Chapter 2

A Distributed Polling Service-Based Medium Access Control Protocol: Prototyping and Experimental Validation

Yingsong Huang, Philip A. Walsh, Shiwen Mao and
Yihan Li

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/53772

1. Introduction

Mobile ad hoc networks and its variations such as wireless mesh networks and wireless LANs (WLAN) have become the ubiquitous connectivity solution in public as well as residential access networks, due to their cost efficiency, reliability and flexibility of deployment and operation. The rapidly proliferation of such wireless access networks are greatly advanced by the distributed multiple access control (MAC) protocols, which is based on random access techniques such as ALOHA, slotted ALOHA, carrier sense multiple access (CSMA) and CSMA with collision avoidance (CSMA/CA). The most important standards for these applications are the protocols in the IEEE 802.11 [1] series, which are widely used as the solution for the "last mile" access problem and become a *de facto* standard for various wireless access networks. The IEEE 802.11 protocol family defines physical layer (PHY) and medium access control (MAC) functions for wireless communication in the ISM bands of 2.4GHz and 5GHz. There are various amendments for the standard 802.11, such as 802.11a/b/g/e/n and the currently working draft of 802.11ac. Most of these amendments focus on the enhancement in PHY, which provides higher link capacity. For example, 802.11g adopts OFDM to leverage the data rate up to 54Mbps in 2.4GHz band. 802.11n [2] further improves the previous standards by adding multiple-input multiple-output (MIMO) antennas and the link capacity is boosted up to 600 Mbps. Although various PHY techniques are added to improve the link capacity, the MAC's they are based on almost remains same, which is based on CSMA/CA.

1.1. IEEE 802.11 MAC's

IEEE 802.11 employs Distributed Coordination Function (DCF) based on a CSMA/CA MAC protocol with binary exponential backoff algorithm. An optional access method by using Point Coordination Function (PCF) is also defined in IEEE 802.11 standard. PCF is designed for the operation in the Access Point (AP) mode, which allows the AP to poll nodes and transmit beacon frames to the nodes. PCF is only supported with AP thus limits its application in ad hoc networks mode. Due to its limitation, only very few products support

PCF in reality. We focus our discussion on DCF in the book chapter, which supports both infrastructure mode and ad hoc network mode, and is fully implemented in all commercial WLAN devices.

DCF follows the CSMA/CA techniques with random backoff algorithms. It also defines an optional handshake of Request-To-Send and Clear-To-Send (RTS/CTS) to reduce frame collisions introduced by the hidden node problem. In a single hop network, a source station with data to transmit senses the channel. After the channel is sensed idle for the period of DCF inter-frame space (DIFS), the station starts the random backoff by decreasing its contention window (CW). At the beginning of the backoff procedure, the source station shall generate a random backoff period for additional deferral before transmitting, unless the backoff timer already contain a nonzero value. 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}. After a successful transmission, CW will be reset to CW_{min}. If the channel is busy before CW reaches 0, CW will be frozen. The transmission will then be deferred and the station restarts to seek the idle status of DIFS interval of the channel. If the channel keeps idle when CW reaches 0, an RTS frame with the transmission duration will be transmitted. 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 also contains the transmission duration, which may contains the DATA frame durations and allow other stations set up their Network Allocation Vector (NAVs) for virtual carrier sensing. The neighbor stations then go to sleep mode and come back to sense the channel after their NAV expires. After receiving the CTS, the source station will transmit one DATA frame to the destination. All other stations will keep silent and wait for the NAV to expire. When the DATA frame is received, an ACK frame is issued by the destination to acknowledge the successfully received frame after the period of short inter-frame space (SIFS). If the source station does not receive the ACK frame within a specified ACK timeout interval, the backoff procedure will be performed at the source station to defer the transmissions. The lost or corrupted frame will be retransmitted at a later time. A typical DCF process and backoff algorithm are shown on Fig. 1 and Fig. 2.

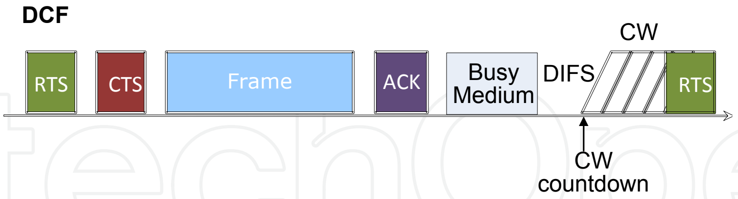

Figure 1. Illustration of DCF.

The original DCF scheme does not differentiate the traffic of different network services. It treats the high priority traffic and low priority traffic equally, which is not capable of satisfying the Quality of Service (QoS) request from the applications, such as voice over wireless LAN and streaming multimedia. To enhance the support of QoS, IEEE 802.11e [3] extends the MAC's by introducing Hybrid Coordination Function (HCF), which divides traffic into different classes and guarantees a QoS to each class. In the service differentiation, traffic in the same class competes the channel fairly like "best effort" transmission scheme, while traffic from different classes obtains different level of service. Although this service differentiation idea does not guarantee hard QoS, such as delay and loss rate, it provides a better response to the QoS requirements for different classes of services. The main techniques

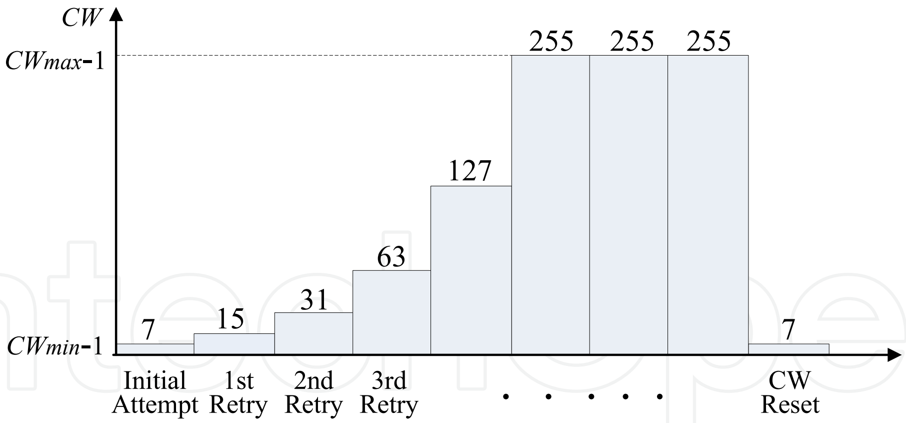

Figure 2. The *CW* backoff procedure.

used in the service differentiation of 802.11e include Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA) components. The former is for contention-based channel access by extending DCF, while the latter is for contention-free transfer by extending PCF.

EDCF classifies the medium access according to the priority of access classes (AC). Intuitively, it can be noticed that the length of DIFS in DCF controls the priority of transmitting RTS frame. In EDCF, an arbitration inter-frame space (AIFS) is defined to specify the minimum number of slots for which the stations in the AC should sense the channel to be free before attempting transmission. The station in higher priority AC is assigned shorter length of AIFS and the *CW* will countdown earlier than lower priority nodes, hence will have a higher success probability. Further, different random backoff window size settings CW_{min} and CW_{max} can be used for different ACs. High priority traffic has a higher transmission chance than the low priority traffic by assigning smaller CW_{min} and CW_{max}.

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 [4]. The polling mechanism has been also incorporated in the HCCA. The hybrid coordinator (HC) polls QoS 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 [5]. This technique gives an opportunity for a receiver 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. Since the NAV duration may be changed in CTS to support the "bidirectional" TXOP, more complex schemes are needed to handle hidden nodes problems.

IEEE 802.11 MAC, although widely used in WLANs, they are well-known for their considerable control overhead, which could consume as much as 40% of the nominal link capacity [6]. For example, the maximum achievable throughput for IEEE 802.11a is 24.7 Mbps, which is about 45.7% of the normianl link capacity. The problem gets even worse in the multi-hop scenario, due to carrier sensing and spatial reuse issues [7]. The compelling demands to support high definition videos, online games, and other real-time applications bring new challenges to the usage efficiency of the link capacity of existing WLANs and stress the new design of more effecient wireless MAC's.

1.2. Polling Service-Based MAC

We presented three polling service-based MAC protocols, termed PSMACs in our prior work [8, 9], which can amortize the control overhead of medium contention/resolution over multiple back-to-back frame transmissions, thus achieving high efficiency in medium access control. The gated service based PSMACs are analyzed and compared with p-Persistent CSMA, which closely approximates the standard IEEE 802.11 DCF [10]. Considerable gains on throughput, delay, energy consumption, and fairness performance are observed in the analysis and simulation studies [9].

