LA-MAC: A Load Adaptive MAC Protocol for MANETs

Weihong Hu Xiaolong Li Homayoun Yousefi'zadeh
Department of EECS
University of California, Irvine
[weihongh,xiaolonl,hyousefi]@uci.edu

***Abstract*—The performance of prevalent MAC protocols in MANETs relies on the level of contention in networks. While contention-based MAC protocols such as CSMA suffer from inefficiency under high contention, slot-based MAC protocols such as TDMA perform in the opposite way. In this paper, we propose a hybrid protocol to which we refer as Load-Adaptive MAC (LA-MAC) protocol for MANETs. By adaptively switching its running mode between CSMA and TDMA, LA-MAC achieves high channel utilization under both high and low contention. We report our implementation of LA-MAC on a MANET testbed formed by a collection of Multiple-Input Multiple-Output (MIMO) Universal Software Radio Peripheral (USRP) software defined radio nodes. We program the PHY layer of USRP nodes using GNU Radio and integrate LA-MAC with the PHY layer implementation of USRP. Through experimental studies, we demonstrate the performance improvements of LA-MAC relative to CSMA and TDMA.**

***Index Terms*—Software Defined Radio, USRP, GNU Radio, Hybrid MAC, CSMA, TDMA, MANET.**

I. INTRODUCTION

Over the past few years, a spectrum of wireless MAC protocols have been proposed to improve the channel efficiency ranging from contention-based to slot-based protocols. With no reliance on any topology or synchrony information, contention-based protocols such as Carrier Sense Multiple Access (CSMA) [1], [2], [3] are typically robust to topology changes. However, their performance can be significantly degraded under high contentions due to high overhead accrued for resolving collisions. In contrast, slot-based protocols such as Time Division Multiple Access (TDMA) [4], [5], [6] utilize synchrony among neighboring nodes to achieve collision-free transmission by assigning transmission time slots to individual nodes. While slot-based protocols can generally achieve high channel utilization under high contentions, they often suffer from inefficiency during low contentions due to the fact that a node can transmit only during its scheduled time slots. Most importantly, slot-based protocols closely rely on the accuracy of information synchrony for high performance delivery of time slots even under high contentions. Thus, their performance can suffer as the result of not being able to keep the synchrony due to factors such as the time varying nature of wireless channel conditions, clock synchronization overhead, and interference irregularity problems. Thus and as illustrated by Fig. 1, utilizing a hybrid protocol that can perform a proper assessment of the tradeoff between the opposite ends of the spectrum is anticipated to combine the strengths of both types of protocols while offsetting their shortcomings. Designing one such hybrid protocol relying on the above-mentioned tradeoff represents the motivation of the work of this paper.

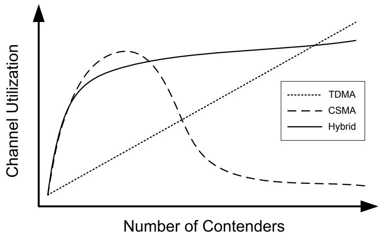

Fig. 1. A comparison of CSMA versus TDMA.

In what follows a brief review of the literature in a close context to the problem space of this paper is provided. Adapting a hybrid behavior between TDMA and CSMA according to the level of contention was first explored in [7]. DRAND [8] is a relatively scalable channel scheduling algorithm that can be used to assign time slots to network nodes. B-MAC [2] is a low power MAC protocol designed for sensor networks. Utilizing the work of [7], [8], and [2], Z-MAC [9] is another hybrid MAC protocol also designed for sensor networks.

Not only hybrid TDMA/CSMA and Z-MAC but many other newly proposed MAC protocols are specifically designed for wireless sensor networks rather than for MANETs due to the following reasons. First, a sensor node is relatively inexpensive and simple. Second, many newly proposed MAC protocols can be easily implemented in a sensor network environment because of the easy-to-use programming interface provided by TinyOS [10]. Given that applications running in sensor networks are characterized with a relatively low throughput and a simple data stream pattern, it is inappropriate to employ a sensor network infrastructure as a testbed of MANETs' MAC protocols.

