

# Experimental Evaluation of Reverse Direction Transmissions in WLAN Using the WARP Platform

Raul Palacios<sup>a</sup>, Francesco Franch<sup>a</sup>, Francisco Vazquez-Gallego<sup>b</sup>, Jesus Alonso-Zarate<sup>b</sup> and Fabrizio Granelli<sup>a</sup>

<sup>a</sup>DISI, University of Trento, Italy, {palaciostrujillo@disi, f.franch@studenti, granelli@disi}.unitn.it

<sup>b</sup>Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Spain, {fvazquez, jesus.alonso}@cttc.es

**Abstract**—This paper describes an experimental implementation of a variation of the Reverse Direction (RD) Medium Access Control (MAC) Protocol (RDP) defined in the IEEE 802.11n using the Wireless Open-Access Research Platform (WARP). The proposed approach, named Bidirectional MAC (BidMAC), allows the receiver of a valid data sequence to perform an RD transmission to the transmitter without contending for the channel. Whereas in RDP the RD transmission must be initiated by the transmitter, in BidMAC it can be dynamically initiated by the receiver according to its traffic requirements. Previous results based on mathematical analyses and computer-based simulations have shown that BidMAC can better balance downlink and uplink transmission opportunities in a Wireless Local Area Network (WLAN) where the Access Point (AP) handles bidirectional data flows for some of its wireless stations (STAs). This paper aims at going one step further and demonstrating that such superior performance can be attained in real environments. Towards this end, an implementation of BidMAC has been carried out in a reference design of WARP compatible with the IEEE 802.11a/g and tested in a proof-of-concept network formed by an AP and two STAs. Experimental results confirm the superior performance of BidMAC when compared to the legacy Distributed Coordination Function (DCF) of the IEEE 802.11 versus the traffic load, packet length, and data rate, yielding gains of up to 60%. \*

## I. INTRODUCTION

Recently, the use of Reverse Direction (RD) transmissions has been proposed in the IEEE 802.11 Standard to improve the throughput and energy efficiency of Wireless Local Area Networks (WLAN) based on the IEEE 802.11 [1]. More specifically, the Reverse Direction Protocol (RDP) has been defined in the IEEE 802.11n as a Medium Access Control (MAC) layer enhancement of the legacy Distributed Coordination Function (DCF) to increase channel utilization. The RDP breaks with the basic operation of the DCF where a wireless station (STA) gains a Transmission Opportunity (TXOP) by competing to get access to the wireless channel in order to transmit data to one arbitrary destination. In RDP, the holder of a TXOP, once it has seized the channel, can allocate the unused TXOP duration to one or more receivers in order to allow data transmissions in the reverse link. For scenarios with bidirectional traffic, this approach is very convenient as it reduces contention in the wireless channel.

The concept of reverse direction (or bidirectional) transmission in WLAN was first introduced by [2], prior to the

standardization of the RDP. Since then, several works have proposed similar approaches with different purposes. Existing RD-based protocols can be classified into two categories: (i) proactive, i.e. RD exchange sequence initiated by the transmitter, or (ii) reactive, i.e. RD exchange sequence initiated by the receiver. Proactive RD protocols [3], [4] allow the transmitter to grant the receiver the remaining time of its TXOP for reverse data transfer, in a way similar to RDP. On the other hand, reactive RD protocols [2], [5]–[9] allow the receiver to reserve the wireless channel for a backward transmission by extending the transmitter’s TXOP time, without needing to compete for the channel. This sort of RD protocols can achieve higher performance in some scenarios because they are more adaptive to the actual needs of a network.

In particular, the work in [7] and our previous works [8], [9] investigate the feasibility of reactive RD exchange operation in infrastructure WLAN, wherein an Access Point (AP) is connected to a cable network infrastructure and provides wireless Internet access for a number of STAs in its coverage area. Results show that reactive RD approaches can effectively address the unbalanced operation of DCF between uplink and downlink traffic when traffic flows are highly bidirectional. Indeed, DCF provides equal channel access opportunities for all STAs, including the AP. Therefore, the AP only receives an equal share of the wireless channel to deliver downlink traffic to all the STAs, while it has data to transmit to all of them. Note that we consider the case when all STAs route all their traffic through the AP. Therefore, by allowing the AP to dynamically initiate RD exchange sequences when receiving data from the STAs, uplink and downlink transmission opportunities can be better balanced, hence improving the overall WLAN performance.

The results presented in the works discussed above are based on theoretical analyses and computer-based simulations. Whereas theoretical models typically adopt simplified assumptions for mathematical tractability, computer-based simulations usually lack PHY-layer modeling accuracy, thus leading to inaccurate results and conclusions. In contrast, real-world implementation can help reveal unexpected challenges to the development of new MAC protocols and also provide new insights in the operation of communication protocols. This is the main motivation for the work presented in this paper where we describe an experimental implementation of a reactive RD MAC protocol, named Bidirectional MAC (BidMAC) [8], [9], for infrastructure WLAN in real hardware. This implemen-

\*This work has been funded by the GREENET research project (PITN-GA-2010-264759) and by the Generalitat de Catalunya through 2014-SGR-1551.

tation has enabled us to test and evaluate the operation of BidMAC in real environments, and to demonstrate its superior performance compared to the legacy DCF.