There are two fundamental differences between the proposed PSMACs and the existing polling approaches in IEEE 802.11 series. First, the schemes adopted in Bluetooth and HCCA 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 [4] 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 802.11 MACs. More importantly, there is a need of both theoretical and experimental study to underpin the scheduling techniques to be adopted in both standards.

In this book chapter, we introduce PSMACs protocols and prototype the PSMACs in a real wireless networking environment [11]. Generally, 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 channels and verify our prior theoretical and simulation studies, but also identify new practical constraints and problems.

Two main contributions are made in this work. First, we implement the PSMACs on the GNU Radio [12] and Universal Software Radio Peripheral (USRP) [13] 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 in both infrastructure mode and ad hoc mode. The experimental results demonstrate the significant improvements that PSMAC can achieve on throughput, delay and fairness, and also validate the theoretic analysis and simulation studies in the prior work [9].

The remainder of this chapter is organized as follows. We first review PSMAC in Section 2. We then provide the system overview in Section 3 and discuss implementation details in Section 4. Our experimental results are presented in Section 5. Related work is discussed in Section 6 and Section 7 concludes the chapter.

2. Polling Service-Based MAC protocol

In this section, we briefly review PSMACs to provide the necessary background for the testbed. We refer interested readers to [9] for more technical details.

PSMAC is motivated by the insights from polling system theory [9]. 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 polls the stations for channel access. In the distributed scenario, the stations contend for channel access 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; (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% [14].

Based on the polling system theory, three polling service-based MAC protocols are introduced in [8, 9]. 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. The operation of PSMACs are shown in Fig. 3. In particular, PSMAC 1 senses a channel with CSMA/CA and uses RTS/CTS frames for contention resolution. All the frames to be transmitted are queued in a common transmission buffer. 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 that are not involved in the transmission be scheduled to sleep for energy conservation. PSMAC 3 extends PSMAC 2 by serving all non-empty virtual queues when a station wins the channel, which may achieve even higher efficiency. Specifically, PSMAC 3 introduces a new control frame *announcement frame* (AF). AF is broadcasted after a sender wins the channel by RTS, which contains the lengths of all the non-empty virtual queues at the sender, as well as the order in which the virtual queues will be served. Thus, each neighbor will realize how many frames it will receive, as well as the starting and ending time for its reception. The sender then starts data transmission by clearing the virtual queues one by one by gated service in the order that announced by AF. The current receiving node is active for the reception, while all other neighbors can be scheduled to sleep and to wake up when its corresponding virtual queue is to be served.

All of PSMACs introduced are based on gated policy in polling theory. Exhaustive policy may achieve higher efficiency, however, it is not practically implementable. This is due to the fact that the new frames may arrive at the buffer after the transmission start. The source node can not determine the exact transmission time before sending RTS. Thus, extra coordination control protocols are needed for the scheduling.

In [8, 9], 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 due to its similarity to the IEEE 802.11 DCF [10]. In addition, PSMACs 2 and 3 can achieve significant energy savings

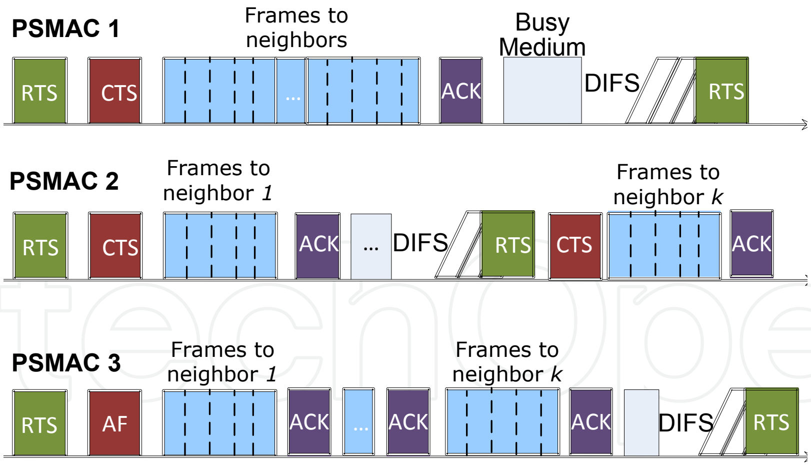

Figure 3. Timeline illustration of PSMACs operation.

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 [8, 9].

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 since it is most compatible to the DCF. 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/USRP platform [12, 13]. 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).

3. Testbed system overview

3.1. 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 WiFi networks. In particular, the implementations integrate CSMA/CA with binary exponential backoff, 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 non-empty virtual queues, the source station will selects a nonempty virtual queue in the *round-robin* fashion and start to sense the channel. After the channel is idle for DIFS interval, the *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 may return a CTS frame to confirm that it is ready for receiving data. The CTS frame contains the transmission duration, which may contains multiple frame durations and allows other stations set up their NAV for virtual carrier sensing. After receiving the CTS, the gated-service will be used for the selected virtual queue, i.e., 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 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 backoff 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 Fig. 3.

The backoff procedure used in the implementation is illustrated in Fig. 2, which follows the IEEE 802.11 DCF specification. In this chapter, we set $CW_{min} = 8$ and $CW_{max} = 256$ as in [1]. 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 the similar manner, except that when the source station wins the channel, 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 and used for performance comparison with the proposed GR-PSMAC.

3.2. Software and hardware platforms

We develop the PSMAC testbed on the Software Defined Radio (SDR) platform consisting of GNU Radio and USRP [12, 13]. SDR is a modern approach to wireless communications [15], which allows dynamic reconfiguration of waveforms by software. GNU Radio [12] 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. GNU Radio applications are usually written in Python scripts, which allows the quick reconfiguration of the protocols, while the compiled C++ codes are used for the signal processing components of physical layer for minimal processing time. USRP [13] is a generic SDR hardware 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 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. 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 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 latency hard to track. It is reported in [16] 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).

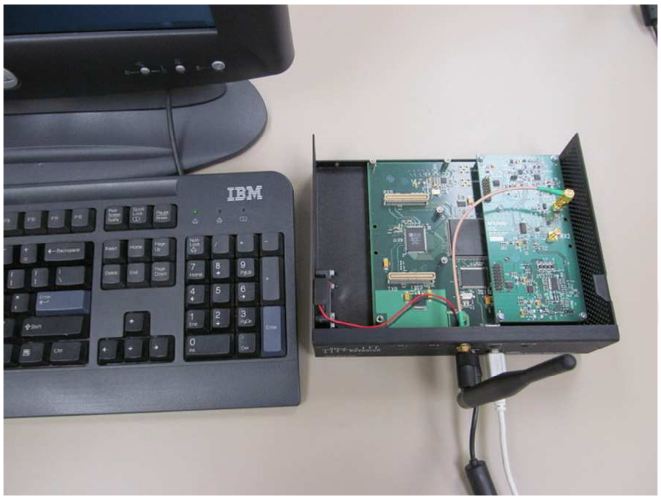

Figure 4. The testbed wireless station setup.

The large latency also negatively affects performance measurement during testbed experiments, especially under high transmission rates. 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. 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 with the given platform. It is worth noting that the Gigabit Ethernet interfance used in the later version of USRP, and the implementing the protocol functions in the FPGAs as in Rice University's Wireless Open-Access Radio Platform (WARP) platform [17], will help to allieviate the latency issue.

4. Testbed implementation description

We develop the MAC protocols on the GNU Radio/USRP platform [12, 13]. Each wireless station in the testbed consists of a USRP1 unit and a laptop (or desktop) computer, as illustrated in Fig. 4. We describe the implementation related issues in this section.

4.1. 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. Since 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 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 (API). This approach is illustrated in Fig. 5.

To implement the MAC layer functions, we design the MAC header as given in Fig. 6, which is similar to that of IEEE 802.11. The header fields are defined as follows.

