Network Coding and Duty Cycling in IEEE 802.11 Wireless Networks with Bidirectional Transmissions and Sleeping Periods

Raul Palacios-Trujillo[a], Jesus Alonso-Zarate[b], Fabrizio Granelli[c],
Frank H. P. Fitzek[d] and Nelson L. S. da Fonseca[a]

[a]Institute of Computing, University of Campinas, Brazil, *{rpalacios, nfonseca}@ic.unicamp.br*
[b]Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Spain, *jesus.alonso@cttc.es*
[c]Department of Information Engineering and Computer Science, University of Trento, Italy, *granelli@disi.unitn.it*
[d]Communication Networks Group, Technical University of Dresden, Germany, *frank.fitzek@tu-dresden.de*

Abstract*—In this paper, we propose an energy-efficient solution for implementing Network Coding (NC) in wireless networks based on the IEEE 802.11 Standard. The proposed mechanism, called GreenCode, allows nodes to duty cycle by switching to a low-power (sleep) state when they overhear coded packet transmissions that will not provide any new information for them. To facilitate the sleep operation, bidirectional transmissions involving both coded and non-coded packets between pairs of sender-receiver nodes are integrated into the operation of GreenCode. Both analytical and simulation results presented in this paper show the high energy efficiency of GreenCode with gains of up to 360% when compared to the existing mechanisms based on the IEEE 802.11 Standard.*

I. INTRODUCTION

Maximizing energy efficiency in wireless networks has become a major challenge over the recent years and Network Coding (NC) constitutes a promising technology to increase the energy efficiency of wireless networks [1]. The basic idea is to allow intermediate nodes between a group of sources and potential destinations to transmit combined packets to multiple receivers simultaneously by exploiting the broadcast nature of the wireless channel. More specifically, the packets can be coded (and decoded) by applying linear operations (e.g., XOR) and can contain information from one or several sources, thus being referred to as intra- or inter-session NC, respectively. An encoding vector is then added to the header of the coded packets to perform successful decoding at the receivers. Despite this coding overhead, there is a gain that stems from the fact that each packet transmission conveys more information and the total number of channel accesses from source to destination is reduced.

Motivated by this fact, the work in [2] proposed the first implementation of an inter-session NC protocol in Wi-Fi networks based on the IEEE 802.11 Standard [3]. The techniques proposed therein were specifically designed to cope with the inherent NC unawareness of the mandatory Medium Access Control (MAC) protocol of the Standard, referred to as Distributed Coordination Function (DCF). This MAC protocol is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) in combination with a Binary Exponential Backoff (BEB) algorithm and has been widely optimized for many years due to several inefficiencies of the legacy specification.

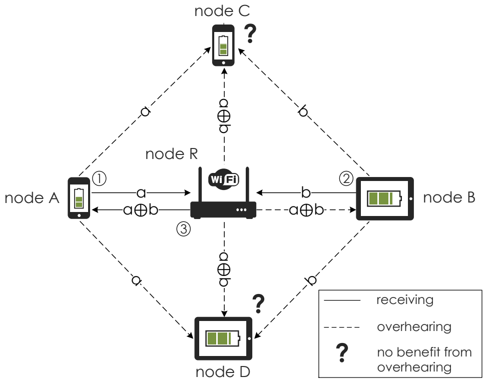

Fig. 1. Example of the overhearing problem when NC is enabled in the canonical cross network.

The basic operation of NC is exemplified in Fig. 1. In this example, pairs of source nodes A-B and C-D exchange data packets through an intermediate node R. All nodes operate in promiscuous mode, thus listening to all transmitted packets regardless of their intended destination. Node R encodes two received packets from each pair of nodes together and sends the coded packets to one of the intended receivers. The coded packets include a new header that contains the MAC address of other potential receivers to ensure successful decoding of overheard coded packets. From the energy point of view, this operation may yield low energy efficiency due to the continuous overhearing of the wireless channel, being some packets not useful for some nodes. As shown in Fig. 1, when node R sends a coded packet containing information from nodes A and B to node A, nodes C and D do not benefit from overhearing that coded packet, thus spending energy.