There are various available wireless platforms for prototyping at the MAC layer [10]. We have selected the Wireless Open-Access Research Platform (WARP) [11] because it offers an available open-source reference design that can interoperate with commercial IEEE 802.11a/g devices, acting as either AP or STA. The DCF MAC source code of the reference design has been modified to implement BidMAC. The focus has been put on the evaluation of the energy efficiency, which has been measured in each node using Energino [12]. In order to validate the accuracy of the experimental implementation, the maximum achievable energy efficiencies of DCF and BidMAC have been theoretically derived and compared to the experimental results, taking into account various values for relevant system parameters such as the traffic load, packet length, and data rate.

The rest of this paper is structured as follows. The system model is presented in Section II. For completeness, Section III briefly describes the DCF and BidMAC protocols. In Section IV, the protocols under consideration are analyzed in terms of energy efficiency. Section V provides an overview of the experimental implementation of the protocols. The results are discussed in Section VI. Finally, Section VII concludes this paper.

## II. NETWORK LAYOUT AND SYSTEM MODEL

A Basic Service Set (BSS) composed of an AP and  $N$  associated STAs in the Basic Service Area (BSA) is considered. All devices are equipped with IEEE 802.11g wireless interfaces since the selected reference design of WARP implements the IEEE 802.11a/g based on the specifications given in [1]. The size of the BSA allows all the STAs of the BSS to overhear the transmissions between each STA and the AP in both directions. The AP operates as a regular STA and thus can deliver downlink data to any STA in the BSS. In addition, all data packets transmitted have a fixed and common length for all devices.

The Physical (PHY) layer specification of the IEEE 802.11g is called Extended Rate PHY (ERP), which uses the Orthogonal Frequency Division Multiplexing (OFDM) modulation and provides 8 transmission modes with different modulation schemes and coding rates. The ERP-OFDM transmission rates are 6, 9, 12, 18, 24, 36, 48, and 54 Mbps and the Number of Data Bits Per OFDM Symbol ( $N_{DBPS}$ ) are 24, 36, 48, 72, 96, 144, 192, and 216, respectively. Note that data transmissions can be performed using any of these rates whereas acknowledgment (ACK) packets must be transmitted at the rates that use 1/2 rate coding, i.e., 6, 12, and 24 Mbps, based on the BSS basic rate selection rules specified in [1].

The expression to compute the transmission time of each packet using the ERP-OFDM PHY is given in [1] as

$$T_x = T_{pre} + T_{sig} + T_{sym} \left\lceil \frac{L_{serv} + 8 \cdot L_x + L_{tail}}{N_{DBPS}} \right\rceil + T_{sigEx} \quad (1)$$

TABLE I  
SYSTEM PARAMETERS

| Parameter  | Value        | Parameter    | Value     |
|------------|--------------|--------------|-----------|
| $T_{slot}$ | 9 $\mu s$    | $T_{sig}$    | 4 $\mu s$ |
| $T_{SIFS}$ | 10 $\mu s$   | $T_{sym}$    | 4 $\mu s$ |
| $T_{DIFS}$ | 28 $\mu s$   | $T_{sigEx}$  | 6 $\mu s$ |
| $T_{EIFS}$ | 88 $\mu s$   | $L_{serv}$   | 16 b      |
| $CW_{min}$ | 15           | $L_{tail}$   | 6 b       |
| $CW_{max}$ | 1023         | $L_{ACK}$    | 14 B      |
| $T_{BO}$   | 67.5 $\mu s$ | $L_{MAChdr}$ | 24 B      |
| $T_{pre}$  | 16 $\mu s$   | $L_{LLChdr}$ | 8 B       |

where  $x$  is the packet type and may correspond to DATA or ACK.  $T_{pre}$ ,  $T_{sig}$ ,  $T_{sym}$ , and  $T_{sigEx}$  denote the preamble time, the signal time, the OFDM symbol period, and the signal extension period, respectively. The ceiling function  $\lceil \cdot \rceil$  contains the sequences of services bits ( $L_{serv}$ ) and of tail bits ( $L_{tail}$ ).  $L_x$  is the MAC packet length and may correspond to the length of a data packet ( $L_{DATA}$ ) or MAC Protocol Data Unit (MPDU) or an ACK packet ( $L_{ACK}$ ). The MPDU includes a frame body or MAC Service Data Unit (MSDU) together with a MAC header ( $L_{MAChdr}$ ) and a Frame Check Sequence (FCS) represented by  $L_{FCS}$ . For purposes that will be clarified in the implementation part (Section V), a Logical Link Control (LLC) header ( $L_{LLChdr}$ ) is added after the MAC header. All the above parameters and their values are provided in Table I.

When the radio transceiver is on, the IEEE 802.11g wireless interface can be in one of the following three operational states: transmitting, receiving (or overhearing), and idle. In the first two states, the radio transceiver is actively used to transmit and receive information. In the idle state, the wireless interface is ready to transmit and receive but no signal is present in the radio transceiver. Let  $P_t$ ,  $P_r$ , and  $P_i$  denote the power consumed by the wireless interface in each of those states. The values of power consumption associated with each operational state for the WARP hardware have been obtained through experimental measurements performed on the platform, as it will be detailed later in Section V.