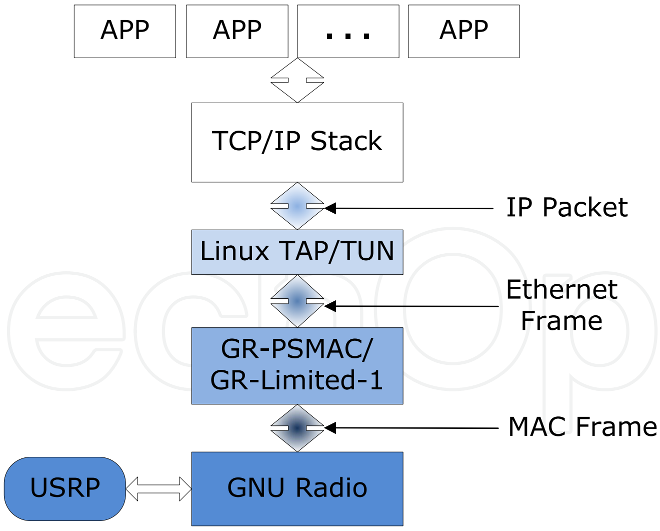

Figure 5. Protocol architecture of the GNU Radio testbed.

- Frame Control: four least significant bits define the frame type (RTS/CTS/DATA/ACK); other bits are reserved for future use.
- Destination Address: address of the destination node.
- Source Address: address of the source node.
- Next Hop Address: address of the next hop node; only valid for DATA frames and is used for the access point mode or multi-hop mode.
- Duration: 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: sequence number of transmitted DATA frame; in ACK: sequence number of the first received DATA frame.
- Count: in RTS/CTS/DATA: number of transmitted frames; in ACK: number of correctly received DATA frames.
- Option: reserved for future use.

The PSMAC header contains eight fields and is 16-bytes long in total. Although some of the fields are compatible with the header definition of IEEE 802.11; the header format is different from the standard Ethernet header. For example, standard 48-bit MAC addresses are used for the Linux TAP/TUN frame, but two-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 Fig. 5. Similarly, GR-PSMAC and GR-Limited-1 also map PSMAC headers back to the Ethernet header for received frames.

4.2. 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. The design of the two paths is shown in Fig. 7 and outlined below.

Frame Control (1 Byte)	16-Byte Header
Destination Address (2 Bytes)	
Source Address (2 Bytes)	
Next Hop Address (2 Bytes)	
Duration (4 Bytes)	
Sequence Number (2 Bytes)	
Count (2 Bytes)	
Option (1 Byte)	

Figure 6. GR-PSMAC/GR-Limited-1 Header Format.

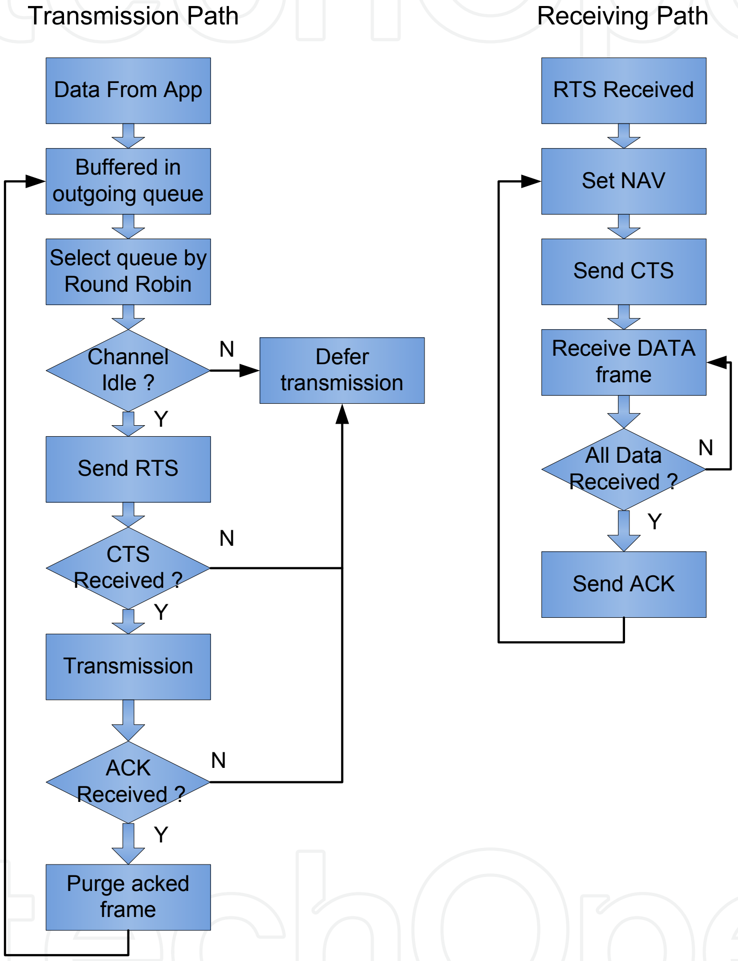

Figure 7. Illustration of the transmission and receiving operation.

4.2.1. 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 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 a predefined limit, GR-PSMAC will reset *CW* and serve the next nonempty outgoing virtual queue for the 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 backoff the transmission and try to serve the next nonempty outgoing virtual queue.

4.2.2. 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.

Since 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 need 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.

4.3. 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. 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 the 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 (see Fig. 6). 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 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 retransmissions, but with a slightly higher control overhead (i.e., longer ACK frames) and complexity.

4.4. 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 (PTP) daemon that implements the IEEE 1588 standard [18] 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 PTP daemon. When a DATA frame is enqueued at the source node, a time stamp will 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 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 stored time stamps.

During the testbed experiments, we use the above 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.

5. Experiments and results

5.1. 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 Fig. 4. 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).

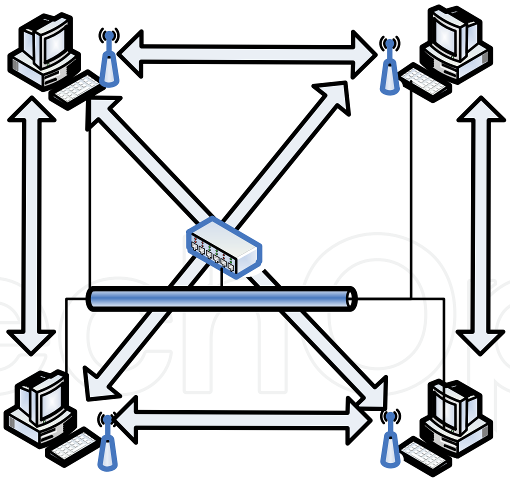

Figure 8. Network topology setups for the testbed experiments: ad hoc mode with a uniform traffic pattern.

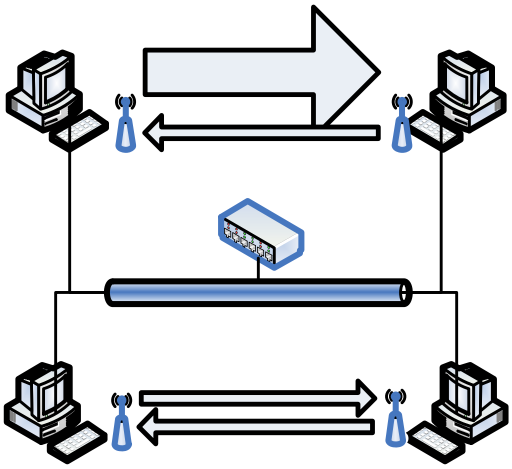

Figure 9. Network topology setups for the testbed experiments: ad hoc mode with a non-uniform traffic pattern.

We run the experiments using two network topologies as shown in Figs. 8–10: (i) a single-hop ad hoc network topology where each nodes communicates with every other node, as shown in Figs. 8 and 9, 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 Fig. 10. 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 below.