To overcome this problem, the DutyCode protocol was presented in [4]. This protocol lets nodes duty cycle by switching to a low-power (sleep) state, in which they turn off their radio transceivers, when they overhear coded packets

---

*This work has been funded by the GREENET European research project under grant PITN-GA-2010-264759, the CNPq Brazilian research agency, and the Catalan Government under grant 2014-SGR-1551.

that have already been decoded before. The focus of that work was on dissemination applications in flooding-based wireless sensor networks where the redundancy of coded packets being transmitted is likely to occur. Specifically, nodes group packets into logical entities, called pages, and only encode packets of the same page. Then, all the packets of a page are sent together into a stream using CSMA. Based on the first received packets, nodes decide whether a stream is useful and they stay awake to receive all the packets or they can sleep for an estimated duration of the stream. The time to sleep and the sleep duration are randomly chosen using elastic intervals. Unfortunately, this operation may induce nodes to sleep when useful packets are being transmitted. In addition, DutyCode has not been tested in other scenarios different from that based on flooding.

In this paper, we propose a more general-purpose energy-efficient solution for implementing NC in wireless networks based on the IEEE 802.11 Standard. The proposed protocol, called GreenCode, allows nodes to go to sleep when they overhear coded packet transmissions that will not provide any new information for them, hence saving energy and increasing energy efficiency. However, the sleep operation will only be feasible if the packet transmission time is longer than the duration of the transitions of radio transceivers between on and off states [5]–[7]. Therefore, nodes can also perform receiver-initiated (reactive) bidirectional, also referred to as Reverse Direction (RD), transmissions involving both coded and non-coded packets, thus increasing the total packet transmission time and enabling the duty cycling. GreenCode is backwards compatible with the IEEE 802.11 Standard since its operation is based on the combination of two already standardized mechanisms. One is the IEEE 802.11ac Transmission Opportunity Power Save Mode (TXOP PSM) to enable sleeping periods. The other is the IEEE 802.11n RD Protocol (RDP) to enable transmitter-initiated (proactive) RD transmissions.

The operation of GreenCode differs from that of the previously proposed DutyCode protocol in various aspects. Firstly, GreenCode provides NC awareness at the MAC layer and allows sleeping periods in a packet-per-packet basis, in contrast with DutyCode where random sleeping periods are determined by an NC-aware application layer. This operation allows nodes to precisely determine the sleep duration without missing useful coded packets transmissions. Secondly, GreenCode does not require sending packets into bursts and applying NC to a specific group of packets, as it is the case for DutyCode to efficiently work in dissemination activities. Instead, GreenCode has been designed for a general use and NC can be applied to any packets. For this reason, GreenCode has not been compared to DutyCode. In addition, the integration of reactive RD coded data transmissions into the operation of GreenCode to facilitate the sleep operation can further increase the energy efficiency, as shown in [8], [9]. The results presented in this paper show that GreenCode can increase the energy efficiency by up to 360% and 130% when compared to legacy mechanisms alone and combined with NC, respectively.

The remainder of this paper is organized as follows. Section II provides a detailed description of the operation of GreenCode. The derivation of the maximum achievable energy efficiency of GreenCode is presented in Section III. The evaluation of its energy efficiency is analyzed and quantified by means of analytical and simulation results in Section IV, considering the reference scenario shown in Fig. 1. For this evaluation, various system parameters, such as, the traffic load, packet length, and data rate, have been tested. In addition, the performance of GreenCode is also compared in this section to those of the IEEE MAC protocol (DCF) with and without NC. Finally, Section V concludes this paper by summarizing the most relevant results and outlining possible future work.

