# A GNU RADIO TESTBED FOR DISTRIBUTED POLLING SERVICE-BASED MEDIUM ACCESS CONTROL

**Saranyadevi.K.Brindha R., Karpagam College of Engineering**

*ABSTRACT—***Medium access control (MAC) protocols play a vitalrole in wireless networking. It is well-known that the high control overhead of IEEE 802.11 MAC is the limiting factor on wireless networks capacity. In our prior work, polling service-based MAC protocols (PSMACs) are developed to amortize the high control overhead over multiple frame transmissions, thus achieving higher efficiency. Both analysis and simulations are conducted to validate the efficacy of the proposed protocols. In this paper, we extend this work by implementing the distributed version of PSMAC, i.e., PSMAC 2, on the GNU Radio and Universal Software Radio Peripheral (USRP) platform. We discuss various design considerations and challenges of prototyping PSMAC 2, and carry out extensive experimental studies with the PSMAC testbed. Our experimental results are found to be consistent with the theoretical study reported in our prior work, and validate the advantages of PSAMC under a realistic setting.**

## I. INTRODUCTION

Wireless LANs (WLANs) based on the IEEE 802.11 stan-dard [1] have become the ubiquitous connectivity solution in public as well as residential areas. The compelling demands to support, e.g., high definition videos, online games and other real-time applications, greatly stress the capacity of existing WLANs. The IEEE 802.11 MAC, however, is well-known for its considerably high control overhead, which could consume as much as 40% of the nominal link capacity [2]. The problem gets even worse in the multi-hop scenario, due to carrier sensing and spatial reuse issues.

In our previous work [2], [3], we present three polling service-based MAC protocols, termed PSMACs, that can amortize the control overhead of medium contention/resolution over multiple back-to-back frame transmissions, thus achiev-ing high efficiency. The gated service based PSMACs are an-alyzed and compared with p-Persistent CSMA, which closely approximates the standard IEEE 802.11 distributed coordina-tion function (DCF). Considerable gains on throughput, delay, energy consumption, and fairness performance are observed in the analysis and simulation studies [2].

In this paper, we extend the prior work by prototyping the PSMACs in a real wireless networking environment. Gener-ally, testbeds can provide useful insights that computer-based simulations cannot offer, since they capture the complex real-world radio propagation effects as well as distributed network dynamics, which are often greatly simplified in simulation and theoretical studies to make the problem manageable. By prototyping PSMACs, we can not only evaluate the MAC

protocols under realistic wireless network settings and verify our prior theoretical and simulation studies, but also identify new practical constraints and problems.

Therefore, the goal of the implementation effort is to create a reconfigurable and completely functional PSMAC testbed, as well as evaluating the PSMAC performance under realistic wireless network settings. We make two main contributions in this paper. First, we implement the PSMACs on the GNU Radio [4] and Universal Software Radio Peripheral (USRP) [5] platform. We integrate the key functions of 802.11 DCF and the gated service policy in the implementation, such as gated service scheduling, CSMA/CA, virtual carrier sensing, RTS/CTS handshake, automatic repeat request (ARQ), random backoff mechanism, and distributed clock synchronization using IEEE 1588. Second, we conduct extensive experiments with various traffic types and traffic patterns, to evaluate the real system performance of the PSMAC testbed. The experimental results demonstrate the significant improvements that PSMAC can achieve on throughput, delay and fairness, and validate the theoretic analysis and simulation studies in our prior work [2].

The remainder of this paper is organized as follows. We describe the system overview in Section II and discuss im-plementation details in Section III. The experiment results are presented in Section IV. Related work is discussed in Section V and Section VI concludes the paper.

## TESTBED SYSTEM OVERVIEW

### A. Polling Service-Based MAC Protocol

We first briefly review PSMAC to provide the necessary background, and refer interested readers to [2] for more details.

PSMAC is motivated by the insights from polling system theory [2]. Generally, a polling system may have either a centralized or a distributed structure. In the centralized case, a server polls the stations and controls channel access. In the distributed scenario, stations contend for channel access using a distributed mechanism. In either case, one of the three service policies can be used to serve the frames for a wining station: (i) *Exhaustive* policy, where the server serves a station until its buffer is emptied; (ii) *Gated* policy, where the server serves for a station the frames that have backlogged in the buffer when the service begins; (iii) *Limited-k* service, where a station is served for up to k frames or until the queue is empty, whichever comes first. It has been shown that both

exhaustive service and gated service are more efficient than limited-$k$ service, and they can guarantee bounded delay as long as the offered load is strictly less than 100% [2].

