

# WiFiRE: Wireless Broadband Access for Rural Area

Second Stage Report

Submitted in partial fulfillment of the requirements  
of the degree of  
**Master of Technology**

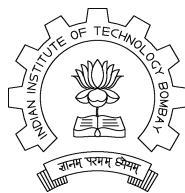
by

**Sameer Kurkure**  
(Roll No. 05239025)

&

**Shravan Kumar Hullur**  
(Roll No. 05239018)

Under the guidance of  
**Prof. Anirudha Sahoo and Prof. Sridhar Iyer**  
Kanwal Rekhi School of Information Technology



Kanwal Rekhi School of Information Technology  
Indian Institute of Technology, Powai, Mumbai  
2006-2007

## Abstract

The report covers the background knowledge of *WiFi-Re* with basic working of the protocol and its implementation in C. Problems associated with design and implementation and their plausible solutions are covered as a part of report. Additionally, it also comprises of sequence diagrams, flow diagrams, state diagrams etc. of working components of *WiFi-Re* along with design model in C sockets and describes the issues and challenges involving implementation of the projects.

## 1 Introduction and Motivation

Now-a-days the use of Internet and mobile communication has grown to a large extent such that it became mandatory for daily usage of life. The statistics in India shows that there are more than 100 million mobile users in India [as per June 10th of 2006] which shows its importance in daily routine. Major population in India resides in remote areas where access to basic amenities like telephony, internet etc. are difficult to provide. Broadband wireless access (BWA) can become the best way to meet escalating business demand for rapid Internet connection and integrated data, voice and video services. BWA can extend fiber optic networks and provide more capacity than cable networks or digital subscriber lines (DSL). But deployment of BWA (WiMAX) compatible devices are much complex and costlier.

Rural areas are sparsely populated and their distances varies in few kilometers, unlike urban areas. Installation of more base stations will probably not solve this problem, which also costs more. Wireless Fidelity - Rural Extension (WiFi-Re) introduces the concept of wireless communication over WiFi IEEE 802.11b physical layer (PHY) and WiMAX IEEE 802.16 MAC layer using low cost chip sets. 802.11b PHY has better availability of low cost chip sets which can operate on unlicensed 2.4GHz frequency band and WiMAX has potential to work over larger distances of 30-40km.

Almost every rural area can avail fixed phone lines, but mobile communication and broadband are difficult to deploy. For this, *WiFi-Re* can provide a very good solution. *WiFi-Re* uses WiFi PHY which has got a free license band spectrum (IEEE 802.11b, 2.4 GHz Band), the easy availability of WiFi chip sets, and very good QoS features of WiMAX, which makes it suitable to provide long range communications for rural areas. *WiFi-Re* uses a star topology network, in which main station (S) will be connected to set of Base Stations (BS) which in turn connected to sectorized antennas through which a Subscriber Terminal (ST) will communicate.

Other approaches to solve the problem such as WiMAX, Optical Networks, DSL etc. are not cost effective and did not proved to provide affordable services to rural environment. The concept of *WiFi-Re* seems to be good solution for this scenario and can satisfy bandwidth need at proper price that suits rural people.

The basic organization of the paper is as follows. Section II gives the overview of the problem statement. Section III gives how did we approach to solution. Section IV gives why we are doing it on LAN instead of actual kernel level implementation. Section V design of the project. Section VI gives reusable modules. Section VII the portability issues. Section VIII gives about the individual contribution. Section IX gives conclusion and future work.

## 2 Problem Statement

Implementation of MAC layer of *WiFi-Re* protocol, emulating it on LAN using C socket programming, and finally implementing the same at kernel level.

### 2.1 Description of *WiFi-Re* protocol

The basic design of WiFiRe comprises of a single operator Station (S) which have licensed bandwidth like dedicated lines, fiber PoP etc. This operator provides the communication base for the outside world to rural environment. The total area is being sectorized and each sector will be having Base Station (BS), which is a sectorized antenna of height around 40m that lies near point of presence (PoP). BS are arranged such that they can simultaneously able to transmit or receive within the sectors. There are Subscriber Terminals (ST) situated at the villages which have 10-12m directional antennas. Both BS and ST are fixed where as users with in ST (e.g. building, house, small campus etc) can be either fixed or mobile depending upon the internal network being used.

These are the basic points for the villages from where people will be able to communicate with the outer world. These ST's should be in a height so as to maintain a system gain of 150dB. Users may connect to these ST's using wired or wireless means of communication. The System will be of star topology. The network topology will be as shown in the following figure 1.

