RODEO: Robust and Rapidly Deployable TDM Mesh with QoS Differentiation

Aniesh Chawla*§, Vinay Yadav*§, Vasu Dev Sharma†, Jitin Bajaj‡, Eshan Nanda*, Vinay Ribeiro* and Huzur Saran*
§Joint First Authors
*Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India.
Email: {chawla.aniesh,vinayyadav.iitd,eshannanda31888}@gmail.com, {vinay,saran}@cse.iitd.ernet.in
†Tower Research Capital; Email: vdsiitd2005@gmail.com
‡ Computer Science Department, UCLA, USA; Email: jbajaj@cs.ucla.edu

***Abstract*—We present RODEO, a RObust and rapidly DEplOyable wireless mesh network designed for applications such as disaster management. Natural disasters often destroy existing communication infrastructure, thus forcing rescue teams to deploy their own networks for communication. Such networks must be rapidly deployable, capable of recovering from node failure, have a large coverage area, and at the same time meet the stringent QoS requirements of voice, video and data. RODEO employs TDM scheduling for efficient and predictable use of resources. By employing multiple frequency channels in different clusters, it simplifies synchronization and scheduling, and also trivially allows simultaneous concurrent transmissions in the network. With the help of a fast network entry protocol, it self-configures quickly and also recovers from node failure. In addition, RODEO provides differentiated QoS for various applications. We validate RODEO through experiments conducted on a 4-node prototype built on the Wireless Open Access Research Platform (WARP) platform and present results for throughput, jitter, QoS differentiation, and failure recovery.**

I. Introduction

Natural disasters, such as tsunamis, earthquakes, landslides and floods, often destroy existing communication infrastructure, thus forcing rescue teams to deploy their own networks for communication. Networks in disaster zones must possess several features in order to effectively serve their purpose. First, they must be *rapidly deployable*, preferably on the order of minutes, and not require much technical expertise to setup. Long delays in bringing up the network can hamper rescue efforts, thereby potentially putting lives at risk. Second, they must have a *large coverage* area which must encompasses the entire zone over which the team will fan out. Third, they must be rugged, being able to withstand harsh physical conditions and be *robust to failure* of few network nodes. Fourth, they ideally must support the stringent *QoS requirements* of voice, video and data. A rescue worker may want to stream live video of his surroundings to a central command and receive VoIP instructions in real-time. These design features also apply to military battlefield scenarios where infrastructure may not have existed in the first place or was destroyed in battle.

An obvious consequence of the rapidly deployable criterion is that the network must be wireless. A tower, possibly

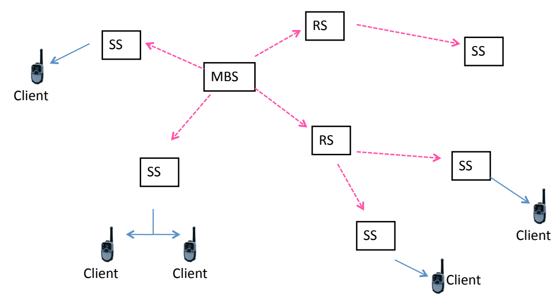

Fig. 1. Tree-structured topology for the backbone. The main base-station (MBS) forms the root, the subscriber stations (SS) the leaves, and relay stations (RS) the rest of the nodes. Clients are connected to backbone nodes through a second tier network.

vehicular mounted, housing a base-station can be erected at a central location and rescue workers with wireless handsets can communicate through it. In case the target coverage area is too large for a single base-station to reach, one can extend the network using relays or wireless routers to form a multi-hop wireless network also termed a *mesh* network.[1] Alternatively, one can go in for a satellite-based network, in which users communicate via low-earth orbit satellites. Terrestrial networks, however, have a significant advantage over satellite networks in terms of performance and cost due to the shorter communication distances involved. Our focus will hence be on terrestrial mesh networks. Note that although mobile ad-hoc networks satisfy the first three criteria of rapid deployability, large coverage area, and robustness, because of rapid topology changes they are ill-suited to provide guaranteed QoS for voice, video, and data applications.