Based on the polling system theory, three polling service-based MAC protocols are introduced in [2], [3]. The main idea is to serve multiple frames after a successful contention resolution, thus amortizing the high control overhead over mul-tiple DATA frames and making the protocols more efficient. In particular, PSMAC 1 senses a channel with CSMA/CA and uses RTS/CTS frames for contention resolution. A winning node will use gated service to serve its backlogged frames. PSMAC 2 introduces multiple virtual queues, one for each neighbor. The gated service is used for one of the non-empty virtual queues when the station wins the channel. This allows other neighbors not involved in the transmissions be scheduled to sleep for energy savings. PSMAC 3 extends PSMAC 2 by serving all non-empty virtual queues when the station wins the channel, which may achieve even higher efficiency.

In [2], [3], the PSMACs are evaluated with analysis and sim-ulations. They are shown to achieve considerable throughput and delay improvements over p-Persistent CSMA. In addition, PSMACs 2 and 3 can achieve significant energy savings by scheduling nodes to sleep. The PSMACs are also shown to be more efficient for handling bursty traffic, and the performance gains are achieved without sacrificing fairness performance.

The limited-1 policy is a special case of limited-$k$, with only one frame served for a winning station. This policy is used in most existing MAC protocols, e.g., $p$-Persistent CSMA and IEEE 802.11 DCF. We focus on the PSMAC 2 protocol in this paper. We also implement a limited-1 based IEEE 802.11 DCF like protocol for comparison purpose.

### B. Software and Hardware Platforms

We built the testbed on the Software Defined Radio (SDR) platform consisting of GNU Radio and USRP. GNU Ra-dio [4] is an open-source software development toolkit under the GNU General Public License (GPL). It provides signal processing runtimes and processing blocks to implement SDR on RF hardware and commodity processors. USRP [5] is a generic SDR hardware device that natively integrates with GNU Radio. The motherboard of USRP 1 consists of four 64 MS/s ADCs and four 128 MS/s DACs. It has an FPGA for processing baseband and IF signals. The RFX2400 RF front-end daughterboard supports transmission and receiving from 2.3 GHz to 2.9 GHz in the ISM band.

During the implementation, we observe that the main limita-tion of GNU Radio for MAC development is the high latency. Most MAC protocols rely on precise receiving and trans-mission timing. For example, IEEE 802.11 requires precise timing for the virtual carrier sensing mechanism. However, GNU Radio introduces a non-negligible latency due to the general-purpose processor and USB interface. In addition, the bus system to transfer the samples between a radio front-end and the processor also introduces extra latency. Finally, the Python script environment, kernel/user space switch and process scheduling of the operation system also make the
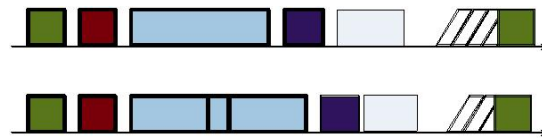


Fig. 1.  Illustration of the protocol operation: GR-Limited-1 vs. GR-PSMAC.

latency hard to track. It is reported in [6] that the modulation, spreading, demodulation and despreading procedures could introduce an additional 22.5 ms delay, which is quite large comparing to the standard timing setting in IEEE 802.11 (generally in the µs scale).

The large latency also negatively affects performance mea-surement during testbed experiments, especially under high transmission rates. To solve this problem, we use a relatively small link rate along with a large frame size to alleviate the impact of latency on transmissions. For example, using a 125 kbps link capacity with 1,500-byte frames, the frame transmission delay is about 96 ms, which is about 70% of the total transmission latency. With reduced link rates, we can conduct full functional tests for the MAC protocols and obtain precise normalized performance results.

### C. GR-PSMAC and GR-Limited-1

We implement GR-PSMAC and GR-Limited-1 by extending the IEEE 802.11 DCF. GR-PSMAC integrates CSMA/CA with binary exponential backoff, virtual carrier sense, RTS/CTS handshake, and ARQ for link error control.

