink Sockets in Linux (Assuming Control Plane is in User Space and Data plane exists in the kernel) from either a configuration like organized in a standard CSV

or XML format or directly by Control Plane Protocol entities. The framework also provides a remote API so that the database can be populated by user land application which is not collocated with the framework i.e. not on the same machine

- Contains provision for access technology specific parameters, to be used by the data path and access hardware for transmission/reception. It is intended that all external control of the wireless SoC be implemented through this data-base, without needing any further interactions between the control plane and the hardware.

3.1.2 Resource Allocator

The Resource Allocator is responsible for

- Appropriate allocation of radio resources between various contenders based on various factors like the bandwidth requested by the subscriber or a flow or outstanding information pending to be transmitted on a particular connection, the agreed QoS for that connection, the link configuration and quality for that connection and the most importantly the available radio resources
- Scheduling these resources in time, frequency or both for them to be transmitted over the air. There could be various algorithms used for the scheduling the radio resources.

The method by which it does allocation of the physical radio resources; depends on the access Technique being used by a particular Radio technology. For e.g. If the Access Technique used is OFDM than the Resource Allocator procedures may be similar across all Radio technologies which use OFDM. The proposed design framework provides an API by means of which most kinds of resource allocator can be plugged into the data path stack with minimal changes.

The MAC Data path has the following Components:

3.1.3 Network Interface

Assuming the Network layer currently used is typically IPv4/IPv6 all the operating systems provide a standard well defined interface to plug the Wireless MAC from the existing network Layer. A pseudo network device driver interface can be used for achieving such functionality in case of Linux as well any BSD based Operating System. A standard interface like netfilter, iptables or by creating a pure static routing entry into the kernel routing table is sufficient to make all the IP PDUs travel through the Wireless MAC Layer.

Currently, in the Linux the wi-fi MAC layer is implemented very similarly. A network device is setup for each physical wifi interface available typically named a wifi0 and so on. When an IP packet is routed by the Linux IP stack it checks for the outgoing interface in its forwarding information base (FIB or the routing table). In case the outgoing interface is the a wifi interface it is the packet is handed over to the hard_start_xmit API which is initialized to the MAC processing API of the wireless device. Hence this kind of Ethernet like interface can be extended to any kind of wireless MAC for providing a neat and clearly defined API interface to make an entry into its MAC processing.

However, there could be some differences on how these entry point APIs are invoked. Currently, a transmit routine is invoked whenever an IP packet is available for delivery. However in case of a technology like WiMAX the transmit routine may be invoked as and when the driver indicates that downlink bandwidth is available for a part or the complete IP packet to be transmitted.

3.1.4 Queue Management

Once the IP PDUs reach the MAC layer enqueueing them for further processing based on the flow they belong to and what treatment should be given to each such flow becomes a key functionality to be exhibited by the MAC layer. Each queue may be organized based on per connection basis. This connection will have a unique numeric identifier, which will map to the CID in case of WiMAX, a RABID in case of 3G/LTE or an MS ID in case of GPRS; the data path is agnostic to the original source of the id.

Each flow may belong to a particular connection. The queue depth of each connection shall be based on the QoS parameters agreed for that particular connection, the outstanding bytes on that connection and the kind of flow control that has to be offered on the ingress side.

3.1.5 Composition

In the downlink side each SDU that has to be transmitted over the physical media may have to be segmented based on the number of bytes scheduled on the air. In case the number of bytes scheduled on the air is less than size of the next SDU available for transmission it shall do segmentation otherwise the MAC layer shall try to pack as many SDUs as can be accommodated in the number of bytes scheduled for transmission by the resource allocator. In the uplink side the re-assembly procedure shall be executed. The segmentation and re-assembly state machines are common across most of the technologies and can be reused with minimal or no changes.

The composition of the SDU's into an actual wireless data units that is transmitted OTA is the key role played by any wireless MAC and varies across multiple technologies. This component deals with physically constructing the payload and appropriately inserting the information in the form of special headers which indicate the type of fragment, the layer 2 addressing and sequence number in case some flow control or retransmission is supported by the MAC. The illustration in Figure 3 and Figure 4 gives example of composed SDUs for WiMAX and LTE