Each BS can cover up to 15-20km range, covering around 100 villages. Each BS will be responsible for all the communication that takes place in its sector range. Each ST will be connected to voice and data terminals in the village by a local area network. As mentioned earlier these ST will be directional and will be connected to corresponding BS covering the sector, thus providing reliable data transfer. Chances of interference with the other transceivers can be solved by locking up ST with the BS with highest signal strength. BSs in the system (S) are configured to operate alternatively or diagonally opposite BS for non-overlapping transmission. WiFiRE supports time division duplex (TDD) over single channel with multi-sector TDM (MSTDM) mechanism, which supports about 25Mbps (for both uplink and downlink) for a cell. In TDD, the uplink (ST to BS) and downlink

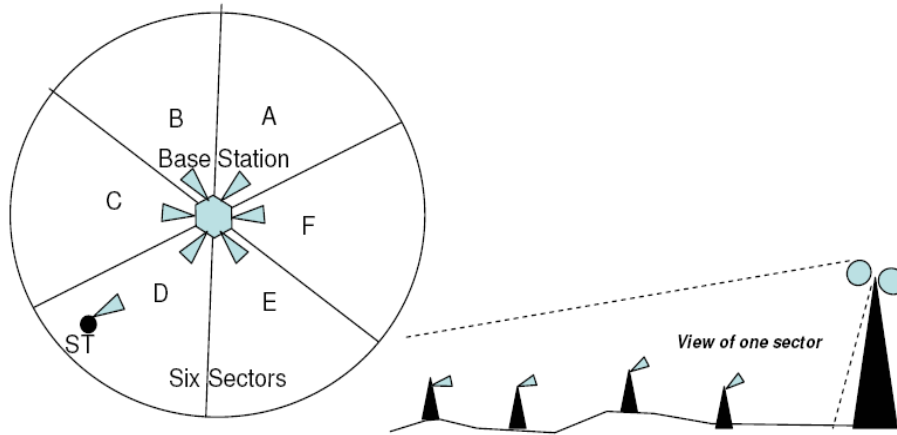


Figure 1: WiFi-Re Topology

(BS to ST) share the same frequency but are activated at different time. BS and ST operate with synchronization with each other. Time is divided into frames, which is further divided into DownLink (DL) and UpLink (UL) segments, which may not be of equal time intervals. In each DL slot one or zero transmissions can take place in each sector. Multiple BS antennas can transmit simultaneously provided they do so in a non-interfering manner.

Figure 2 is sequence diagram for basic working of WiFi-Re protocol. Beacons are being transmitted at the start of each DL segment, which contains information for time synchronization of the ST(s) in that sector, information regarding the DL and UL slots allocations (which are called DL and UL maps respectively) for that frame, and other control information. These DL and UL maps are computed online because there may be site dependent or installation dependent losses and different time varying requirements at each point of time.

The basic assumptions for working for WiFi-Re protocol are stated as:

- Wireless links in the system are fixed, single hop, with a star topology. Mobility and multi-hop wireless links are not considered.
- Fixed carrier frequency and WiFi radios operating at 11Mbps, except PHY operating at 1 or 2 Mbps.
- Various components in the system will be having unique IP addresses.
- About 20MHz(1 carrier) of conditionally licensed spectrum is available for niche/rural areas.

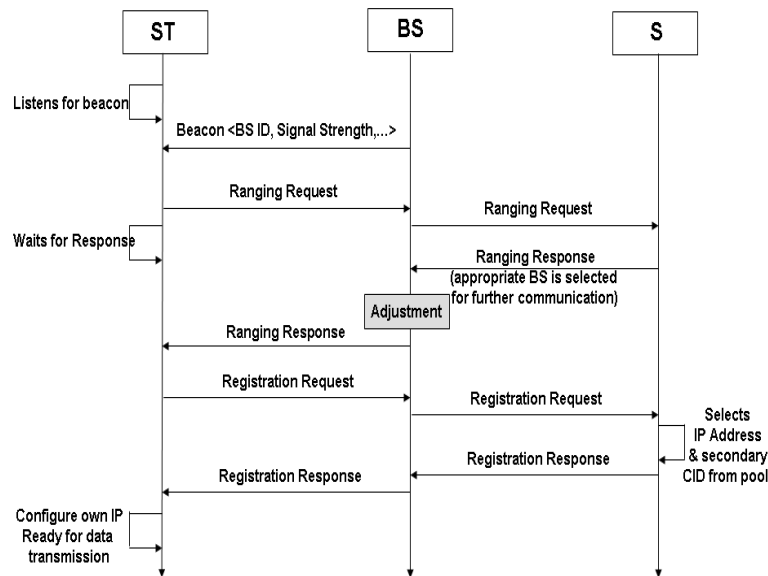


Figure 2: Basic communication sequence diagram

- All nodes in the system are operated by a single operator who owns the conditional license.
- The availability of unlicensed or free spectrum in the 2.4GHz band.
- The existence of point of presence (PoP) every 25km or so, for backbone connectivity.