In GR-PSMAC, a source station selects a nonempty virtual queue in a round-robin fashion and senses the channel before transmission. After the channel is idle for DIFS interval, the contention window (CW ) start to decrease. If the channel remains idle when CW reaches 0, an RTS frame will be transmitted. If the channel is busy, CW will be frozen and the transmission will be deferred. When the destination station receives the RTS, it returns a CTS frame to confirm that it is ready for receiving data. The CTS frame contains the transmission duration, which allows other stations set up their Network Allocation Vector (NAV) for virtual carrier sensing. After receiving the CTS, the source station will transmit its backlogged DATA frames back-to-back to the destination following the gated service policy (see Fig. 1). All other stations will keep silent and wait for the NAV to expire (or be scheduled to sleep for energy conservation). When the last frame is received, an ACK frame is issued to acknowledge all the received frames. If frames are not correctly received after the transmission phase, the backoff procedure will be performed at the source station to defer the transmissions.

The backoff procedure follows the IEEE 802.11 DCF spec-ification [1]. After DIFS medium idle time, the source station shall generate a random backoff period for additional deferral before transmitting. The backoff period is randomly generated in [0, CW − 1]. CW takes an initial value of $CW_{MIN}$, and is doubled after each collision or unsuccessful transmission, until it reaches $CW_{MAX}$. We set $CW_{MIN} = 8$ and $CW_{MAX} = 256$.
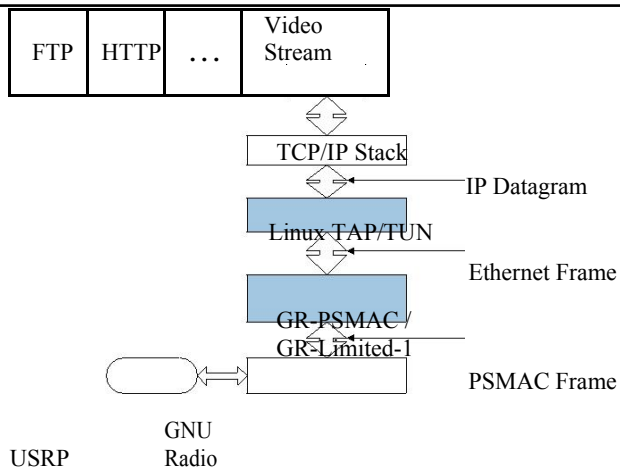
Fig. 2.Protocol architecture of the GNU Radio testbed.

After each successful transmission or the number of RTS retries reaches a predefined value, CW is reset to $CW_{MIN}$ .

GR-Limited-1 is implemented in the similar manner, except that only up to one DATA frame will be transmitted for a winning station (as shown in Fig. 1). This is consistent with the standard IEEE 802.11 DCF and its performance is comparable to IEEE 802.11 DCF.

## III.  TESTBED IMPLEMENTATION DESCRIPTION

### A. Network Protocol Architecture

GR-PSMAC and GR-Limited-1 are implemented as Layer 2 protocols from the point of view of network protocol archi-tecture. Both protocols are written in Python script running in the user space of Linux. We resort to Linux TAP/TUN virtual network interface that provides the bridge between GNU Radio and Linux TCP/IP kernel. A virtual Ethernet interface is created, which can be configured with an IP address. Applications can then use the MAC protocols implemented in GNU Radio transparently as a standard network API. This is illustrated in Fig. 2.

The GR-PSMAC and GR-Limited-1 header format is given in Table I. The header contains eight fields and is 16-bytes long. Most of the fields are compatible with the header defini-tion of IEEE 802.11, but are different from the standard Ether-net header. Therefore, frames from the upper layer through the TAP/TUN driver will require a mapping from Ethernet header to PSMAC header, as illustrated in Fig. 2. Similarly, GR-PSMAC/GR-Limited-1 map PSMAC header back to Ethernet header for received frames.

### B. Transmission and Receiving Path

The GR-PSMAC is implemented as two execution paths, namely, the transmission path and the receiving path. We adopt multithreading and each path is controlled by a thread. The design of the two paths is outlined below.

*1) Transmission Path:* When GR-PSMAC receives aDATA frame from the upper protocol stack, it replaces the Ethernet header with the PSMAC header and buffers the frame in the outgoing queue. If the channel is sensed busy, the frame is held in the outgoing queue and the transmission is deferred. GR-PSMAC maintains a virtual queue for each of its neighbors, and the frames are enqueued according to their destination MAC addresses.