We envision a two-tier wireless network for disaster management (see Figure 1). The first tier consists of a long-range backbone mesh network consisting of high-power transmitter nodes covering several tens of square kilometres. The nodes in this network are initially taken to their designated geographical positions following which the network self-configures. After

[1]We distinguish a mesh network from a mobile ad-hoc network (MANET). In a mesh network, nodes are essentially static once setup in their locations. In MANETS nodes are highly mobile.

978-1-4673-0298-2/12/$31.00 © 2012 IEEE

this, the nodes are more-or-less stationary and at most move in a nomadic fashion. The second tier consists of low-power devices carried in backpacks or handheld by rescue workers. These act as subscribers to the nearest backbone node which performs the role of their base-station. In a more general case, the second tier nodes can form a MANET or mesh network among themselves.

In this paper we focus on the design and implementation of the tier-1 backbone mesh. Tier-1 nodes must preferably be equipped with omnidirectional antennas. Omni-directional antennas have two advantages over directional ones in the present context. First, they require less time for alignment and hence are better suited for rapid deployment. Second, the target coverage region of a base-station in a disaster zone may not be known a priori and hence it is better to go in for circular coverage area provided by omnidirectional antennas than a sectored coverage area resulting from directional antennas. On the other hand, however, omnidirectional antennas have shorter communication range than directional ones. This shortcoming can be partially obviated by increasing transmit power and using lower frequencies for extending range [1].

We christen our network RODEO, which is short for RObust and rapidly DEplOyable TDM mesh. RODEO, to the best of our knowledge, is the first wireless network with a prototype implementation that has all of the following features. Section VI gives a more detailed comparison with other implementations [2]–[8].

1) Rapidly deployable: A node, say X, which has not yet joined the mesh network initially listens for SYNC messages from the mesh. On receiving a SYNC message from node Y, the new node X performs a network entry operation and becomes a child of X. This procedure results in a tree-structured network. The network entry procedure for each node is fast, of the order of tens of milliseconds in our implementation. As a result, RODEO is rapidly deployable. Details of the network entry procedure are provided in Section II.
2) Large coverage area: By their very nature, mesh networks have a large coverage area compared to point-to-multipoint wireless networks.
3) Robust to node failure: When a particular RODEO node fails, its descendants in the tree rejoin the network using the network entry procedure to reconstruct a new tree.
4) Multiple channels to reduce interference: Each RODEO node uses a different channel to communicate with its children than that used by other nodes. Consequently, several nodes can simultaneously transmit thereby improving the efficiency of the network. Since we confine ourselves to single transceiver nodes, each node must switch between two channels, one to communicate with its parent and the other to communicate with its children. The impact of using multiple channels on network entry and on the MAC are described in Sections II and III respectively.
5) TDM scheduling: We employ a TDM MAC which allows reservation of time-slots for communication between nodes. Well-designed TDM MAC protocols can circumvent the collisions and hidden terminal problems inherent to CSMA-based MAC protocols which degrade performance [9]. TDM MACs however require accurate synchronization between mesh nodes. We achieve this using SYNC messages embedded in the MAC frame structure. These and other MAC details are found in Section III.
6) QoS differentiation: In RODEO we give scheduling priority to packets of certain applications over others, which is elaborated in Section III-B.

We have built a prototype of RODEO using the wireless open-access research platform (WARP) [10]. The WARP platform is a software-defined radio that can be used in stand-alone mode, has an associated open source software repository which includes an implementation of orthogonal frequency division multiplexing (OFDM), and has the computational power to support data rates of several tens of Mbps. We describe our implementation in Section IV and present results from a four node testbed in Section V.

II. Network Architecture Overview

In this section, we describe RODEO's network topology, its use of multiple channels, and its network entry and routing protocols.

A. Topology

RODEO forms a tree-structured backbone topology as shown in Figure 1. All control and data communication takes place along branches of this tree. Every node in the tree has a unique node ID. Nodes are classified as follows.

1) *Main Base Station (MBS)*: The root node is called the Main Base Station (MBS). The MBS synchronizes the whole network by generating a SYNC packet which propagates down the tree.
2) *Relay Station (RS)*: All nodes responsible for relaying information (whether data or control) to nodes not in the direct range of the MBS over multi-hops are termed relay stations (RS). A relay node has a parent and at least one child in the tree.
3) *Subscriber Station (SS)*: We refer to all leaf nodes of the tree as subscriber stations. An SS can be directly connected to the MBS or to an RS. Whenever a new node joins the network as a child of an SS, the SS node becomes an RS.

