Emulation of WiFiRe protocol on LAN

Janak Chandarana, Ranjith Madalapu, Sameer Kurkure, Shravan Hullur
Anirudha Sahoo and Sridhar Iyer
Department of Computer Science and Engineering, IIT Bombay

Email: {janak, ranjith, sameers, shravan, sahoo, sri}@it.iitb.ac.in

***Abstract*— WiFi Rural Extension (WiFiRe) is a MAC layer protocol especially designed to provide broadband communication in rural areas of India. It supports long range communication using 802.11b PHY and a MAC similar to 802.16d. In this paper, we describe the implementation of WiFiRe MAC software stack as an emulation over Ethernet. Implementation is done using C sockets and allows a user to test basic MAC functions such as connection establishment, packet flow and header construction in absence of WiFiRe hardware. We believe that, WiFiRe protocol implemented as part of LAN emulation can be easily ported to the WiFiRe hardware.**

I. Introduction

A. Motivation

Long range wireless for data and voice connectivity is being considered as a viable and affordable solution for rural India. As described in [1], the average user cannot spend more than Rs. 300 per month for Internet. There is a need for affordable and easily available system, using which broadband access can be spread to every village in India. WiFiRe [2] presents one such solution, using cheap 802.11b chip-set but modifying the MAC for longer range, support for QoS and better efficiency. Initiatives reported in [3] and [4] address very similar issues.

B. WiFiRe architecture

WiFi Rural Extension (WiFiRe) [2] uses the licence-free 2.4 GHz spectrum and cheap 802.11b RF chip-set to provide long-range wireless communication. It replaces 802.11b MAC mechanism (DCF/PCF) with a MAC similar to 802.16d, while retaining 802.11b PHY. WiFiRe architecture is based on star topology (Figure 1) - a Base Station (BS) at the fiber Point of Presence (PoP) and Subscriber Terminal (ST) in the villages nearby. BS has six sectorised antennas covering area of 15-20 km. ST is connected to end users with different LAN technologies [5] and communicates with BS through directional antennas. Within a sector, WiFiRe follows

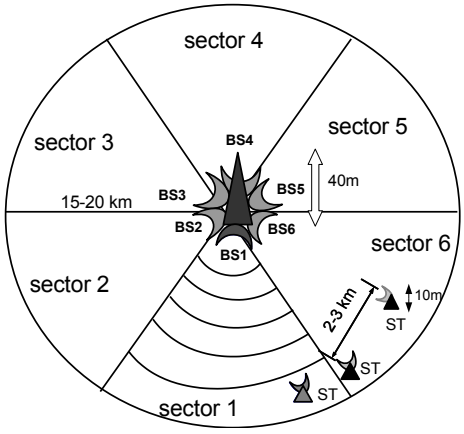

Fig. 1. WiFiRe Architecture

time division multiplex (TDM) frame structure similar to 802.16d.

WiFiRe is promoted by CEWiT India[1] and the project is spread across IIT Bombay, IIT Madras and IISc Bangalore.

C. WiFiRe MAC

WiFiRe supports time division duplex (TDD) over single channel with multi-sector TDM mechanism to provide about 25Mbps (for both uplink and downlink) for a cell [2]. In TDD, the uplink (ST to BS) and downlink (BS to ST) share the same channel but are activated at different times. BS and ST operate in synchronization with each other. Time is divided into frames, which is further divided into downlink (DL) and uplink (UL) segments, which need not be of equal time intervals. DL is further divided into two major parts: beacons and actual downlink PDU. Every Beacon message (Figure 2) have control information followed by UL-MAP and DL-MAP in first DL pdu. These MAPs have the information about total available slots for STs in all the sectors. All

[1]The Center of Excellence in Wireless Technology (CEWiT) is an independent research organization set up by the Ministry of Information Technology, Government of India (MIT) in partnership with industry.

