INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2014; **27**:3901–3921
Published online 12 June 2013 in Wiley Online Library (wileyonlinelibrary.com). **DOI**: 10.1002/dac.2584

A distributed polling service-based MAC protocol testbed‡

Yingsong Huang[1], Phillip A. Walsh[2], Yihan Li[1] and Shiwen Mao[1,*,†]

[1]*Auburn University, 200 Broun Hall, Auburn, AL 36849-5201, USA*
[2]*QUALCOMM Inc., 5775 Morehouse Drive, San Diego, CA 92121, USA*

SUMMARY

Medium access control (MAC) protocols play a vital role in wireless networking. It is well-known that the high control overhead of IEEE 802.11 MAC is the limiting factor on the throughput and delay performance of wireless networks. In our prior work, three polling service-based medium access control 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 (GNU Radio/USRP) platform. We discuss various design considerations and challenges of prototyping PSMAC 2 and carry out extensive experimental studies with the GNU Radio/USRP PSMAC testbed. Our experimental results are found to be consistent with the theoretical study reported in our prior work and validate the advantages of PSMAC under a realistic wireless channels. Copyright © 2013 John Wiley & Sons, Ltd.

Received 28 November 2012; Revised 2 May 2013; Accepted 8 May 2013

KEY WORDS: GNU Radio; medium access control; testbed; polling systems; gated service

1. INTRODUCTION

Wireless LANs (WLANs) based on the IEEE 802.11 standard [2] have become the ubiquitous connectivity solution in public as well as residential access networks. 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 medium access control (MAC), however, is well-known for its considerable control overhead, which could consume as much as 40% of the nominal link capacity [3]. The problem gets even worse in the multihop scenario, because of carrier-sensing and spatial reuse issues [4].

In our previous work [5, 6], we present three polling service-based MAC protocols, termed polling service-based medium access control protocols (PSMACs), that can amortize the control overhead of medium contention/resolution over multiple back-to-back frame transmissions, thus achieving high efficiency in MAC. The gated service-based PSMACs are analyzed and compared with p-persistent carrier sense multiple access (CSMA), which closely approximates the standard IEEE 802.11 distributed coordination function (DCF) [7]. Considerable gains on throughput, delay, energy consumption, and fairness performance are observed in the analysis and simulation studies [6].

Polling has been adopted in wireless MAC protocols. For example, the master-driven architecture of Bluetooth piconets provides an ideal setting for applying polling-based scheduling. Polling is adopted in Bluetooth piconets, but the actual scheduling policy has not been prescribed in the current standard [8]. In addition, a polling mechanism has been incorporated in the IEEE 802.11e

*Correspondence to: Shiwen Mao, Auburn University, 200 Broun Hall, Auburn, AL 36849-5201, USA.
†E-mail: smao@ieee.org
‡This work was presented in part at IEEE MILCOM 2011, Baltimore, MD, on November 2011 [1].

hybrid coordination function (HCF) [9]. In the HCF controlled channel access mode, the hybrid coordinator polls the quality of service enhanced stations (QSTA) to assign them transmission opportunities (TXOP). A TXOP is a bounded time interval in which a QSTA is allowed to transmit one or more frames. Again, the specific scheduling policy has not been specified. Recently, the reverse-direction protocol has been suggested for IEEE 802.11n to support higher speed and higher throughput [10]. This technique provides an opportunity for a recipient to transmit data to a sender during the sender's TXOP, which is suitable for the highly asymmetrical traffic network applications, such as FTP and HTTP. Because the network allocation vector (NAV) duration may be changed in clear to send (CTS) message to support the 'bidirectional' TXOP, more complex schemes are needed to handle the hidden terminal problem.

There are two fundamental differences between the proposed PSMAC and the aforementioned existing approaches. First, the schemes adopted in Bluetooth and IEEE 802.11e are centralized ones, where a master or base station polls other stations. They are designed for relatively simple network topologies (e.g., a piconet with one master and seven slaves [8] or a single-hop WLAN). However, there may be no such master/base station in distributed wireless networks. These centralized approaches are quite different from the random access and fully distributed approach taken in PSMAC. Second, even for single-hop networks, the specific scheduling policy is not specified in either Bluetooth or IEEE 802.11e. More importantly, there is a need for both theoretical and experimental studies to underpin the scheduling techniques to be adopted in both standards.

In this paper, we extend the prior work by prototyping the PSMACs in a real wireless networking environment. Generally, testbeds can provide useful insights that computer-based simulations cannot offer, because 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 channels and verify our prior theoretical and simulation studies but also identify new practical constraints and problems.

Therefore, the goal of the implementation is to create a reconfigurable and completely functional PSMAC testbed, as well as evaluate the PSMAC performance under realistic wireless channels. We make two main contributions in this paper. First, we implement the PSMACs on the GNU radio [11] and universal software radio peripheral (USRP) [12] platform. We integrate the key functions of 802.11 DCF and the gated service policy in the implementation, such as gated service scheduling, carrier sense multiple access/collision avoidance (CSMA/CA), virtual carrier sensing, request to send/clear to send (RTS/CTS) handshake, automatic repeat request (ARQ), random back-off mechanism, and distributed clock synchronization, by 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. These are consistent with the trend observed in the theoretic analysis and simulation studies in our prior work [5, 6] and thus further validate the superior performance of PSMAC in a realistic wireless network setting.

The remainder of this paper is organized as follows. We first describe the system overview in Section 2 and discuss implementation details in Section 3. Our experimental results are presented in Section 4. Related work is discussed in Section 5, and Section 6 concludes the paper with a discussion of future work.

2. TESTBED SYSTEM OVERVIEW

2.1. Polling service-based medium access control protocol

In this section, we briefly review PSMAC to provide the necessary background for the testbed. We refer interested readers to [6] for more details.

Polling service-based medium access control protocols are motivated by the insights from polling system theory [6]. Generally, a polling system consists of a shared resource (i.e., the wireless channel) and multiple stations (i.e., the wireless nodes). Polling systems may have either a centralized or a distributed structure. In the centralized case, a server maintains state information of the stations and

DOI: 10.1002/dac

polls the stations for channel access. In the distributed scenario, the stations contend for channel access by using a distributed mechanism. In either case, one of the following three types of service policies can be used to serve the frames for a wining station: (i) *exhaustive* policy, where a station is served until its buffer is emptied; (ii) *gated* policy, where a station is served until all the frames that have backlogged in its buffer when the service begins are transmitted; and (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% [13].

On the basis of the polling system theory, three PSMACs are introduced in [5, 6]. The main idea is to serve multiple frames after a successful contention resolution, thus amortizing the high control overhead over multiple 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 nonempty virtual queues when the station wins the channel. This allows other neighbors that are not involved in the transmissions be scheduled to sleep for energy conservation. PSMAC 3 extends PSMAC 2 by serving all nonempty virtual queues when a station wins the channel, which may achieve even higher efficiency.

In [5, 6], the PSMACs are evaluated with analysis and simulations. They are shown to achieve considerable throughput and delay improvements over p-persistent CSMA, which is used as a proper benchmark for the performance evaluation because of its similarity to the IEEE 802.11MAC [7]. In addition, PSMACs 2 and 3 can achieve significant energy savings by scheduling nodes to sleep, when they are not involved in the transmission of a packet train. The PSMACs are also shown to be more efficient for handling bursty traffic types and asymmetric traffic patterns, and the performance gains are achieved without sacrificing fairness performance [5, 6].

When $k = 1$, the limited-1 policy is a special case of limited-k with only up to 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 because it is most compatible to the 802.11 DCF MAC. We also implement a limited-1-based IEEE 802.11 DCF-like protocol for performance comparison purpose. Both implementations are based on the GNU Radio and universal software radio peripheral (GNU Radio/USRP) platform [11, 12]. We call the PSMAC 2 and limited-1 MAC implementations GR-PSMAC and GR-Limited-1, respectively, in the rest parts of the paper (where GR stands for GNU Radio).

2.2. *GR-PSMAC and GR-Limited-1*

We implement GR-PSMAC and GR-Limited-1 by extending the IEEE 802.11 DCF, which is the *de facto* protocol for Wi-Fi networks. In particular, the implementations integrate CSMA/CA with binary exponential back off, virtual carrier sense, RTS/CTS handshake, and ARQ for link error control to make full operational MAC protocols.