Clients can be connected directly to one of the backbone nodes through wired or wireless media, or through a tier-2 network.

B. Multiple channels

Each parent node and its immediate child nodes together form a *cluster*. We call the parent node the clusterhead.

Every cluster uses a different frequency channel for communication. One advantage in doing this is that it greatly simplifies scheduling. Any clusterhead can schedule a transmission in a particular slot to or from one of its children independently of surrounding transmissions, provided that it is itself not scheduled to communicate with its own parent in that slot. Note that in a mesh network where all nodes use the same channel, before scheduling a transmission in a particular slot, one must take into account all other scheduled transmissions in that slot within an interference vicinity. To do this correctly, one needs an accurate interference map as well as the entire existing schedule within the vicinity. For example, this vicinity is assumed to be two hops in the 802.16e mesh mode [11]. Note that all RS nodes belong to two clusters and hence must switch between the channels of these clusters as and when required. We refer to the channel of a particular cluster as the *clusterhead channel* of the parent node in that cluster.

Another advantage is that the synchronization does not have to be network-wide, unlike the case where all nodes use the same channel. Each node only has to be synchronized with its parent in the tree to ensure that it can communicate with it. In the case where all nodes use the same channel, a delayed transmission in a particular time-slot in one part of the network may interfere with a scheduled transmission in the next time slot in another part of the network. Hence all nodes must tightly track some global reference clock to eliminate inter-slot interference.

A disadvantage of using different channels in different clusters is that the maximum number of nodes in the network equals the total number of available channels. This disadvantage can be remedied by allowing reuse of channels in clusters which are separated by a large distance. Doing this will require detailed information of interference maps [12], which is difficult to obtain quickly in rapidly deployable scenarios.

C. Network Entry Protocol

A node uses the network entry protocol in two cases: (i) when it initially joins the network, and (ii) when it get disconnected from his parent and has to rejoin the network. The protocol must be fast and automated in order to ensure rapid deployability and robustness to node failure.

Every node which has already joined the network acts as a clusterhead and sends out a SYNC message in a designated slot in every TDM frame and in its clusterhead channel. For simplicity, assume that one node is designated as the MBS and that it transmits a SYNC at a designated frequency f_0. A node desiring to join the network has a list of all frequency channels which can be used in the network. It starts scanning for SYNC messages starting with f_0. If it does not receive a SYNC within a time interval T (set to a multiple of the maximum frame size), it switches to the next frequency in the list. On receiving a SYNC from a clusterhead, the node registers with that cluster by sending a network entry (NE) packet in the initial ranging slot of the frame and receives a confirmation piggybacked on the next SYNC. The node replies in the allotted slot with an NE DONE packet, acknowledging its network entry. Thus network entry can be as short as two frame durations.

Note that the clusterhead allots two slots to the new node in the frame, one in order to transmit to and the other to receive data from it. The confirmation message it sends to the new node indicates which slots these are in the TDM frame.

In case the MBS itself fails, all nodes will not receive a SYNC message at any frequency. They must hence perform a distributed leader election algorithm to choose a new MBS. Our first prototype of RODEO does not incorporate such a distributed election algorithm.

D. Routing Protocol

Because RODEO uses a tree to route packets, there is only one possible path between any pair of source and destination nodes. Hence the question of choosing an optimal path from among several available paths does not arise.

In order to forward a packet correctly, a node needs to know whether to send it up the tree to its parent, down the tree via one of its children, or to send it to a client via the tier-2 network. It makes its forwarding decision based on the following tables.

1) *Client Table:* This table contains a list of the IP addresses of the clients connected to this node.
2) *Next-hop Table:* This table has two fields: (i) a destination IP field and (ii) the corresponding next-hop node ID. The table has entries for each client in the sub-tree of this node. The default entry (the last entry) in this table, for all nodes except for the MBS, has its parent as the next-hop.