- Samples per symbol:	2
- Carrier frequency:	2.401 GHz
- Modulation scheme:	GMSK
- RTS Retry Limit:	5
- DATA frame size:	1,500 Bytes

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 TCP congestion control, thus focusing on the MAC performance. The following three traffic models are used in the experiments:

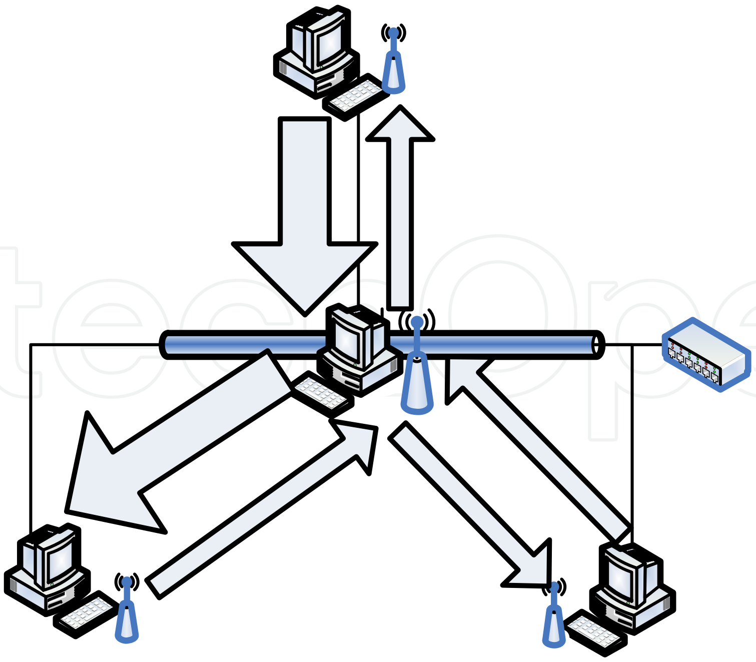

Figure 10. Network topology setups for the testbed experiments: AP mode with a non-uniform traffic pattern.

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 Markovian model with geometrically distributed on and off periods. The average on period is five, while the average off period is tuned to achieve different offered loads, for the results reported in this section.

iii) *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 exhibit long range dependence [19]. The Hurst parameter is chosen to be $H = 0.7$ for the results reported in this chapter.

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 a useful for modeling computer data traffic, which is shown to be self-similar [19]. The LRD model is much more bursty than 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 non-uniform traffic pattern, one source-destination pair has much higher load than others.

For each offered load, we run the testbed experiment for ten times. Each experiment lasts for 300 s when the i.i.d. Bernoulli and On-Off bursty traffic models are used, and 3,000 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 ten tests, while the 95% confidence intervals are plotted as error bars.

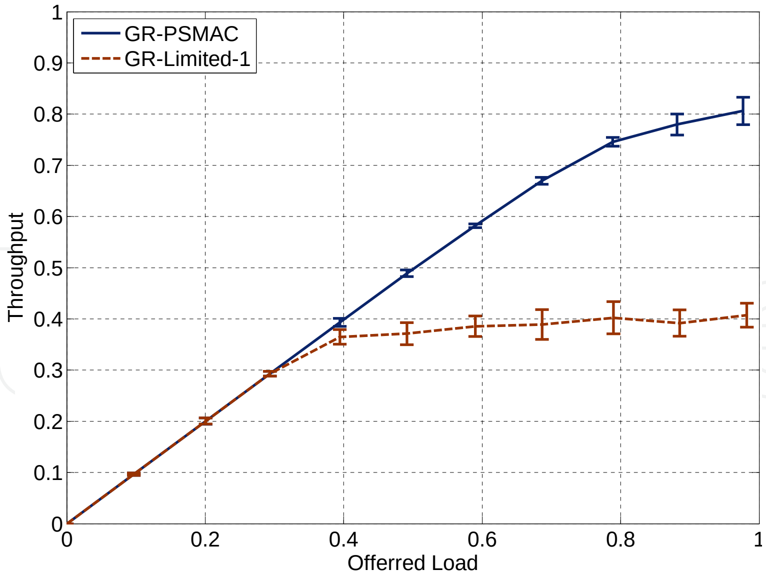

Figure 11. Normalized throughput under uniform i.i.d. Bernoulli traffic: ad hoc mode.

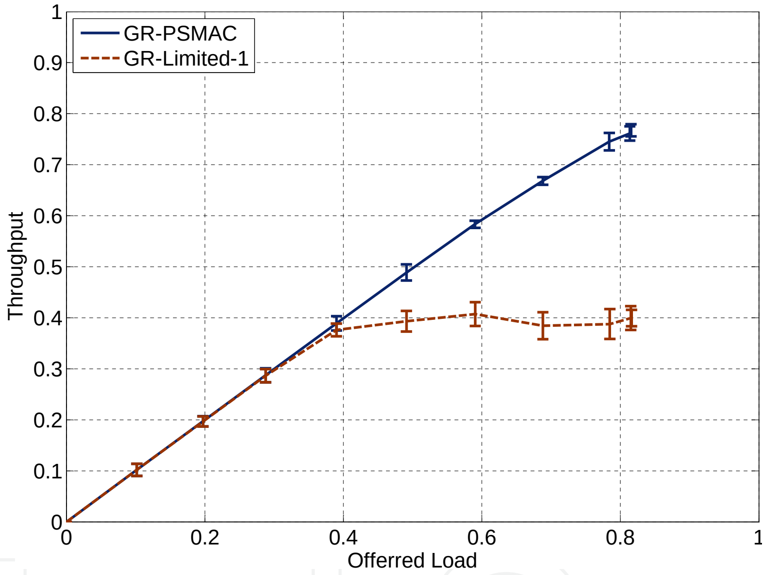

Figure 12. Normalized throughput under uniform On-Off busty traffic: ad hoc mode.

5.2. Experimental results

5.2.1. Performance under SRD Traffic: Ad Hoc Mode

5.2.1.1. Throughput and Delay

We first examine the network-wide throughput under the uniform Bernoulli and On-Off bursty traffic models and uniform traffic pattern. As shown in Fig. 8, 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. 11 for the uniform Bernoulli traffic case and in Fig. 12 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

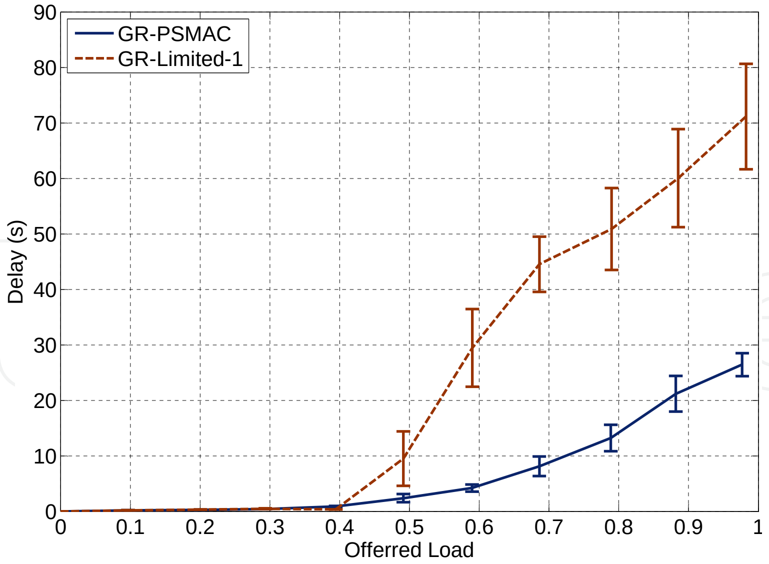

Figure 13. Average frame delay under uniform i.i.d. Bernoulli traffic: ad hoc mode.

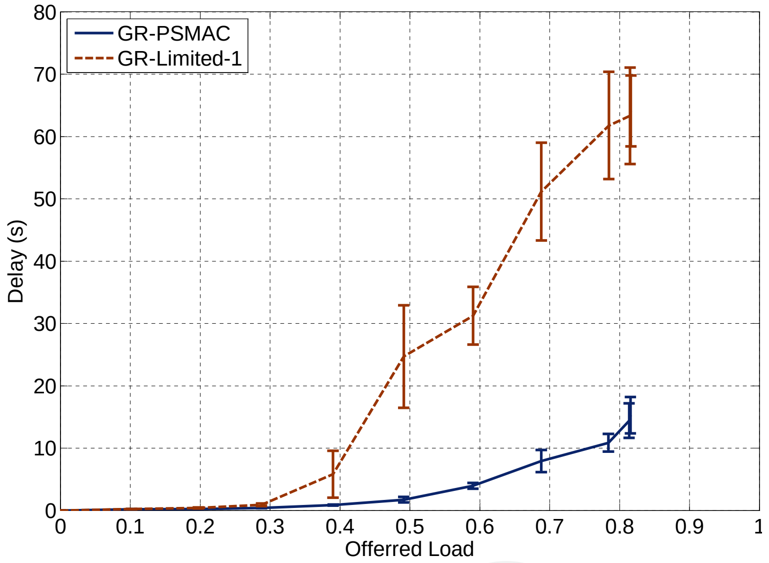

Figure 14. Average frame delay under uniform On-Off bursty traffic: ad hoc mode.

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 the frame delay under the same setup as in Fig. 8. The average delay for successfully received DATA frames are plotted in Figs. 13 and 14 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%.