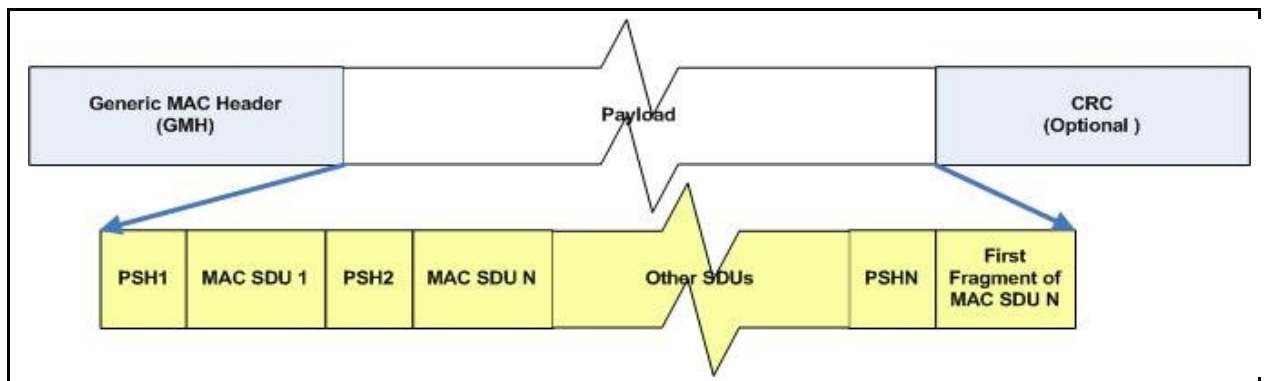


Figure 3 WiMAX MAC Data Unit

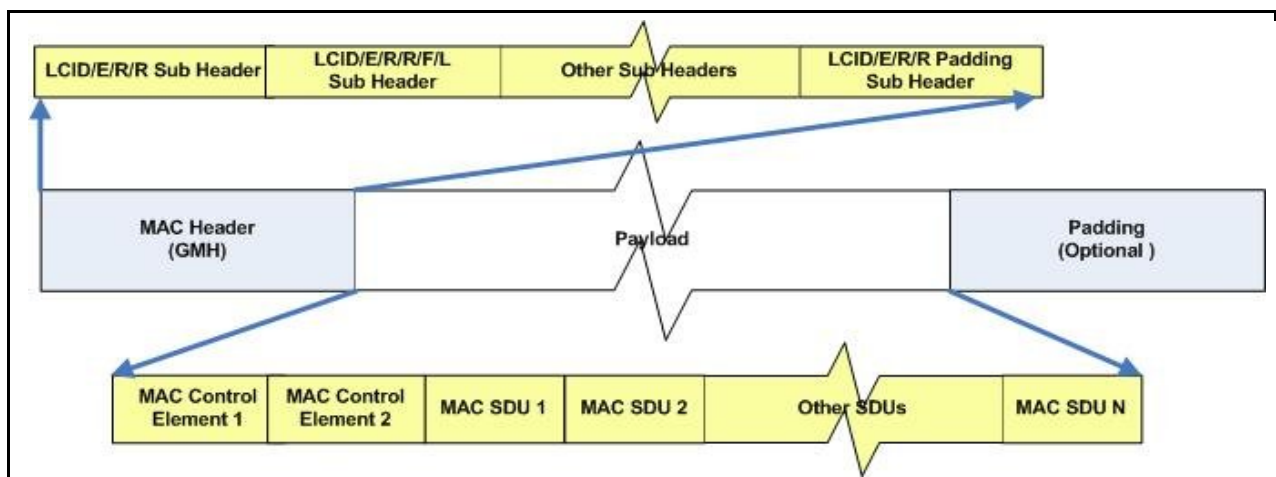


Figure 4 LTE MAC Data Unit

Across each of the components of the Wireless Data MAC the existing network buffer management of an Operating system can be used to get a buffer chain once the MAC processing is complete which may easily be used by the MAC PHY interface for transmission over the physical media. In case of Linux the socket buffer or skb's can be used and in case of BSD the mbuf buffer chain can be used. The benefits of using a standard buffer management scheme are

- Complete zero copy. The IP stack is designed in such a way that generally the traffic which is originated from a network node or is en-route from the network traverses the multiple layers. Consider the example where the IP packet is en-route a particular node. In this case it is copied from the buffer rings of the device to a standard buffer chain and then the same buffer chain is passed across all the layers i.e. layer 2 mac processing, layer 3 forwarding/routing , layer 2 processing again and then finally transmission. During the entire life cycle of the packet within the stack the buffer chain is pushed from one layer 2 other as it is, the duration for which the memory occupied should be live is generally maintained by means of the reference count. The concept of loaning memory from one layer to another or one module to another by means of the reference count enables the IP stack to ensure zero copy. In case the multiple modules the IP packet traverses during its lifecycle had different buffer management schemes, there would have been a need of multiple memory allocation at each level. This would have introduced additional latencies and memory leakages.. Additionally, buffering is primarily needed as IP stack functions in a way that the data is available to be sent at any point of time and not only at the point when the device is ready to do the transmissions on the physical media.
- IP fragmentation is handled more efficiently within the IP stack and the MAC interface may easily treat each fragment as a new IP packet
- The infrastructure is already present and is stable minimizing the testing effort