Whenever a new client gets connected to any of the nodes, the corresponding node adds this client to its Client table and then sends a control message (with the IP address of the client) to its parent node, which then forwards it to its own parent and so on till the message reaches the MBS. Each node that receives this packet creates (or updates) an entry in its Next-hop table with the destination IP set to the IP in the control message and the next hop set to the node ID of the node from which it has received this message.

Whenever a node detects that one of the clients has left the network (this is known after a certain number of Hello packets are not acknowledged by the client), it sends a control message meant for deleting the corresponding entry from the Next-hop table of all parent nodes in the routing tree. On receiving this delete message all the intermediate nodes remove the entry corresponding to this client IP from their tables.

When a node gets a request for transmission from a source client to a destination client IP, it searches for the destination IP in its Client table. If the destination is a client attached to this node (via the tier-2 network), then it forwards the packet to the node (via the tier-2 network). If there is no match in the Client table, then the node searches for the destination IP in its Next-hop table and forwards the packet to the appropriate next-hop node ID (the default being its parent node). If when the packet reaches the MBS, it does not have the desired destination IP in its Next-hop table then the packet is dropped and MBS sends a control packet to the source node informing it that the desired destination does not exist.

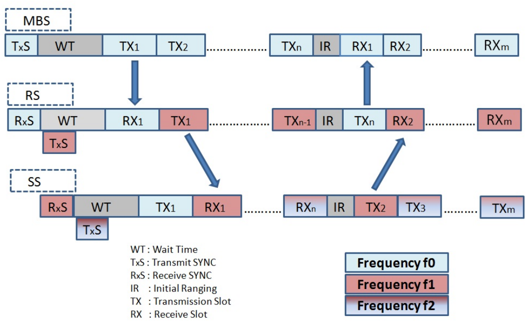

Fig. 2. Frame Structure

III. TDM MAC

This section describes RODEO's MAC frame structure as well as its QoS differentiation mechanism. RODEO's frame structure and synchronization is similar to the non-transparent mode of operation of the IEEE 802.16j standard [13]. As of yet, we are unaware of any prototype implementation of 802.16j.

A. Frame Structure

The frame structure for the MBS, RS, and SS nodes is depicted in Figure 2. The MBS begins the frame by sending out a SYNC message which is received by its children who immediately switch to their clusterhead channels and send out SYNC messages themselves. In this way all nodes know when the frame begins.

Various slots and their functions are given below.

1) **SYNC Slot**: The SYNC packet generated by MBS and relayed by other nodes is sent in this slot.
2) **WAIT Slot**: This slot immediately follows the SYNC slot. In this slot nodes wait for the SYNC packet to percolate down to the bottom of the tree.
3) **TX Slot**: During a TX slot, a node transmits data either to its parent or child. The size of a TX slot is flexible and can be changed to accommodate more or less data.
4) **RX Slot**: During an RX slot, a node receives data either from its parent or child. This slot size is also flexible.
5) **Initial Ranging (IR) Slot**: The IR slot is used by nodes in the network entry protocol as described in Section II-C. The position of this slot in the frame is sent in the SYNC packet. The size of this slot is the same as that of one SYNC slot.

B. QoS Differentiation

Recall from Section II-C that every node has one slot reserved to transmit to and one slot to receive data from its parent in every frame. It can so happen that several packets await transmission in a particular TX slot. RODEO maintains two FIFO queues for each TX slot. Higher priority packets are stored in the first and lower priority ones in the second. The two queues are serviced using *strict priority scheduling*, that is, packets are serviced from the second queue only when the first is empty.

IV. Prototype Implementation

We have built a four node prototype of RODEO using WARP boards. We describe the WARP platform and other details of our implementation in this section.

A. Platform

To build our prototype, we wanted a platform that could operate at the speed of a hardware solution and also provide the flexibility of a software solution. We eventually chose Rice University's WARP, a scalable and extensible programmable wireless platform. Our prototype uses the OFDM reference design available in the WARP repository.

The WARP motherboard is equipped with a Xilinx Virtex-4 FPGA (older versions use the Virtex-2) and can be interfaced to 4 daughterboards. Each RF daughterboard operates in the 2.4GHz and 5GHz ISM bands.