In GR-PSMAC, a station maintains multiple *virtual queues*, one for each of its neighbors. That is, DATA frames for different neighbors are enqueued into different virtual queues. When there is one or more nonempty virtual queues, the source station selects a nonempty virtual queue in a *round-robin* fashion and starts to sense the channel. After the channel is idle for DCF inter-frame space interval, the contention window (*CW*) starts to decrease. If the channel remains idle when *CW* reaches 0, a request to send (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 may return a CTS frame to confirm that it is ready for receiving data. The CTS frame contains the transmission duration, which may contain multiple frame durations and allow other stations to set up their NAV for virtual carrier sensing. After receiving the CTS, the gated service will be used for the selected virtual queue, that is, the source station will transmit its backlogged DATA frames back to back to the destination, following the gated service policy. All other stations will keep silent and wait for the NAV to expire (or they may be scheduled to sleep for energy conservation). When the last frame is received, an

DOI: 10.1002/dac

acknowledgement packet (ACK) frame is issued by the target receiver to acknowledge all the successfully received frames, which will be removed from the virtual queue at the source station. If some frames are not correctly received after the transmission phase, the back-off procedure will be performed at the source station to defer the transmissions. The lost or corrupted frames will be retransmitted at a later time. This procedure is illustrated in Figure 1.

The back-off procedure used in the implementation is illustrated in Figure 2, which follows the IEEE 802.11 DCF specification [2]. After the DCF inter-frame space medium idle time, the source station shall generate a random back-off period for additional deferral before transmitting, unless the back-off timer already contains a nonzero value. The back-off 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}. In this paper, we set $CW_{\mathrm{min}}=8$ and $CW_{\mathrm{max}}=256$ as in [2]. After each successful transmission or when the number of RTS retries reaches a predefined maximum value, CW will be reset to CW_{min}.

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

2.3. Software and hardware platforms

We develop the PSMAC testbed on the software-defined radio (SDR) platform consisting of GNU Radio and USRP [11, 12]. SDR is a modern approach to wireless communications [14], which allows dynamic reconfiguration of waveforms by software. GNU Radio [11] is an open-source software development toolkit under the GNU General Public License. It provides signal processing runtimes and processing blocks to implement SDR on radio frequency (RF) hardware and commodity processors. GNU Radio applications are usually written in Python scripts, which allow the quick reconfiguration of the protocols, whereas the compiled C++ codes are used for the signal-processing components of physical layer for minimal processing time. USRP [12] is a generic SDR hardware

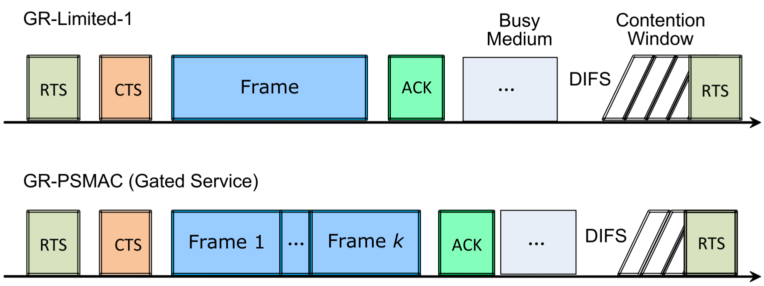


Figure 1. Illustration of the protocol operation: GR-Limited-1 vs. GR-PSMAC. RTS, request to send; CTS, clear to send; ACK, acknowledgement packet; DIFS, distributed interframe space.

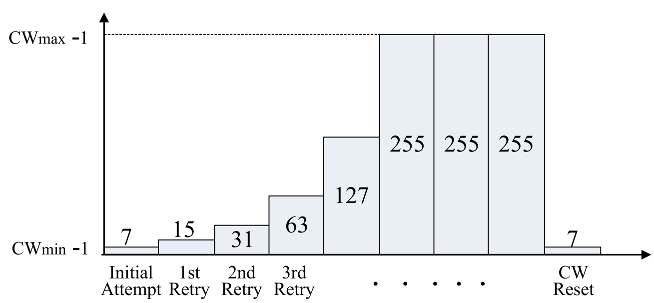


Figure 2. The *CW* back-off procedure. CW, contention window.

DOI: 10.1002/dac

device that natively integrates with GNU Radio. We use USRP1 as the hardware platform for prototyping. The motherboard of USRP1 consists of four 64-MS/s analog-to-digital converters and four 128-MS/s digital-to-analog converters. It has a field-programmable gate array (FPGA) for processing baseband and intermediate frequency signals. The RFX2400 RF front-end daughterboard supports transmission and receiving from 2.3 to 2.9 GHz in the industrial, scientific, and medical band. Integrated with USRP, GNU Radio provides a compelling software platform for prototyping wireless communications and networking protocols.

During the implementation, we observe that the main limitation of GNU Radio for MAC development is the high latency. Most MAC protocols rely on precise receiving and transmission timing. For example, IEEE 802.11 requires precise timing for the virtual carrier-sensing mechanism. However, GNU Radio introduces a non-negligible latency because of 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 latency hard to track. It is reported in [15] that the modulation, spreading, demodulation, and despreading procedures could introduce an additional 22.5-ms delay, which is quite large compared with the standard timing setting in IEEE 802.11 (generally in the microsecond scale).

The large latency also negatively affects performance measurement during testbed experiments, especially under high transmission rates. The slow processing in the stations does not match the high data rate in the channel. To tackle this problem, we use a relatively small link rate along with a large frame size to mitigate the impact of latency on transmissions to make the PSMAC implementation work. For example, with using a 125-kbps link capacity with 1500-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 with the given platform. It is worth noting that the Gigabit Ethernet interface used in the later version of USRP and implementing the protocol functions in the FPGAs as in Rice University's wireless open-access radio platform [16] will help to alleviate the latency issue.

3. TESTBED IMPLEMENTATION DESCRIPTION

We develop the MAC protocols on the GNU Radio/USRP platform [11, 12]. Each wireless station in the testbed consists of a USRP1 unit and a laptop computer, as illustrated in Figure 3. We describe the implementation-related issues in this section.

3.1. Network protocol architecture

Both GR-PSMAC and GR-Limited-1 are implemented as Layer 2 protocols from the point of view of network protocol architecture. Both protocols are written in Python scripts and are running in the user space of Linux (Linux.org). Because there is no explicit interface to directly access the MAC from the user space, we resort to the Linux TAP/TUN virtual network interface that provides the

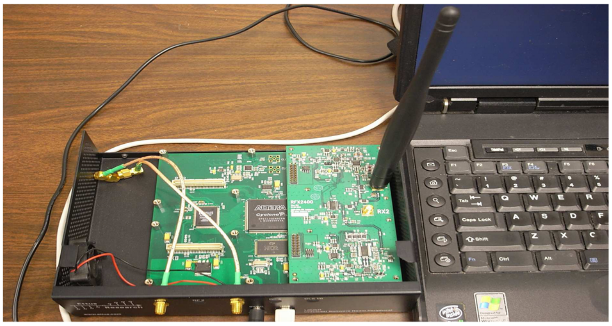


Figure 3. The testbed wireless station setup.

bridge between GNU Radio and Linux TCP/IP kernel. Specifically, we create a *virtual* Ethernet interface, termed gr_0, which can be configured with an IP address. Applications can then use the MAC protocols implemented in GNU Radio transparently as a standard network application programming interface. This approach is illustrated in Figure 4.

To implement the MAC layer functions, we design the MAC header as given in Table I, which is similar to that of IEEE 802.11. The PSMAC header contains eight fields and is 16-byte long in total. Although some of the fields are compatible with the header definition of IEEE 802.11, the header format is different from that of the standard Ethernet header. For example, standard 48-bit MAC addresses are used for the Linux TAP/TUN frame, but 2-byte addresses are used to identify the USRP hardware in PSMAC. Therefore, frames from the upper layer through the TAP/TUN driver will require a mapping from Ethernet header to PSMAC header, as illustrated in Figure 4. Similarly, GR-PSMAC and GR-Limited-1 also map PSMAC headers back to the Ethernet header for received frames.

Note that using this customized header makes it incompatible with legacy 802.11 systems. For such a hybrid system, maybe a dual protocol stack is needed: packets are classified as 802.11 and PSMAC packets according to their header format and then sent to different MAC units for processing.