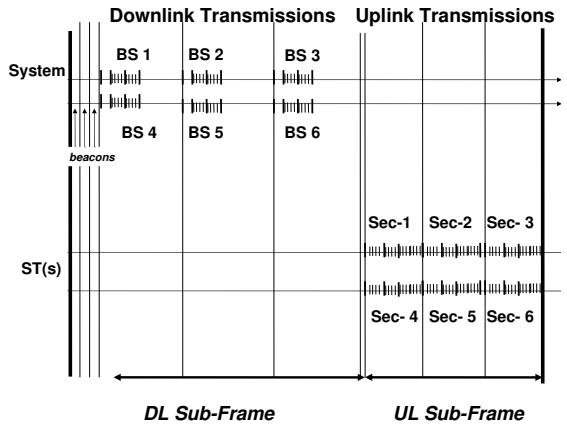

Fig. 2. WiFiRe MAC frame

STs will read the DL-MAP and search for their own ST-ID. If ST-ID is found, ST will wake up in that particular slot and receive the packet. ST transmits packets to BS according to UL-MAP [2]. STs follow multi-sector TDD mechanism as illustrated in Figure 2. Connections established between BS and ST are identified by a 16 bit Connection Identifier (CID). Refer [2] for more details about WiFiRe MAC.

D. WiFiRe implementation

Simulations have been done to check WiFiRe protocol's correctness, performance, scheduler design and few PHY related issues like antenna behavior [6] . We would like to test protocol behavior in actual implementation and emulation on LAN is the first step towards it.

We are using layer-2 functionalities of C sockets to implement WiFiRe MAC and test it on DIX[2] Ethernet based LAN. Emulator is carefully designed so that some modules are PHY dependent and some are PHY independent. PHY dependent modules will be replaced appropriately when WiFiRe hardware becomes available.

E. Contributions

There are two main contributions of this paper.

(A) By doing emulation, we expect to encounter many implementation and system related issues which did not show up in simulation. This would help to improve WiFiRe protocol's correctness. We describe maintenance of soft timers, packet classification, VoIP on wireless, QoS and few other WiFiRe specific issues in subsequent sections.

(B) We exploit the Ethernet network's parameters like frame size, MTU, collisions, propagation delay, NIC to emulate a wireless link, even though they are different from 802.11. Such a link can be used to test new wireless protocol proposal, in absence of wireless hardware.

[2]DIX frame (named after DEC, Intel, and Xerox) is a commonly used Ethernet standard today and has 16-bit type field

II. Architecture

A. WiFiRe MAC testbed

Implementation of WiFiRe MAC is done using C sockets on Linux platform. Program runs in user space (see section II-E for a discussion). BS and ST are Linux terminals with two NICs each. Network parameters like IP address, ARP cache, DNS, proxy settings are kept fixed to keep the implementation simple.

WiFiRe testbed (Figure 3) includes a BS with two NICs, an ST with two NICs, a server which acts as gateway, FTP, proxy and web server. Several clients are connected to ST through Ethernet switch. BS is connected to ST using DIX based Ethernet cross cable. This link is treated as wireless link of WiFiRe. Subsequently, it will be replaced with the hardware for WiFiRe PHY. ST's *eth0* is connected with end users via switch. BS is connected to server using *eth0* and sends data to ST by other interface (*eth1*). *eth0* of ST and BS have standard 802.3 MAC based connectivity and it makes the network transparent for end users and Server (S) respectively. End-user clients send packets to ST, ST follows WiFiRe MAC slot structure defined in the UL-MAP and sends packets to BS in designated slots. BS receives these packets, processes them using WiFiRe modules and forwards them to S as normal Ethernet packets. Server sends reply to BS, which forwards it to appropriate ST and finally, client receives the packets.

B. Assumptions and limitations

- Both BS and ST are connected using a cross cable, which eliminates problems of propagation delays, ranging, synchronization and other such wireless specific issues. We assume that such issues will be taken care by underlying hardware in actual implementation.

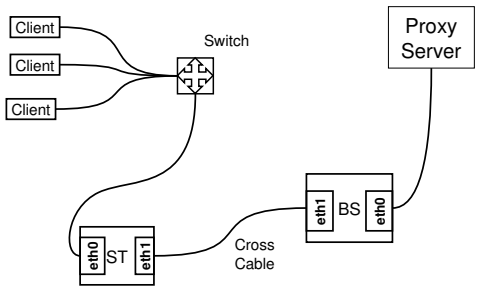

Fig. 3. WiFiRe Test Bed

- Implementation does not perform real ranging procedure as test-bed uses Ethernet cable where propagation delay is not variable. In this scenario, Beacon transmission is enough for synchronization. Purpose of ranging here is to assign and transfer basic and primary CIDs only.
- As our emulation runs on Ethernet, and PCs are connected directly using RJ-45 cables, it cannot get total flexibility on frame structure, size, CRC etc. This Ethernet link restricts size of individual packets and WiFiRe MPDUs as well.
- Due to absence of real hardware clock, software timers are used in implementation which guarantees precision only up to milliseconds.