B. Frame and Slot Duration

We implement QPSK modulation in all OFDM sub-channels over a 10MHz channel. Given various parameters of the WARP RF implementation, we find that a maximum sized MAC packet of 1488 bytes can be transmitted in 1056μs. We introduced a guard interval of 150μs (60μs is time for a node to move from receive to transmit mode and 90us to take any propagation delays) in each TX/RX slot, which increased its size to 1206μs.

C. Testbed

The test bed for our experiments consists of 4 nodes in a topology illustrated in Figure 3. Each node consisted of a WARP board with a client PC attached through it via Ethernet.

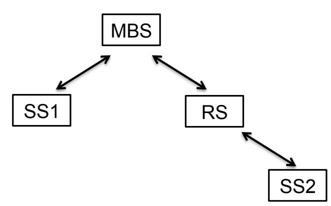

Fig. 3. Testbed for experiments

The frame structure for four nodes in our test bed consists of a SYNC slot, an IR slot, a WAIT slot, 2 TX slots, and two 2 RX slots. The WAIT slot duration is set as 300μs. Thus overall the each frame is of about 5 milliseconds duration. The duration of the SYNC and IR slots are a few tens of μs each, and do not contribute much to the frame duration.

V. Performance Evaluation

We perform experiments to determine the maximum single-hop and two-hop throughput possible in the testbed shown in Figure 3. In addition, we reduce the TX/RX slot size and evaluate its impact on RTT. Finally we provide QoS differentiate between TCP and UDP packets and demonstrate how it reduces jitter for the higher priority data. In our experiments we employ the non-overlapping and interference free channels in the 5GHz band. The various nodes were placed a few metres apart in an indoor lab setting.

We note that in our lab setting if the SS2 is made to start scanning the channels from the base frequency f_0, then it will join the tree with the MBS as its parent and we will not have the topology as shown in Figure 3. We hence make SS2 start scanning frequencies from the clusterhead channel of the RS and place f_0 second on its list of frequencies to be scanned.

A. Throughput over a single hop

Recall that a single slot in the frame is used for transmission from one node to another. Taking guard intervals and framing headers into account, we obtain the maximum transmission rate over a single hop to be 2.16 Mbps. The raw bit-rate on the link for QPSK modulation is 11.22 Mbps.

Table I captures the UDP throughput results for single hop communication. The result are the same no matter which two neighboring nodes are considered for the experiment. We used `iperf` to vary the offered UDP load while keeping packet size constant at 1484 bytes. The upper limit for single hop throughput was found to be 2.11 Mbps and offered loads higher than that result in increased losses. This throughput is very close to our calculated upper bound.

Offered Load (Mbps)	Throughput (Mbps)	Error (%)	Jitter (ms)
1.0	1.0	0.9	5.71
1.5	1.49	0.4	7.49
2.0	1.99	0.5	7.07
2.12	2.11	0.5	6.87
2.5	2.11	15	7.3
3	2.11	30	7.27
4	2.11	47	8.24

TABLE I
UDP TEST RESULTS(ONE HOP)

The TCP throughput was marginally less and was found to be 1.97 Mbps.

B. Throughput over two hop

Table 2 depicts the UDP results for a two-hop communication (MBS-RS-SS2). We experimentally found the upper limit for UDP throughput to be 1.39 Mbps. Note that in theory the maximum throughput for a two-hop communication is the same as that of the one-hop. In every frame, the MBS has one slot reserved for transmission to the RS, and the RS has one slot reserved for transmission to SS2 (at a different frequency).

We speculate that this discrepancy between single-hop and two-hop throughput is likely caused due to a computational and frequency switching overheads at the RS node. Note that the RS needs to change channels within a frame, strip and recreate packet headers etc. As part of future work we will identify the exact cause of this discrepancy. The TCP throughput was found to be 1.2 Mbps for the same two-hop path.

Offered Load (Mbps)	Throughput (Mbps)	Error (%)	Jitter (ms)
1.0	1.0	0.22	4.98
1.2	1.2	0.4	5.29
1.3	1.3	0.34	5.07
1.4	1.39	0.25	7.07
1.5	1.38	6.3	10.3

TABLE II
UDP TEST RESULTS(TWO HOP)

C. Slot size granularity

The time slot granularity affects performance in more than one way. First, small data packets can be transmitted in a short duration of time. Thus using large time-slots to transmit small packets is inefficient. Second, reducing each time-slot size shortens the duration of each frame, provided the overall frame structure is kept the same. Smaller frames translate into shorter round-trip-times (RTT) which can benefit applications such as VoIP.