II. GREENCODE: A NEW NC-AWARE ENERGY-EFFICIENT MAC PROTOCOL

In this section, we introduce a new protocol, called GreenCode, that integrates NC awareness, duty cycling (i.e., enabling sleeping periods during which the radio transceiver is switched off to save energy) and bidirectional (RD) transmissions at the MAC layer to increase the energy efficiency of wireless networks based on the IEEE 802.11 Standard.

A. Protocol Description

The proposed GreenCode protocol allows nodes to switch to a low-power (sleep) state when they do not benefit from overhearing coded packets being transmitted, thus reducing energy consumption due to idle channel listening and overhearing. In addition, the operation of GreenCode integrates receiver-initiated (reactive) RD exchanges involving both coded and non-coded data packets between pairs of sender-receiver nodes, hence increasing the duration of each single channel access and facilitating the sleep operation.

The use of RD transmissions also guarantees a much higher share of the wireless channel for congested intermediate nodes as they can send coded packets when receiving data packets, in a contention-free manner. This characteristic of GreenCode represents a key advantage over existing mechanisms based on the basic operation of the IEEE 802.11 DCF MAC protocol (i.e., CSMA/CA and BEB) when alone or combined with NC. This is due to the fact that in DCF all contending nodes are constrained by long-term channel access fairness regardless of the capability of intermediate nodes to convey more information in a single transmission by using NC. As a result, the proposed GreenCode protocol can significantly improve the end-to-end network performance, thus also further increasing the energy efficiency.

GreenCode seamlessly integrates a linear (XOR) inter-session NC protocol on top of the IEEE 802.11 protocol stack, based on the framework presented in [2]. Specifically, all nodes receive all transmitted data packets regardless of their intended destination and store both transmitted and received packets for a limited time. Also, all received packets are held for a given time, called holding time, before forwarding them without coding in order to increase coding opportunities. Then, pairs of data packets received from different sources and addressed to other destinations are coded together into a single coded packet that is sent to one of the intended destinations to produce a synchronous Acknowledgment (ACK) response. This operation simplifies decoding at the receivers and is more robust against packet losses at overhearing nodes, since they only need to correctly receive a coded packet to successfully decode it by using a previously transmitted data packet. Furthermore, most of overhearing nodes can sleep during RD coded data transmissions where they are not involved as coded packets are always intended for two potential receivers. One is the actual destination of the RD transmission and the other is indicated in the header of the transmitted coded packet as an overhearing node that is able to decode it.

The operation of the proposed GreenCode protocol at the MAC layer is backwards compatible with the IEEE 802.11 Standard since it is based on the combination of already standardized mechanisms whose operation parameters are adjusted to efficiently work with NC. In particular, GreenCode integrates three MAC-layer techniques for energy saving at nodes, while being aware of the NC approach: *i)* opportunistic sleeping, *ii)* dynamic RD exchanges, and *iii)* enhanced Request-To-Send/Clear-To-Send (RTS/CTS) operation.

1) Enabling Sleeping Periods: GreenCode adopts the IEEE 802.11ac TXOP PSM operation when nodes overhear useless coded packet transmissions, as determined by the enhanced RTS/CTS exchange operation described later in this section. When the wireless channel is sensed busy, nodes back off and execute a virtual carrier sense mechanism. This mechanism allows them to determine the amount of time that the wireless channel will remain busy (i.e., the total transmission time or TXOP duration). They read the duration field of overheard control and data packets and set or update their Network Allocation Vectors (NAVs) with the duration values. Then, they can sleep until the NAV timers expire provided that the NAV values are longer than the duration of the on/off radio transitions, which is hundreds of microseconds [5]–[7].