## III. MAC PROTOCOLS DESCRIPTION

The DCF MAC specification of the IEEE 802.11 Standard defines a basic access method that is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism in combination with a Binary Exponential Backoff (BEB) algorithm. In each transmission cycle, the transmitter, either the AP or a STA, senses the wireless channel for a DCF Interframe Space (DIFS) and, if the wireless channel is sensed busy, may wait for a random backoff time based on a Contention Window (CW). If the wireless channel is sensed idle after a DIFS or the backoff timer expires, the transmitter initiates data transfer to the receiver, indicating the expected occupancy time of the wireless channel in the header of transmitted data packets. Upon successful reception of data, the receiver responds with an ACK packet after a Short Interframe Space (SIFS). If no ACK packet is received within a given period of time, the transmitter waits for an Extended Interframe Space (EIFS) and then executes the BEB procedure for retransmission. The transmitter's CW size

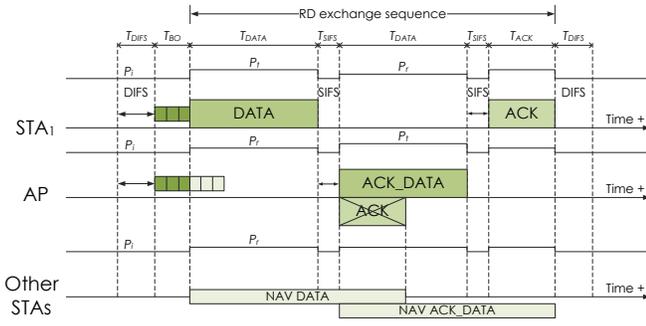


Fig. 1. Example of operation of the BidMAC protocol.

doubles each failed retransmission attempt up to a maximum value ( $CW_{max}$ ), and is reset down to a minimum value ( $CW_{min}$ ) after successful transmission (see Table I). All other STAs not involved in data transmission update their Network Allocation Vectors (NAV) with the value of the duration field contained in the MAC header of overheard DATA and ACK packets. They will not attempt channel access for the duration of the NAV.

The operation of BidMAC is based on the legacy DCF, but it also allows RD exchange sequences between the AP and the STAs with a single channel access invocation. Specifically, the receiver of a data packet, either the AP or a STA, is able to transmit a data packet of arbitrary length (from 0 to 2312 bytes of payload) with a piggybacked ACK whose destination is the transmitter of the received data packet. The transmission rate of the data packet is kept constant for both the forward and reverse transmissions. Also, the value of the duration field in the transmitted data packet is extended to reserve the wireless channel for the duration to complete the RD exchange sequence, including the transmission time of the ACK packet from the transmitter. To protect against hidden STAs, the optional Request-To-Send/Clear-To-Send (RTS/CTS) access method defined in the legacy DCF can be enabled in BidMAC. The operation of BidMAC can also be extended to support batch transmission, aggregation, and block ACK, which are features defined in the IEEE 802.11 Standard.

Fig. 1 shows an RD exchange sequence between STA 1 and the AP when BidMAC without RTS/CTS is executed. As it can be seen, the AP does not need to gain a TXOP to transmit data to STA 1, as it would happen when using the DCF. Instead, with BidMAC the AP uses the TXOP of STA 1 to send its data packet along with the ACK packet to it by extending the TXOP time through the NAV information carried in control and data packets. As a result, access delays can be reduced, hence improving throughput and energy efficiency.

#### IV. ENERGY EFFICIENCY ANALYSIS

The expressions of the maximum achievable energy efficiency for the protocols considered in this paper are derived from three different perspectives: entire network, AP, and average per STA.

The energy efficiency of a given protocol  $x$  ( $\eta_x$ ) is defined as the amount of bits contained in an MSDU ( $L_{MSDU}$ ) divided by the energy consumption ratio ( $E_x$ ) required to transmit a

data packet that includes the MSDU:

$$\eta_x [\text{Mb/J}] = \frac{8 \cdot L_{MSDU}}{E_x} \quad (2)$$

where  $E_x$  is defined as the product of power consumed and time spent in transmission over the total amount of transmitted data packets, and is split into three energy consumption components, namely, transmitting ( $E_t$ ), receiving and overhearing ( $E_r$ ), and idle ( $E_i$ ).

In order to compute the upper bound of the theoretical energy efficiency within a BSS in idealistic conditions, the following assumptions are made: (i) the wireless channel is ideal, (ii) the probability of collision is negligible, (iii) the propagation delay is neglected, (iv) the transmit queues are never empty, (v) no packets are lost due to queue overflow, (vi) no management packets, such as beacons and association requests, are transmitted, and (vii) fragmentation is not used.

In the following,  $T_{SIFS}$ ,  $T_{DIFS}$ , and  $T_{slot}$  denote the SIFS and DIFS intervals and the slot time, respectively, and  $T_{DIFS} = T_{SIFS} + 2T_{slot}$ . Since we consider no collisions, the backoff period ( $T_{BO}$ ) is an average value obtained from  $CW_{min}$  and  $T_{slot}$  as  $T_{BO} = (\frac{CW_{min}}{2}) T_{slot}$ . For the same reason,  $CW_{max}$  and the EIFS interval ( $T_{EIFS}$ ) are not considered herein (although they will be used in the experimental part), and  $T_{EIFS} = T_{DIFS} + T_{SIFS} + T_{ACK}$  (6Mbps). These variables and their values are shown in Table I.