However, there are some limitations in the standard buffer management scheme due to which implementers may chose an internal proprietary buffer management scheme. These limitations are

- Even for small things like sub headers we will have to reserve the entire buffer
- May lead to freeing complexity by manipulation of reference counts
- Extensive pointer manipulation at all levels of the datapath

3.1.6 Unification/Phy Handling

The PHY handling component of this framework needs to be totally customized or ported for transmission/reception of the data payload and hence varies across wireless technologies. In addition to the normal PHY handling procedures like setting up devices and readying them up for transmission or reception this component may also need to do a unification function. The unification function may include flattening up a network buffer chain supplied by the MAC Layer into a format which is understood by the PHY interface in the downlink direction and converting it into a network buffer chain which can traverse the entire OS in the uplink direction (RX case). Any kind of customization or optimization to make the whole MAC interrupt driven/polling driven or both may be done as a part of this component

3.1.7 Triggering Criteria

Last but not the least an important requirement for any Wireless MAC is the need of a triggering criteria which shall kick off the entire MAC processing. This component shall also supply the timing information to the entire implementation and may be tightly coupled to the control plane for setting up the data path. An example of such a criteria could be the frame interrupt in case of WiMAX which may either be 5ms or 10ms depending upon the specifications the MAC is using

4.0 WIMAX CASE STUDY

802.16e WiMAX MAC data plane has been chosen to prove this design framework. The entire data path was made to run over the existing Ethernet driver.

The simulation of this design has been done using the Linux Kernel 2.6.21 wherein the entire MAC was running as a kernel Module. The Connection database is populated through a neatly defined NETLINK interface from the user plane. The information is stored in the XML files which can be hooked to any kind of GUI.

The triggering criteria have been established by means of a high resolution timer library which is capable of generating 1ms and above interrupts.

4.1.1 How we did it for WiMAX

All the components of Wirelss MAC design framework in the following fashion to achieve the functionality of the WiMAX MAC data path

4.1.1.1 Queue Management & Composition

The Composer of the SDU was made to trigger by means of high resolution timer library which simulated the frame interrupts. This was module was responsible for filling the data burst. And hence in the process of filling data (in PDU format) it does fragmentation and packing if the connection (i.e.CID) allows so. The composer maintains per CID circular queue and provides API for en-queuing of IP packets to the IP stack.

4.1.1.2 Unfication/Phy Handling

The Unfication/Phy handling module contained the logic of building downlink frame and the corresponding interface to the physical layer for transmitting the same. The main function include frame planning (identifying zones to be transmitted), DL bandwidth scheduling (bandwidth allocation to each connection), DL burst building (creating two dimensional allocation plan covering the dimensions of each zone) and burst delivery in real time. It took the help of Composition function for filling DL bursts. **The composition and Unfication/Phy handling modules thus served the functionality of CPS sublayer 802.16e MAC.**

4.1.1.3 Network Interface

A pseudo Network device driver named as WiMAX0 was used to en-queued the IP Packets in per CID queue.

4.1.1.4 Connection Database

A Provisionary Application was implemented for reading configuration parameters from configuration file and populating the data in Connection database using Netlink interface. It was also apply classifier rules and routing table updates at IP layer. Hence Network Interface and Provisionary application serve the functionality of CS sublayer of 802.16e MAC.

The entire data path was implemented as Linux Loadable Kernel Modules (LKM) and executed from the context of the softlrd. It is recommended that for implementations that use this framework should execute the entire data path as part of kernel a single kernel context

- avoid race conditions and latencies introduced due to locks and mutexes
- as the kernel context have better privileges of access to operating system resources and priority of execution

The provisionary application written to populate the connection database was written as a user space process

To attain throughput data points it was mandatory to establish an end to end data connection and hence there was a need to write a verifier so that it could re-assemble the packets transmitted by this MAC over the Ethernet on the other side. Using a proprietary protocol known as WiMAX over Ethernet or WoE the end to end data connection has been established.