3.2. Transmission and receiving path

The GR-PSMAC is implemented as two execution data paths, namely, the *transmission path* and the *receiving path*. We adopt multithreading, and each path is controlled by a thread. We use the

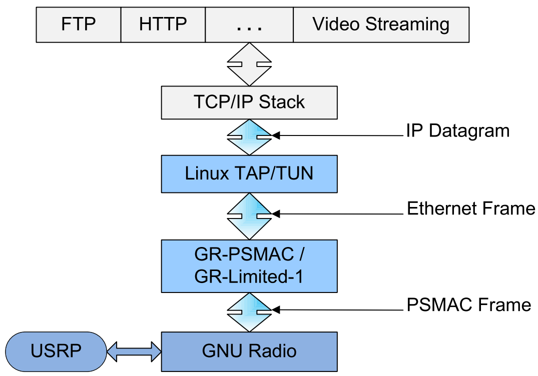


Figure 4. Protocol architecture of the GNU Radio testbed. USRP, universal software radio peripheral; PSMAC, polling service-based medium access control protocol; IP, internet protocol.

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 address
Source address	2	Source address
Next hop address	2	Address of the next hop, only valid for DATA frames, and is used for the access-point mode or multihop mode
Duration	4	Multipurpose 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

RTS, request to send; CTS, clear to send; ACK.

DOI: 10.1002/dac

threading library in Python to manage the synchronization of the two paths. The design of the two paths is outlined in the following texts.

3.2.1. Transmission path. When GR-PSMAC receives a DATA 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. As discussed, GR-PSMAC maintains a virtual queue for each of its neighbors. The DATA frames are enqueued to the virtual queues according to their destination MAC addresses.

If the channel is sensed idle, the station selects a nonempty 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 back off the transmission and increase the RTS retry counter by 1. Furthermore, if the RTS retry number exceeds a predefined limit, GR-PSMAC will reset *CW* and serve the next nonempty outgoing virtual queue for fair operation among the virtual queues.

On the other hand, if a CTS frame is successfully received, GR-PSMAC will reset its *CW*, transmit the DATA frames in a row that had been backlogged in the selected virtual queue when the RTS was sent, and wait for ACK. If an ACK frame is received before timeout, GR-PSMAC will purge the acknowledged frames from the outgoing virtual queue. Otherwise, it will back off the transmission and try to serve the next nonempty outgoing virtual queue.

3.2.2. Receiving path. When a station receives an RTS destined for itself (i.e., carrying its MAC address as destination), it sets its NAV according to the *duration* field value in the RTS. 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 the successfully received sequence numbers back to the source station. The source station, once receiving the ACK, will remove all the successfully transmitted frames from its outgoing virtual queue.

Because both the transmission path (i.e., sending DATA frames) and receiving path (i.e., clearing acknowledged DATA frames) need to access the outgoing virtual queues, the multi-thread control needs to be designed. A fast thread synchronization lock is introduced to protect the access conflict of the common resources of the transmission path and receiving path.

The basic flowcharts of the transmission and receiving paths are illustrated in Figure 5.

3.3. Acknowledgement and retransmission mechanisms

We also implement the acknowledgement 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. That is, a stop-and-wait ARQ mechanism is sufficient in this case. 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 station 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) by reusing the duration field (Table I). All the frames received after the first missing frame will be discarded and 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 and reduce the retransmission

DOI: 10.1002/dac

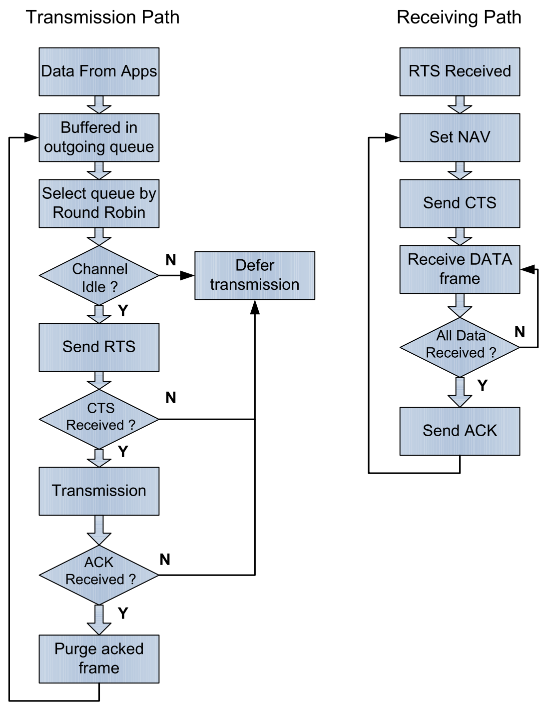


Figure 5. The flowcharts for the transmission and receiving paths. RTS, request to send; CTS, clear to send; ACK, acknowledgement packet; NAV, network allocation vector.

cost, we also implement the selective repeat protocol (SRP). In SRP, the ACK issued by the destination node contains an explicit list of the sequence numbers of successful received frames; only the missing frames need to be retransmitted. SRP is generally more efficient than Go-Back-N protocol, because it can reduce the number of retransmission, but with a slightly higher control overhead (i.e., longer ACK frames) and complexity.

3.4. Synchronization for distributed delay measurement

In a distributed network scenario, the CPU clocks may not be precisely synchronized. This may introduce frame delay measurement errors. To address the synchronization issue, we adopt the precision time protocol daemon that implements the IEEE 1588 standard [17] to synchronize the testbed nodes. IEEE 1588 provides real-time clock synchronization for distributed systems with sub-microsecond precision. Such precision is sufficient for experiments and delay measurement in the PSMAC testbed.

We implement the delay measurement in the MAC layer as follows. The testbed nodes are connected with an Ethernet hub and are then synchronized with the Precision Time protocol daemon. When a DATA frame is enqueued at the source node, a timestamp will be stored at the source node. When a DATA frame is successfully received, the destination node will attach a timestamp in the ACK frame that records the time when the DATA frame was received, along with the list of sequence numbers. The MAC layer can directly monitor the outgoing queues and the event of frame receptions, which is free from the extra scheduling latency in the upper layers. The source station can compute the one-way delay as the difference between the received (i.e., in the ACK frame) and the stored timestamps.

During the testbed experiments, we use the aforementioned mechanism frame delays for evaluation of the proposed schemes. For normal operation of the PSMAC implementation, however, such synchronization (and the Ethernet connections) is not required. Furthermore, with SRP ARQ, each ACK frame of GR-PSMAC carries more than one sequence numbers and timestamps of all the correctly received DATA frames, for the purpose of one-way delay measurement. In the normal operation mode, the ACK frame can be much shorter by carrying the sequence numbers of missing frames only and by not carrying the timestamps. Therefore, the control overhead could be further reduced, and better throughput and delay performance could be achieved.

4. EXPERIMENTS AND RESULTS

4.1. Experiment setting

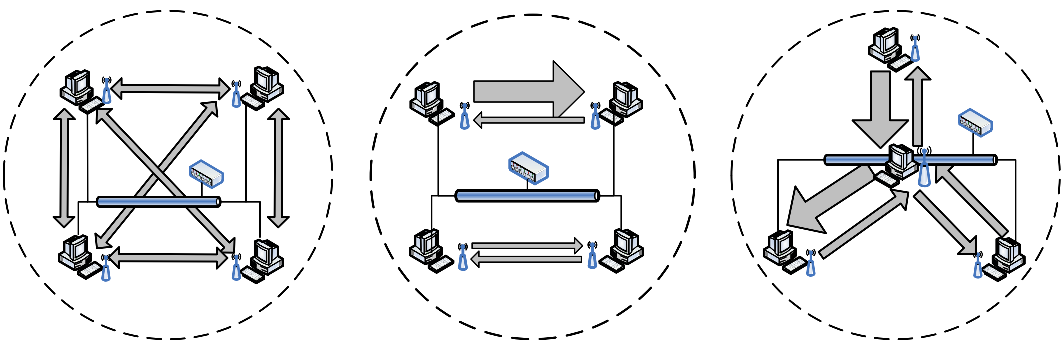
The GNU Radio PSMAC testbed consists of four USRP1 kits, each connected to a general purpose computer through a USB 2.0 port, as shown in Figure 3. GR-PSMAC and GR-Limited-1 are implemented in GNU Radio 3.3 with Ubuntu Linux OS. As discussed, we also connect all the computers to an Ethernet hub and synchronize their clocks with IEEE 1588 (for accurate measurement of one-way frame delays).

We run the experiments by using two network topologies as shown in Figure 6: (i) a single-hop ad hoc network topology where each node communicates with every other node, as shown in Figure 6(a) and (b); and (ii) an access point (AP) mode, where one station serves as the AP to relay traffic for the other three stations, as shown in Figure 6(c). During the tests, the four testbed stations share the 125-kbps nominal link capacity. The more efficient SRP ARQ scheme and the RTS/CTS virtual carrier-sensing mechanism are used. The default parameter settings are listed in Table II.