The network energy consumption ratios of DCF ( $E_{DCF}^{net}$ ) and BidMAC ( $E_{BidMAC}^{net}$ ) are computed as follows. In each transmission cycle of DCF, the transmitter and the receiver, respectively, consume energy to transmit and receive a data packet and to receive and transmit an ACK packet. Meanwhile, the  $N-1$  STAs not involved in the transmission consume energy to overhear the exchange of packets. The AP and the  $N$  STAs also consume energy for listening to the wireless channel for the DIFS, average backoff, and SIFS intervals. Similarly, in BidMAC an additional data transmission from the receiver to the transmitter (as an implicit ACK) with the response of an ACK packet from the transmitter, followed after a SIFS, is performed, which is overheard by the other  $N-1$  STAs. Hence,  $E_{DCF}^{net}$  and  $E_{BidMAC}^{net}$  can generally be expressed as

$$\begin{aligned} E_x &= \alpha (E_t + E_r + E_i) \\ E_t &= (\beta T_{DATA} + \gamma T_{ACK}) P_t \\ E_r &= (\delta T_{DATA} + \epsilon T_{ACK}) P_r \\ E_i &= \zeta (T_{DIFS} + T_{BO} + \kappa T_{SIFS}) P_i \end{aligned} \quad (3)$$

where  $\alpha = \beta = \gamma = \kappa = 1$ ,  $\delta = \epsilon = N$ , and  $\zeta = N + 1$  for  $E_{DCF}^{net}$  and  $\alpha = \frac{1}{2}$  and  $\beta = \kappa = 2$ ,  $\gamma = 1$ ,  $\delta = 2N$ ,  $\epsilon = N$ , and  $\zeta = N + 1$  for  $E_{BidMAC}^{net}$ .

The AP energy consumption ratios of DCF ( $E_{DCF}^{AP}$ ) and BidMAC ( $E_{BidMAC}^{AP}$ ) are calculated as described next. When the DCF is executed,  $E_{DCF}^{AP}$  shows a minimum value ( $E_{DCF}^{APmin}$ ) and a higher stable value under saturation ( $E_{DCF}^{APsat}$ ) due to the long-term fairness of the DCF. To compute  $E_{DCF}^{APmin}$ , the case when the STAs can transmit  $N$  data packets to the AP and the AP can deliver  $N$  data packets to them is

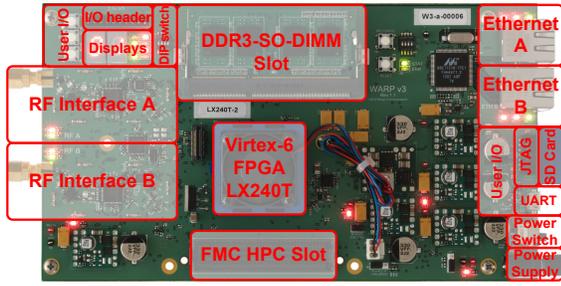


Fig. 2. WARP v3 and its hardware features [13], [14].

analyzed. In contrast,  $E_{DCF}^{APsat}$  is obtained when the AP can perform data transmission only once every  $N$  transmissions from the STAs. When BidMAC is enabled, the AP is able to transmit a data packet after receiving a data packet from each STA and when it gains a TXOP, wherein the receiving STA can also send back a data packet.  $E_{DCF}^{AP}$  and  $E_{BidMAC}^{AP}$  are thus written using (3), where  $\alpha = \frac{1}{N}$ ,  $\beta = \gamma = \delta = \epsilon = N$ ,  $\zeta = 2N$ , and  $\kappa = 1$  for  $E_{DCF}^{APmin}$ ,  $\alpha = \beta = \epsilon = \kappa = 1$ ,  $\gamma = \delta = N$ , and  $\zeta = N + 1$  for  $E_{DCF}^{APsat}$ , and  $\alpha = \frac{1}{N+1}$ ,  $\beta = \delta = \zeta = N + 1$ ,  $\gamma = 1$ ,  $\epsilon = N$ , and  $\kappa = 2$  for  $E_{BidMAC}^{AP}$ .

Finally, the average per-STA energy consumption ratios of DCF ( $E_{DCF}^{STA}$ ) and BidMAC ( $E_{BidMAC}^{STA}$ ) are explained as follows. In the DCF, a STA acts as a transmitter once every  $N$  transmissions from  $N-1$  STAs and the AP. During  $N-1$  transmissions, a STA overhears. When the AP gains a TXOP, a STA can be the actual receiver with probability  $1/N$  (assuming uniform traffic distribution), whereas with probability  $1-1/N$  it is not the intended destination. A similar explanation can be given for BidMAC, except that in BidMAC each transmission is bidirectional and so a STA receiving a data packet from the AP (with probability  $1/N$ ) can respond with a data packet. As a result,  $E_{DCF}^{STA}$  and  $E_{BidMAC}^{STA}$  are given using (3), where  $\alpha = \beta = \kappa = 1$ ,  $\gamma = \frac{1}{N}$ ,  $\delta = N$ ,  $\epsilon = N + 1 - \frac{1}{N}$ , and  $\zeta = N + 1$  for  $E_{DCF}^{STA}$  and  $\alpha = (1 + \frac{1}{N})^{-1}$ ,  $\beta = 1 + \frac{1}{N}$ ,  $\gamma = \frac{1}{N}$ ,  $\delta = 2N + 1 - \frac{1}{N}$ ,  $\epsilon = N$ ,  $\zeta = N + 1$ , and  $\kappa = 2$  for  $E_{BidMAC}^{STA}$ .

## V. IMPLEMENTATION

This section describes the implementation of the proposed BidMAC protocol based on the IEEE 802.11 reference design of WARP. A brief overview of the WARP platform is first provided, followed by a description of the reference design considered. Finally, the details of the experiments framework are presented, such as equipment used, measurement tools, and methodology.