While there are few commercially available radio platforms such as WARP [11], CalRadio [12], and KUAR [13] that can be used for the formation of MANETs, such platforms are either very expensive, or have a large Size Weight and Power (SWaP), or lack mobility. Therefore, the evaluation of performance of MAC protocols designed for MANETs has been mainly performed by means of simulations so far.

In our prior work of [14], we describe how to form a MANET testbed utilizing a collection of radio nodes. Each radio node consists of a General Purpose Processing (GPP) host PC running on Linux Operating System (OS) connected to a USRP [15] motherboard and a pair of programmable RF front-end boards. Our study Utilizes GNU Radio [16]

This work was sponsored by a research grant from the Boeing Company.

framework to implement a MIMO-capable PHY layer and a simple CSMA-based MAC protocol in DATA LINK layer of USRP. Hydra [17] is a flexible testbed similar to our testbed developed at UT Austin. A Hydra node is also comprised of a GPP and a USRP system. However, Hydra uses the Click module [18] for its MAC implementation. The work of [19] ports Click module, created for packet processing, to GNU Radio. This integrated framework allows USRP to support MAC protocol development.

The protocol proposed in this work, builds on the design ideas of [7], [8], [20], [2], and [9] for **MANETs**. We emphasize on the fact that the cross-layer design methodology of our proposed protocol allows for leveraging the capabilities offered by MIMO-capable SDR nodes such as USRP nodes. We propose a hybrid CSMA/TDMA MAC protocol to which we refer as Load-Adaptive MAC (LA-MAC) protocol. LA-MAC is capable of leveraging information although possibly imprecise about interference and synchrony in MANETs to improve performance. The protocol is intended to behave similar to CSMA under low contention and TDMA under high contention conditions by dynamically switching its running modes between CSMA and TDMA. We use the number of lost ACK messages at the MAC layer along with the neighboring nodes information to determine the contention level. By adopting the time slot assignment algorithm of USAP [20], the hidden terminal problems are avoided without using RTS/CTS handshakes. Furthermore, the synchronization problem commonly faced by slot-based MAC protocols is also investigated.

The key contributions of this paper are as follows. First, we propose a solution to strengthen the advantages of CSMA and TDMA while overcoming their shortcomings. Second, we provide a cross-layer design and implementation methodology capable of exchanging information between PHY and DATA LINK layers of a USRP-based MANET node. We reiterate that, to the best of our knowledge, leveraging advance capabilities of such MANET node for a MAC protocol has not been previously explored. Thus, we investigate how the use of USRP can facilitate MAC protocol design and implementation. Third, we measure the performance of our proposed protocol in a MANET testbed formed by a collection of USRP nodes. We conduct extensive experimental studies through which we demonstrate that our proposed protocol exhibits a significant performance improvement compared to CSMA or TDMA in terms of throughput.

The remainder of this paper is organized as follows. Section II describes our hybrid MAC protocol. The implementation of LA-MAC is presented in Section III. In Section IV, we describe our experimental studies and the analysis of our results. Finally, we conclude the paper and discuss future work in Section V.

II. LA-MAC OVERVIEW

LA-MAC may operate in three modes: CSMA, TDMA, and HYBRID. In one extreme, the protocol operates in CSMA mode when nodes have no knowledge about the network topology or when they lose synchronization. In the other extreme, the protocol switches to TDMA mode when the synchronization information is available and all nodes are restrained to transmit in their own time slots. In any situation other than two extreme cases, LA-MAC operates in a HYBRID mode, blending CSMA and TDMA modes of operation. In the following subsections, each mode of operation mode will be described.

A. CSMA Mode

In CSMA mode, LA-MAC simply employs a random backoff mechanism similar to IEEE 802.11 MAC protocol. Specifically, within a contention window a node waits for a random backoff period before transmitting. Then it senses the channel. If the channel is clear, the node sends its data. Otherwise, it waits and repeats the above process until it can acquire the channel. After receiving the data, a receiver sends an acknowledge (ACK) to the sender indicating whether the data was received correctly. If no ACK is received, the sender retransmits the data automatically. Notably, in LA-MAC a node is always required to perform this carrier sensing process before transmitting data regardless of its operation mode. In essence, CSMA mode is the baseline of the protocol operation. However, it is also possible for two nodes operating in different modes to communicate.