We develop a UDP client–server application in C++ that can generate traffic to drive the experiments. UDP is chosen to avoid the complex rate variations caused by the transmission control protocol congestion control, thus focusing on the MAC performance. The following three traffic models are used in the experiments:

(1) *Independent identically distributed* (*i.i.d.*) *Bernoulli* traffic—a frame is generated in each time slot with a predefined probability.
(2) *On–off bursty* traffic—frames are generated according to an on–off Markovian model with geometrically distributed on and off periods. The average on period is five, whereas the average off period is tuned to achieve different offered loads, for the results reported in this section.
(3) *Long-range dependent* (LRD) traffic—frames are generated according to an on–off traffic model with Pareto-distributed on and off periods. It is shown that such source exhibits long-range dependence [18]. The Hurst parameter is chosen to be $H = 0.7$ for the results reported in this paper.

The first two traffic models belong to the class of *short-range dependent* (SRD) models and are sufficient for modeling voice over IP traffic, and the LRD model is useful for modeling computer data traffic, which is shown to be self similar [18]. The LRD model is much more bursty than



(a) Ad hoc mode with a uniform traffic pattern
(b) Ad hoc mode with a non-uniform traffic pattern
(c) AP mode with a non-uniform traffic pattern

Figure 6. Network topology setups for the testbed experiments. AP, access point.

DOI: 10.1002/dac

Table II. Testbed parameters and values.

Carrier frequency	2.401 GHz
Modulation scheme	GMSK
Samples per symbol	2
RTS retry limit	5
DATA frame size	1500 bytes

RTS, request to send; GMSK, Gaussian-filtered minimum-shift keying.

the first two traffic models, and the experiments with LRD traffic model take much longer time to converge to the steady state.

In the tests, we also consider different traffic patterns by controlling the traffic rates at the source stations and the destination address of the generated DATA frames. With the *uniform* traffic pattern, the destination of each DATA frame is uniformly distributed among all the neighbors; with the nonuniform traffic pattern, one source–destination pair has much higher load than others.

In the remainder of this section, the offered load is defined to be the ratio of the rate that packets are generated (at all the stations) and the packet rate that the channel can transmit. During each experiment, we let each station record the amount of packets it receives. The normalized throughput is defined to be the total amount of packets received at all the stations divided by the total amount of packets the channel can transmit during the same time interval. Note that if a packet is forwarded (as in the infrastructure mode), it is only counted once. For each offered load, we run the testbed experiment for 10 times. Each experiment lasts for 300 s when the i.i.d. Bernoulli and on–off bursty traffic models are used and 3000 s when the LRD traffic model is used. The offered load is increased from 0.1 to 1.0 in steps of 0.1 for the test scenarios. In the figures presenting experimental results, each point is the average of the 10 tests, whereas the 95% confidence intervals are plotted as error bars.

4.2. *Experimental results*

4.2.1. *Performance under SRD traffic: ad hoc mode.*

4.2.1.1 ***Throughput and Delay.*** We first examine the network-wide throughput under the uniform Bernoulli and on–off bursty traffic models and the uniform traffic pattern. As shown in Figure 6(a), 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 performances are presented in Figure 7 for the uniform Bernoulli traffic case and in Figure 8 for the uniform on–off traffic case. It can be seen that when the

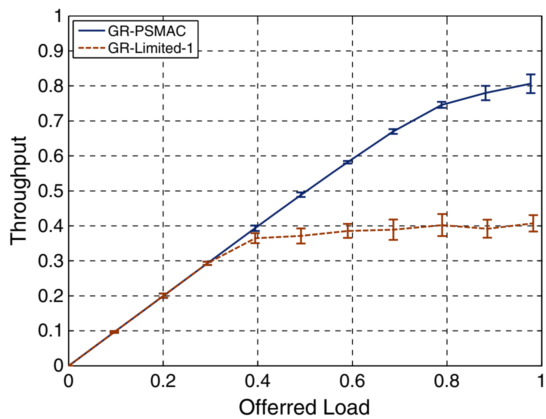


Figure 7. Normalized throughput under uniform i.i.d. Bernoulli traffic: ad hoc mode.

DOI: 10.1002/dac

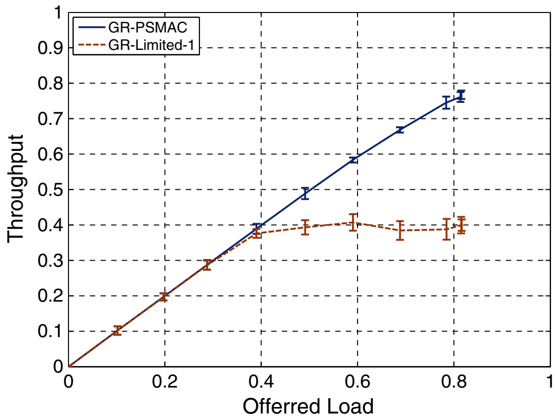


Figure 8. Normalized throughput under uniform on–off busty traffic: ad hoc mode.

offered load is low, the achieved network-wide throughput is almost identical to the offered load. However, the normalized throughput saturates at about 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 Figure 6(a). The average delay for successfully received DATA frames is plotted in Figures 9 and 10 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%.

4.2.1.2 ***Fairness.*** A common myth about gated or exhaustive polling service is that although the throughput/delay performances are superior, the fairness performance may not be good, because a heavily loaded node could use a larger fraction of the link capacity. To validate this common belief, we next examine the fairness performance with a nonuniform traffic pattern, as illustrated in Figure 6(b). In this setting, the link from station 1 to station 2 takes 85% of the offered load, whereas the other three links share the remaining 15% offered load. Both the i.i.d. Bernoulli traffic and on–off bursty traffic models are tested. We use the

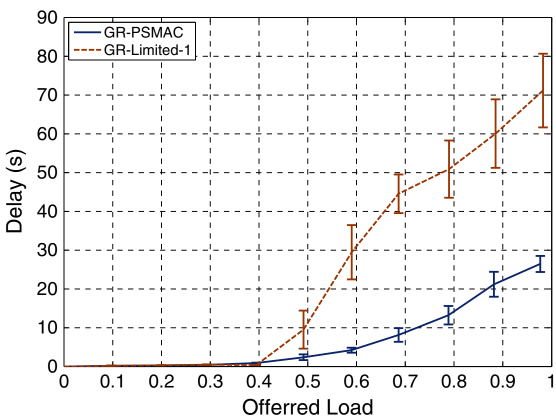


Figure 9. Average frame delay under uniform i.i.d. Bernoulli traffic: ad hoc mode.

DOI: 10.1002/dac

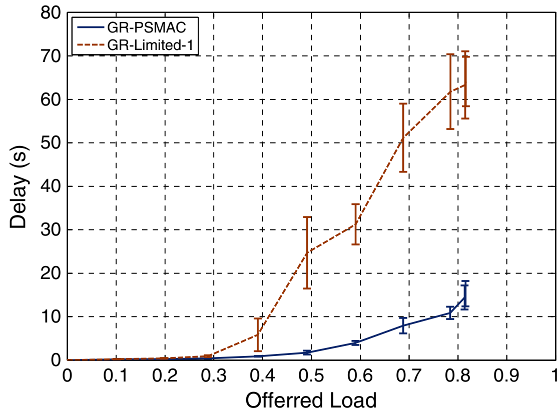


Figure 10. Average frame delay under uniform on–off bursty traffic: ad hoc mode.

fairness index defined in [19]. For a system with N stations, the fairness index is

$$f(D_1, D_2 \cdots, D_N) = \frac{(D_1 + D_2 + \cdots + D_N)^2}{N\left(D_1^2 + D_2^2 + \cdots + D_N^2\right)},$$

where D_i is the average delay for the frames transmitted by station i, for $i = 1, 2, \cdots, N$. It can be verified that f is always between 0 and 1. In the fairest case, all the nodes have the same average delay, that is, $D_1 = D_2 = \cdots = D_N$, and we have $f = 1$. In the worst case when one station's delay is dominant, that is, $D_i >> D_j$, for all $j \neq i$, we have $f \approx 1/N$ (and $f = 0$ as $N \to \infty$).

The fairness indices achieved by GR-PSMAC and GR-Limited-1 are plotted in Figures 11 and 12. It can be observed that all the fairness index curves drop as the offered load is increased, indicating the negative effect of congestion on fairness performance. In most cases, the GR-PSMAC fairness index is above 80% even under very high offered load, except for one point in the on–off traffic case. On the other hand, the GR-Limited-1 fairness index curves drop to around 30% when the offered load exceeds 60% under both traffic patterns.