### A. WARP platform overview

WARP is a high-performance programmable wireless platform to implement PHY, MAC, and network layer protocols. It was originally developed by Rice University within the WARP Project [13] and is currently manufactured and distributed by Mango Communications [14]. The latest generation of WARP hardware is WARP v3 (see Fig. 2). This is a Field Programmable Gate Array (FPGA) board composed of a Xilinx Virtex-6 FPGA with an embedded PowerPC processor, two programmable Radio Frequency (RF) interfaces each with

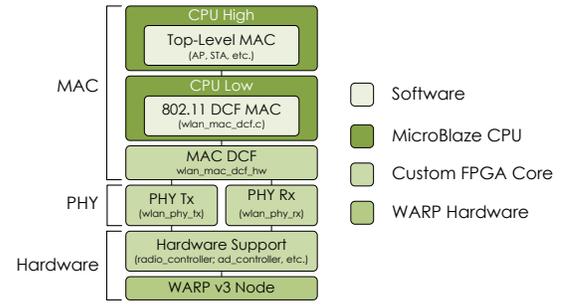


Fig. 3. Architecture of the IEEE 802.11 reference design [13], [14].

a 2.4/5GHz transceiver (40MHz RF bandwidth), and multiple peripherals, such as two Gigabit Ethernet interfaces. Further information about WARP v3 can be found in [13], [14].

### B. IEEE 802.11 reference design

The WARP Project provides an open-source repository of C-coded reference designs and support materials for the WARP hardware. In particular, the Mango 802.11 reference design is a real-time FPGA implementation of the DCF MAC and OFDM PHY from IEEE 802.11a/g for the WARP v3 hardware, which can operate as an AP or a STA. The overall architecture of this reference design is illustrated in Fig. 3. PHY processing is performed by the PHY Tx/Rx cores, or Central Processing Units (CPU). MAC functions are mainly implemented in software running in two MicroBlaze CPUs (i.e. CPU High and CPU Low), with an intermediate core interfacing to the PHY Tx/Rx cores (i.e., MAC DCF) and a support core (i.e., Hardware Support) to achieve accurate inter-packet timing.

Focusing on the MAC layer of the design, the MAC software implementation is split into two pieces: the upper-level MAC and the lower-level MAC, which communicate with each other via inter-processor mailbox. The upper-level MAC code contains the AP/STA implementations (`wlan_mac_ap.c` and `wlan_mac_sta.c`) and a collection of their shared inter-packet behaviors that are not time critical, referred to as MAC High Framework. The interactions between the different upper-level MAC implementations and this framework may involve notification of wired/wireless reception and command of wired/wireless transmission. Also, the framework provides a Local Traffic Generator (LTG) to generate data packets of arbitrary length up to 1500 bytes (LTG Payload) at periodic or uniform random intervals (LTG Schedule). Note that LTG data packets include an LLC header to avoid that non-WARP devices, e.g. laptops and smartphones, can process them.

On the other hand, the lower-level MAC code (`wlan_mac_dcf.c`) handles intra-packet states that are time critical for the DCF via the MAC DCF core (`wlan_mac_dcf_hw`) in order to perform wireless transmission and reception. This core directly connects to the Tx/Rx PHY control and status signals and implements the timers and state machines required to meet the IEEE 802.11 channel access timing requirements. For instance, in this core a small state machine, called Auto Tx or auto-responder, that initiates a PHY transmission in response to a valid PHY

reception is integrated to enable transmission of ACK packets immediately after a SIFS.

### C. Description of the BidMAC implementation based on the reference design of WARP

The BidMAC protocol is mainly implemented in the lower-level MAC of the 802.11 reference design, i.e., the C code in the CPU Low MicroBlaze core (`wlan_mac_dcf.c`). The proposed BidMAC implementation allows the AP to transmit a data packet (with an implicit ACK) of the same length and with the same transmission rate as those of the received data packet back to the transmitting STA after a SIFS, upon successful data reception. The main modifications to the existing MAC software of the reference design are described below. The reader may refer to [15] to see the new pieces of code.

In the `wlan_mac_dcf.c` file, the static MAC addresses of the WARP v3 nodes considered, namely, an AP and several STAs, are defined to determine which of them is the receiver of a data packet inside the `frame_receive` function. In this function, when a packet of type DATA destined to the AP is received with a valid FCS, the auto-responder state machine is enabled and configured with a new type of data packet, called `ACK_DATA`. This packet type is defined in the `wlan_mac_802_11_defs.h` (MAC High Framework) and the packet is created via a new function, called `wlan_create_ack_data_frame`, that sets the packet type in the `frame_control_1` field contained in the header of the packet. Since DATA packets are created by the upper-level MAC (`wlan_mac_packet_types.c`), this function is required in the lower-level MAC to prepare an `ACK_DATA` packet for transmission before reception completes, thus respecting the SIFS requirement. Also, the reception of an `ACK_DATA` packet with a valid FCS by a STA involves processing the received packet as an ACK and DATA packet, generating an ACK auto-response and notifying data reception to the upper-level MAC. Specifically, a condition is included in the `mpdu_rx_process` function contained in the `wlan_mac_sta.c` file to account received `ACK_DATA` packets as DATA packets and update reception statistics.

### D. Experimental setup

An experiment framework called WARPnet [13] is used for the experimental evaluation of the DCF and BidMAC implementations. WARPnet is a Python-coded environment that allows performing real-time experiments with multiple WARP nodes through an experiment controller running on a host PC. Specifically, the WARPnet module implemented for the 802.11 reference design is called `wlan_exp`. This framework enables low-level visibility and control of MAC and PHY behaviors of the reference design in real-time.