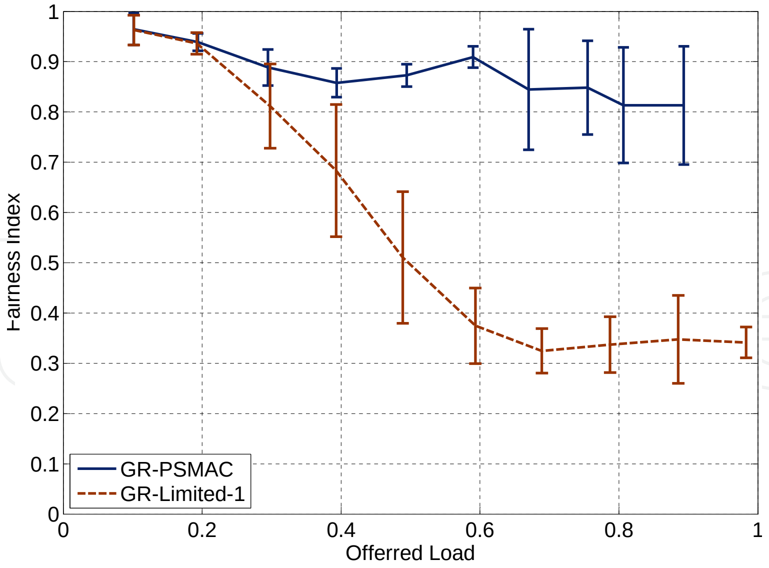

Figure 15. Fairness performance under non-uniform i.i.d. Bernoulli traffic: ad hoc mode.

5.2.1.2. Fairness

A common myth about gated or exhaustive polling service is that although the throughput/delay performance are superior, the fairness performance may not be good, since 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 non-uniform traffic pattern, as illustrated in Fig. 9. In this setting, the link from station 1 to station 2 takes 85% of the offered load, while the other 3 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 fairness index defined in [20]. For a system wit N stations, the fairness index is:

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

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 zero and one. In the fairest case, all the nodes have the same average delay, i.e., $D_1 = D_2 = \cdots = D_N$, and we have $f = 1$; in the worst case when one station's delay is dominant, i.e., $D_i >> D_j$, 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 Figs. 15 and 16. It can be observed that all the fairness index curves drops 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-PASMAC and GR-Limited-1 under the non-uniform Bernoulli traffic in Fig. 17. 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

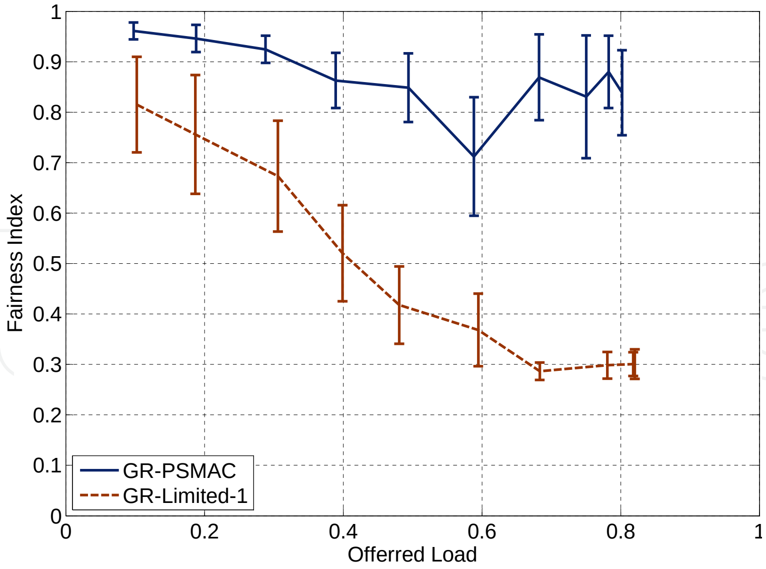

Figure 16. Fairness performance under non-uniform On-Off bursty traffic: ad hoc mode.

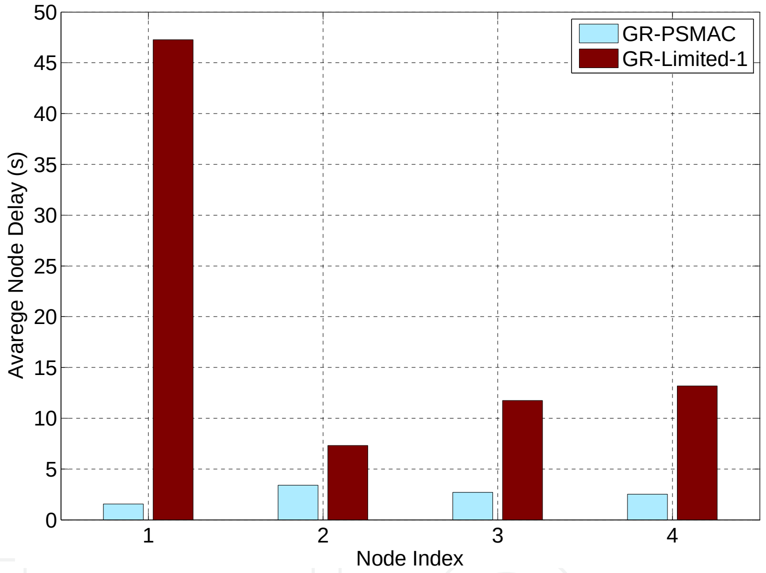

Figure 17. Average delay for each station under non-uniform i.i.d. Bernoulli traffic when the offered load is 60%: ad hoc mode.

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 s, 3.40 s]. Under GR-Limited-1, the heavily loaded station 1 has an average delay of 47.29 s, while the other three lightly loaded nodes have much lower average delays (all less than 14 s). GR-PASMAC achieves not only lower per station average delay than GR-Limited-1, but also more evenly distributed average delays among the stations than 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.

5.2.2. 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 computer data and VBR video traffic are *self-similar*, with Hurst parameters ranging from 0.5 to 1.0 [19]. For such traffic type, the class of SRD traffic models are 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 Figs. 18, 19, and 20. 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 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 Fig. 18, the throughput becomes saturated when the offered load exceeds 30%, which 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 Fig. 19, the SRD delay starts to diverge before the offered load reaches 30%, which is earlier than the SRD case in Fig. 14. 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 Fig. 20, 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 demonstrate 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 Fig. 18 is similar to that in Fig. 12, and the delay curve in Fig. 19 is also similar to that in Fig. 14, for the range of offered load examined. In Fig. 20, the GR-PSMAC fairness indices are all higher than 0.9, which is slightly better than that of the SRD case shown in Fig. 16.

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 are generally very difficult to management and control.

5.2.3. Performance in the AP Mode

Finally, we examine the performance of GR-PASMAC with an infrastructure-based wireless network topology. As shown in Fig. 10, 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, since the field of next hop address in all outgoing frame header 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,

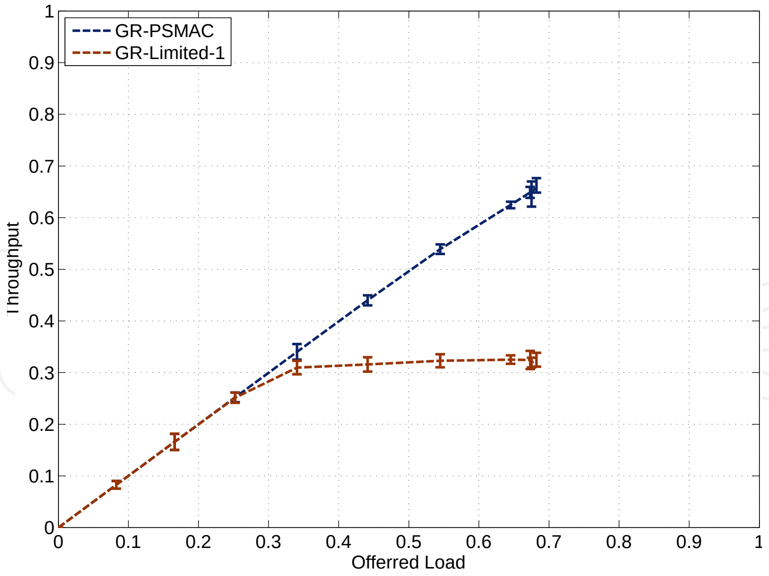

Figure 18. Normalized throughput under uniform LRD traffic: ad hoc mode.