2) Integrating RD Transmissions: GreenCode integrates a variation of the IEEE 802.11n RDP to be used in conjunction with NC, presented and evaluated in [8], [9], to facilitate the TXOP PSM execution. This mechanism allows the receiver of a valid data packet to reserve the wireless channel for an RD transmission (as an implicit ACK) by extending the transmitter's TXOP time, without contending for channel access. The RD transmission can be initiated by the receiver (and not by the transmitter as it is the case in RDP) only if there is a coded packet ready to be transmitted or a data packet whose holding time has expired or when the immediately received data packet can be combined with a stored data packet in the queue. A reactive RD exchange extends the total transmission time to include both forward and backward transmissions, thus enabling the TXOP PSM operation.

3) Enhanced RTS/CTS exchanges: GreenCode defines an enhanced RTS/CTS operation to let overhearing nodes know when upcoming RD (coded packet) transmissions will be useful for them and they need to remain awake to receive the packets or they can sleep for the duration of the RD exchanges. Specifically, the proposed mechanism, coined CTS-awake, allows the receiver of an RTS packet to send a CTS packet to another node different from the transmitter when such node needs to be awake to decode the upcoming transmitted coded packet within the RD exchange. In this way, the node will not attempt to sleep after setting or updating its NAV with the duration value of the overheard CTS packet extended to account for the total RD exchange sequence. Then, the transmitter of the RTS packet can interpret the overheard CTS packet as a transmission grant from the receiver, since it is waiting for a CTS packet destined to its address.

B. Example of Operation

Fig. 2 shows an example of the operation of GreenCode in the cross network depicted in Fig. 1. In this example, nodes A and B exchange a pair of data packets through node R, while nodes C and D are listening to the wireless channel. Node R implements a zero holding time, which means that packets are opportunistically coded without holding them for a given time. The packet exchange sequence between nodes A, R, and B can be described in two phases as follows.

1) Node A sends packet a to node R: Nodes A and B sense the wireless channel for a DCF Interframe Space (DIFS) and then invoke the backoff (BEB) procedure based on a Contention Window (CW). Due to a shorter backoff period, Node A seizes the wireless channel earlier and sends an RTS packet to node R. Nodes C and D overhear the RTS packet and set their NAVs, while node B continues the backoff countdown as it is out of transmission range with node A. Node R receives the RTS packet and replies with a CTS packet after a Short Interframe Space (SIFS). Node B overhears the CTS packet, halts its backoff timer, and sets its NAV. After a SIFS, node A sends packet a and node R responds with an ACK packet, after a SIFS. This operation follows the standard DCF rules.

2) Node B sends packet b to node R and node R responds with coded packet $a \oplus b$: Node R initiates a backoff procedure to send packet a to node B after a DIFS. However, node B is first to complete the backoff countdown and sends an RTS packet to node R. After receiving the RTS packet, node R identifies an opportunity for coding with stored packet a and for exploiting the RD transmission mode to send the possible coded packet. Thus, it prepares a CTS packet with the value of the duration field extended to cover the RD transmission. The CTS packet is transmitted, after a SIFS, to node A to force it to stay awake and overhear the potentially coded packet. Nodes C and D go to sleep after updating their NAVs with the duration value of the overheard CTS packet. Node B also overhears the CTS packet from node R and, after a SIFS, sends packet b to it. Then, node R responds with coded packet $a \oplus b$ (as an implicit ACK), after a SIFS. Node B completes the RD exchange sequence by sending back an ACK packet, after a SIFS. Node A identifies its address in the header of the overheard coded packet. Therefore, both nodes A and B can retrieve packets b and a, respectively, by using their own packets and the received coded packet.