The testbed used to perform the experiments with the `wlan_exp` module consists of two systems: wireless and wired (see Fig. 4). The wireless system implements an IEEE 802.11g WLAN composed of three WARP v3 nodes, an AP and STA 1 and STA 2, that are placed at 1-meter distance from each other, forming an equilateral triangle, in a zone free of wireless

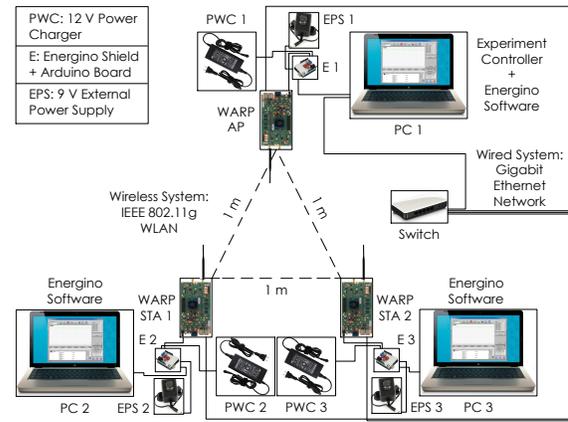


Fig. 4. The testbed layout.

interferences. Each WARP v3 node is equipped with a single common Wi-Fi 2.4 GHz antenna and a 12 V power charger. The wired system, instead, implements a Gigabit Ethernet network that connects the WARP v3 nodes to a PC (i.e., the experiment controller) through a switch. The experiment controller launches custom-design Python scripts that exploit various features of the `wlan_exp` experiment framework. The scripts generate traffic flows between the AP and the STAs through the LTG framework and calculate the throughput as the number of delivered bits of information over a given trial time, using Tx/Rx packet counts at each node.

Specifically, three different scripts have been developed. The first script generates bidirectional symmetric traffic flows of different periodic inter-packet arrival intervals (from long to short) between the AP and each STA with a constant data payload length (MSDU) of 1400 bytes and a fixed PHY data rate of 54 Mbps. Note that for BidMAC only unidirectional data flows from each STA to the AP are configured, since the AP will automatically generate an `ACK_DATA` packet for each STA in response to successful data reception. The second script varies the MSDU length from 50 to 1500 bytes with a 250-byte interval and considering zero inter-packet arrival interval (i.e., fill up the transmit queues to reach the saturation state) and a fixed PHY data rate of 54 Mbps. Finally, the third script tunes the PHY data rate from 6 to 54 Mbps with zero inter-packet arrival interval and a constant MSDU length of 1500 bytes. In all these scripts, the trial time for each experiment is set to 30 s and the throughput results are obtained as an average value of 10 repetitions per experiment.

In order to compute the energy efficiency results, the throughput results are divided by the power consumption data of the WARP v3 boards, gathered during the experiments from the Energino meters via custom-design software. Energino [12] is an Arduino-based energy consumption monitoring platform, designed and developed by the iNSPIRE group at CREATE-NET within the Energino Project [16], that provides real-time precise energy consumption statistics for any DC appliance. The electronic components of Energino are shown in Fig. 5, whose main building blocks are a voltage sensor implemented using a voltage divider, a current sensor based on the Hall effect, and a management module implemented

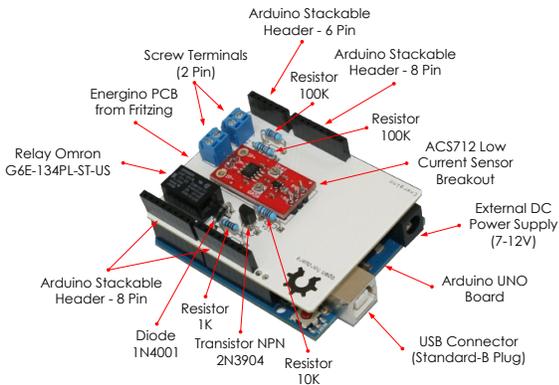


Fig. 5. Energino shield on top of the Arduino UNO board [12], [16].

using a mechanical relay. The specific features of Energino can be found in [16]. Three Energino shields on top of Arduino UNO boards are built following the instructions given in [16] and redesigned in software to achieve sampling rates of 15 kHz. Each Energino shield is connected to the WARP v3 board's power supply and its power charger using the screw terminals. The Arduino UNO board assembled below each Energino shield is connected to a PC using the USB interface. Also, an additional external power source of 9 V is used to supply the Arduino UNO board (see Figs. 4 and 5).

A custom program developed in LabVIEW is executed in each PC to control Energino and acquire samples of voltage, current, and power for each WARP v3 board during a selected period of time. This software provides an easy-to-use visual interface (see Fig. 6) and also allows averaging the samples values and calibrating the voltage and current sensors of Energino. For instance, the average value of power consumption measured in the WARP v3 boards when transmitting ( $P_t$ ), receiving ( $P_r$ ), and being idle ( $P_i$ ) during the experiments is 18.95 W (each board). This value is used in (3) to obtain the theoretical energy efficiency results for the protocols analyzed. Also, note that the Energino meters start sampling 5 s before the beginning of a new experiment in order to gather the power consumption data exactly during the 30 s that each experiment takes. More details about how to use the custom LabVIEW program are provided in [15].