**Alternatives** There are alternatives present for this but most of them are not cost effective. The following are some of the alternatives

- WiMAX-d (IEEE 802.16d), can provide an alternate solution as it has got high gain and a good spectral efficiency, which can carry 80Mbps over-the-air per base station with a 20MHz allocation. The main drawback is deploying WiMAX solutions are difficult which also need complex and costly hardware that is not available easily.
- WiFi (IEEE 802.11b) can provide for short distances of about few meters but not for long distances. In 802.11 based networks, contention algorithm like Distributed Coordination Function (DCF) mechanism does not provide any delay guarantees and are more distributed in nature, while the Point Coordination Function (PCF) mechanism is efficient only for small number of nodes.
- Mobile cellular technologies cannot provide broadband services with high bandwidth need.

- 802.11 based Mesh Network [3], where it doesn't use the existing CSMA/CA technology in 802.11, instead it uses 2-phase TDMA based protocol. But the problem with current approach is MAC of 802.11b. 802.11b doesn't provide any quality of service except PCF. The outdoor long-distance use of 802.11 requires a revisit to the protocols at various layers of the OSI stack, as well as various system design issues.

## 2.2 What our Problem Focuses on

Implementing working MAC layer protocol which holds basic communication between BS & ST. Steps involved in our implementation are:

**Beacon Transmission** :- Broadcasting beacon to all the ST of a sector which is implemented by the concept of multi-unicast.

**Ranging** :- Synchronizing clock and other physical layer parameters with respect to the System S are being done. It is also being performed periodically so that ST will keep in-sync with S. Here the ST will be given Basic and Primary Connection Identifiers(CID) by which the further communication between BS and ST take place. This step is not required in LAN implementation point of view as the propagation delay is negligible on ethernet. Detailed explanation is being given in the following Design section.

**Registration** :- This steps ensures that the ST can establish a connection for data exchange as registered ST known to BS. This step is mandatory before any actual data transfer between ST and BS. Here operational parameters and capabilities are being exchanged. Detailed explanation is being given in the following Design section.

**Data Connection Creation** :- In this phase control packet requesting for data connection (DSA) is sent by ST to BS for initiating actual data exchange. BS will assign a data CID to ST for further data communication which informs the nature of the bandwidth request service to be used with the connection.

**QoS Management** :- It allows the existing CID's to change the nature of the bandwidth allocation or for a new CID which does not have any specified/allocated bandwidth resource. This feature is currently not implemented as part of demonstration.

**Data Connection Termination** :- In this phase the entity (BS or ST) which wants to terminate a data connection exchanges a management message to inform the peer entity.

### 3 Approach to the Solution

Here design and implementation a WiFi-Re MAC layer will be over application layer using C sockets on ethernet LAN where it will construct, process and execute the packets on the WiFi-Re MAC and will pass the packet to socket layer assuming it to be the PHY layer of WiFi 802.11.

### 4 Why Emulating on LAN?

- To understand and ensure that steps involved in WiFi-Re protocol works.
- Design and implement structures and small working modules in order to test and reuse them with minimum changes while going into kernel level.
- WiFi-Re hardware is not ready and in order to test the protocol, there is need of already setup network infrastructure (i.e. LAN in this case).
- It is comparatively easy to debug and make changes on application layer rather than at kernel level.

The concept of emulating the protocol on LAN using C sockets by assuming that the Application Layer of the Ethernet as the MAC layer of our protocol and assuming the layers down to it as the PHY. Here the characteristics of PHY layer can be ignored while implementing the MAC layer through C sockets as the device driver will take care of the PHY at lower levels.

Figure 3 is the overview of the LAN level emulation of the protocol.

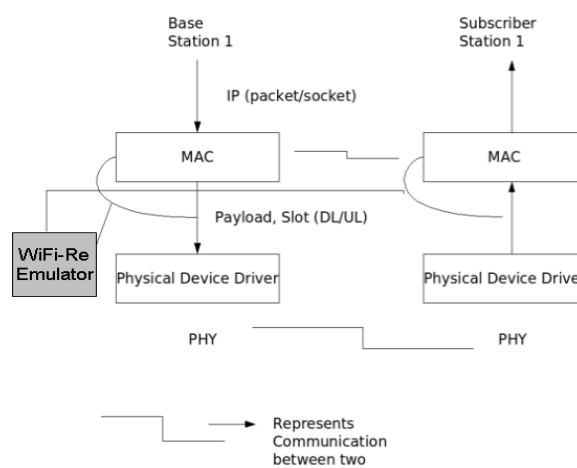


Figure 3: Overview of LAN emulation

## 5 Design and Implementation Issues

Design part of the project comprises of flow charts of main procedures, state diagram and design of basic structures like beacon message, control message packets, data packets, generic MAC packet etc. Figure 4 describes the implementation overview of protocol in C. This follows the client server architecture as BS acts as a server, it creates socket and starts listening on designated port where as ST being a client can come up any time and connects to the BS. Whenever a connection request comes to BS, it creates a thread and assigns an socket file descriptor to ST for further communication.