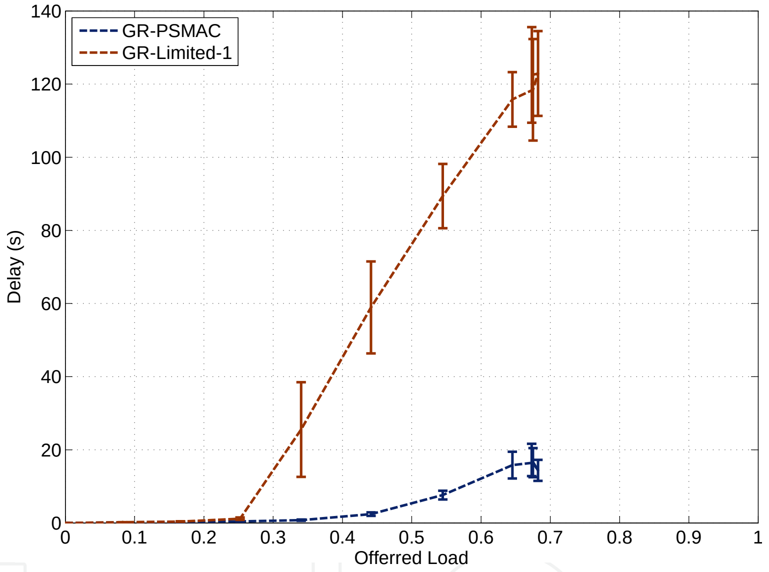

Figure 19. Average frame delay under uniform LRD traffic: ad hoc mode.

while the traffic flow from station 2 to station 3 and the traffic flow from station 3 to station 1 takes 10% of the offered load, respectively. Since the AP relays traffic, each frame will be transmitted twice. This scenario can also represents 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 Fig. 21, we plot the normalized throughput for the AP topology. We observed that GR-PSMAC still achieves considerably higher throughput under different offered loads, while the throughput of GR-Limited-1 becomes saturated when the offered load exceeds 40%. The average frame delays are plotted in Fig. 22. It can be observed that the GR-PSMAC delays are consistent with the previous experiments, while the GR-Limited-1 delay shoots up when the offered load exceeds 30%. In Fig. 23, the GR-PSMAC fairness indices are constantly above 0.8, while the GR-Limited-1 fairness index curve drops when the offered load exceeds

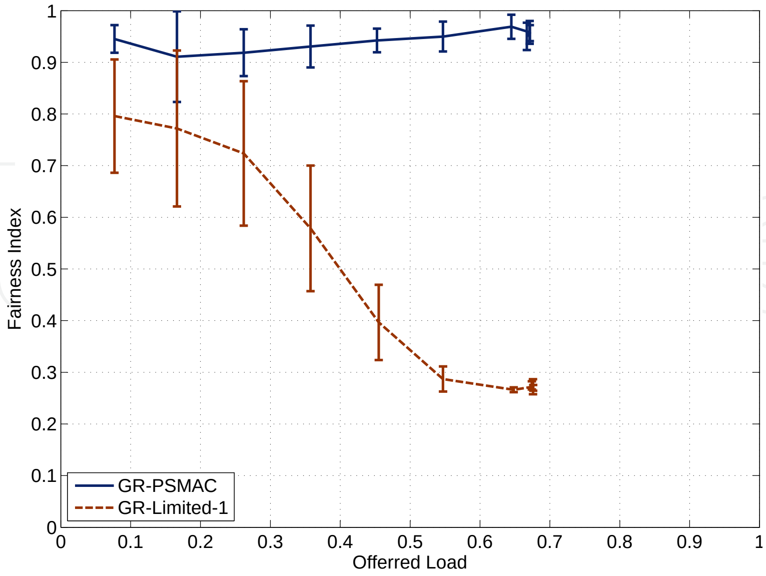

Figure 20. Fairness performance under non-uniform LRD traffic: ad hoc mode.

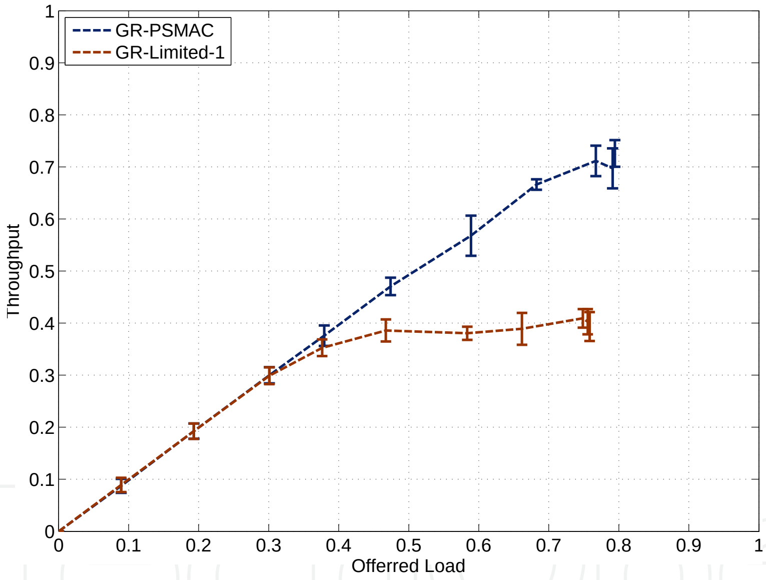

Figure 21. Normalized throughput with On-Off bursty non-uniform traffic: AP mode.

20%, 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 Fig. 24. 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, while 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.

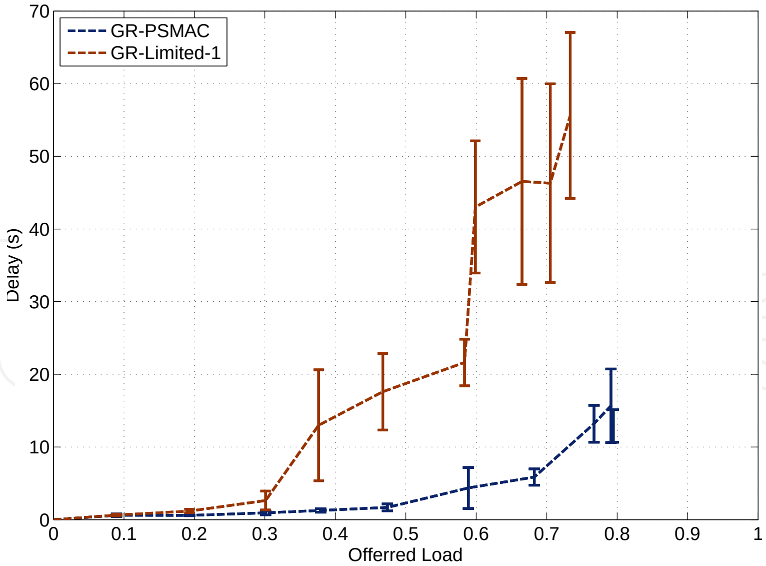

Figure 22. Average frame delay with On-Off bursty non-uniform traffic: AP mode.

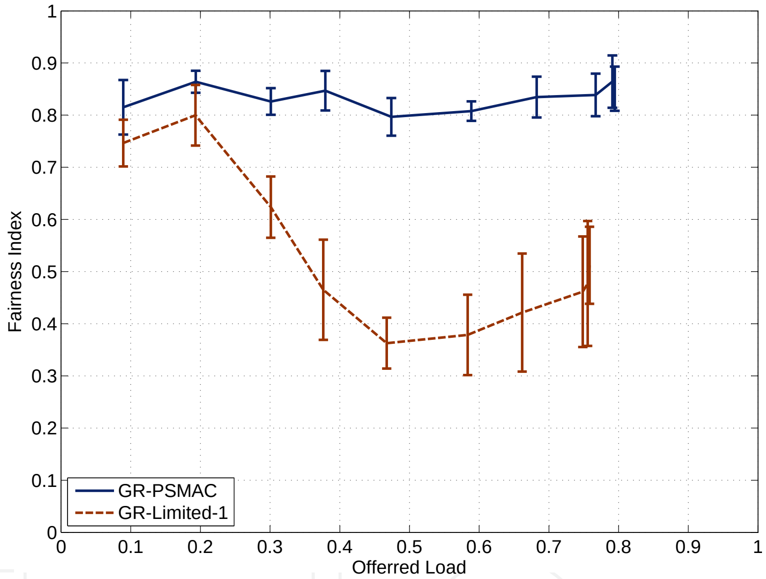

Figure 23. Fairness performance with On-Off bursty non-uniform traffic: AP mode.