III. ENERGY EFFICIENCY ANALYSIS

In this section, we present and explain the mathematical expression that computes the maximum achievable energy

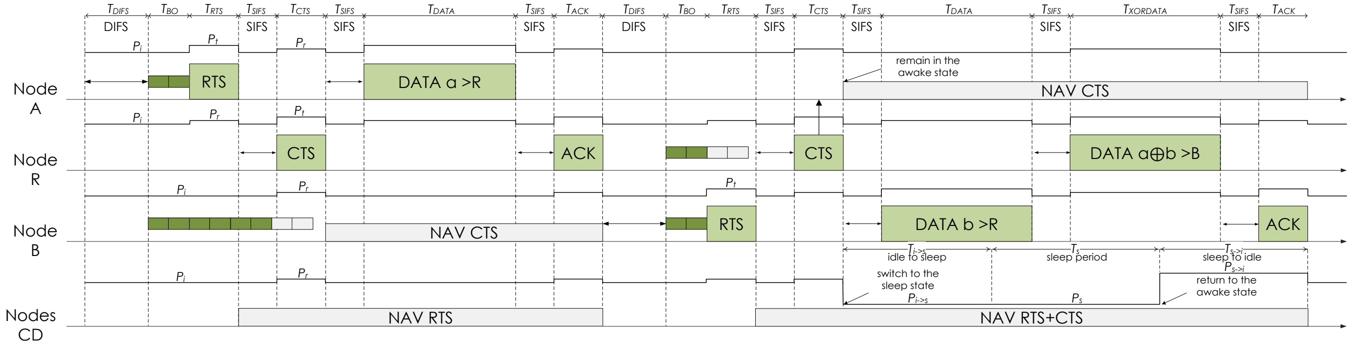

Fig. 2. Example of operation of GreenCode in the cross network when nodes A and B exchange two data packets through node R. Nodes C and D can sleep during the RD exchange between nodes B and R, while node A remains awake to overhear (and decode) the coded data transmission from node R to node B.

efficiency of GreenCode.

A. System Model

The reference scenario considered for the analysis of GreenCode is the canonical cross network shown in Fig. 1. In this network, nodes A, B, C, and D act as pairs of independent sources/destinations and node R acts as a common relay for all of them. All the nodes are equipped with IEEE 802.11n radio interfaces enabling ad hoc communication mode with a single omnidirectional antenna for communications, i.e., a Single-Input Single-Output (SISO) communications system. The source nodes are able to detect the transmission signals of their respective destination nodes (i.e., they can perform carrier sensing). However, they cannot receive any packets correctly from transmitters at two-hop distance. They can only receive packets with no errors from transmitters at one-hop distance. Node R only forwards the received data packets from the sources to their respective destinations. Hence, this scenario can be modeled as N source nodes (N=4) and a relay node.

B. Assumptions

We assume that the considered network is at the best-case scenario. In any transmission cycle, one single node is active and always has a data packet ready to be transmitted. This node contends alone for channel access and waits an average backoff time before transmitting. Other nodes can only receive data packets and respond with ACK or data packets, hence avoiding collisions. All data packets have constant byte length and are transmitted over an ideal wireless channel. Therefore, no packet errors occur due to channel variations and the propagation delay is neglected. In addition, coding/decoding XOR operations require no extra time and energy.

C. System Parameters

The system parameters that have been used to analyze the energy efficiency of GreenCode and their variables are defined as follows. The average backoff time and DIFS and SIFS intervals are denoted by T_{BO}, T_{DIFS}, and T_{SIFS}, respectively. The transmission times of RTS, CTS, data, coded data, and ACK packets are expressed as T_{RTS}, T_{CTS}, T_{DATA}, $T_{XORDATA}$, and T_{ACK}, respectively. The byte-length of the frame body or MAC Service Data Unit (MSDU) contained in a data packet is referred to as L_{MSDU}. We also denote P_t, P_r, P_i, and P_s as the power consumed by the radio interface of a node when transmitting, receiving or overhearing, idle channel listening, and sleeping, respectively. The radio transition times from idle (awake) to sleep and vice versa are referred to as $T_{i\to s}$ and $T_{s\to i}$, respectively. The power consumptions of these radio transitions are denoted as $P_{s\to i}$ and $P_{i\to s}$, respectively.

