Outline
Introduction and Motivation
WiFiRe Protocol Details
LAN Emulation
Scheduler
Scheduler Implementation Details
Implementation Issues
Future Extensions

Design and Implementation of MAC Layer of *WiFiRe* protocol

H Shravan Kumar

Under the guidance of:

Prof. Sridhar Iyer and Prof. Anirudha Sahoo

Kanwal Rekhi School of Information Technology

July 16, 2007

Introduction and Motivation
- Introduction
- Problem Statement

WiFiRe Protocol Details

LAN Emulation
- What is LAN Emulation?
- Why LAN Emulation?
- What did we achieve by doing LAN Emulation?

Scheduler
- What is Scheduling?
- Service Flows and Types of Grants
- Timing Sequence

Scheduler Implementation Details

Scheduler Implementation Tables

Implementation Issues

Future Extensions

Introduction

- *WiFiRe* is WIMAX like MAC over WiFi like PHY.
- It uses a free license band spectrum (IEEE 802.11b, 2.4 GHz Band) for communication.
- Its a star topology network.
- It follows Multi-Sector TDM approach.

Problem Statement

Our problem focuses on implementing the MAC layer of the protocol, which primarily focuses on the basic communication between BS and ST.
The following are the steps in WiFiRe Protocol-

- *Beacon Transmission*
- *Ranging*
- *Registration*
- *Data Connection Creation*
- *Data Connection Termination*