B. TDMA Mode

Noting that neighbor information collection and slot assignment are two important steps that need to be taken during the setup phase of TDMA, this section concentrates on operation in running phase of TDMA. Simply put, the discussion focuses on TDMA dynamics including topology acquisition and dynamic time slot assignment under the assumption that all of the initialization steps have been completed upon the startup of the protocol.

As a node joins the network or starts up, it needs to acquire the knowledge of its neighbors. In LA-MAC, each node collects its one-hop neighbor information utilizing a periodic Ping-like broadcast mechanism. Then each node exchanges its one-hop neighbor list with its one-hop neighbors. Eventually all nodes will constitute their two-hop neighborhood information. Importantly, there is no need to gather neighbor information beyond two-hop as the collision domain of nodes is limited in their two-hop neighborhood. Similar to Z-MAC, LA-MAC uses two-hop neighborhood information to assign slots for each node. Moreover, transmission control messages are delivered via neighborhood information messages.

To avoid collision without using RTS/CTS handshakes, LA-MAC's uses a Dynamic Time Slot Assignment (DTSA) algorithm that guarantees no two nodes within a two-hop neighborhood are assigned to the same slot. Recall that a conventional TDMA scheme accrues a high overhead related to propagating the maximum slot number necessary for the determination of frame length upon any network topology change. Adapting the time slot assignment algorithm of USAP, DTSA can overcome these shortcomings by changing the frame length adaptively. More specifically, a node is allowed to pick its own frame length according to the number of nodes

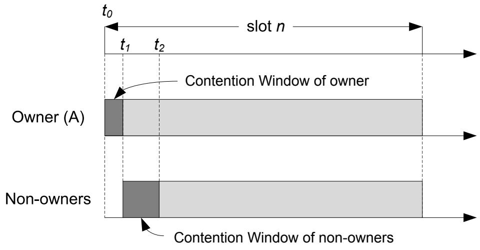

Fig. 2. LA-MAC slot competition by priorities.

in its contention area. Notably, such variable frame length mechanism yields two benefits as: i) fully utilizing time slots; and ii) significantly reducing overhead of control messages as a local network topology change no longer causes a global change. Once a DTSA operation completes, every node will have the slot and frame information of its one-hop and two-hop neighbors. At this point, all nodes are ready to run the transmission control protocol.

C. Hybrid Mode

In HYBRID mode, a node can be in two states: Low Contention (LC) and High Contention (HC). A node is in HC state when it receives a notification message from its one-hop or two-hop neighbors. Otherwise, the node is running in LC state. In LA-MAC, a node can stay in HC state for a pre-defined duration of T_{HC}. If a node does not receive new notification messages during that period, it will go back to LC state automatically.

While in LC state, all nodes are allowed to transmit in any time slot, implying that all nodes run in CSMA mode. Nonetheless and unlike conventional CSMA, each node is assigned a priority and needs to compete for a time slot according to its priority. The owner of the slot possesses the highest priority against other nodes. Specifically, the priority of a node is associated with its Contention Window (CW). Suppose slot n begins at time t_0 and the owner of the slot is node A. Then the CW for node A is defined as $[t_0, t_1]$ while the CW for other competitors is $[t_1, t_2]$, where $t_0 < t_1 < t_2$. Fig. 2 shows the mechanism of slot competition by priorities.

Such dual-state operation can improve the channel utilization significantly. Given a time slot, if the state of the owner is HC then the owner's two-hop neighbors are not allowed to transmit in this slot. In other words, only the owner and its one-hop neighbors can compete for the slot access. Thus, the collisions caused by hidden terminal problem can be significantly reduced. Similarly, each node is assigned a priority in HC state and the slot competition follows the same strategy as that of LC state.