For further insights, we plot the per-station average delay for GR-PSMAC and GR-Limited-1 under the nonuniform Bernoulli traffic in Figure 13. We focus on the knee point when the offered load is 60%. It can be seen that with GR-PSMAC, every station has an average delay smaller than 4 s. Although station 1 is transmitting at a rate 17 times as high as that of the other three stations, their average delays are close to each other, ranging between 1.60 and 3.40 s. Under GR-Limited-1, the heavily loaded station 1 has an average delay of 47.29 s, whereas the other three lightly loaded

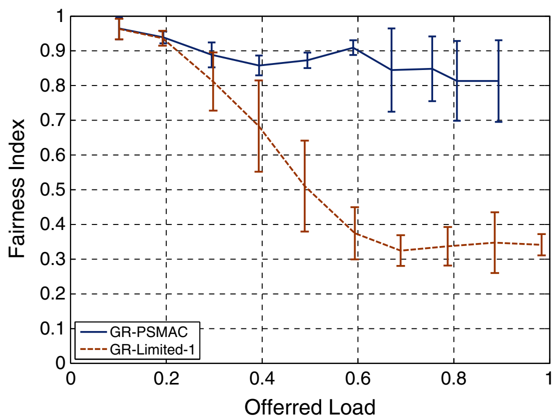


Figure 11. Fairness performance under nonuniform i.i.d. Bernoulli traffic: ad hoc mode.

DOI: 10.1002/dac

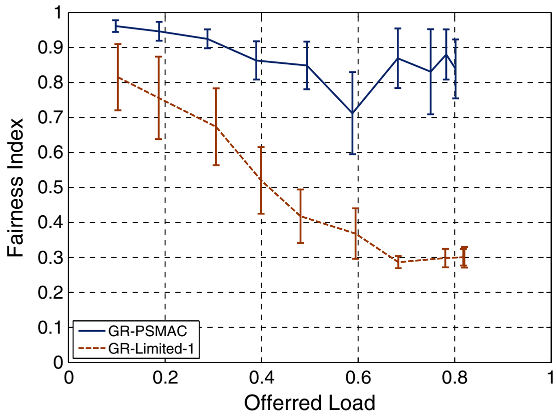


Figure 12. Fairness performance under nonuniform on–off bursty traffic: ad hoc mode.

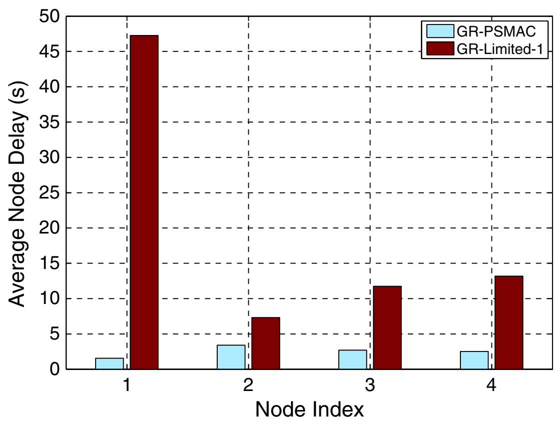


Figure 13. Average delay for each station under nonuniform i.i.d. Bernoulli traffic when the offered load is 60%: ad hoc mode.

nodes have much lower average delays (all less than 14 s). GR-PSMAC achieves not only lower per-station average delay than that by GR-Limited-1 but also more evenly distributed average delays among the stations than those by GR-Limited-1.

Therefore, the use of gated service in GR-PSMAC does not result in poor fairness. On the contrary, it achieves better fairness performance than limited-1-based schemes. This is largely due to the high efficiency and greatly reduced control overhead of PSMAC. All the virtual queues are efficiently served. The benefit introduced by gated service to a heavily loaded station does not significantly increase the delays of other lightly loaded nodes.

4.2.2. Performance under LRD traffic: ad hoc mode. In addition to i.i.d. Bernoulli and on–off bursty traffic models, we also investigate the testbed performance under the LRD traffic model. It has been well known that compute data and variable-bit-rate video traffic are *self similar*, with Hurst parameters ranging from 0.5 to 1.0 [18]. For such traffic type, the class of SRD traffic models is inadequate to capture the complex autocorrelation structure. We adopt the on–off traffic model with Pareto-distributed on/off periods, which is an accurate model for LRD sources. By tuning the average duration of the off periods, the LRD process has a Hurst parameter of 0.7 for the experiments.

The simulation results with the LRD sources are presented in Figures 14–16. These results are obtained with the same topology and setting as the previous experiments, except that the traffic source is now the LRD source. In general, all the performance curves with the LRD sources have

DOI: 10.1002/dac

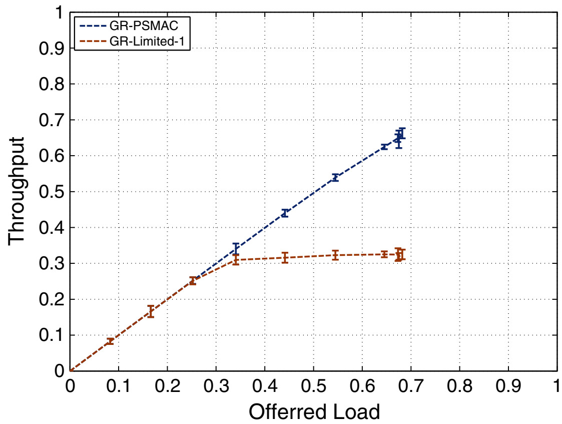


Figure 14. Normalized throughput under uniform LRD traffic: ad hoc mode.

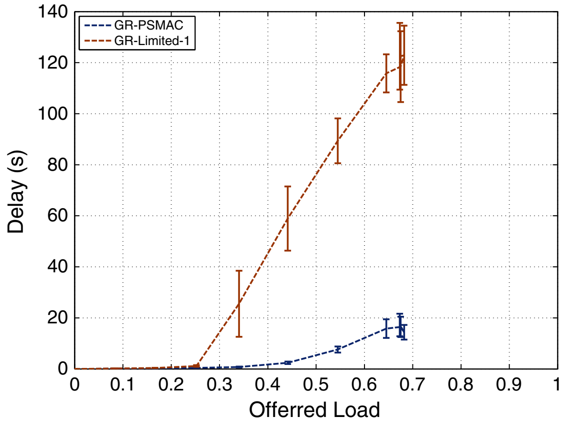


Figure 15. Average frame delay under uniform LRD traffic: ad hoc mode.

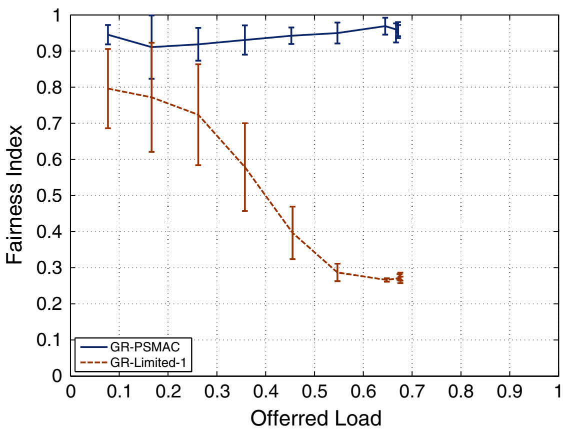


Figure 16. Fairness performance under nonuniform LRD traffic: ad hoc mode.

the same trend as those in the SRD case, and significant performance gains in throughput, delay, and fairness are achieved by GR-PSMAC over GR-Limited-1.

Furthermore, GR-Limited-1 has worse performance under LRD traffic than that under SRD traffic. In Figure 14, the throughput becomes saturated when the offered load exceeds 30%, which

DOI: 10.1002/dac

is earlier than the 40% offered load in the SRD case. The saturated throughput is 30%, which is also lower than the 40% saturated throughput in the SRD case. In Figure 15, the SRD delay starts to diverge before the offered load reaches 30%, which is earlier than the SRD case in Figure 10. When the offered load is 50%, for example, the GR-Limited-1 achieves an average delay of 76 s in the LRD case, a big increase from 25 s in the SRD case. In Figure 16, the LRD fairness performance of GR-Limited-1 is similar to that in the SRD case, although the fairness index in the high offered load range is slightly lower than that in the SRD case. Such performance degradation clearly demonstrates the negative impact of LRD traffic on the network system performance.

On the other hand, we do not observe any performance degradation when the traffic sources are LRD. For example, the throughput curve in Figure 14 is similar to that in Figure 8, and the delay curve in Figure 15 is also similar to that in Figure 10, for the range of offered load examined. In Figure 16, the GR-PSMAC fairness indices are all higher than 0.9, which is slightly better than that of the SRD case shown in Figure 12.