### TABLE I
### GR-PSMAC/GR-LIMITED-1 HEADER FORMAT

| Field | Bytes | Description |
|---|---|---|
| Frame Control | 1 | Four least significant bits define the frame type (RTS/CTS/DATA/ACK); Other bits are reserved for future use. |
| Destination Address | 2 | Destination MAC address. |
| Source Address | 2 | Source MAC address. |
| Next Hop Address | 2 | MAC address of the next hop. Only valid for DATA frames and is used for access point mode or multi-hop mode. |
| Duration | 4 | Multi-purpose field. In RTS/CTS/DATA: number of frames to be transmitted. In ACK: sequence number of the last received DATA frame. |
| Sequence Number | 2 | Sequence number of transmitted DATA frame. In ACK: sequence number of the first received DATA frame. |
| Count | 2 | In RTS/CTS/DATA: number of transmitted frames. In ACK: number of correctly received DATA frames. |
| Option | 1 | Reserved for future use. |

If the channel is sensed idle, the station selects a non-empty virtual queue in the round-robin manner, and issues an RTS frame to the neighbor corresponding to the chosen virtual queue. The requested transmission time in the RTS is equal to the duration for transmitting all the backlogged frames in the selected virtual queue. If a CTS frame is not returned before timeout, GR-PSMAC will backoff the transmission and increase the RTS retry counter by one. Furthermore, if the RTS retry number exceeds the limit, GR-PSMAC will reset CW and serve the next nonempty outgoing virtual queue.

On the other hand, if a CTS frame is successfully received, GR-PSMAC will reset its CW , transmit the backlogged DATA frames in the selected virtual queue in a row, and wait for ACK. If an ACK frame is received before timeout, GR-PSMAC will purge the acknowledged frames from the outgo-ing virtual queue. Otherwise, it will backoff the transmission and serve the next nonempty outgoing virtual queue.

*2) Receiving Path:* When a station receives an RTSdestined for itself, it sets NAV according to the *Duration* field. Then it returns a CTS frame with the duration equal to the original duration minus the CTS frame duration. Other neighbors that receive the CTS frame will set their NAV according to the *Duration* field and enter the sleep mode.

During the following transmission period, the destination station receives one or more back-to-back DATA frames. It maps the PSMAC headers back to Ethernet headers, and forwards the Ethernet frames to the upper layer. The sequence numbers of received DATA frames are recorded in a list. After all the frames are received or when there is a timeout, an ACK frame is issued with successfully received sequence numbers back to the source station. The source station, on receiving the ACK, will remove all the successfully transmitted frames from its outgoing virtual queue.

### C. Acknowledge and Retransmission Mechanisms

We also implement the acknowledge and retransmission mechanisms for GR-PSMAC. IEEE 802.11 DCF uses limited-1 service that transmits only up to one DATA frame each

time, such that a subsequent ACK acknowledges the successful DATA transmission. In GR-PSMAC, there may be multiple DATA frames transmitted in a row during the transmission period. Therefore a default ACK frame is not sufficient for acknowledging multiple DATA frames.

We implement two ARQ options for GR-PSMAC. The first one is Go-Back-N . The destination records the received sequence numbers in increasing order. When timeout happens or the last frame is received, the destination sends an ACK carrying the first received sequence number in the batch, as well as the last received sequence number right before the first missing frame (if any). All frames received after the gap are discarded and will be retransmitted.

Although Go-Back-N ARQ is easy to implement, it is not efficient when the number of transmitted frames is large or when the frame loss rate is low. To improve efficiency, we also implement the Selective Repeat Protocol (SRP). In SRP, the ACK contains an explicit list of the sequence numbers of successfully received frames; only the missing frames need to be retransmitted. SRP can reduce the number of retransmissions, but has a slightly higher control overhead (i.e., longer ACK frames) and complexity.

### D. Synchronization for Distributed Delay Measurement

In a distributed network scenario, the CPU clocks may not be precisely synchronized. This may introduce large frame delay measurement errors. To address the synchronization issue, we adopt the Precision Time protocol (PTP) daemon that implements the IEEE 1588 standard [7] to synchronize the testbed nodes. Note that such tight synchronization is used for data collection during experiments (i.e., delay measurement), rather than for the normal operation of the PSMAC testbed.

IEEE 1588 provides real-time clock synchronization for distributed systems with sub-microsecond precision. Such pre-cision is sufficient for experiments and delay measurement in the PSMAC testbed. The testbed nodes are connected with an Ethernet hub, and are then synchronized with the PTP daemon. When a DATA frame is enqueued at the source node, a time stamp will also be stored at the source node. When a DATA frame is successfully received, the destination node will attach a time stamp in the ACK frame that records the time when the DATA frame was received, along with the list of sequence numbers. The source node can compute the one-way delay as the difference between the received (i.e., in the ACK frame) and stored time stamps.