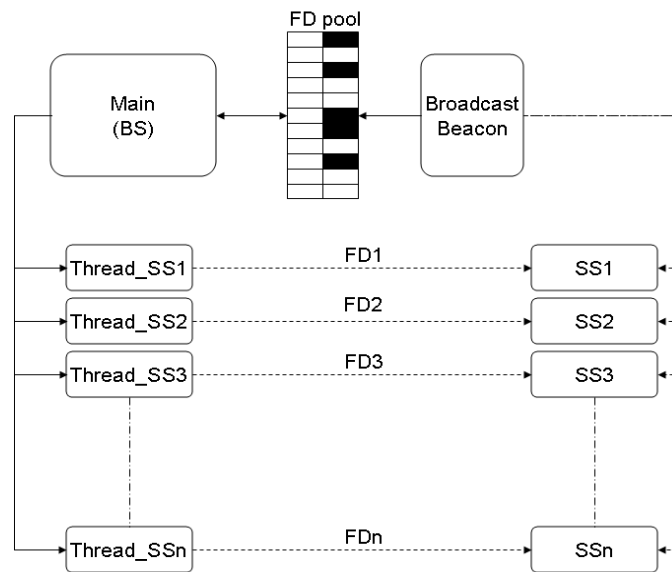


Figure 4: Implementation in C socket

### 5.1 Beacon Broadcasting

When ever a new ST comes up, it is in IDLE state until a beacon is received, BS associates a thread and a socket file descriptor to that ST and at the same time it maintains pool of FDs assigned to STs in a bitmap like structure which is used to broadcast beacon to all currently connected STs. Here (in LAN scenario) there is need to maintain TCP connection (connection oriented) with all STs currently active in the system, so broadcasting is not possible over pre-established socket connections. So concept of multi-unicast came into picture in which unicast of beacon packet is done over all those FDs stored in pool. The state of an ST changes to BCNR (Beacon Received) whenever it successfully receives and interprets beacon message. When ST goes down it sends at terminating signal to BS which kills the associated thread and deletes corresponding FD from the pool.

## 5.2 Ranging

Ranging is done in order to synchronize clocks and other physical parameters with the system (S). Figure 5 and figure 6 gives the overview of ranging procedure done at Subscriber Terminal(ST) and Base Station (BS) at design level. The state of ST changes to RNGG (Ranging) when ever it sends ranging request (RNG\_REG) to BS and state changes to RNGD (Ranged) when it is successfully ranged by BS and receives ranging response (RNG\_RSP). Over LAN scenario there is no need of ranging as the propagation delay is negligible so this procedure get by passed currently. There might be need of ranging at ethernet also when the hardware is not time synchronized.