Figure 4 plots round trip time (RTT) for *ping* packets over a single-hop (MBS-SS1-MBS) for various slot sizes. Observe that as we decrease the slot size, the RTT decreases, though not in a linear fashion as expected. In addition, we found that our time slot can be decreased to as low as 100μs. Note that we decreased guard interval per TX/RX slot to 60μs while conducting this experiment.

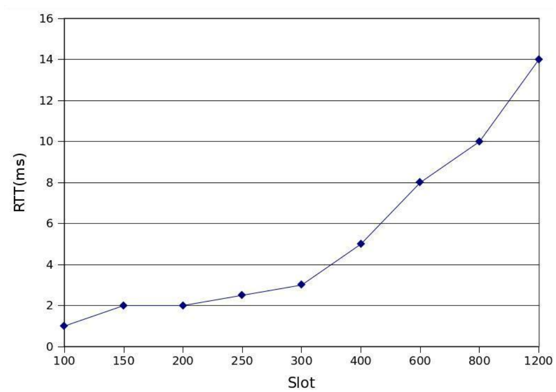

Fig. 4. Slot size (in μs) vs. RTT (in ms)

D. Quality of Service

We finally prove the efficacy of RODEO's QoS differentiation. Using iperf, we transmit UDP and TCP traffic simultaneously between the MBS and SS1 (see Figure 3). In one experiment, no priority is given to any traffic, that is packets are serviced in FIFO order irrespective of their transport protocols. In another experiment, the MBS QoS scheduler identifies the transport protocol of each packet from the protocol field in the IP header, and gives strict priority to UDP over TCP traffic.

Figure 5 compares the jitter of UDP traffic for various UDP offered loads for the two cases. Observe that when UDP is given strict priority, its jitter remains low unlike the case when it is not given priority over TCP traffic. Similar results were obtained when the strict priority ordering was reversed, that is TCP was given priority over UDP.

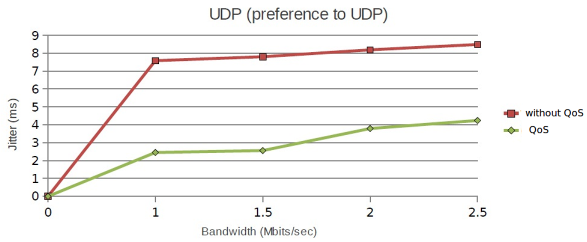

Fig. 5. Jitter vs. Offered Load when UDP is given strict priority

E. Robustness to node failure

To test if RODEO is robust to node failure, we streamed video using VLC from the MBS to SS2 via RS and then switched off the power of the RS node. SS2 then rejoined the network directly at MBS and the video continued to stream to SS2. No visual glitch was observed at SS2 during this switch over.

VI. Related Work

We here describe some other TDM-based mesh networks with prototype implementations which have been developed in the past few years. *TDM MAC* [2] implements a multi-hop network using proprietary hardware and uses in-band synchronization like RODEO. However it does not use mutliple channels as we do. In addition, its synchronization using beaconing differs from ours. *PIP* [14] uses multiple channels in its relay stations like RODEO. It has been designed for sensor networks and in particular unidirectional flows, whereas our requirement is for bidirectional voice, video, and data transfer. *Soft-TDMAC* [3] is a TDMA protocol built using commodity WiFi hardware. It achieves synchronization of the order of few μs and employs 16μs sized time slots. It does not use multiple channels like RODEO. *Overlay MAC* [4] is more an overlay layer on already exiting MAC and not a complete MAC layer itself. Thus though being a TDMA-based MAC it has to take care of collisions in the network because of the underlying CSMA. *FreeMAC* [5] is a kernel based approach built over Mad Wifi using a software TDMA MAC. It uses an out-of-band synchronization technique unlike RODEO. Like RODEO it uses multiple channels for data transmission in the network. *WiLDNet* [6] uses the directional antennas in order to increase its range. It is specially designed for very long range networks. *FRACTAL* [7], [8] is a mesh network specially designed for rural communication and uses commodity WiFi hardware. It uses a tree-based topology for synchronization like RODEO. Unlike RODEO, it uses arbitrary paths in the mesh for data communication and does not use multiple channels for communication. *LiT MAC* [12] uses TDM scheduling, and multiple channels for transmission like RODEO. It differs from RODEO in that it is connection-oriented and uses centralized dynamic scheduling.