Accordingly, it is the sender who controls when to switch its operation mode. If a node is experiencing high collision rate, the packet loss rate tends to increase accordingly. As LA-MAC does not use RTS/CTS mechanism to avoid collisions, current contention level is proportional to the collision rate. Therefore, the contention level can be estimated by counting the number of lost ACKs. If a node misses N_{th} (Mode switch threshold) ACKs consecutively, a notification message is sent to all of its two-hop neighbors informing them not to act as hidden terminals such that the collisions can be avoided. Accordingly, the node switches to the HC state.

III. LA-MAC IMPLEMENTATION

In this section, we present the implementation of LA-MAC on USRP and GNU Radio. First, a fundamental conception known as inband signaling is introduced. Then three key mechanisms namely carrier sensing, collision avoidance and clock synchronization are presented.

In our MANET nodes built on USRP and GNU Radio [14], GNU Radio runs on the host and USRP is connected to the host via a USB connection. While USRP functions as the RF frontend and the baseband processing unit, the PHY and DATA LINK layers are implemented in GNU Radio. Generally, the FPGA chipset on USRP system converts the signal between the IF band and the baseband, resamples the signal, and then transfers the I/Q samples across the USB. All other signal processing functions such as modulation/demodulation and framing are completed on the host. However, such architecture introduces significant latencies as the speed of signal processing is limited by the computing capability of the host. Moreover, data transfer through the USB is very slow. Given the fact that mainstream MAC protocols such as CSMA/CA and TDMA require a very strict timing for the transmission control, such approach is not suitable for the development of MAC protocols. While such timing requirements could be roughly satisfied by offloading time-critical functions to FPGA on USRP, there are other issues. For example, the earlier versions of GNU Radio do not support control or status information exchanges between the host and USRP. USRP interprets all data over the USB bus as samples and GNU Radio can only handle fixed length data with no meta-data. Thus, until the concept of message blocks [21] was proposed to support variable length data and meta-data, packet processing in GNU Radio has been quite challenging. In light of the introduction of message blocks, a technique called inband signaling [22] emerged allowing for duplex communication between FPGA and GNU Radio components.

As illustrated by Fig. 3, in inband signaling the samples over the USB are encapsulated in a new packet structure along with additional information about the samples.

This technique provides a command channel between host and FPGA thus enabling the exchange of information such as status. In our implementation, key functions such as carrier sensing, collision avoidance and time synchronization are all built on top of the inband signaling. We will discuss them in the following subsections.

A. Carrier Sensing

In CSMA, a node senses the channel by computing the signal strength or noise level of the channel. If the signal strength exceeds some given threshold, the channel is busy. GNU Radio offers a basic Python-based implementation of CSMA to which we refer as GR-CSMA. The host keeps computing the signal strength of the channel. Once it senses that the channel is idle, packets are sent out immediately.

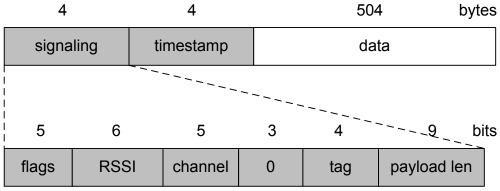

Fig. 3. The format of USB packets used for USRP inband signaling.

However, due to the transmission latency, it could be time-consuming to get a signal sample through USB and compute the signal strength. Thus, when a host realizes that the channel is idle, the channel has been actually idle for a while. The delay is the interval from the moment when signals are captured by the RF frontend to the moment when the processing of samples finishes on the host. We note other nodes are likely allowed to transmit during this interval. If the host also sends packets at this time, collisions will occur.

Inband signaling provides a potential solution to this problem. As illustrated by Fig. 3, there is an “RSSI” field in the meta-data indicating the received signal strength as reported by the RF front end. When a host receives samples, the channel state can be extracted from the RSSI information in meta-data rather than computed from the samples. While this mechanism removes signal processing time, the transmission delay through the USB still cannot be ignored.

Coupled with inband signaling, LA-MAC’s implementation approach calls for offloading the process of carrier sensing to FPGA. As soon as the channel becomes idle, the FPGA sends out the packets immediately. This translates to a $1\!-\!persistent$ CSMA system where a node transmits with a probability of 1 once it senses an idle channel. However, such approach may cause collisions when there are simultaneous transmissions. To avoid collisions, LA-MAC employs a random backoff algorithm. The host determines and signals the backoff time to FPGA relying on inband signaling.