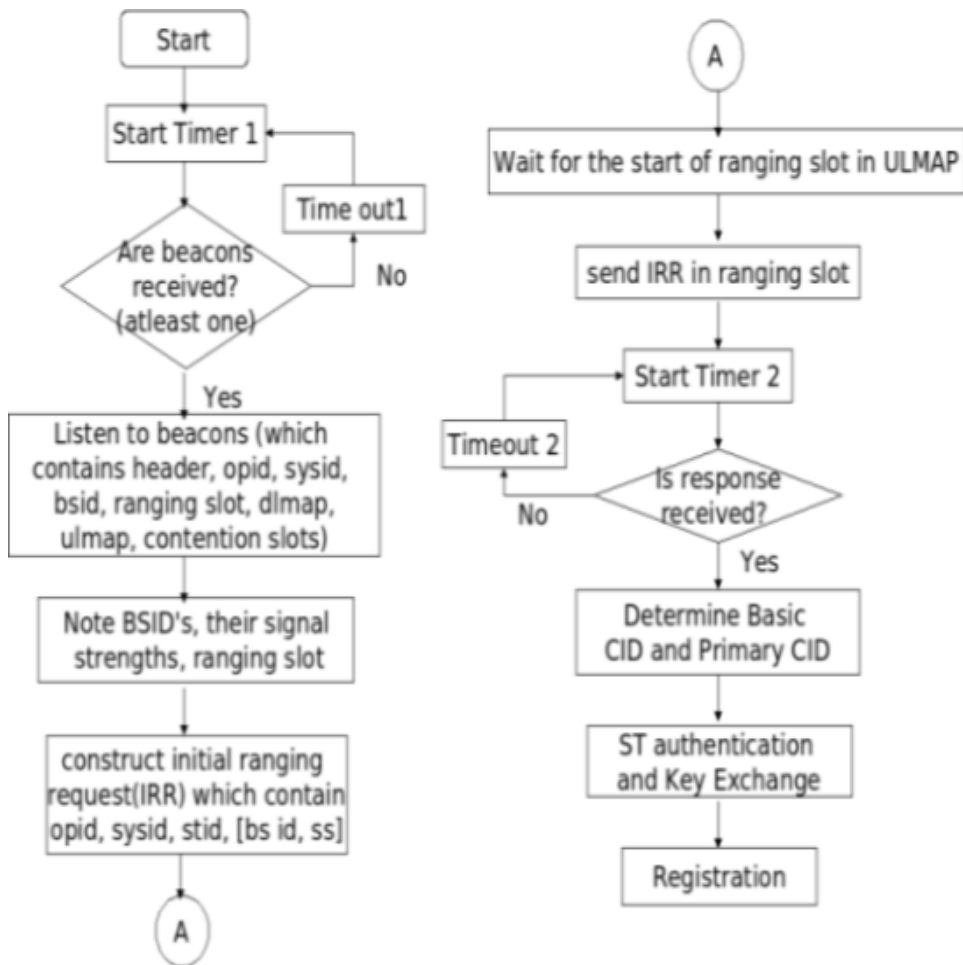


Figure 5: Ranging at ST

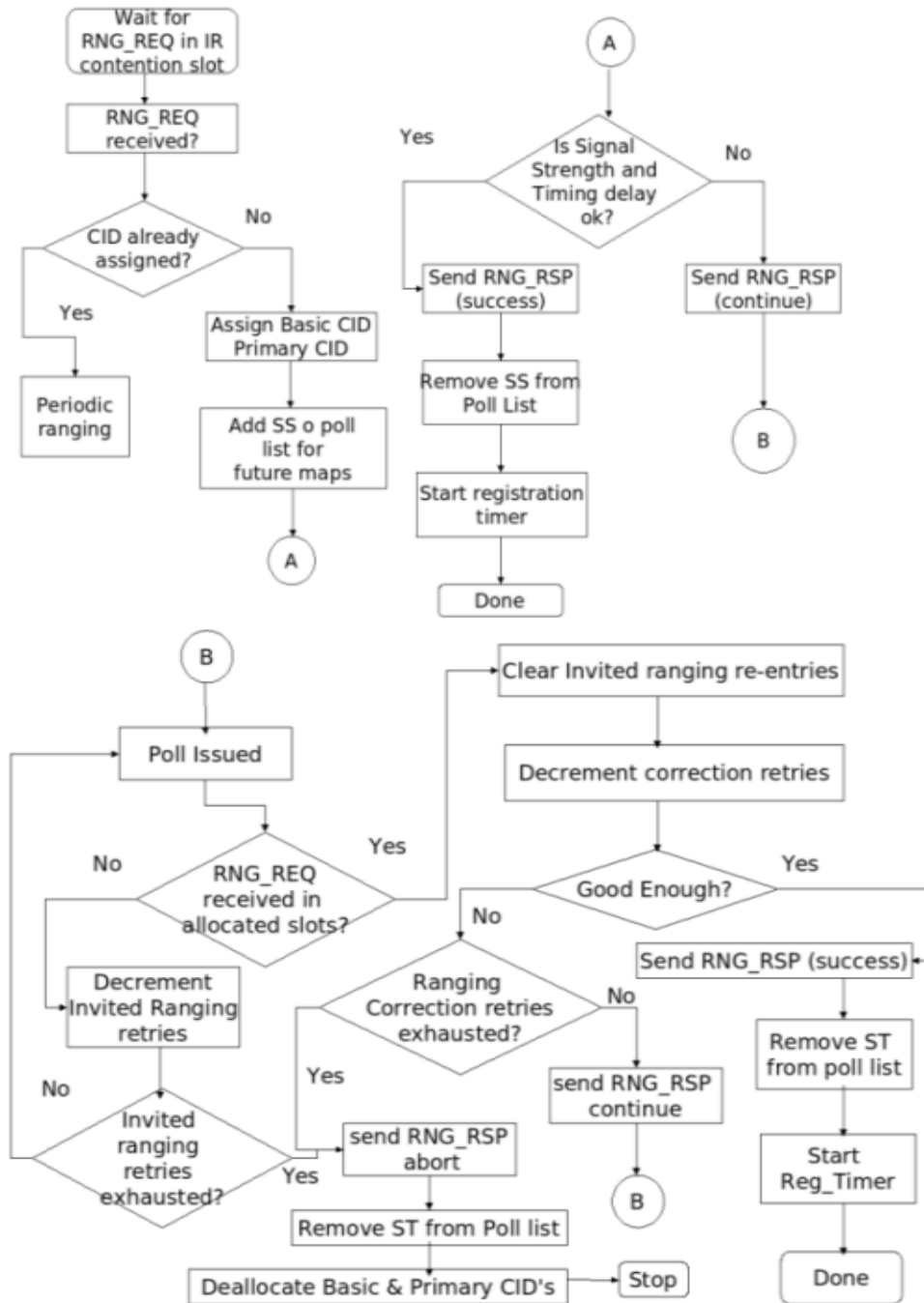


Figure 6: Ranging at BS

### 5.3 Registration

Registration happens for getting an IP address, the details of IP version from S, which can be used for further communication. This process involves a Registration request from ST and registration response from S. During this process the operational parameters and capabilities are being exchanged. State of an ST changes to REGG (Registering) when it sends REG\_REQ and is in the process of registering to BS. After reception of registration response message (REG\_RSP) state changes to REGD (Registered). Figure 7 and figure 8 give the overview of the registration process at BS and ST.