D. Energy Efficiency and Energy Consumption

The energy efficiency of a given protocol x (η_x) is defined herein as the amount of bits contained in an MSDU divided by the energy consumption required to deliver a data packet that includes the MSDU from end to end of the network:

$$\eta_x = \frac{8 \cdot L_{MSDU}}{E_x} \tag{1}$$

where E_x is defined as the product of power consumption and time spent in transmission over the total amount of data packets delivered from source to destination through a relay.

The expression of E_x to compute the energy efficiency of GreenCode is presented and explained as follows.

As shown in Fig. 2, the exchange of a pair of data packets from end to end in GreenCode includes two channel accesses. In the first channel access, a unidirectional data transmission is performed from a source node to the relay node using standard DCF rules, where all overhearing nodes remain awake. In the second channel access, a bidirectional data exchange is performed between a source node and the relay node using reactive RD rules, in which the forward transmission contains a normal data packet and the backward transmission carries a coded data packet. During the RD exchange sequence, some overhearing nodes, except one, can sleep. Thus, given N source nodes communicating into pairs around the relay node, $\frac{N}{2}$ unidirectional data transmissions and $\frac{N}{2}$ bidirectional coded and non-coded data transmissions are required in GreenCode to forward N data packets from end to end (in both directions).

The duration of a unidirectional data transmission comprises a DIFS interval, an average backoff period, the RTS, CTS, data, and ACK transmissions, and three SIFS intervals. All the nodes (N+1) consume energy to listen to the wireless channel for the DIFS interval, the average backoff period, and all the SIFS intervals. The sender and receiver nodes, a source node and the relay node or vice versa, consume energy to transmit and receive the RTS and data packets and to

$$\begin{aligned}
E_{GreenCode}&=\tfrac{1}{N}\left(E_t+E_r+E_i+E_{sw}+E_s\right)\\
E_t&=\left(N\left(T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}\right)+\tfrac{N}{2}T_{XORDATA}\right)P_t\\
E_r&=\left(N\left(N-1\right)T_{RTS}+N^2T_{CTS}+\tfrac{N}{2}\left(N-1+1\right)T_{DATA}+\tfrac{N}{2}ST_{XORDATA}+\tfrac{N}{2}\left(N+1\right)T_{ACK}\right)P_r\\
E_i&=\left(N\left(N+1\right)\left(T_{DIFS}+T_{BO}\right)+\tfrac{N}{2}\left(4\left(N+1\right)+3\left(S+1\right)\right)T_{SIFS}+N\left(T_{RTS}+T_{DATA}\right)+\tfrac{N}{2}T_{ACK}\right)P_i\\
E_{sw}&=\tfrac{N}{2}\left(N-S\right)\left(T_{i\to s}P_{i\to s}+T_{s\to i}P_{s\to i}\right)\\
E_s&=\tfrac{N}{2}\left(N-S\right)T_sP_s,\, where\ T_s=T_{DATA}+T_{XORDATA}+T_{ACK}+3\cdot T_{SIFS}-\left(T_{i\to s}+T_{s\to i}\right)
\end{aligned} \tag{2}$$

receive and transmit the CTS and ACK packets, respectively. The rest of source nodes consume energy to overhear all the transmitted packets, except one that only overhears the CTS and ACK transmissions when the relay node acts as a receiver (or the RTS and data transmissions when the relay node is transmitting data). Such node also consumes energy for idle channel listening during the RTS and data transmissions when a source node acts as a transmitter (or the CTS and ACK transmission when a source node is receiving data).

The duration of a bidirectional coded data transmission adds a coded data transmission and a SIFS interval to the duration of a unidirectional data transmission. For easy comprehension, the energy consumption of GreenCode during a bidirectional coded data transmission is described in the following points:

1) Transmission period: A source node consumes energy to transmit the RTS, data, and ACK packets to the relay node. Then, the relay node consumes energy to transmit the CTS packet to the destination node pair of the transmitting source node and the coded packet to the transmitting source node.