C. Modules at BS and ST

As shown in Figure 4, there are various modules in BS dedicated to different functions of WiFiRe MAC. ST modules are explained in next paragraph but diagram is omitted due to space constrains. As expected, BS and ST have significantly different functionality in WiFiRe MAC. BS MAC contains packet classifier, CID generator, packet controller, scheduler and timer clock. Packet classifier detects packet type, QoS requirements and connection details. It makes appropriate entries in the mapping table (like BS_TABLE or MC_TABLE). CID generator module generates 16-bit CID for each new connection. These 16 bits are divided in subgroups, such that each group of bits represents ST-ID, Type of Service (ToS) and client. Packet controller sends packets to outgoing queues, buffer queues or simply drops them based on MAC requirement. Scheduler keeps track of all connection's requirements and generates DL-MAP and UL-MAP. Both MAPs are sent in first slot of downlink segment to all STs. Scheduler keeps dedicated slots for real-time traffic like VoIP. Timer clock reads UL-MAP and generates a sequence for packet reception.

ST has simpler MAC compared to BS. ST has MAP parser, packet controller, connection classifier and CS layer. MAP parser reads both MAPs and prepares ST for uplink and downlink slots. Packet controller maintains incoming and outgoing queue, filled with raw IP packets. Packet controller takes inputs from parser for packet sequence. Connection classifier detects packets coming from different end-users and makes new dynamic service addition (DSA) request for each new connection. CS layer understands Ethernet and WiFiRe header, converts packets appropriately from WiFiRe to Ethernet and vice-versa.

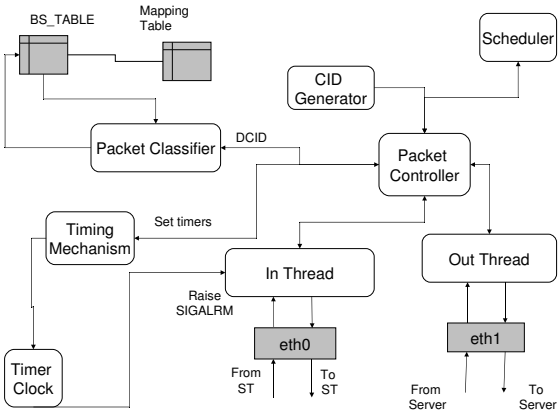

Fig. 4. WiFiRe modules in BS

BS and ST have similar modules for packet transmission and reception, CID tables and buffer queues. Implementation of these modules is done using well-known strategies [7].

D. Control and Data flow for various events

Initially BS comes up and starts execution of its routine procedure of beacon broadcast, which is handled by packet controller. Whenever an ST comes up, it starts listening for a beacon. ST will go through ranging procedure, and ST finally get registered. ST is now ready to serve its client's requests, which were not served earlier. Timing mechanism module generates local sequence based on timer from DL-MAP/UL-MAP. This timing sequence is fed to the Clock to raise interrupts in the beginning of designated slots to wake up ST at required slot.

Each ST maintains ST-TABLE which includes MAC address of client, CID and ToS. Each row of the table indicates unique connection from ST to BS. BS also maintains two tables: BS-TABLE and MC-TABLE. BS-TABLE entry includes BSID (indicates the sector), STID (indicates the ST) and CID while MC-TABLE entry includes CID, client IP and ToS. Here, we give an example (Figure 5) of typical HTTP transaction to illustrate actions taken in WiFiRe MAC at ST and BS.

Whenever ST receives Ethernet packets from client(s), following actions are performed in uplink:

1) Packet classifier checks if received packet belongs to any existing connection by scanning table entries. If matched, goto to step 4. If no match found, then make new entry in the table with temporary CID.
2) ST creates DSA request and waits for DSA response from BS. Ethernet packet will be stored in the buffered queue till the response arrive.