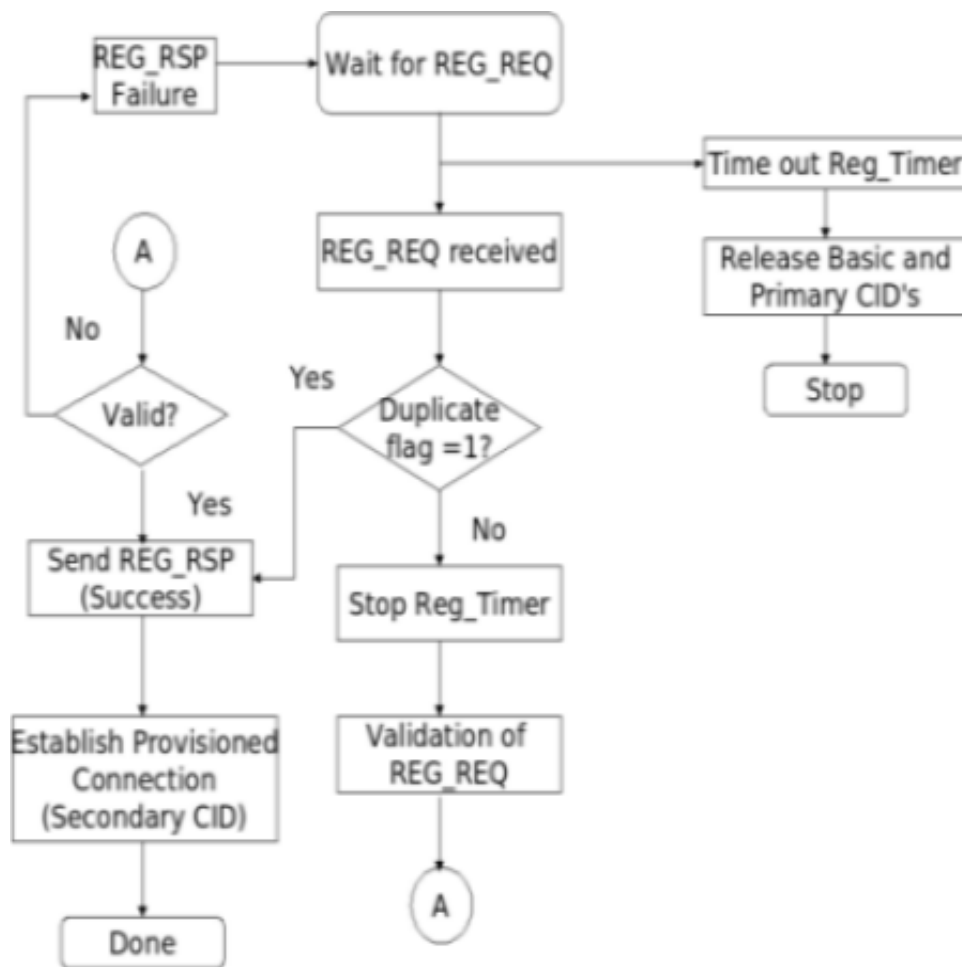


Figure 7: Registration at BS

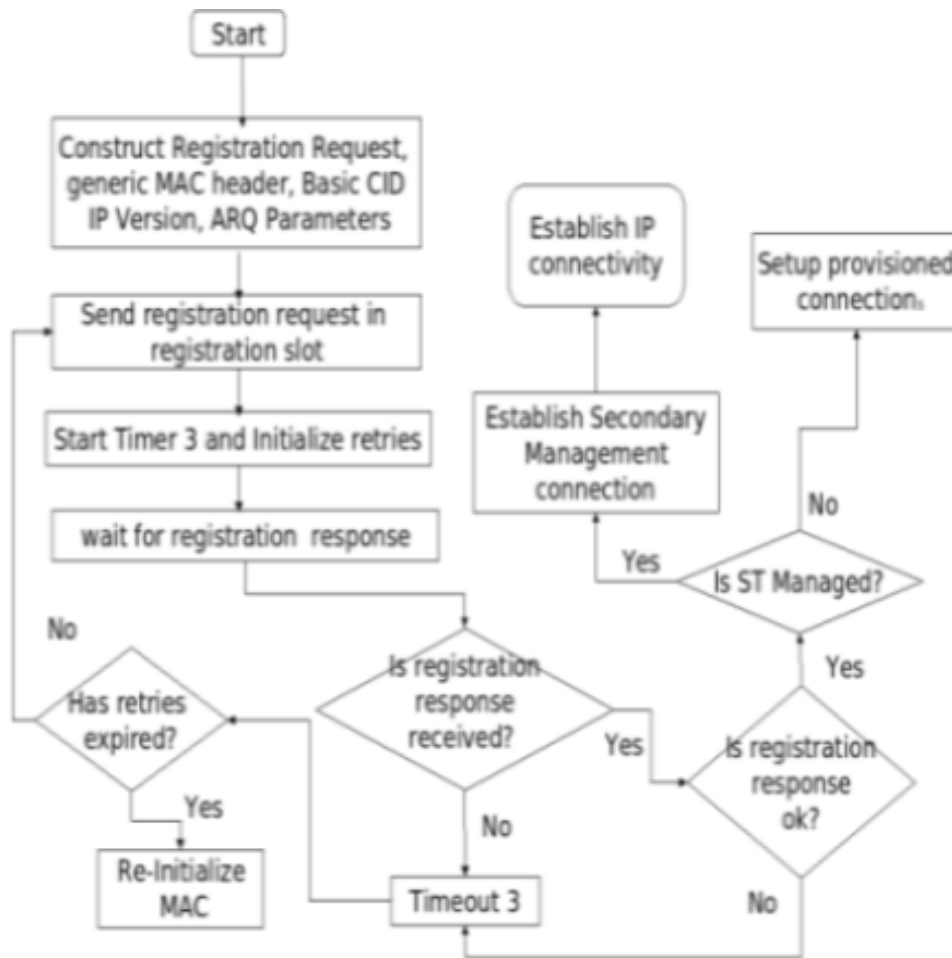


Figure 8: Registration at ST

#### 5.4 Data Connection Handling

A registered ST can request further for the data connection by sending DSA\_REQ to BS, changing its own state to DCIN (Data Connection Initiating) and settles down to DCED (Data Connection Establish) state after receiving positive response message (DSA\_RSP). ST switches to DCCG (Data Connection Changing) when it requires to change the connection parameters. And finally goes through DSDG (Data Connection Deleting) state when ST request to terminate its existing connection and on termination it settles to REGD (Registered) state. Figure 9 gives state diagram of ST during whole course of protocol working, description of state is mentioned in table 1