B. Collision Avoidance

As mentioned earlier, an RTS/CTS handshake may be used to deal with the problem of hidden terminals.

First, the sender issues an RTS control packet which includes the destination and the duration of the whole data transmission. Every node receiving the RTS has to set its Network Allocation Vector (NAV) in accordance with the duration field indicating that it will not try to access the channel during that period. Notably, this mechanism involves the time-critical control. While it is inappropriate to implement the RTS/CTS handshake on host due to the excessive latency, offloading is not an option either due to the difficulties of implementing such complicated functions as modulation and demodulation in FPGA. Therefore, collision avoidance on GNU Radio remains an open issue. In LA-MAC, hidden terminal problems can be mitigated as much as possible by running in HYBRID mode. As mentioned in Section II, the probability of collisions caused by hidden terminals is small if the contention level is low. If the contention level is high, the collisions can be reduced by notifying the hidden terminals not to transmit. As shown in Section IV, this hybrid mechanism is an effective approach to avoid collisions.

C. Clock Synchronization

While LA-MAC does not require maintaining global clock synchronization, LA-MAC requires a node to maintain clock synchronization with other nodes in its contention area. The timestamp field in the meta-data is used for synchronization. The field is set by a 32-bit counter in FPGA that is incremented by the A/D sample clock. If the packet is sent toward the host by the FPGA, the timestamp indicates the time by which the first sample of the packet was produced by the A/D converter. If the packet is sent away from the host, the timestamp indicates the time at which the first sample of the packet should go out of the D/A converter. A node synchronizes with its neighbors by sending synchronization (SYN) messages. The sender schedules a SYN message to be transmitted at time T by specifying the timestamp field in the meta-data. In the meantime, the sender also set the timestamp in SYN message to time T. When the receiver receives the SYN message, it knows the time by which the packet is received relative to its local clock. Then, it can compare the difference between its local time and the sender’s time. Our experiments results show that the inaccuracy of this synchronization is less than 10 microseconds, which may be safely ignored for a link with a channel capacity of 1 Mbps.

IV. PERFORMANCE EVALUATION

In this section, we present our experimental studies to evaluate the performance of LA-MAC protocol against GR-CSMA and TDMA. Our experiments rely on a MANET testbed in which each test node consists of a host PC and a USRP motherboard hosting a pair of frontend RF daughter boards. Since each daughter board is attached to a single antenna, each MANET node is equipped with a pair of antennas. When transmitting, each MANET node utilizes Space-Time Block Coding (STBC) method of [23]. When receiving, it utilizes Maximum Ratio Combining (MRC). Due to shortage of space, we do not describe the details of implementing those coding schemes using signal processing blocks of GNU Radio. Fig. 4 shows the system architecture of a MANET node. The host PC includes a 2.4GHz Intel Core 2 Duo CPU, 1GB of memory, and runs Fedora Core 10 distribution of Linux OS. Our MAC protocol as well as GNU Radio runs in the user space of Linux. Other major hardware configurations and protocol parameters are summarized in Table I.

Two different network topology scenarios are used in our experiments. While a one-hop scenario is used to evaluate the effect of CSMA Mode on maximum achievable data throughput, the effect of hidden terminals problem is investigated in a two-hop scenario.

In the one-hop scenario, 16 nodes are distributed within a one-hop distance of one another. This guarantees that there are no hidden terminals in the entire network and any pair of nodes can communicate with one another directly. For simplicity, only one node is designated as the receiver and the others are designated as senders in this scenario. Since

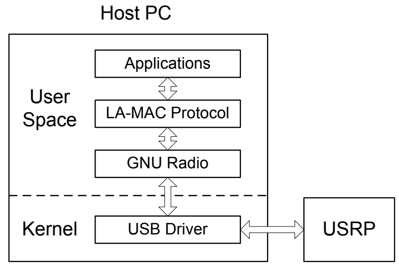

Fig. 4. The system architecture of a MANET testbed node.