Under LRD sources, it is more likely that the backlogged frames are concentrated in a small number of virtual queues. With gated service, such backlogged virtual queues can be quickly cleared out during one service period. Therefore, GR-PSMAC is more effective in support LRD traffic, which has high rate variations and is generally very difficult to manage and control.

4.2.3. Performance in the AP mode. Finally, we examine the performance of GR-PSMAC with an infrastructure-based wireless network topology. As shown in Figure 6(c), one station is configured to operate as the AP and the remaining three stations WLAN nodes that communicate with each other through the AP. With PSMAC, each WLAN node maintains a single outgoing queue, because the field of next hop address in all outgoing frame headers is fixed to the address of the AP. The AP buffers the incoming packets in three different virtual queues, one for each of the WLAN nodes. We configure a nonuniform traffic pattern and use the on–off burst traffic model for this star topology. Specifically, the traffic flow from station 1 to station 2 takes 30% of the offered load, whereas the traffic flow from station 2 to station 3 and the traffic flow from station 3 to station 1 take 10% of the offered load, respectively. Because the AP relays traffic, each frame will be transmitted twice. This scenario can also represent a multihop wireless topology, in which all the none-AP nodes are two hops from each other, and the AP becomes a hotspot of the multihop network.

In Figure 17, we plot the normalized throughput for the AP topology. We observed that GR-PSMAC still achieves considerably higher throughput under different offered loads, whereas the throughput of GR-Limited-1 becomes saturated when the offered load exceeds 40%. The average frame delays are plotted in Figure 18. It can be observed that the GR-PSMAC delays are consistent with the previous experiments, whereas the GR-Limited-1 delay shoots up when the offered load exceeds 30%. In Figure 19, the GR-PSMAC fairness indices are constantly above 0.8, whereas the GR-Limited-1 fairness index curve drops when the offered load exceeds 20%,

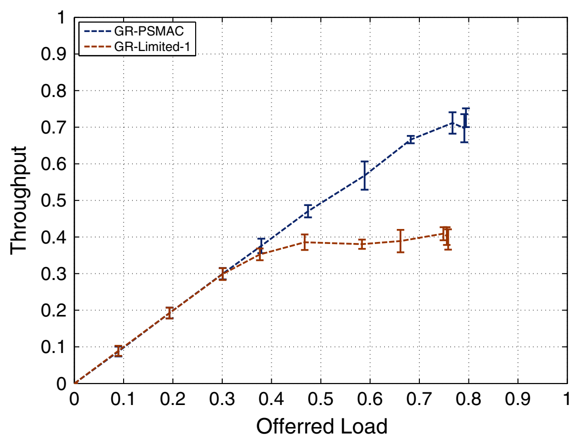


Figure 17. Normalized throughput with on–off bursty nonuniform traffic: AP mode.

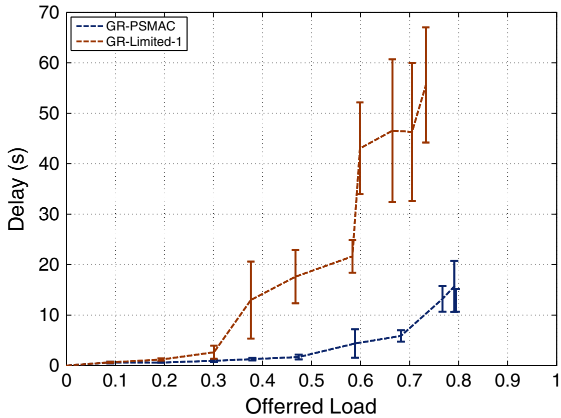


Figure 18. Average frame delay with on–off bursty nonuniform traffic: AP mode.

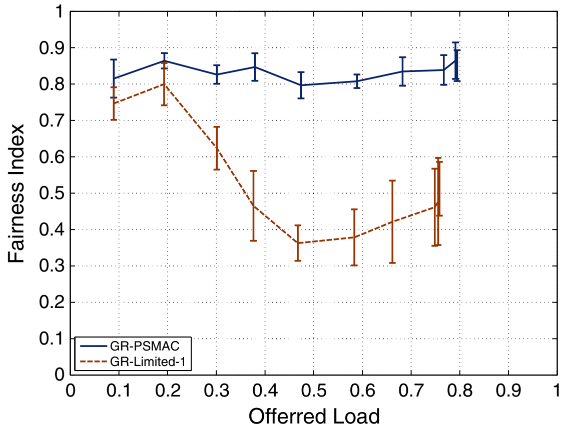


Figure 19. Fairness performance with on–off bursty nonuniform traffic: AP mode.

to about 0.4 for the high offered load region. All these AP results are consistent with those for the ad hoc topology.

We further plot the average backlog length of the virtual queues at the AP in Figure 20. The offered load is 60%. The ith virtual queue stores frames to be transmitted by the AP to the ith WLAN node. The average virtual queue backlogs for GR-PSMAC are 10.4, 25.5, and 2.3, whereas

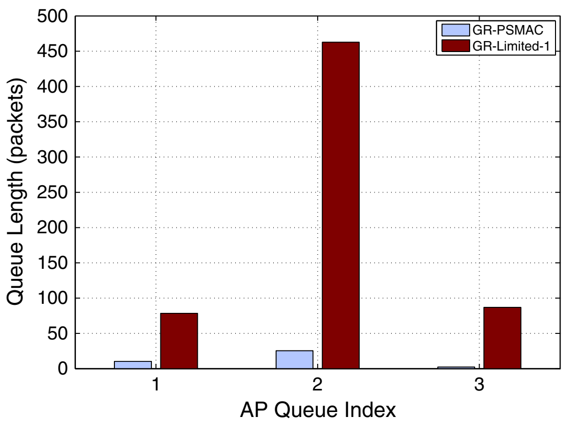


Figure 20. Virtual queue lengths at the AP with on–off bursty nonuniform traffic when the offered load is 60%: AP mode.

DOI: 10.1002/dac

the GR-Limited-1 virtual queue backlogs are 78.5, 462.8, and 86.9. Clearly, the gate service incorporated in PSMAC is much more effective in clearing the backlogs at the AP node. The high efficiency of GR-PSMAC brings about significant benefits to alleviate congestion at the wireless hotspot.

5. RELATED WORK

The performance analysis of IEEE 802.11 series protocols has been widely investigated [20–28]. This work is closely related to the research effort on improving the efficiency of MAC protocols. In particular, PSMAC that incorporates gated or exhaustive services was first introduced in [5, 6]. Limited-k service was used in the concatenation mechanism (CM) MAC [29], where k is equal to the *concatenated threshold*. In IEEE 802.11e HCF-controlled channel access, the hybrid coordinator (i.e., the AP) can assign 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, but the specific service discipline or algorithm for determining how many TXOPs to assign to a node is not specified. In addition, a centralized controller is required to poll the secondary nodes, which is different from the random access and fully distributed approach taken in this work.

GNU Radio/USRP is a popular platform for prototyping wireless systems. A main branch of prototyping work focus on the physical layer (PHY) due to the configurable signal processing ability offered by GNU Radio [30–36]. In [30], an implementation of a multiple-input and multiple-output (MIMO) PHY, is reported. In [31], the authors develop a new wireless carrier-sensing approach termed *LinkSense* to obtain fine-grain indications of channel activity. LinkSense utilizes a few OFDM subcarriers for conveying the link signature in each symbol, enabling sensing of active links at any time instant. The feasibility of LinkSense is then demonstrated on the GNU Radio/USRP platform with OFDM implementation. In [32], the authors present a software-defined IEEE 802.11b receiver and channel impulse response (CIR) measurement system. A USRP/GNU Radio testbed is designed to validate the CIR measurement system. The match filters are implemented in FPGA, whereas the Python code collects data from the USB, demodulates the packet, and records results.

In [33], the authors verify a multihop and multirate adaptation mechanism with a small-scale USRP/GNU Radio-based testbed. In [34], a cooperative communication testbed for both single-relay cooperation and multi-relay cooperation was reported on the basis of GNU Radio and USRP2. The significant performance enhancements for link reliability and end-to-end throughput of cooperative transmissions were observed. In addition, USRP and GNU Radio are also used to facilitate the prototype of RF front-end hardware. In [35], an RF front end with 50–2.5-GHz frequency range is designed and tested.