### IV. EXPERIMENTS AND RESULTS

#### A. Experiment Setting

The testbed consists of four USRP 1 kits, each connected to a general purpose computer through a USB 2.0 port, as shown in Fig. 3(a). GR-PSMAC and GR-Limited-1 are implemented in GNU Radio 3.3 with Ubuntu Linux OS. As discussed, we also connect all the computers to an Ethernet hub and synchronize the clocks with IEEE 1588.

We run the tests using a one-hop ad hoc network topology. The four testbed nodes share the 125 kbps nominal link

capacity. The SRP ARQ scheme and the RTS-CTS virtual carrier sensing mechanism are used. The DATA frame size is set to 1,500 bytes and the carrier frequency is set to 2.401 GHz. The GMSK modulation is used. We choose two samples per symbol and the RTS retry limit is set to five.

We develop a UDP client-server application that can gen-erate traffic to drive the experiments. UDP is chosen to avoid the complex rate variations caused by TCP congestion control, thus focusing on the MAC performance. The following two traffic models are used in the experiments:

i) i.i.d. Bernoulli traffic: a frame is generated in each time slot with a predefined probability.

ii) On-off bursty traffic: frames are generated according to an on-off model with geometrically distributed on and off periods. The average on period is five for the results reported in this section.

We also consider the uniform traffic pattern, where the des-tination of each DATA frame is uniformly distributed among the neighbors, and the non-uniform traffic pattern, where one link has much higher load than other links.

For each offered load, we run the testbed for ten times, each lasts for 300 s. The offered load is increased from 0.1 to 1.0 in steps of 0.1 for each scenario. In the figures presenting experimental results, each point is the average of ten tests, while the 95% confidence intervals are plotted as error bars on the curves.

#### B. Experimental Results

*1) Throughput and Delay:* We first examine the network-wide throughput under the uniform Bernoulli and On-off bursty traffic models. As shown in Fig. 3(b), each node uniformly sends UDP datagrams to all of its neighbors, and the offered loads for all the nodes are identical.

The network-wide normalized throughput performance are presented in Fig. 4 for the uniform Bernoulli traffic case and in Fig. 5 for the uniform On-off traffic case. It can be seen that when the offered load is low, the achieved network-wide throughput is almost identical to the offered load. However, the normalized throughput saturates at 40% when GR-Limited-1 is used in both Bernoulli and On-off traffic cases, indicating congestion when the offered load exceeds 40%. On the other hand, the GR-PSMAC throughput keeps increasing even when the offered load is close to 100%. The maximum throughput of GR-PSMAC is about twice as high as that of GR-Limited-1.

We next evaluated the frame delay under the same setup as in Fig. 3(b). The average delay for successfully received DATA frames are plotted in Figs. 6 and 7 for the uniform Bernoulli and On-off traffic models, respectively. It can be seen that the GR-PSMAC delay is consistently much lower than the GR-Limited-1 delay for the entire range of offered loads. Under uniform Bernoulli traffic, the GR-PSMAC delay is only 37.16% of the GR-Limited-1 delay when the offered load is 98%. Under uniform On-off bursty traffic, the GR-PSMAC delay is only 23.86% of the GR-Limited-1 delay when the offered load is 81.5%.

(a) The testbed node setup pattern
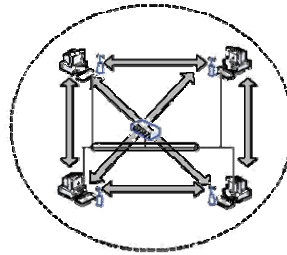
(b) Uniform traffic pattern
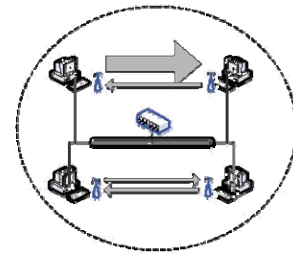
(c) Non-uniform traffic

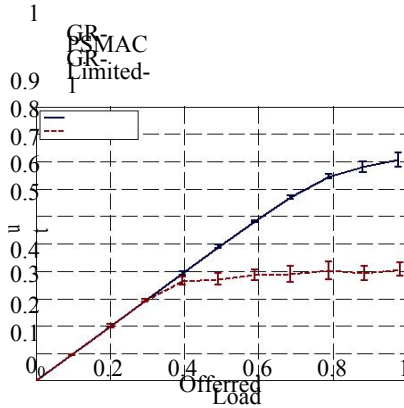Fig. 3.Testbed node and network topology setup for the testbed experiments.