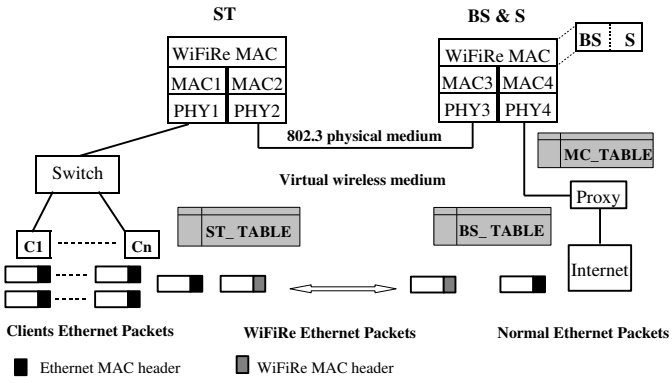

Fig. 5. Layering approach in WiFiRe emulation

3) Once ST receives DSA response, it updates the table entry in ST-TABLE with newly allocated CID.
4) ST removes existing Ethernet header from the packet. ST's CS layer creates a WiFiRe header (based on CID of the table entry) for raw IP packet PDUs.
5) ST transmits WiFiRe packet to BS through outgoing socket according to UL-MAP.

Action at BS:

1) BS captures the packet from *eth1* (Figure 3) using socket.
2) BS reads the WiFiRe header and updates the table with IP address to CID mapping entry.
3) Packet controller removes the WiFiRe header after making table entry and constructs an Ethernet MAC packet for proxy.
4) Transmits Ethernet packet to proxy, which ultimately forwards these packets to Internet and responds back to BS.
5) BS buffers the responses and send them in downlink frame according to DL-MAP.

Similar steps are followed for downlink as well.

E. Discussion

Implementation is done in C sockets instead of kernel for following reasons:

- A user level communication component, which provides direct access to low level communication (like putting bytes directly on PHY) mechanisms, bypasses operating systems complexity (like interrupts, sock-buff) in critical paths of communication.
- Overheads of kernel traps and memory copies along with various dependencies between user space and kernel space are avoided.
- To develop a prototype when WiFiRe protocol still evolving and changing, it is difficult to change code and debug in kernel module frequently.

III. Integrating with hardware

In a typical implementation, such as WiMAX (802.16d), all the modules (MAC, RF, antenna, modem, and memory) are in same *box* [7]. Every component is closely coupled with each other and they exchange configuration parameters with each other. For example, memory has driver and API through which, it can store/retrieve data to/from buffer. This buffer is shared by MAC and PHY.

In current WiFiRe implementation plan, WiFiRe MAC and BS-PHY are separate from each other (Figure 6). PHY boards are being developed independently and require MAC frame to be delivered on Ethernet cable. Eventually, MAC and PHY will be integrated as single entity. Current hardware keeps six different NICs for six BSs. These six NICs are connected by single switch.

This leads to Ethernet cable as single communication medium. All the non-WiFiRe control messages such as 2 byte control data (explained later in this section) have to be transferred using same link. Extra control modules have to be written to handle such messages which adds to the complexity in implementation. WiFiRe frames are to be sent to this switch with destination address as a given BS-PHY MAC address. Frame structure and slot size is fixed. There is a centralized module which instructs all six BS-PHYs to synchronize with each other and send data as and when required.

In our approach both BS-PHY and BS-MAC are separate from each other (Figure 6). There are various ways (broadcast, indexing, bitmap and consecutive allocation) to transmit a frame which is built by WiFiRe MAC from

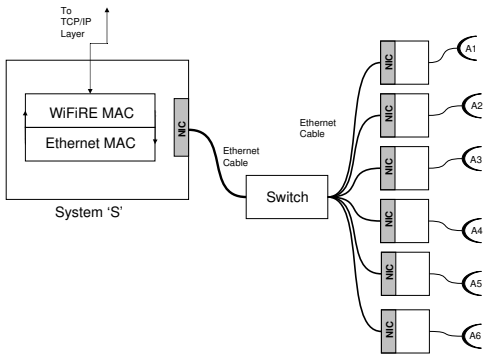

Fig. 6. BS hardware: proposed WiFiRe real system component

BS-MAC to respective BS-PHY such that each BS-PHY should know when to transmit the data frame.

We also need to optimize the PHY overhead, which is of 4 slots, for every new transmission at BS-PHY. Among the available methods, concatenation has low PHY overhead. Using concatenation we allocate all slots which belong to same BS consecutively, thereby reducing PHY overhead. We use a 2 byte control packet per DL-TB (down link transport block), having information <starting slot, number of consecutive slots> and then transmitted to BS-PHY. At BS-PHY, it reads the information present in control packet and acts accordingly.

Figure 7 shows how this approach works. Total DL frame is divided into groups (B1, B3, B5) and each group belongs to the respective BS-PHY. Beacons are transmitted to every BS-PHY. Down link Transport Blocks (DL-TB is a group of slots which belongs to the same BS. Each BS will have zero or more DL-TBs and this number depends on Ethernet packet size.) are created before inserting the WiFiRe frame into Ethernet packets.