Because of the flexibility and full accessibility to the PHY and MAC, GNU Radio/USRP platform has been used to prototype MAC protocols that exploit PHY features [37–45]. The Hydra project [37] is a flexible wireless network testbed developed at UT Austin. The project exploits the Click modular router [46], GNU Radio, and C++ codes to prototype a cross-layer design of a rate adaptive MAC protocol. CoopMAC [38] is a programmable cooperative communication testbed developed at Polytechnic Institute of NYU. The testbed implements cooperative protocols in both PHY and MAC layers on the GNU Radio/USRP platform. The testbed experiment results verified significant benefits of cooperation in wireless networks. In [39], a load-adaptive MAC protocol is designed that switches between code-division multiple access and time-division multiple access on the basis of traffic loads. Its performance is evaluated with a multiple-input and multiple-output MANET testbed implemented with USRP-based SDR nodes. In [40], the authors study the performance of the IEEE 802.11 MAC under channel-oblivious and channel-aware jamming by theoretical analysis and extensive simulations via a GNU Radio/USRP testbed. An 802.11b ad hoc network with UDP traffic flows is established, where the sender, the receiver, and the jammer are all implemented with USRP and GNU Radio. In [41], Dhar *et al.* present a simple framework for joint design of MAC and PHY layers with the GNU Radio and Click platform. In [46], a software framework is presented, in which GNU Radio functions are encapsulated as a single Click element to provide PHY layer functionality. Because of the primarily goal of GNU Radio for supporting signal processing, the functions of MAC protocol are not fully supported. One of the main concerns in MAC prototyping is precise timing in carrier sensing. The latency of GNU Radio/USRP brings about a significant challenge for high-speed data rates. This issue is analyzed in [15], among others.

DOI: 10.1002/dac

The transmit and receive latencies were evaluated, and the impact on network performance was characterized under an IEEE 802.15.4 implementation.

As the interest in cognitive radio (CR) networks increases, GNU Radio/USRP has become popular in developing CR systems [47, 48]. In [47], an adaptive interference avoidance transform-domain communication system-based CR was demonstrated. In [48], the authors implement an adaptive spectrum-sensing scheme that exploits primary network traffic information with GNU Radio/USRP. USRP is shown to be amenable to implementing spectrum-sensing algorithms.

In this paper, we mainly focus on a proof-of-concept prototyping of PSMAC to evaluate the performance under the realistic wireless channels and networks. It is possible to extend this work to a more product-like system by adopting the recently developed platforms. For example, besides GNU Radio and USRP, FPGA-based software radio platform, such as Airblue [36], was also designed to support high-performance wireless protocols and cross-layer experiments. A very recent study promotes the wireless MAC processor concept [49], which provides engines for reconfigurable MAC protocol implementation. The processor defines the programming interface through actions, events, and conditions to support full-custom MAC protocol programming. The effectiveness of the wireless MAC processor is evaluated by AirForce54G chipset and proved that the processor can be implemented over an ultra-cheap commodity WLAN card.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented the design and implementation of PSMAC, a gated service-based MAC protocol, as well as a limited-1-based IEEE 802.11 DCF-like MAC for comparison purpose. The testbed was developed on the GNU Radio/USRP platform. We discussed related design issues on the prototyping process. In addition, we also presented extensive experimental results under various traffic models and traffic patterns. The experimental study validated the analysis and simulation studies presented in our prior work and demonstrated the advantages of PSMAC under a realistic wireless network setting.

This work can be extended in many ways. For example, it would be interesting to examine the impact of the transport layer protocols (i.e., transmission control protocol and UDP) on the impact of the proposed PSMAC protocol. It would also be interesting to investigate how to improve the testbed system with respect to QoS provisioning. Because there are many enhancements to the 802.11 MAC in the literature, it would also be interesting to extend the comparison studies and to investigate how to incorporate the scheduling scheme with more advanced PHY technologies.

ACKNOWLEDGEMENTS

Part of this work was conducted when Phillip Walsh was pursuing a master's degree at Auburn University. This work is supported in part by the US National Science Foundation (NSF) under Grants CNS-0953513, CNS-1247955, ECCS-0802113, and DUE-1044021, and through the NSF Wireless Internet Center for Advanced Technology at Auburn University. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

REFERENCES

1. Huang Y, Walsh P, Li Y, Mao S. A GNU Radio testbed for distributed polling service-based medium access control. *Proceedings of the IEEE MILCOM'11*, Baltimore, MD, 2011; 519–524.
2. IEEE. Wireless LAN media access control (MAC) and physical layer (PHY) specifications, 2007.
3. Xiao Y, Rosdahl J. Throughput and delay limits of IEEE 802.11. *IEEE Communications Letters* 2002; **6**(8):335–357.
4. Li J, Blake C, Coute DD, Lee H, Morris R. Capacity of ad hoc wireless networks. *Proceedings of the ACM MOBICOM'01*, Rome, Italy, 2001; 61–69.
5. Li Y, Mao S, Panwar S. PSMAC: polling service-based medium access control for wireless networks. *Proceedings of the WiOpt'07*, Limassol, Cyprus, 2007; 1–10.
6. Li Y, Mao S, Panwar S, Midkiff S. On the performance of distributed polling service-based medium access control. *IEEE Transactions on Wireless Communications* 2008; **7**(11):4635–4645.
7. Bruno R, Conti M, Gregori E. Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs. *IEEE Transactions on Mobile Computing* 2002; **1**(1):10–31.
8. Bluetooth Special Interest Group. Specification of the Bluetooth system, Nov. 2003, version 1.2. Available from: http://www.bluetooth.com.