VII. Conclusions and Future Work

We have in this paper described the design and prototype implementation of RODEO. RODEO has several features, such as rapid deployability and TDM scheduling, which make it apt for applications such as disaster management. Although RODEO was designed for disaster management and military applications, it can be used for other urban and rural applications as well. If we relax the requirement of rapid deployability, then we can employ directional antennas to improve peformance.

Our prototype implementation on WARP boards validates RODEO. The code for our implementation will in due course be added to the WARP software repository to benefit the research community.

As future work, we will perform outdoor experiments and test RODEO's long-range capabilities. We will experiment with sub-GHz bands and power amplifiers to achieve better range. The latest PHY layer coding schemes will be incorporated to reduce bit error rate.

Acknowledgment

This work was funded through a research grant from the Department of Information Technology, Government of India (RP02177).

References

[1] S. Deb, V. Srinivasan, and R. Maheshwari, "Dynamic spectrum access in DTV whitespaces: design rules, architecture and algorithms," ser. MobiCom '09. New York, NY, USA: ACM, 2009, pp. 1–12.
[2] D. Koutsonikolas, T. Salonidis, H. Lundgren, P. LeGuyadec, Y. C. Hu, and I. Sheriff, "TDM MAC Protocol Design and Implementation for Wireless Mesh Networks," *CoNEXT*, 2008.
[3] P. Djukic and P. Mohapatra, "Soft-TDMAC: A Software TDMA-based MAC over Commodity 802.11 hardware," *INFOCOM*, 2009.
[4] A. Rao and I. Stoica, "An overlay mac layer for 802.11 networks," in *ACM MobiSys*, Seattle,WA, USA, Jun. 2005, pp. 135–148.
[5] A. Sharma and E. M. Belding, "FreeMAC: Framework for Multi-Channel MAC Development on 802.11 Hardware," *PRESTO*, Aug. 2008.
[6] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer, "WiLDNet: Design and Implementation of High-Performance Wifi-based Long Distance Networks," in *Proc. NSDI*, Cambridge, MA, USA, Apr. 2007.
[7] K. Chebrolu and B. Raman, "Fractel: a fresh perspective on (rural) mesh networks," ser. NSDR. New York, NY, USA: ACM, 2007, pp. 8:1–8:6.
[8] A. Dhekne, N. Uchat, and B. Raman, "Implementation and evaluation of a TDMA MAC for WiFi-based rural mesh networks," *NSDR*, 2009.
[9] M. Garetto, T. Salonidis, and E. Knightly, "Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 4, pp. 864 –877, Aug. 2008.
[10] "Rice University, WARP Project," http://warp.rice.edu.
[11] P. Mogre and M. Hollick, "The IEEE 802.16-2004 MESH Mode Explained," Multimedia Communications Lab, Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Merckstr. 25, 64283 Darmstadt, Germany, Tech. Rep. KOM-TR-2006-08, Dec 2006.
[12] V. Gabale, B. Raman, K. Chebrolu, and P. Kulkarni, "LiT MAC: addressing the challenges of effective voice communication in a low cost, low power wireless mesh network," ser. ACM DEV '10. New York, NY, USA: ACM, 2010, pp. 5:1–5:11.
[13] V. Genc, S. Murphy, Y. Yu, and J. Murphy, "IEEE 802.16j relay-based wireless access networks: an overview," *Wireless Communications, IEEE*, vol. 15, no. 5, pp. 56 –63, october 2008.
[14] B. Raman, K. Chebrolu, S. Bijwe, and V. Gabale, "PIP: A Connection-Oriented, Multi-Hop, Multi-Channel TDMA-based MAC for High Throughput Bulk Transfer," *SenSys*, 2010, Zurich, Switzerland.