6. Related work

This work is closely related to the research effort on improving the efficiency of MAC protocols. There are many techniques that exploit new capabilities of the wireless system to achieve this goal, such as adopting downlink MIMO (DL MIMO) [21], exploiting new spectrum allocation [22–24], exploiting spectrum opportunities in underutilized licensed band [25, 26], and exploit location information to schedule concurrent transmissions [27–29]. Exploiting efficient scheduling of the transmissions is a useful approach that is complimentary to the above techniques.

In particular, PSMAC that incorporates gated or exhaustive services was first introduced in [8, 9]. Limited-*k* service was used in the CM MAC [30], where *k* is equal to the *concatenated threshold*. In IEEE 802.11e HCF (hybrid coordination function) controlled channel access

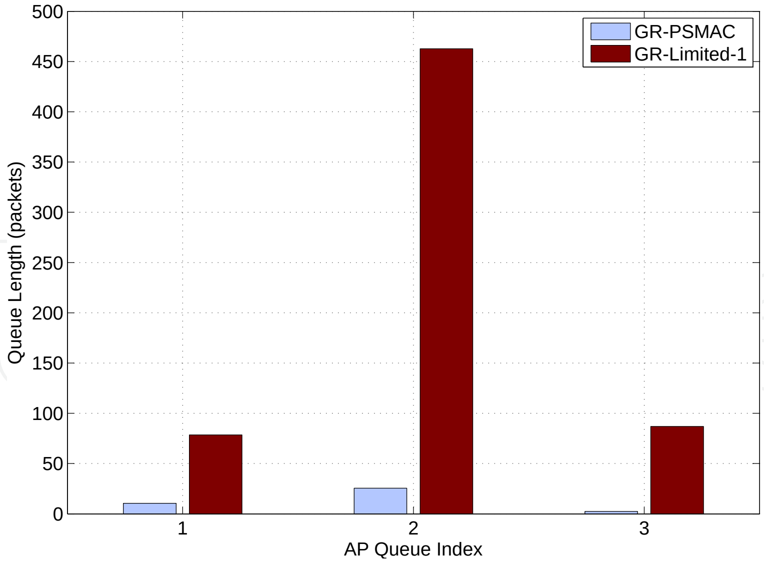

Figure 24. Virtual Queue lengths at the AP with On-Off bursty non-uniform traffic when the offered load is 60%: AP mode.

(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, but the specific service discipline or algorithm for determining how many TXOPs to assign to a node, are 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. In [31], the authors discuss the general implementation issues of the prototyping of wireless systems with USRP and GNU Radio. A main branch of prototyping works focus on the PHY, due to the configurable signal processing ability offered by GNU Radio [32–40]. In [32], an implementation of a MIMO PHY is reported. In [33], the authors developed 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 an OFDM implementation. In [34], the authors presented a software-defined IEEE 802.11b receiver and channel impulse response (CIR) measurement system. A USRP and GNU Radio testbed is designed to validate the CIR measurement system. The match filters are implemented in FPGA, while the Python code collects data from the USB, demodulates the packet, and records results.

In [35], the authors verify a multihop, multirate adaptation mechanism with a small scale USRP/GNU Radio-based testbed, while in [36], an implementation of Angle-of-arrival-assisted Relative Interferometric (ARI) RADAR transceiver is proposed based on GNU Radio and USRP. In [37], USRP, GNU Radio and OSSIE [41] are integrated to prove the concept of Government Reference Architecture (GRA), which is a standard for establishing a modular open system architecture for a family of Above 2GHz (A2G) Tactical Military Satellite Communications (MILSATCOM) terminals operating over several radio frequency bands. In [38], a cooperative communication testbed for both single-relay cooperation and multi-relay cooperation was reported based on GNU Radio and USRP2.

The significant performance enhancement 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 [39], an RF front-end with 50 MHz – 2.5 GHz frequency range is designed and tested.

Due to 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 [42–46]. The Hydra project [42] is a flexible wireless network testbed developed at UT Austin. The project exploits the Click modular router [47], GNU Radio, and C++ codes to prototype a cross-layer design of a rate adaptive MAC protocol. CoopMAC [43] 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 [44], a load-adaptive MAC protocol is designed that switches between CDMA and TDMA based on traffic loads. Its performance is evaluated with a MIMO MANET testbed implemented with USRP based SDR nodes. In [45], the authors studied 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 [46], Dhar et al. presented a simple framework for joint design of MAC and PHY layers with the GNU Radio and Click platform. In [47], a software framework is presented, in which GNU Radio functions are encapsulated as a single Click element to provide PHY layer functionality. Due to the primarily goal of GNU Radio for supporting signal processing, the functions of MAC protocol are not fully supported. One of the main concern 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 is analyzed in [16], among others. 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 [48–51]. In [48], an adaptive interference avoidance Transform-domain Communication System (TDCS) based cognitive radio was demonstrated. In [49], GNU Radio/USRP is used to set up a testbed for service discovery and device identification in CR networks, which may achieve better spectral efficiency and also enhance wireless security. In [50], a cognitive receiver is designed, which includes a universal classifier, synchronizer, and demodulator. The performance is verified with the GNU Radios/USRP platform together with MATLAB-enabled Anritsu MS2781A Signature Signal Analyzer. In [51], the authors implemented 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 book chapter, we focus on the fast prototyping of the PSMACs to evaluate the performance under the realistic wireless channels and networks. It could be possible to extend this work to more efficient industrial product by the recently developed frameworks. For example, besides GNU Radio and USPR, FPGA-based software radio platform, such as Airblue [52] was also designed to support high performance wireless protocols and cross-layer experiments. A very recent study promotes the wireless MAC processor concept [53], 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 are evaluated by AirForce54G chipset and proved that the processor can be implemented over an ultra-cheap commodity WLAN card.

7. Conclusion and future directions

In this book chapter, 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. In future, it would be interesting to extend this work by integrating the PSMACs into IEEE 802.11 framework, such as *mac802.11* framework [54] and *Madwifi* [55] in Linux kernel, for a realistic addition to the industrial wireless MAC standard.

Acknowledgements

This work is supported in part by the US National Science Foundation (NSF) under Grants 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 foundation.

Author details

Yingsong Huang[1], Philip A. Walsh[2],
Shiwen Mao[1] and Yihan Li[1]

1 Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA
2 QUALCOMM Inc., San Diego, CA, USA

References

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

[2] IEEE. Part 11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 5: Enhancements for Higher Throughput, Oct. 2009. IEEE Std 802.11n.

[3] 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.

[4] Bluetooth Special Interest Group, “Specification of the Bluetooth system,” Nov. 2003, version 1.2. [online]. Available: http://www.bluetooth.com.

[5] M. Ozdemir, D. Gu, A. B. McDonald, and J. Zhang. Enhancing MAC performance with a reverse direction protocol for high-capacity wireless LANs. In *Proc. VTC’06 Fall*, pages 1–5, Montreal, Canada, Sept. 2006.

[6] Y. Xiao and J. Rosdahl. Throughput and delay limits of IEEE 802.11. *IEEE Commun. Lett.*, 6(8):335–357, Aug. 2002.

[7] J. Li, C. Blake, D. D. Coute, H. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proc. ACM MOBICOM’01*, pages 61–69, Rome, Italy, July 2001.

[8] Y. Li, S. Mao, and S. Panwar. PSMAC: Polling service-based medium access control for wireless networks. In *Proc. WiOpt’07*, pages 1–10, Limassol, Cyprus, Apr. 2007.

[9] Y. Li, S. Mao, S. Panwar, and S. Midkiff. On the performance of distributed polling service-based medium access control. *IEEE Trans. Wireless Commun.*, 7(11):4635–4645, Nov. 2008.

[10] R. Bruno, M. Conti, and E. Gregori. Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs. *IEEE Trans. Mobile Computing*, 1(1):10–31, Jan./Mar. 2002.

[11] Y. Huang, P. Walsh, Y. Li, and S. Mao. A GNU radio testbed for distributed polling service-based medium access control. In *Proc. IEEE MILCOM 2011*, pages 519–524, Baltimore, MD, Nov. 2011.