TABLE I
HARDWARE CONFIGURATIONS AND PROTOCOL PARAMETERS

Host PC	Dell Latitude E6400
RF frontend	USRP & XCVR2450
GNU Radio	Revision 10663
Modulation	GMSK
Channel bandwidth	1 Mbps
CW size for owners	1 ms
CW size for non-owners	2 ms
Time slot size	15 ms
Mode switch threshold (N_{th})	5
HC state duration (T_{HC})	2 s
Center frequency	2.48GHz

the data transmitted by a node can be received by all other nodes in this scenario, this designation of a single or multiple receivers makes no difference on performance profiling results.

In the two-hop scenario, we evaluate the impact of the hidden terminal problem. The network topology used in the two-hop scenario is shown in Fig. 5. As illustrated by the figure, the senders are separated into two clusters for this scenario. Each cluster is a one-hop network topology consisting of 7 nodes. A pair of nodes belonging to different clusters cannot directly communicate. Two receivers are placed in the union of the two clusters. It has to be guaranteed that those two receivers can communicate with all senders directly. In addition, it has to be guaranteed that the nodes in one cluster cannot sense a transmission made by a sender in another cluster. To satisfy these requirements, the transmission power and the receive gain are tuned carefully.

In our experiments, we analyze performance using two metrics:

- *Round Trip Time (RTT):* Essentially, RTT is a translation of the transmit latency caused by host signal processing and USB data transfer. Further, the efficiency of MAC protocol may introduce additional latency. For example, in TDMA, a node has to wait for its time slot to transmit data. We note that it is difficult to directly measure the transmit latency due to complications associated with measuring the exact time when a packet is sent out by the RF frontend. Therefore, RTT is used as an alternative of transmit latency.
- *Maximum Achievable Data Throughput (MADT):* MADT is defined as the aggregation of all data traffic delivered to the receivers per second when the senders transmit at best effort. Note that the contention level is proportional to the number of senders. A larger number of senders yields a higher contention level. When measuring MADT, the payload portion of data traffic is considered valid but not the frame preamble, frame header, and CRC bytes.

A. The Effect of CSMA Mode on Transmit Latency

In this subsection, the average RTT is measured comparing the transmit latency of GR-CSMA and the performance of LA-MAC when operating in CSMA mode as forced by the topology of the experiment. We refer to the latter as LA-CSMA. The experiment is conducted on a point-to-point link with a capacity of $500Kbps$. A "Ping-Pong"-like program runs on both sides and sends $64Byte$ long packets. The measured results illustrate average RTT values of $29.8ms$ and $18.7ms$ for GR-CSMA and LA-CSMA, respectively. The measurements demonstrate that LA-CSMA achieves a transmit latency that is lower than 30% of that of GR-CSMA. The performance gain is mainly due to the fact that LA-CSMA takes an FPGA carrier sensing approach significantly reducing the transmit latency caused by host signal processing and USB data transfer. Furthermore, the choice of the backoff algorithm contributes to the performance gain. While the use of an exponential backoff mechanism in GR-CSMA can reduce the possibility of collisions, it increases transmit latency in contrast to the use of a random backoff mechanism used by LA-CSMA.

B. The Effect of CSMA Mode on MADT

In the one-hop scenario, we gradually increase the number of senders capturing a scenario of increased contention. Each sender transmits as quickly as possible. We observe how the throughput changes as a function of the contention level. Fig. 6 compares the MADT achieved by GR-CSMA and LA-CSMA. As illustrated by the figure, the throughput of LA-CSMA is nearly independent of the number of senders because no collisions occur in the absence of hidden terminals and every time slot can be utilized to transmit. In contrast, GR-CSMA illustrates a MADT drop when the number of senders exceeds three. When 15 senders transmit simultaneously, the MADT of GR-CSMA is approximately 60% less than that of LA-CSMA. While not shown in the paper, the receiver experiences a large number of corrupt packets due to collisions. This observation demonstrates that sensing the carrier on host introduces not only additional transmit latency but also collisions, both of which consequently degrade the performance.