9. IEEE. Part 11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), July 2003. ANSI/IEEE Std 802.11e, Draft 5.0.
10. Ozdemir M, Gu D, McDonald AB, Zhang J. Enhancing MAC performance with a reverse direction protocol for high-capacity wireless LANs. *Proceedings of the VTC'06 Fall*, Montreal, Canada, 2006; 1–5.
11. GNU Radio. Available from: http://www.gnu.org/software/gnuradio.
12. Ettus Research LLC. USRP Family Brochure. Available from: http://www.ettus.com/downloads/.
13. Takagi H, Kleinrock L. A tutorial on the analysis of polling systems. Computer Science Department, UCLA, Tech. Rep. No. 850005, Feb. 1985.
14. Reed JH. *Software Radio: A Modern Approach to Radio Engineering*. Prentice Hall: Indianapolis, Indiana, 2002.
15. Schmid T, Sekkat O, Srivastava M. An experimental study of network performance impact of increased latency in software defined radios. *Proceedings of the WinTECH'07*, Montreal, Canada, 2007; 59–66.
16. WARP: wireless open-access research platform, Rice University. Available from: http://warp.rice.edu/index.php.
17. IEEE. IEEE 1588TM-2002 standard for a precision clock synchronization protocol for networked measurement and control systems. Available from: http://www.nist.gov/el/isd/ieee/ieee1588.cfm.
18. Park K. The internet as a complex system. In *SFI Studies in the Sciences of Complexity*, Chap. 1, Park K, Willinger W (eds). Oxford University Press: Oxford, UK, 2005.
19. Jain R, Chiu D, Hawe W. A quantitative measure of fairness and discrimination for resource allocation in shared systems. DEC Research Report TR-301, 1984.
20. Chatzimisios P, Boucouvalas AC, Vitsas V. Performance analysis of the IEEE 802.11 MAC protocol for wireless LANs. *International Journal of Communication Systems* 2005; **18**(6):545–569.
21. Jayaparvathy R, Anand S, Dharmaraja S, Srikanth S. Performance analysis of IEEE 802.11 DCF with stochastic reward nets. *International Journal of Communication Systems* 2007; **20**(3):273–296.
22. Xu K, Wang Q, Hassanein H. Performance analysis of differentiated QoS in IEEE 802.11e WLANs. *International Journal of Communication Systems* 2005; **18**(7):619–637.
23. Wu C, Hou T, Leou M, Liaw Y, Chan M. Adaptive backoff scheme for ad hoc networks based on IEEE 802.11. *International Journal of Communication Systems* 2010; **23**(12):1632–1650.
24. Ke C, Wei C, Lin KW, Ding J. A smart exponential-threshold-linear backoff mechanism for IEEE 802.11 WLANs. *International Journal of Communication Systems* 2011; **24**(8):1033–1048.
25. Lin Y, Yeh J, Yang T, Ku C, Tsao S, Lai Y. Efficient dynamic frame aggregation in IEEE 802.11 s mesh networks. *International Journal of Communication Systems* 2009; **22**:1319–1338.
26. Gupta V, Gong M, Dharmaraja S, Williamson C. Analytical modeling of bidirectional multi-channel IEEE 802.11 MAC protocols. *International Journal of Communication Systems* 2011; **24**(5):647–665.
27. Ge X, Huang K, Wang CX, Hong X, Yang X. Capacity analysis of a multi-cell multi-antenna cooperative cellular network with co-channel interference. *IEEE Transactions on Wireless Communications* Oct 2011; **10**(10):3298–3309.
28. Chen M, Leung V, Mao S, Li M. Cross-layer and path priority scheduling based real-time video communications over wireless sensor networks. *Proc. IEEE VTC-Spring'12*. Singapore, 2008; 2873–2877.
29. Xiao Y. IEEE 802.11 performance enhancement via concatenation and piggyback mechanisms. *IEEE Transactions on Wireless Communications* 2005; **4**(5):2182–2192.
30. Li X, Hu W, Yousefi'zadeh H, Qureshi A. A case study of a MIMO SDR implementation. *Proceedings of the IEEE MILCOM'08*, San Diego, CA, 2008; 1–7.
31. Santhapuri N, Choudhury R, Nelakuditi S. Link sense: beyond wireless carrier sensing. *IEEE Transactions on Communications Letters* 2011; **15**(4):470–472.
32. Maas D, Firooz M, Zhang J, Patwari N, Kasera S. Channel sounding for the masses: low complexity GNU 802.11b channel impulse response estimation. *IEEE Transactions on Wireless Communications* 2011; **PP**(99):1–8.
33. Yu C, Shen T, Shin K, Lee J, Suh Y. Multihop transmission opportunity in wireless multihop networks. *Proceedings of the IEEE INFOCOM'10*, San Diego, CA, 2010; 1–9.
34. Zhang J, Jia J, Zhang Q, Lo E. Implementation and evaluation of cooperative communication schemes in software-defined radio testbed. *Proceedings of the IEEE INFOCOM'10*, San Diego, CA, 2010; 1307–1315.
35. Bruno M, Murdy M, Perreault P, Wyglinski A, McNeill J. Widely tunable RF transceiver front end for software-defined radio. *Proceedings of the IEEE MILCOM'09*, Boston, MA, 2009; 1–6.
36. Fuxjager P, Costantini A, Valerio D, Castiglione P, Zacheo G, Zemen T, Ricciato F. IEEE 802.11p transmission using GNURadio. *Proceedings of the WSR'10*, Karlsruhe, Germany, 2010; 83–86.
37. Mandke K, Soon-Hyeok C, Gibeom K, Grant R, Daniels R, Wonsoo K, Heath R, Nettles S. Early results on Hydra: a flexible MAC/PHY multihop testbed. *Proceedings of the IEEE VTC'07*, Dublin, Ireland, 2007; 1896–1900.
38. Korakis T, Knox M, Erkip E, Panwar S. Cooperative network implementation using open-source platforms. *IEEE Communications Magazine* 2009; **47**(2):134–141.
39. Hu W, Yousefizadeh H, Li X. Load adaptive MAC: a hybrid MAC protocol for MIMO SDR MANETs. *IEEE Transactions on Wireless Communications* 2011; **10**(11):3924–3933.
40. Bayraktaroglu E, King C, Liu X, Noubir G, Rajaraman R, Thapa B. On the performance of IEEE 802.11 under jamming. *Proceedings of the IEEE INFOCOM'08*, Phoenix, AZ, 2008; 1265–1273.
41. Dhar R, George G, Malani A, Steenkiste P. Supporting integrated MAC and PHY software development for the USRP SDR. *Proceedings of the 2006 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, 2006; 68–77.

DOI: 10.1002/dac

42. Chung J, Yun Y, Choi S. Prototype implementation of adaptive beamforming-MIMO OFDMA system based on IEEE 802.16e WMAN standard and its experimental results. *International Journal of Communication Systems* 2011; **24**(12):1627–1646.
43. Liu W, Jin H, Wang X, Guizani M. A novel IEEE 802.11-based MAC protocol supporting cooperative communications. *International Journal of Communication Systems* 2011; **24**(11):1480–1495.
44. Shi Q, Comaniciu C, Wang D, Tureli U. Cross-layer MAC design for location-aware wireless sensor networks. *International Journal of Communication Systems* 2011; **24**(7):872–888.
45. Kao HH, Wu PJ, Lee CN. Analysis and enhancement of multi-channel MAC protocol for ad hoc networks. *International Journal of Communication Systems* 2011; **24**(3):310–324.
46. Kohler E, Morris R, Chen B, Jannotti J, Kaashoek M. The Click modular router. *ACM Transactions on Computer Systems* 2000; **18**(3):263–297.
47. Zhou R, Han Q, Cooper R, Chakravarthy V, Wu Z. A software defined radio based adaptive interference avoidance TDCS cognitive radio. *Proceedings of the IEEE ICC'10*, Cape Town, South Africa, 2010; 1–5.
48. Shi X, de Francisco R. Adaptive spectrum sensing for cognitive radios: an experimental approach. *Proceedings of the IEEE WCNC'11*, Quintana-Roo, Mexico, 2011; 1408–1413.
49. Tinnirello I, Bianchi G, Gallo P, Garlisi D, Giuliano F, Gringoli F. Wireless MAC processors: programming MAC protocols on commodity hardware. *Proceedings of the IEEE INFOCOM'12*, Orlando, FL, 2012; 1269–1277.

AUTHORS' BIOGRAPHIES

Yingsong Huang received the BS degree in Automation in 2000 and the MS degrees in Control Theory and Control Engineering in 2003, both from Chongqing University, Chongqing, China. He was a senior R&D engineer and vice supervisor with Advantech Co. Ltd., Beijing and Taipei, China, from July 2003 to July 2007. Since 2007, he has been pursuing the PhD degree in the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA. His research interests include modeling, control, and optimization in computer networks, multimedia communications, and smart grid.

Phillip A. Walsh received the MS degree in Electrical and Computer Engineering from Auburn University, Auburn, AL, USA, in 2010 and the BS degree in Electrical and Computer Engineering from Clemson University, Clemson, SC, USA, in 2008. Since 2010, he has been a systems engineer in the Government Technologies Division, QUALCOMM Inc., San Diego, CA, USA. His research interests include design and prototyping of advanced wireless networking systems and 4G and beyond wireless networks.

Yihan Li received the BE and ME degrees from Tsinghua University, Beijing, China in 1993 and 1997, respectively, both in Electrical Engineering. She also received the MS degree in System Engineering in 2000 and the PhD degree in Electrical and Computer Engineering in 2004 from Polytechnic University (now Polytechnic Institute of New York University), Brooklyn, NY, USA.

Currently, she is a visiting assistant professor in the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA. She was a research scientist in the Department of Electrical and Computer Engineering at Polytechnic University from 2004 to 2006 and a postdoctoral fellow in the Department of Electrical and Computer Engineering at Auburn University from 2006 to 2009.

Dr. Li's research interests include scheduling in wired and wireless networks, wireless ad hoc networks, and high-speed packet switching. She is a member of Tau Beta Pi.

DOI: 10.1002/dac

Shiwen Mao received PhD in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA, (now Polytechnic Institute of New York University) in 2004. He was a research staff member with IBM China Research Lab from 1997 to 1998. He was a postdoctoral research associate/research scientist in the Bradley Department of Electrical and Computer Engineering at Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, from 2003 to 2006. Currently, he is the McWane associate professor in the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA.

His research interests include cross-layer optimization of wireless networks and multimedia communications, with current focus on cognitive radio networks, femtocell networks, 60-GHz networks, free-space optical networks, and smart grid. He is on the Editorial Board of IEEE Transactions on Wireless Communications, IEEE Communications Surveys and Tutorials, Elsevier Ad Hoc Networks Journal, Wiley International Journal of Communication Systems, and ICST Transactions on Mobile Communications and Applications. He is the director of E-Letter of IEEE Communications Society's Multimedia Communications Technical Committee.

Dr. Mao is a coauthor of *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004). He was awarded the McWane Endowed Professorship in the Samuel Ginn College of Engineering for the Department of Electrical and Computer Engineering, Auburn University in August 2012. He was named the 2012 Exemplary Editor of IEEE Communications Surveys and Tutorials. He received the US National Science Foundation *Faculty Early Career Development Award* (CAREER) in 2010. He is a co-recipient of *The 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems* and *The Best Paper Runner-up Award* at the Fifth International Conference on Heterogeneous Networking for Quality, Reliability, Security, and Robustness (QShine) in 2008. He also received *Auburn Alumni Council Research Awards for Excellence–Junior Award* in 2011 and two *Auburn Author Awards* in 2011. Dr. Mao holds one US patent.

DOI: 10.1002/dac