## VI. RESULTS

The results of energy efficiency obtained from the analysis and experiments described in the previous sections for the DCF and BidMAC protocols are presented and discussed in this section. They are summarized in Fig. 7. In general, it can be seen that in all the graphs the experimental results are in line with the analytical results for both protocols. The differences between analytical and experimental results in DCF are due to channel errors and collisions that may occur during the experiments. On the contrary, in BidMAC the upper bounds obtained experimentally are slightly higher than those derived analytically. The reason for this variation is that the proposed BidMAC implementation only allows the AP (and not the STAs) to exploit RD transmissions and also do not require the AP to compete for the channel access, in contrast with the

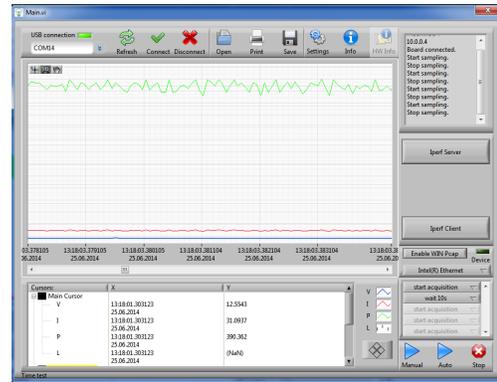


Fig. 6. Visual interface of the custom-design software in LabVIEW to control Energino [15].

general BidMAC operation considered in the analysis.

Figs. 7a, 7b, 7c, respectively, show the network, AP, and average per-STA energy efficiencies of DCF and BidMAC versus the total offered traffic load. When either the DCF or BidMAC is executed and the traffic load is low, the AP and the two STAs can transmit all their data packets normally. Note that the AP transmits twice more data packets than the STAs as it delivers downlink traffic that is symmetric to the uplink traffic received from them. As the traffic load increases, the AP and the STAs transmit more frequently and so their energy efficiencies increase due to the increase of data transmitted and the reduction of idle periods which waste energy. The AP achieves the highest energy efficiency when the total traffic load is almost 30 Mbps (see Fig. 7b), where the AP captures half of the channel accesses and each of the two STAs obtains a quarter (half in total) of the channel accesses.

When the traffic load increases above that value until reaching the saturation point (below 40 Mbps), the channel share of the AP is reduced down to one third whereas those of the STAs increase up to two thirds (one third each), due to the DCF MAC fairness. As a result, the AP experiences a significant reduction of its energy efficiency that affects the STAs in terms of a lower amount of received downlink packets. In contrast, BidMAC allows the AP to initiate contention-free channel accesses to deliver downlink data to the STAs after each successful data reception, thus increasing the amount of downlink packets transmitted and reducing energy consumption due to unnecessary backoff periods. Therefore, when the traffic load is high the energy efficiency of the AP improves by 98%, as well as the network energy efficiency by 31%, with minimum impact on the energy efficiencies of the STAs.

The network energy efficiencies of DCF and BidMAC under saturation (i.e., the AP and the STAs have always data ready to be transmitted) versus the MSDU length and the PHY data rate are reported in Figs. 7d and 7e, respectively. It can be seen in both figures that BidMAC outperforms DCF for all MSDU lengths and PHY data rates considered, showing significant gains. Whereas in Fig. 7d the gain of BidMAC versus DCF decreases from 63% to 29% as the MSDU length increases, Fig. 7e shows that the gain varies between 15% and 29% with increasing PHY data rates. The reason for these behaviors is

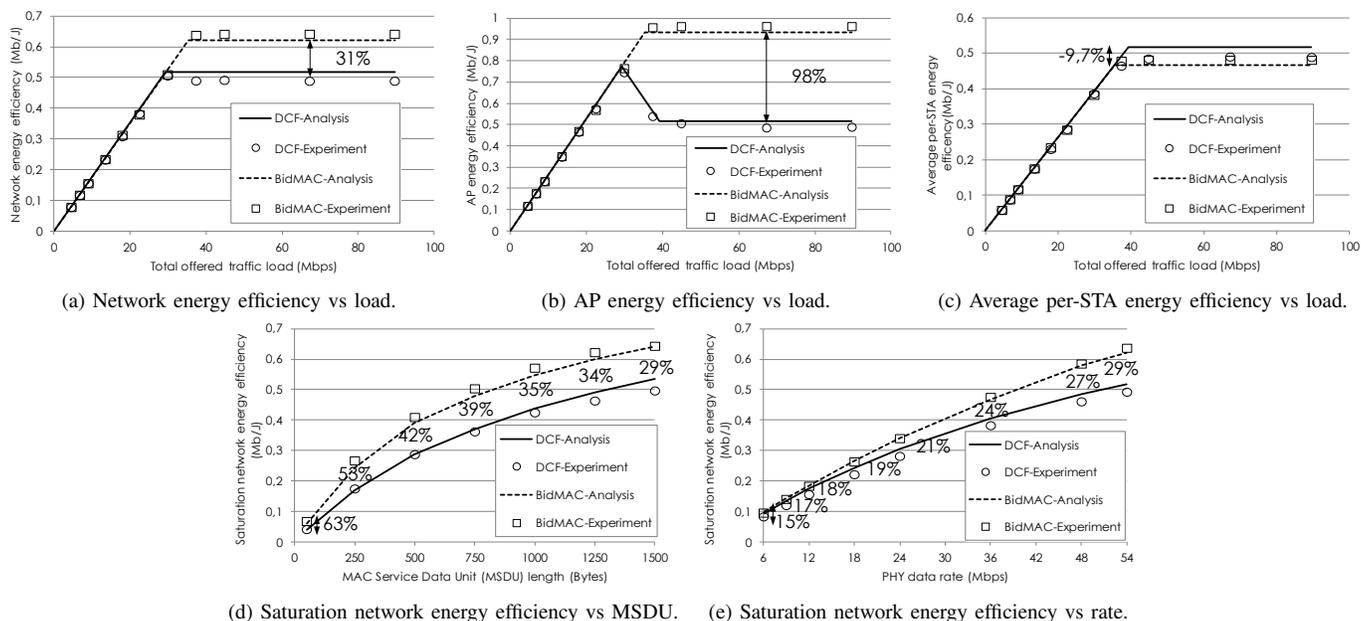


Fig. 7. Network, AP, and average per-STA energy efficiencies of DCF and BidMAC versus the total offered traffic load with an MSDU length of 1400 bytes and a PHY data rate of 54 Mbps and saturation network energy efficiencies of DCF and BidMAC versus the MSDU length (with 54 Mbps) and PHY data rate (with 1500 bytes).