Basic communication overview of WiFiRe

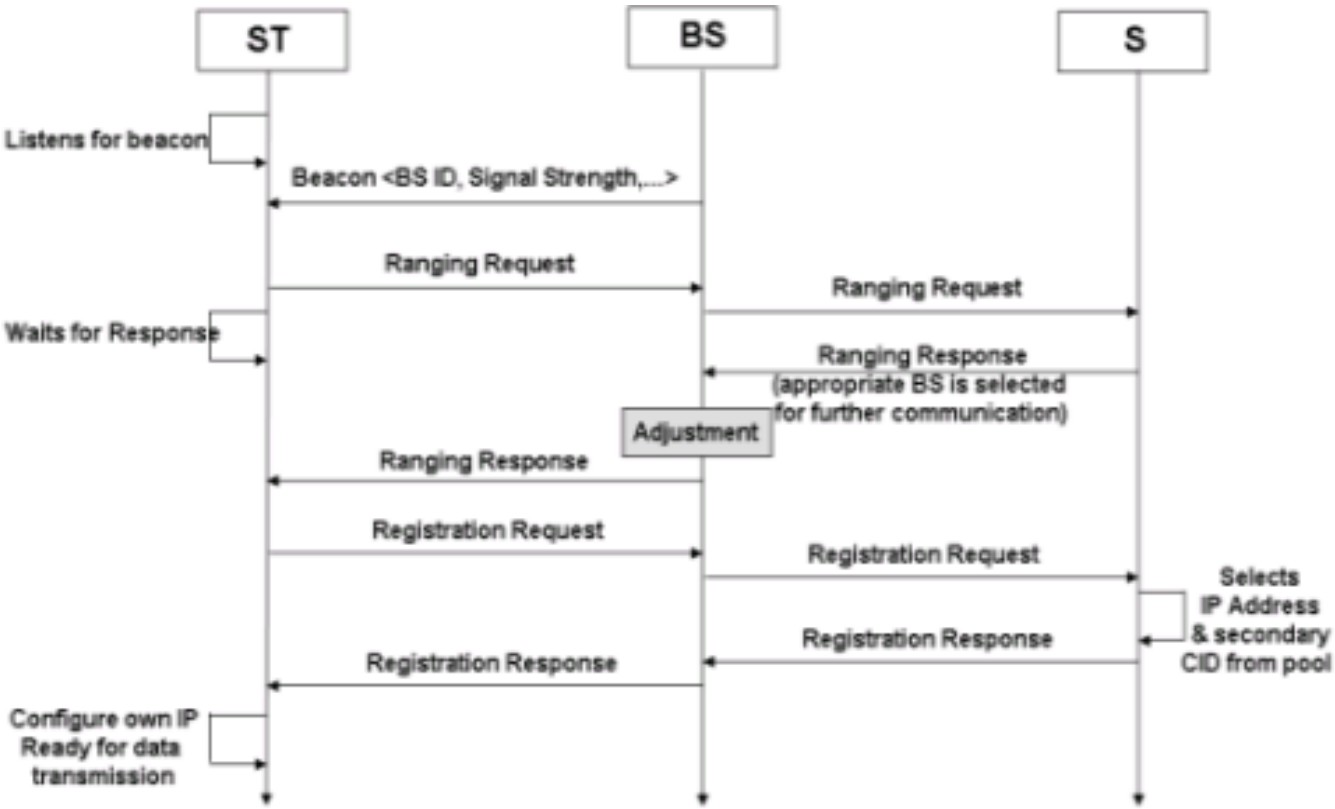


Figure: Basic communication sequence diagram

Structure of WiFiRe MAC PDU

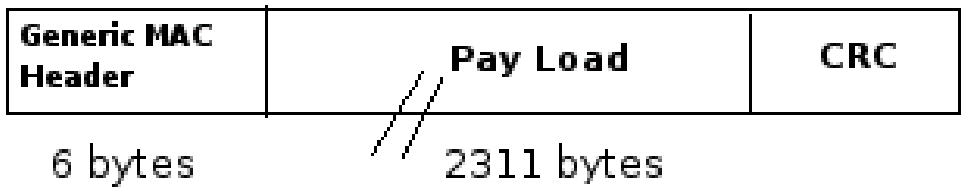
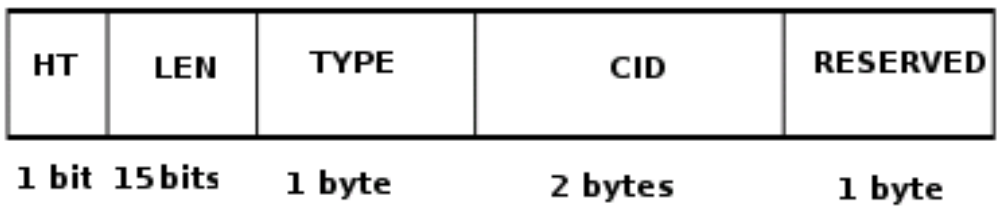


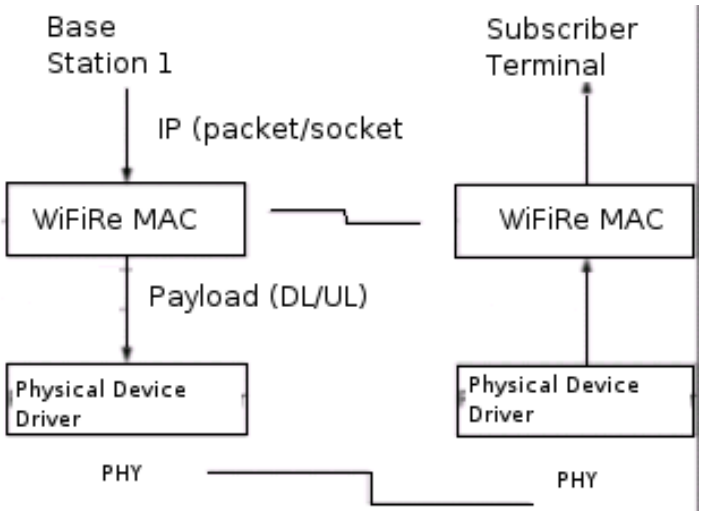
Figure: MAC PDU Format

Generic MAC Header



What is LAN Emulation

WiFiRe MAC will be process packets at the application layer and send it to the below layer considering remaining layers as normal PHY. Figure 4 is the overview of the LAN emulation of the protocol.



Why LAN Emulation?

- To understand and ensure that steps involved in WiFiRe protocol works.
- It is comparatively easy to debug and make changes at the application layer rather than at kernel level.
- WiFiRe hardware is not ready. In order to test the protocol, there is need of already setup network infrastructure (i.e. We chose LAN for our case).
- Design and implement data structures, small working modules in order to test. Reuse them with minimal changes when implementing the protocol in the kernel level.

What we have achieved?

- We have shown that WiFiRe protocol works.
- We have emulated Beacon Broadcast, Registration, Data Service Addition and Data Connection Termination.
- We also have emulated Ranging Procedure but it is used for getting Primary and Basic CID's, not for synchronization.
- We have minimized the number of CPU cycles by using static buffers instead of dynamic memory allocation.