For example, when BS-PHY receives an Ethernet packet having a control packet <05,15>, it transmits from slot 5 to 19. This method has advantage of fixed size memory requirement (PHY buffer need to store single frame at any given time) at BS-PHY, minimal control and PHY overhead. However it increases the complexity of scheduler. Scheduler has to take care of constructing DL-TBs.

IV. CHALLENGES

In this section, we describe a few challenges that we faced during implementation of WiFiRe and our future directions.

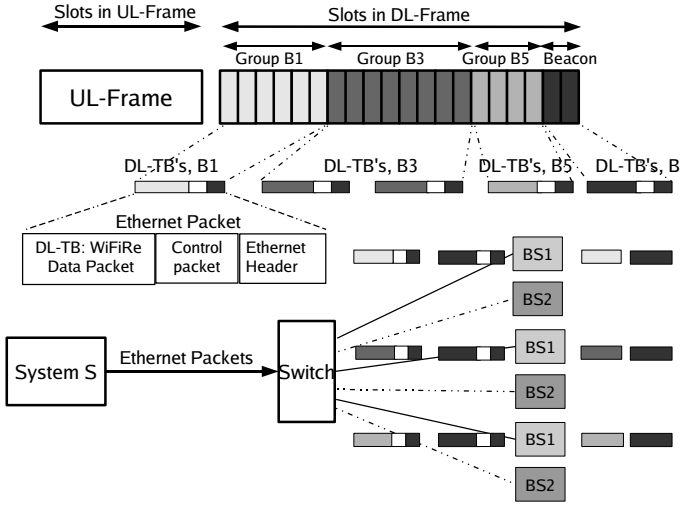

Fig. 7. Consecutive allocation of DL slots for each BS

Detecting different traffic classes for QoS is not trivial. We created TCP-port based classification for RTP packets. We used 802.1Q aware clients and detected packets using MAC headers where cross-layering was not possible.

BS and ST in our testbed are supposed to work as layer-2 devices. Because of Linux kernel inside BS and ST, services (like ARP and DNS) generate periodic packets which will get passed to WiFiRe NICs directly. These packets are misinterpreted as WiFiRe packets. Disabling these services is not an option, as they are shared by other 802.3 based NIC. Additional WiFiRe modules to drop such packets were inserted into WiFiRe MAC.

If we consider G.729 codec, for 10 ms voice, there is 40 byte of header (RTP, UDP and IP) and 10 bytes of voice data. For WiFiRe system, this overhead is very high and some solutions are considered. VoIP gateways, which add headers for outgoing packets and remove headers from incoming packets and VoIP compression using advanced codecs (like GIPS and iLBC) have been considered as an option for future implementation.

V. CONCLUSION

In this paper we have described an implementation of WiFiRe as a LAN emulation over an Ethernet test bed. WiFiRe packets can be monitored over the link between BS and ST to verify factual working of the protocol. We have tested it for multiple clients with single ST system.

We have done emulation for one sector, for single BS and ST but it can be modified easily to accommodate multiple sectors and this work is underway.

REFERENCES

[1] Bhaskar Ramamurthi, Anand Kannan, Ashok Jhunjhunwala , "Interim 3.5G broadband wireless system for India: Framework, requirement, performance needs," *CEWiT Tech Report*, 2005.

[2] Sridhar Iyer, Krishna Paul, Anurag Kumar, Bhaskar Ramamurthy, "WiFiRe: Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *CEWiT Tech Report, India*, 2006.

[3] P Bhagwat, B Raman, D Sanghi, "Turning 802.11 Inside-Outs," *ACM SIGCOMM Computer Communication Review*, 2004.

[4] Rabin Patra, Sergiu Nedevschi etc, "WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks," *USENIX NSDI*, April, 2007.

[5] J Chandarana, Sravana K, S Perur, R Rangarajan, S Sahasrabuddhe and S Iyer , "VoIP-based Intra-village Teleconnectivity: An Architecture and Case Study," *WISARD, COMSWARE*, 2007.

[6] Anitha Varghese, "Design of a TDD Single Channel Multisector TDM MAC for a Single Cell WiFiRE System," *MTech Thesis, IISc Bnagalore*, 2006.

[7] Intel Corporation, "Intel PRO/Wireless 5116 Broadband Interface," 2007.