C. The Effect of HYBRID Mode on MADT

In this experiment, MADT is measured as the number of senders varies in the two-hop scenario. Fig. 7 demonstrates that LA-MAC represents a reasonable tradeoff between CSMA and TDMA. With one sender, LA-MAC operates in CSMA mode and yields similar MADT as GR-CSMA. As a larger number of senders join, LA-MAC significantly outperforms both TDMA and GR-CSMA by switching to HYBRID mode and stays at a relatively high level of performance without much oscillation. In contrast, while the MADT dramatically drops in the case of GR-CSMA due to collisions resulting from the hidden terminal problem, TDMA suffers from significant wastage of time slots. When the number of senders

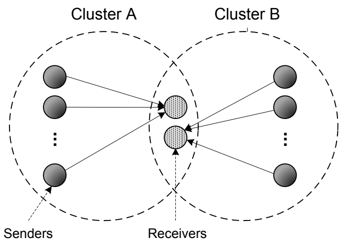

Fig. 5. The topology of the two-hop experiment.

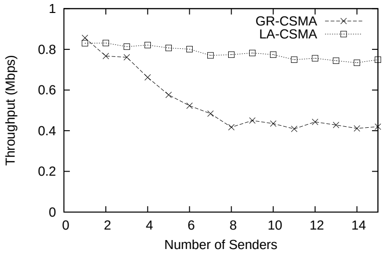

Fig. 6. The throughput comparison of the one-hop experiment.

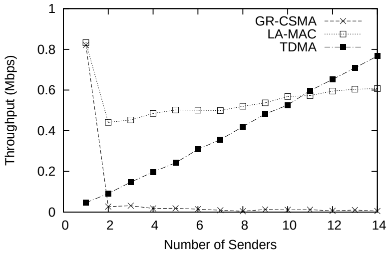

Fig. 7. The throughput comparison in the two-hop experiment.

exceeds 11, LA-MAC further improves its MADT. However, it is outperformed by TDMA due to overhead associated with sending notification messages and inevitable packet loss associated with mode switching. Note that the performance gap between LA-MAC and TDMA can shrink by adjusting the parameters T_{HC} and N_{th}. While reducing N_{th} and/or increasing T_{HC} allow LA-MAC to behave more like TDMA and thereby improving MADT in HC state, such adjustments degrade the performance of LA-MAC when operating under LC state.

V. CONCLUSION

In this paper, we present Load-Adaptive MAC (LA-MAC) protocol for MANETs capable of switching modes of operation. While LA-MAC operates in CSMA mode for light traffic loads, it switches to TDMA mode of operation for heavy traffic loads and increased contention conditions. We implement and host LA-MAC in a MANET testbed formed by a collection of USRP SDRs. Our implementation approach uses GNU Radio and inband signaling technology. Our experimental results show that LA-MAC achieves a better performance measured in terms of throughput than CSMA and TDMA over either one-hop or two-hop MANET topologies. We demonstrate that while certain critical protocol features such as carrier sensing and clock synchronization can be implemented in GNU Radio, implementing more sophisticated transmission control strategies such as p-persistence CSMA, RTS/CTS handshakes in FPGA, and detecting the preamble of data frame in FPGA require further research. Currently, we are extending the functionality of LA-MAC as an anycast MAC protocol such that it can pass link quality information to routing protocols hosted in the NETWORK layer thereby improving routing performance in a DATA LINK-NETWORK cross layer design.

REFERENCES

[1] W. Ye, J. Heidemann, and D. Estrin, “An Energy-Efficient MAC protocol for Wireless Sensor Networks,” in *Proc. the IEEE Infocom*, New York, NY, USA, June 2002, pp. 1567–1576.

[2] J. Polastre, J. Hill, and D. Culler, “Versatile Low Power Media Access for Wireless Sensor Networks,” in *Proc. the 2nd international conference on Embedded networked sensor systems*, Baltimore, MD, USA, 2004, pp. 95–107.