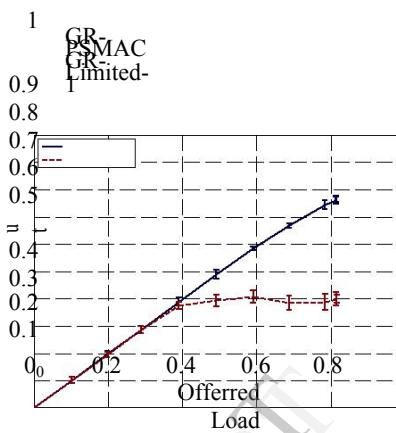Fig. 4.Normalized throughput under uniformi.i.d. Bernoulli traffic.



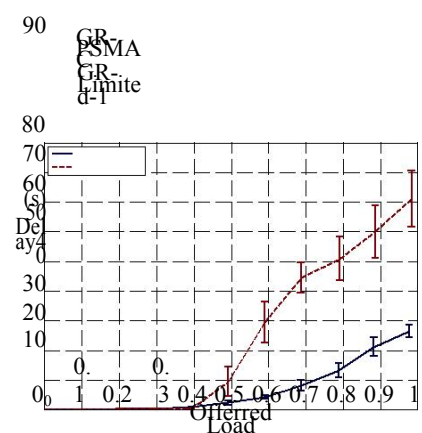Fig. 5.Normalized throughput under uniformOn-off busty traffic.



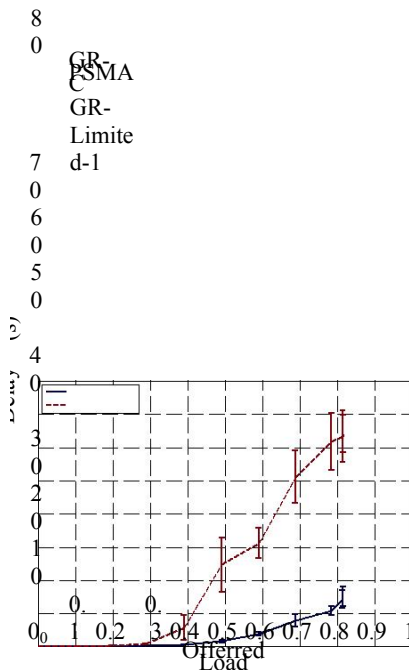Fig. 6.Average frame delay under uniformi.i.d. Bernoulli traffic.



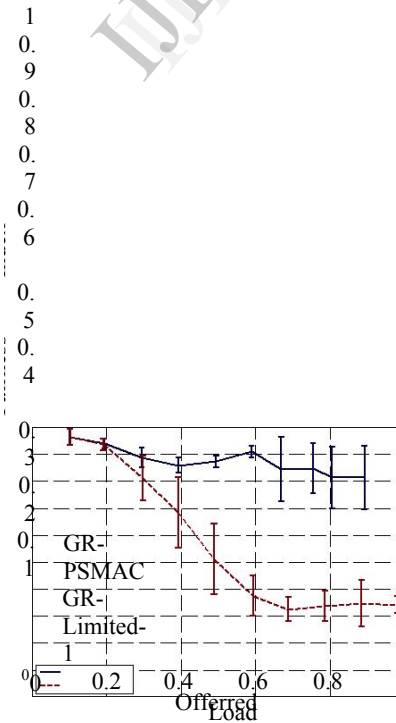Fig. 7.Average frame delay under uniformOn-off bursty traffic.



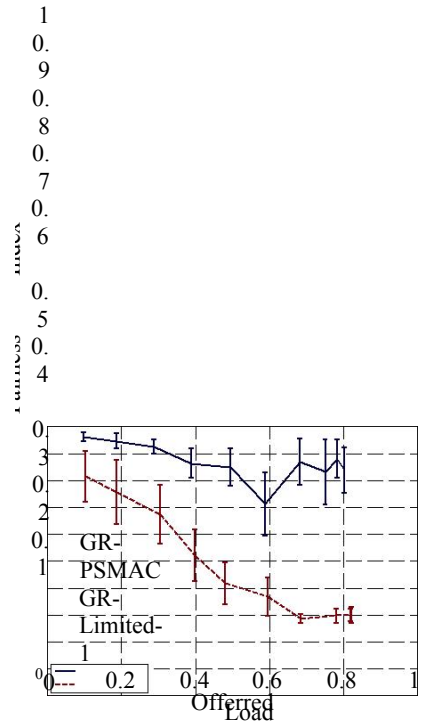Fig. 8.Fairness performance under non-uniform i.i.d. Bernoulli traffic.