What is Scheduling?

- Scheduling of the frame is basically defined as allocating slots to ST's which are currently requesting slots for data transfer.
- Each frame is divided into slots for beacon, Downlink map and Uplink map.
- Normally Downlink slots will be much more than the uplink slots.

Service Flows and Grants

Types of service flows supported by WiFiRe are

- Unsolicited Grant Service (UGS)
- Real time Polling Service (rtPS)
- Non real time Polling Service (nrtPS)
- Best Effort (BE)

We have provided nrtPS and BE service flows.
Types of grants supported by WiFiRe are

- Grant Per Connection Mode
- Grant per Service Flow type
- Grant Per Subscriber Terminal Mode

Timing Diagram

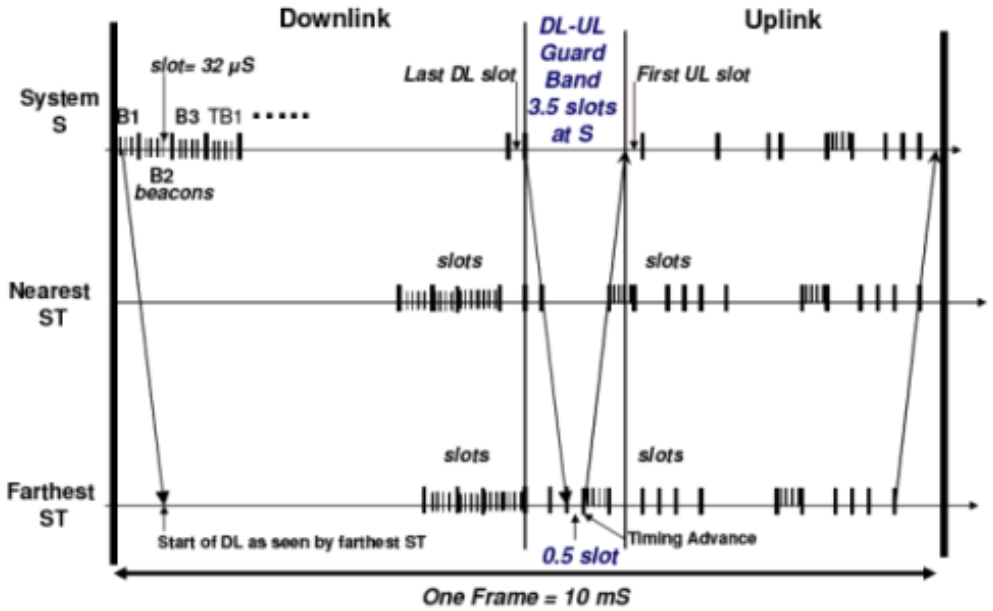


Figure: Timing Sequence

Algorithms which we used

Round Robin Scheduler

- This is simpler and straight forward technique, which schedules the packets in a round robin fashion.
- There is no priority.
- This is starvation free scheduler.

Contd..

Our Scheduler

- We used RR scheduler at our initial stage, but due to its burstiness in scheduling the slots prompted us to use another scheduler.
- We also have tried to implement SRR scheduler which allocates slots in $\Theta(1)$ complexity.
- Our current scheduler uses Grant per Serive Flow mechanism.
- It allocates some % of slots for UGS flows, allocates equal share of slots to rtPS and nrtPS flows, small amount of slots if available to BE, and some contention slots.

Tables Stored at BS

Primary CID	Basic CID	ST ID
2 bytes	2 bytes	4 bytes

While creating ranging response

Primary CID	Data CID	Timer
2 bytes	2 bytes	1 byte

While assigning Data CID to ST

Figure: Tables maintained at BS

Implementation Surprises

- Structure alignment problem.
- Design of a BS scheduler is a complex task because of the dynamic allocation of DL and UL slots in the time frame.
- Broadcasting the packets to all STs.
- Dynamic Memory allocation was consuming more cycles, which created delay in creating packets.

Future Work

Future extension

- Implement the concept of Timers such that the when a packet is lost while transmitting it is retransmitted.
- Ranging part should be completed so that it handles the synchronization.
- Enhancing it to multi sector BS.
- Implementing it actually into the board i.e. Physical Hardware.

Thank you !