[3] T. van Dam and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in *Proc. the 1st international conference on Embedded networked sensor systems*. Los Angeles, CA, USA: ACM, 2003, pp. 171–180.

[4] “WiMAX Protocol Specification,” available at http://ieee802.org/16/published.html.

[5] C. D. Young, “USAP: A Unifying Dynamic Distributed Multichannel TDMA Slot Assignment Protocol,” in *Proc. IEEE MILCOM ’96*, vol. 1, McLean, VA, USA, Oct. 1996, pp. 235–239.

[6] C. David Young, “USAP Multiple Access: Dynamic Resource Allocation for Mobile Multihop Multichannel Wireless Networking,” in *Proc. IEEE MILCOM ’99*, vol. 1, Atlantic City, NJ, USA, Nov. 1999, pp. 271–275.

[7] B. A. Sharp, E. A. Grindrod, and D. A. Camm, “Hybrid TDMA/CSMA Protocol for Self Managing Packet Radio Networks,” in *Universal Personal Communications. 1995. Record., 1995 Fourth IEEE International Conference on*, Tokyo, Japan, Nov. 1995, pp. 929–933.

[8] I. Rhee, A. Warrier, J. Min, and L. Xu, “DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks,” in *Proc. the 7th ACM international symposium on Mobile ad hoc networking and computing*, Florence, Italy, 2006, pp. 190–201.

[9] I. Rhee, A. Warrier, M. Aia, and J. Min, “Z-MAC: a Hybrid MAC for Wireless Sensor Networks,” in *Proc. the 3rd international conference on Embedded networked sensor systems*, San Diego, California, USA, 2005, pp. 90–101.

[10] “TinyOS: An open-source OS for the networked sensor regime,” http://www.tinyos.net/.

[11] Rice University, “WARP: Wireless Open Access Research Platform,” http://warp.rice.edu/.

[12] UCSD, “CalRadio 1a,” http://calradio.calit2.net/calradio1a.htm/.

[13] G. J. Minden, J. B. Evans, L. Searl, D. DePardo, V. R. Petty, R. Rajbanshi, T. Newman, Q. Chen, F. Weidling, J. Guffey, D. Datla, B. Barker, M. Peck, B. Cordill, A. M. Wyglinski, and A. Agah, “KUAR: A Flexible Software-Defined Radio Development Platform,” in *Proc. the IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Apr. 2007, pp. 428–439.

[14] X. Li, W. Hu, H. Yousefi’zadeh, and A. Qureshi, “A Case Study of A MIMO SDR Implementation,” in *Proc. IEEE MILCOM 2008*, San Diego, CA, USA, Nov. 2008.

[15] Ettus Research LLC, http://www.ettus.com/index.html.

[16] E. Blossom, “Exploring GNU Radio,” Nov. 2004, http://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html.

[17] K. Mandke, S. Choi, G. Kim, R. Grant, R. C. Daniels, W. Kim, R. W. H. Jr., and S. Nettles, “Early Results on Hydra: A Flexible MAC/PHY Multihop Testbed,” in *Proc. IEEE Vehicular Tech. Conf.*, Dublin, Ireland, Apr. 2007.

[18] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, “The Click Modular Router,” *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, Aug. 2000.

[19] R. Dhar, G. George, A. Malani, and P. Steenkiste, “Supporting Integrated MAC and PHY Software Development for the USRP SDR,” in *1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, Sept. 2006.

[20] A. Kanzaki, T. Uemukai, T. Hara, and S. Nishio, “Dynamic TDMA Slot Assignment in Ad Hoc Networks.” in *Proc. the 17th International Conference on Advanced Information Networking and Applications*, 2003, pp. 330–335.

[21] GNU Radio Wiki, “Message Blocks,” available at http://gnuradio.org/trac/wiki/MessageBlocks.

[22] GNU Radio, “inband-signaling-usb,” available at http://gnuradio.org/trac/browser/gnuradio/trunk/usrp/doc/inband-signaling-usb.

[23] S. Alamouti, “A simple transmitter diversity scheme for wireless communications,” *IEEE JSAC*, vol. 16, pp. 1451 – 1458, Oct. 1998.