Fig. 9.Fairness performance under non-uniform On-off bursty traffic.

176

*2) Fairness:* A common myth about gated or exhaustivepolling service is that although the throughput/delay perfor-manceare superior, the fairness performance may not be good, since a heavily loaded node could use a larger fraction of the link capacity. We next examine the fairness performance with a non-uniform traffic pattern, as illustrated in Fig. 3(c). In this setting, the link from node 1 to node 2 takes 85% of the link capacity, while the other 3 links share the remaining 15% link capacity. Both i.i.d. Bernoulli traffic and On-off bursty traffic are tested. We use the fairness index defined in [8]

$$DN) = \frac{(D_1 + D_2 + \cdots + D_N)^{2f(D1, D2, }}{N(D1^2 + D2^2 + \cdots + DN^2)}$$

where $D_I$ is the average delay at node i, for $i = 1, 2, \cdots, N$. In the fairest case, all the nodes have the same average delay and $f = 1$; in the worst case when one node delay is dominant, $f \approx 1/N$ (and $f = 0$ as $N \rightarrow \infty$).

The fairness indices achieved by GR-PSMAC and GR-Limited-1 are plotted in Figs. 8 and 9 for i.i.d. Bernoulli and On-off bursty traffic models, respectively. In most cases, the GR-PSMAC fairness index is above 80% (except for one point in Fig. 9) even under very high offered load, while the GR-Limited-1 fairness index drops to around 30% when the offered load exceeds 60%.

In Fig. 10, we plot the per node average delay for GR-PASMAC and GR-Limited-1 under the non-uniform Bernoulli traffic. With GR-PSMAC, every node has a delay smaller than 5 s, although node 1 is transmitting at a rate 17 times as high as that of the other three nodes. Under GR-Limited-1, the heavily loaded node 1 has an average delay of 47.29 s, while the other three lightly loaded nodes have much lower average delays.

Therefore, the use of gated service in GR-PSMAC does not result in poor fairness. On the contrary, it achieves much better fairness performance than limited-1 based schemes, such as IEEE 802.11 DCF. This is largely due to the high efficiency and greatly reduced control overhead of PSMAC. All the

queues are efficiently served. The benefit introduced by gated service to a heavily loaded node does not significantly increase the delays of other lightly loaded nodes.

*3) Discussions:* Note that each ACK frame of GR-PSMACcarries more than one sequence numbers and timestamps of all the correctly received DATA frames, for the purpose of delay measurement. In the normal operation mode, the ACK frame can be shorter by carrying the sequence numbers of missing frames instead, and by not carrying the timestamps. Therefore the control overhead could be lower and better throughput and delay performance could be achieved.

We also implemented the access point (AP) mode for GR-PSMAC and GR-Limited-1, where one station acts as AP and the *Next Hop Address* field in the DATA frame header is fixed to the MAC address of the AP. Each none-AP node maintains a single outgoing queue since all the outgoing frames are destined for the AP. Due to limited space, we omit the AP mode experimental results.

### V. RELATED WORK

Efficient MAC protocols have been widely studied. In par-ticular, PSMAC that incorporates gated or exhaustive services was first

introduced in [2], [3]. Limited-k service was used in the CM MAC [9], where k is equal to the *concatenatedthreshold*. In IEEE 802.11e HCF (hybrid coordination func-tion) controlled channel access (HCCA), the HC (Hybrid Coordinator, i.e., the AP) can assign Transmit Opportunities (TXOP) to a station, to allow the station send multiple frames in a row. This is a centralized approach originally designed to support real-time applications with regular traffic patterns. The service discipline is not specified.

The GNU Radio/USRP platform has been a popular choice for prototyping wireless protocols. In [10], an implementation of MIMO PHY function was reported. In [11], an adaptive interference avoidance TDCS (Transform-domain Communi-cation System) based cognitive radio was demonstrated. The Hydra project [12] is a flexible wireless network testbed de-veloped at UT Austin. The project exploits the Click modular router, GNU Radio, and C++ codes to prototype a rate adaptive MAC protocol. CoopMAC [13] is a programmable cooperative communication testbed that implements cooperative protocols in both PHY and MAC layers on the GNU Radio/USRP platform. In a recent work [14], a cooperative communi-cationtestbed for both single-relay cooperation and multi-relay cooperation was reported. The significant performance enhancement for link reliability and end-to-end throughput of cooperative transmissions were observed.