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

[13] Ettus Research LLC, “USRP family brochure.” [online] Available: http://www.ettus.com/downloads/.

[14] H. Takagi and L. Kleinrock. A tutorial on the analysis of polling systems. Computer Science Department, UCLA, Tech. Rep. No. 850005, Feb. 1985.

[15] Jeffrey H. Reed. *Software Radio: A Modern Approach to Radio Engineering*. Prentice Hall, May 2002.

[16] 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*, pages 59–66, Montreal, Canada, Sept. 2007.

[17] WARP: Wireless Open-Access Research Platform, Rice University, [online] Available: http://warp.rice.edu/index.php.

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

[19] K. Park. The Internet as a complex system. In K. Park and W. Willinger, editors, *SFI Studies in the Sciences of Complexity*, chapter 1. Oxford University Press, Oxford, UK, 2005.

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

[21] M. X. Gong, E. Perahia, R. J. Stacey, Want R, and Mao S. A CSMA/CA MAC protocol for multi-user MIMO wireless LANs. In *Proc. IEEE GLOBECOM 2010*, pages 1–6, Miami, FL, Dec. 2010.

[22] M. X. Gong, R. J. Stacey, D. Akhmetov, and S. Mao. Performance analysis of a directional CSMA/CA protocol for mmWave wireless PANs. In *Proc. IEEE WCNC 2010*, pages 1–6, Sydney, Australia, Apr. 2010.

[23] M. X. Gong, D. Akhmetov, R. Want, and S. Mao. Directional CSMA/CA protocol with spatial reuse for mmWave wireless networks. In *Proc. IEEE GLOBECOM 2010*, pages 1–5, Miami, FL, Dec. 2010.

[24] I. K. Son, S. Mao, M. X. Gong, and Y. Li. On frame-based scheduling for directional mmWave WPANs. In *Proc. IEEE INFOCOM 2012*, pages 2149–2157, Orlando, FL, Mar. 2012.

[25] D. Hu and S. Mao. Design and analysis of a sensing error-aware MAC protocol for cognitive radio networks. In *Proc. IEEE GLOBECOM 2009*, pages 5514–5519, Honolulu, HI, Nov./Dec. 2009.

[26] D. Hu and S. Mao. A sensing error aware MAC protocol for cognitive radio networks. *ICST Transactions on Mobile Communications and Applications*, 1(1), 2012. to appear.

[27] S.-M. Hur, S. Mao, Y. T. Hou, K. Nam, and J. H. Reed. On exploiting location information for concurrent transmission in multi-hop wireless networks. *IEEE Transactions on Vehicular Technology*, 58(1):314–323, Jan. 2009.

[28] I. K. Son, S. Mao, and S.-M. Hur. Medium access control for opportunistic concurrent transmissions under shadowing channels. *MDPI Sensors Journal*, 9(6):4824–4844, June 2009.

[29] S.-M. Hur, S. Mao, K. Nam, and J. H. Reed. On concurrent transmissions in multi-hop wireless networks with shadowing channels. In *Proc. IEEE ICC 2008*, pages 2662–2666, Beijing, P.R. China, May 2008.

[30] Y. Xiao. IEEE 802.11 performance enhancement via concatenation and piggyback mechanisms. *IEEE Trans. Wireless Commun.*, 4(5):2182–2192, Sept. 2005.

[31] D.C. Tucker and C.A. Tagliarini. Prototyping with GNU radio and the USRP - where to begin. In *Proc. IEEE SOUTHEASTCON '09*, pages 50–54, Mar. 2009.

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

[33] N. Santhapuri, R.R. Choudhury, and S. Nelakuditi. Link sense: beyond wireless carrier sensing. *IEEE Trans. Commun. Letters*, 15(4):470–472, Apr. 2011.

[34] D. Maas, M. Firooz, J. Zhang, N. Patwari, and S. Kasera. Channel sounding for the masses: low complexity GNU 802.11b channel impulse response estimation. *IEEE Trans. Wireless Commun.*, PP(99):1–8, Nov. 2011.

[35] C. Yu, T. Shen, K.G. Shin, J. Lee, and Y. Suh. Multihop transmission opportunity in wireless multihop networks. In *Proc. IEEE INFOCOM'10*, pages 1–9, San Diego, CA, Mar. 2010.

[36] J. Friedman, A. Davitian, D. Torres, D. Cabric, and M. Srivastava. Angle-of-arrival-assisted relative interferometric localization using software defined radios. In *Proc. IEEE MILCOM'09*, pages 1–8, Boston, MA, Oct. 2009.

[37] T. Leising, C. Dietrich, M. Gavitt, R. Kim, H. Satake, S. Hoque, and T. Rittenbach. An excursion to define the boundaries of the Government Reference Architecture. In *Proc. IEEE MILCOM'08*, pages 1–6, San Diego, CA, Nov. 2008.

[38] 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*, pages 1307–1315, San Diego, CA, Mar. 2010.

[39] M. Bruno, M. Murdy, P. Perreault, A.M. Wyglinski, and J.A. McNeill. Widely tunable RF transceiver front end for software-defined radio. In *Proc. IEEE MILCOM'09*, pages 1–6, Boston, MA, Oct. 2009.

[40] P. Fuxjager, D. Valerio A. Costantini, P. Castiglione, G. Zacheo, T. Zemen, and F. Ricciato. IEEE 802.11p transmission using GNURadio. In *Proc. WSR'10*, pages 83–86, Karlsruhe, Germany, Mar. 2010.

[41] J. Snyder, B. McNair, S. Edwards, and C. Dietrich. OSSIE: an open source software defined radio platform for education and research. In *Proc. FECS'11*, pages 18–22, Las Vegas, NV, July 2011.

[42] K. Mandke, C. Soon-Hyeok, K. Gibeom, R. Grant, R. Daniels, K. Wonsoo, R. Heath, and S. Nettles. Early results on Hydra: A flexible MAC/PHY multihop testbed. In *Proc. IEEE VTC'07*, pages 1896–1900, Dublin, Ireland, Apr. 2007.

[43] T. Korakis, M. Knox, E. Erkip, and S. Panwar. Cooperative network implementation using open-source platforms. *IEEE Commun. Mag.*, 47(2):134–141, Feb. 2009.

[44] W. Hu, H. Yousefizadeh, and X. Li. Load adaptive MAC: A hybrid MAC protocol for MIMO SDR MANETs. *IEEE Trans. Wireless Commun.*, 10(11):3924–3933, Nov. 2011.

[45] E. Bayraktaroglu, C. King, X. Liu, G. Noubir, R. Rajaraman, and B. Thapa. On the performance of IEEE 802.11 under jamming. In *Proc. IEEE INFOCOM'08*, pages 1265–1273, Phoenix, AZ, Apr. 2008.

[46] R. Dhar, G. George, A. Malani, and P. Steenkiste. Supporting integrated MAC and PHY software development for the USRP SDR. In *Proc. 2006 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, pages 68–77, Sept. 2006.

[47] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M.F. Kaashoek. The Click modular router. *ACM Trans. Comput. Syst.*, 18(3):263–297, Aug. 2000.

[48] 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*, pages 1–5, Cape Town, South Africa, May 2010.

[49] R. Miller, W. Xu, P. Kamat, and W. Trappe. Service discovery and device identification in cognitive radio networks. In *Proc. IEEE SECON'07*, pages 40–47, San Diego, CA, Jun. 2007.

[50] Q. Chen, Y. Wang, and C.W. Bostian. Universal classifier synchronizer demodulator. In *Proc. IEEE IPCCC'08*, pages 366–371, Austin, TX, Dec. 2008.

[51] X. Shi and R. de Francisco. Adaptive spectrum sensing for cognitive radios: an experimental approach. In *Proc. IEEE WCNC'11*, pages 1408–1413, Quintana-Roo, Mexico, Mar. 2011.

[52] M. C. Ng, K. E. Fleming, M. Vutukuru, S. Gross, A. Arvind, and H. Balakrishnan. Airblue: A system for cross-layer wireless protocol development. In *Proc. ANCS'10*, pages 1–11, La Jolla, CA, Oct. 2010.

[53] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli. Wireless MAC processors: programming MAC protocols on commodity hardare. In *Proc. IEEE INFOCOM'12*, pages 1269–1277, Orlando, FL, Mar. 2012.

[54] Linux Wireless, [online]. Available: http://linuxwireless.org/.

[55] Madwifi Project, [online]. Available: http://madwifi-project.org/.