related to the influence of the data transmission time on the total time required to transmit data in DCF and BidMAC. While faster rates or shorter packet lengths lead to shorter data transmission times with lower impact on the total transmission time, slower rates or longer packet lengths imply longer data transmission times with higher impact.

## VII. CONCLUSIONS

An experimental evaluation of the energy efficiency of a reactive RD MAC protocol, named BidMAC, using the WARP platform has been presented in this paper. BidMAC has been implemented on top of the DCF MAC of the IEEE 802.11a/g reference design of WARP to allow the AP to respond with a data packet with an implicit ACK after successful data reception. A testbed composed of a WARP AP and two WARP STAs has been set up to measure the energy efficiency of BidMAC using custom-design Python scripts and Energino meters. A comparison of BidMAC with the legacy DCF has also been provided considering the influence of the traffic load, packet length, and data rate. Analytical and experimental results have shown the high energy efficiency of BidMAC especially for high traffic loads, short packet lengths, and fast data rates. For instance, the maximum energy efficiency gains of BidMAC versus DCF range from 63% to 29% as the packet length grows and from 15% to 29% as the data rate increases.

Future work will be aimed at improving the current BidMAC implementation by incorporating packet aggregation and block ACK and also at evaluating the new implementation with different traffic classes using the WARP's Ethernet interfaces.

## REFERENCES

[1] IEEE, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE 802.11 Std., 2012.

[2] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," in *IEEE INFOCOM 2002*, vol. 2, Jun. 2002, pp. 599–607.

[3] M. Ozdemir, G. Daqing, A. B. McDonald, and J. Zhang, "Enhancing MAC Performance with a Reverse Direction Protocol for High-Capacity Wireless LANs," in *IEEE VTC 2006*, Sep. 2006, pp. 1–5.

[4] D. Akhmetov, "802.11n: Performance Results of Reverse Direction Data Flow," in *IEEE PIMRC 2006*, Sep. 2006, pp. 1–3.

[5] D.-H. Kwon, W.-J. Kim, and Y.-J. Suh, "A Bidirectional Data Transfer Protocol for Capacity and Throughput Enhancements in Multi-rate Wireless LANs," in *IEEE VTC 2004*, vol. 4, Sep. 2004, pp. 3055–3059.

[6] W. Choi, J. Han, B. J. Park, and J. Hong, "BCTMA (Bi-directional Cut-Through Medium Access) Protocol for 802.11-based Multi-hop Wireless Networks," in *ACM ISSADS 2005*, Jan. 2005, pp. 377–387.

[7] N. S. P. Nandiraju, H. Gossain, D. Cavalcanti, K. R. Chowdhury, and D. P. Agrawal, "Achieving Fairness in Wireless LANs by Enhanced IEEE 802.11 DCF," in *IEEE WiMob 2006*, Jun. 2006, pp. 132–139.

[8] R. Palacios, F. Granelli, D. Gajic, and A. Foglar, "An Energy-Efficient MAC Protocol for Infrastructure WLAN Based on Modified PCF/DCF Access Schemes Using a Bidirectional Data Packet Exchange," in *IEEE CAMAD 2012*, Sep. 2012, pp. 216–220.

[9] R. Palacios, F. Granelli, D. Kliazovich, L. Alonso, and J. Alonso-Zarate, "Energy Efficiency of an Enhanced DCF Access Method Using Bidirectional Communications for Infrastructure-based IEEE 802.11 WLANs," in *IEEE CAMAD 2013*, Sep. 2013, pp. 38–42.

[10] F. V. Gallego, J. Alonso-Zarate, C. Verikoukis, and L. Alonso, "A Survey on Prototyping Platforms for the Development and Experimental Evaluation of Medium Access Control Protocols," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 74–81, Feb. 2012.

[11] C. Hunter, J. Camp, P. Murphy, A. Sabharwal, and C. Dick, "A Flexible Framework for Wireless Medium Access Protocols," in *IEEE ACSSC 2006*, Oct. 2006, pp. 2046–2050.

[12] K. M. Gomez, R. Riggio, T. Rasheed, D. Miorandi, and F. Granelli, "Energino: A Hardware and Software Solution for Energy Consumption Monitoring," in *IEEE WiOpt 2012*, May 2012, pp. 311–317.

[13] WARP Project. <http://warpproject.org>.

[14] Mango Communications. <http://mangocomm.com>.

[15] F. Franch, R. Palacios, and F. Granelli, "Performance Evaluation of Energy Efficient and Network Coding-Aware MAC Protocols for IEEE 802.11 Wireless Networking Using the WARP Experimental Platform," Master's thesis, DISI, University of Trento, 2014.

[16] Energino Project. <http://energino-project.org>.