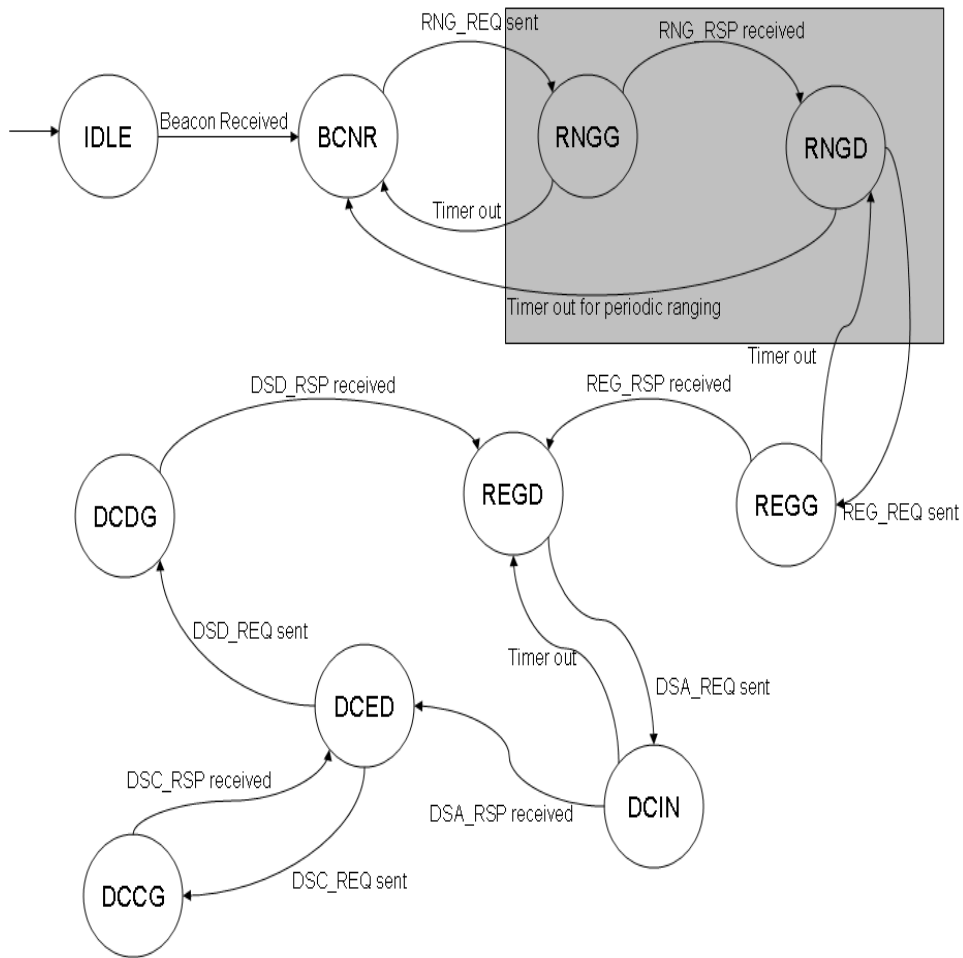


Figure 9: State diagram for ST

State	Description
<i>IDLE</i>	Idle state
<i>BCNR</i>	Beacon Received state
<i>RNGG</i>	Ranging state
<i>RNGD</i>	Ranged state
<i>REGG</i>	Registering state
<i>REGD</i>	Registered state
<i>DCIN</i>	Data Conn Initiating state
<i>DCED</i>	Data Conn Established state
<i>DCCG</i>	Data Conn Changing state
<i>DCDG</i>	Data Conn Deleting state

Table 1: States of ST

## 6 Reusable Modules

Design and implementation so far is modular enough in order to maintain good level of re-usability. All the structures and major subroutines are independently implemented in different files and are coupled properly to other elements in order to have modularity.

## 7 Implementation Issues and Surprises

There were some challenges and surprises which were faced during the implementation. Some of them along with their appropriate solutions are as follows:

**Case 1 :** In the project, BS is implemented as a socket server and ST as socket client. When ever an ST comes up, BS needs to instantiate a separate subroutine which take cares of ST and BS itself should again go back on listening mode on some designated port. At the same time BS need to store the assigned FDs to the pool for broadcasting beacon. This pool is nothing but a data structure which is global to the main BS process. Now whenever a *fork()* is called it creates a new process which have its own text, data and stack segments which even did not allow the global data of one process visible to other process. IPC could be another solution for this. Messages can be passed through pipes but still the problem remains the same. Every process has its own file descriptor table and no process and send or receive data on other's file descriptor set in C. The possible remaining solution are the use of threads. POSIX threads are light weighted process which share the same code and data segment of the process, through which common data can be access through shared memory.

**Case 2 :** Broadcasting the packets to all STs which are currently active over established TCP connections which became difficult to accomplish on a 802.3 ethernet protocol. To overcome this, one of the possible ways could be having a virtual socket at BS end and unicast the packet to this virtual socket. A small subroutine will be initiated which will perform unicast to all the FDs which are currently associated with the BS.

**Case 3 :** Time synchronization is always being a problem in TDM based systems. It can be implemented using 3-sync mechanism similar to that is being used by TCP, but this may give some variance which is not acceptable at granularity of microseconds. Possible solution is using machines which share collision free media for communication like cross cable over LAN, or test experiments can be run over single machine which share system clock.