A profiling tool IPerf has been used to establish TCP connection using the WiMAX data path and get throughput data points. The information mainly consists of data rate or bandwidth achieved on the link for a specific interval of time. Additionally it also provides information about the packet loss over that connection.

4.1.2 Extension to LTE

The Wireless MAC design framework can be ported to achieve the functionality of the LTE MAC data path in the following way

4.1.2.1 Queue Management & Composition

The Composer of the SDU can also be made to trigger by means of high resolution timer library which simulated the frame interrupts. The SDU state machine can be used as it is. Instead of maintaining the per CID queues in LTE we maintain the per RAB ID queue. A customization that needs to be done in the SDU management shall include management of sub headers in case of fragmentation and re-assembly as defined in the specification

4.1.2.2 Unification/Phy Handling

The Unification/Phy handling needs to be totally customized with respect to the LTE PHY

4.1.2.3 Network Interface

The pseudo Network device driver can be used as it is

4.1.2.4 Connection Database

The connection database can be used as it is with only one change wherein the LTE specific parameters will have to be modified. This may include air interface specific parameters like MCS and Channel Coding schemes as well as the context id based on which the information is organized. In case of LTE RAB ID should be used.



PROPRIETARY NOTICE

All rights reserved. This publication and its contents are proprietary to Hughes Systique Corporation. No part of this publication may be reproduced in any form or by any means without the written permission of Hughes Systique Corporation, 15245 Shady Grove Road, Suite 330, Rockville, MD 20850.

Copyright © 2008 Hughes Systique Coporation

CONTACT INFORMATION:

phone: +1.301.527.1629

fax: +1.301.527.1690

email: whitepaper@hsc.com

web: www.hsc.com

APPENDIX A ABOUT HUGHES SYSTIQUE CORPORATION

HUGHES Systique Corporation (HSC), part of the HUGHES group of companies, is a leading Consulting and Software company focused on Communications and Automotive Telematics. HSC is headquartered in Rockville, Maryland USA with its development centre in Gurgaon, India.

SERVICES OFFERED:

Technology Consulting & Architecture: Leverage extensive knowledge and experience of our domain experts to define product requirements, validate technology plans, and provide network level consulting services and deployment of several successful products from conceptualization to market delivery.

Development & Maintenance Services: We can help you design, develop and maintain software for diverse areas in the communication industry. We have a well-defined software development process, comprising of complete SDLC from requirement analysis to the deployment and post production support.

Testing : We have extensive experience in testing methodologies and processes and offer Performance testing (with bench marking metrics), Protocol testing, Conformance testing, Stress testing, White-box and black-box testing, Regression testing and Interoperability testing to our clients

System Integration : As system integrators of choice HSC works with global names to architect, integrate, deploy and manage their suite of OSS, BSS, VAS and IN in wireless (VoIP & IMS), wireline and hybrid networks.: NMS, Service Management & Provisioning .

DOMAIN EXPERTISE:

Terminals

- Terminal Platforms : iPhone, Android, Symbian, Windows CE/Mobile, BREW, PalmOS
- Middleware Experience & Applications : J2ME , IMS Client & OMA PoC,

Access

- Wired Access : PON & DSL, IP-DSLAM,
- Wireless Access : WLAN/WiMAX / LTE, UMTS, 2.5G, 2G , Satellite Communication

Core Network

- IMS/3GPP , IPTV , SBC, Interworking , Switching solutions, VoIP

Applications

- Technologies : C, Java/J2ME, C++, Flash/lite,SIP, Presence, Location, AJAX/Mash
- Middleware: GlassFish, BEA, JBOSS, WebSphere, Tomcat, Apache etc.

Management & Back Office:

- Billing & OSS , Knowledge of COTS products , Mediation, CRM
- Network Management : NM Protocols, Java technologies,, Knowledge of COTS NM products, FCAPS, Security & Authentication

Platforms

- Embedded: Design, Development and Porting - RTOS, Device Drivers, Communications / Switching devices, Infrastructure components. Usage and Understanding of Debugging tools.
- FPGA & DSP : Design, System Prototyping. Re-engineering, System Verification, Testing

Automotive Telematics

- In Car unit (ECU) software design with CAN B & CAN C
- Telematics Network Design (CDMA, GSM, GPRS/UMTS)

BENEFITS:

- **Reduced Time to market :** Complement your existing skills, Experience in development-to-deployment in complex communication systems, with key resources available at all times
- **Stretch your R&D dollars :** Best Shore” strategy to outsourcing, World class processes, Insulate from resource fluctuations