### VI. CONCLUSION

In this paper, we presented the implementation and ex-perimentation of PSMAC, a distributed, gated service based MAC protocol, as well as an IEEE 802.11 DCF like MAC for comparison purpose. The testbed was developed on the GNU Radio/USRP platform. We discussed design issues and presented experimental results under various traffic models and patterns. The experimental study validated the analysis and simulation studies presented in our prior work, and demon-strated the advantages of PSMAC under a realistic setting.
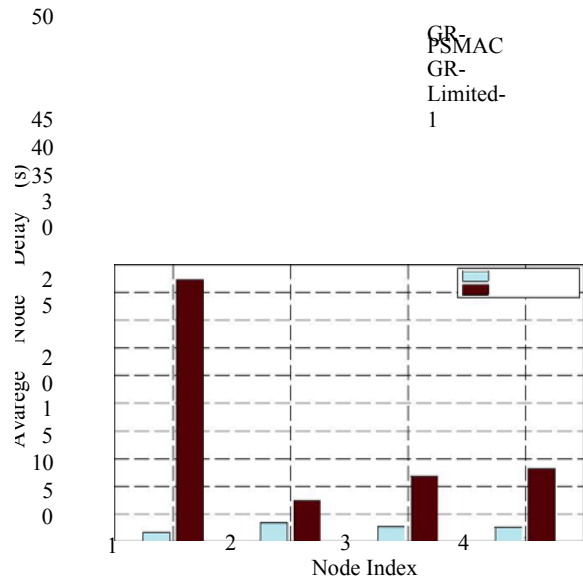
Fig. 10.Average node delay under non-uniform i.i.d. Bernoulli traffic when the offered load is 60%.

REFERENCES

[1] IEEE, "Wireless LAN media access control (MAC) and physical layer (PHY) specifications," 2007.

[2] Y. Li, S. Mao, S. Panwar, and S. Midkiff, "On the performance of distributed polling service-based medium access control," *IEEE Trans.Wireless Commun.*, vol. 7, no. 11, pp. 4635–4645, Nov. 2008.

[3] Y. Li, S. Mao, and S. Panwar, "PSMAC: Polling service-based medium access control for wireless networks," in *Proc. WiOpt'07*, Limassol, Cyprus, Apr. 2007, pp. 1–10.

[4] GNU Radio, [online] Available: http://www.gnu.org/software/gnuradio.

[5] Ettus Research LLC, "USRP family brochure." [online] Available: http://www.ettus.com/downloads/.

[6] T. Schmid, O. Sekkat, and M. Srivastava, "An experimental study of network performance impact of increased latency in software defined radios," in *Proc. WinTECH'07*, Montreal, Canada, Sept. 2007, pp. 59– 66.

[7] IEEE, "IEEE 1588TM-2002 standard for a precision clock synchroniza-tion protocol for networked measurement and control systems," [online] Available: http://www.nist.gov/el/isd/ieee/ieee1588.cfm.

[8] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared systems," 1984, DEC Research Report TR-301.

[9] Y. Xiao, "IEEE 802.11 performance enhancement via concatenation and piggyback mechanisms," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2182–2192, Sept. 2005.

[10] X. Li, W. Hu, H. Yousefi'zadeh, and A. Qureshi, "A case study of a MIMO SDR implementation," in *Proc. IEEE MILCOM'08*, San Diego, CA, Nov. 2008, pp. 1–7.

[11] R. Zhou, Q. Han, R. Cooper, V. Chakravarthy, and Z. Wu, "A software defined radio based adaptive interference avoidance TDCS cognitive radio," in *Proc. IEEE ICC'10*, Cape Town, South Africa, May 2010.

[12] K. Mandke, C. Soon-Hyeok, K. Gibeom, R. Grant, R. Daniels, K. Won-soo, R. Heath, and S. Nettles, "Early results on Hydra: A flexible MAC/PHY multihoptestbed," in *Proc. IEEE VTC'07*, Dublin, Ireland, Apr. 2007, pp. 1896–1900.

[13] T. Korakis, M. Knox, E. Erkip, and S. Panwar, "Cooperative network implementation using open-source platforms," *IEEE Commun. Mag.*, vol. 47, no. 2, pp. 134–141, Feb. 2009.

[14] J. Zhang, J. Jia, Q. Zhang, and E. Lo, "Implementation and evaluation of cooperative communication schemes in software-defined radio testbed," in *Proc. IEEE INFOCOM'10*, San Diego, CA, Mar. 2010, pp. 1307– 1315.