**Case 4** : Design of a BS scheduler is a complex task because of the dynamic allocation of DL and UL slots in the time frame. So the possible solutions could be fetching the schedules from the static read only files in which schedules are defined depending upon the number of active STs or designing our own scheduling routine using simple scheduling strategies like round-robin etc. (as in our case).

**Case 5** : Structure alignment problem, when a packet is sent over network, the original size of packet does not match with the length field specified inside the packet because of the extra alignment bytes padded by the compiler. One solution could be sending each element individually over network but this will result in increase in system calls and highly depends on structure of packet. Packing the structures to one byte (smallest size data type) will solve this as number of bytes required for padding will be zero.

**Case 6** : DEBUG LEVEL, in order to keep track on the execution of the protocols, debug levels has been introduced to get output information at user defined level of depth. Table 2 defines the levels and their purpose of execution.

Debug Level	Purpose
0	No Output
1	Event And High-Level Information Messages
2	Track The Flow Of Command Processing
3	For Inner Loops, Table Traversals, Etc
4	I/O Debug - Mover Messages Traces
5	Trace-Level Debug
6	Redundant Information

Table 2: Debug Levels

## 8 Conclusions and future work

Emulation of WiFi-Re protocol using C sockets seems working for more than one ST with one data connection per ST using simple BS scheduler. Still issues like portability of code on different architectures like 64-bit machines, timers corresponding to various control messages in the protocol need to be taken care of. There are different beacon structure for different BSs that are associated with sectors having different ST set, therefore corresponding beacons need to be forwarded to the BSs which can be counted as a future task. Modules are designed in such a way that they can be reused with minimal changes in kernel level implementation.

## References

- [1] Sridhar Iyer, Krishna Paul, Anurag Kumar, Bhaskar Ramamurthy, *WiFiRe: Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, CEWiT, India, May 2006.
- [2] Bhaskaran Raman, and Kameswari Chebrolu, *Revisiting MAC Design for an 802.11-based Mesh Network*, San Diego, CA, USA: HotNets-III, Nov 2004.
- [3] LAN/MAN standards Committee, and IEEE Microwave Theory and Techniques Society, *Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE-SA Standards Board, June 2004.
- [4] LAN/MAN standards Committee, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE-SA Standards Board, June 2003.
- [5] H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.