2) Reception period: $N-S$ overhearing source nodes only consume energy to overhear the RTS and CTS packets as they can go to sleep to save energy. S denotes the number of active source nodes and S=2. The transmitting source node consumes energy to overhear the CTS packet addressed to its destination node and to receive the coded data packet from the relay node. Then, the relay node consumes energy to receive the RTS, data, and ACK packets from the transmitting source node. Finally, the destination node consumes energy to receive the CTS and overheard coded data packets from the relay node.

3) Idle period: The $N+1$ nodes consume energy to listen to the DIFS interval, the average backoff period, and a SIFS interval. Then, $S+1$ nodes consume energy to listen to the remaining SIFS intervals. Also, the destination node consumes energy for idle channel listening during the RTS, data, and terminating ACK transmissions.

4) Switch period: The $N-S$ overhearing source nodes consume energy during the transition from idle to sleep and during the transition from sleep to idle.

5) Sleep period: The $N-S$ overhearing source nodes can sleep during the RD exchange sequence expect for when they switch between idle and sleep states. The sleep operation is feasible provided that the sleep period (T_s) is greater than zero. Otherwise, none of the overhearing source nodes can sleep.

Based on the explanations given above, the energy consumption of GreenCode ($E_{GreenCode}$) can be split into five energy consumption components: transmitting (E_t), receiving and overhearing (E_r), idle channel listening (E_i), switching between idle and sleep states (E_{sw}), and sleeping (E_s). Therefore, it is possible to compute $E_{GreenCode}$ as (2), and its energy efficiency by (1).

IV. PERFORMANCE EVALUATION

The energy efficiency of GreenCode has been evaluated by means of the analysis presented in the previous section and computer-based simulations. The results are presented and discussed in this section. In addition, and for the purpose of comparison, the performance of GreenCode has been compared to those of the DCF and DCF with NC, referred to as DCF+NC hereafter. The performance results of these protocols have been presented in our previous works [8], [9].

A. Simulation Scenario and Setup

Computer-based simulations have been conducted in a proprietary event-driven simulator coded in Python, where the protocol rules have been implemented. We have considered the RTS/CTS exchange to simulate the operations of all the protocols. The simulation scenario has been based on the system model and assumptions described in the previous section. However, in the simulation, collisions of packets due to identical randomly selected backoff counters have been taken into account, following an Extended Interframe Space (EIFS). Data packets have been generated following a Poisson distribution by which packets are generated on average at a given rate but the packet generation time is random. Then, the packet generation rate has been increased up to saturation, i.e., when all nodes always have data packets to send.

All simulation results have been averaged over 10 simulation runs for the duration of 20 s each. The confidence intervals have been obtained with a confidence level of 95% derived by the method of replication and the width of the confidence intervals has been 2% of the mean value. Therefore, they have been omitted in the figures for the sake of visualization.

B. System Parameters

We have selected the Extended Rate PHY layer (ERP) specification with Orthogonal Frequency Division Multiplexing (OFDM) modulation for SISO communications, defined in the IEEE 802.11n amendment of the Standard. This specification provides 8 transmission rates ranging from 6 to 54 Mbps with Number of Data Bits Per OFDM Symbol (N_{DBPS}) from 24 to 216, respectively. Note that RTS and data transmissions can be performed using any of these rates whereas CTS and ACK packets must be transmitted at the basic rates 6, 12, and 24 Mbps, as specified by the basic rate selection rules in [3].

TABLE I
SYSTEM PARAMETERS

Parameter	Value	Parameter	Value
T_{slot}	9 μs	L_{serv}	16 b
T_{SIFS}	10 μs	L_{tail}	6 b
T_{DIFS}	28 μs	L_{RTS}	20 B
T_{EIFS}	88 μs	L_{CTS}=L_{ACK}	14 B
CW_{min}, CW_{max}	15, 1023	L_{MAChdr}	30 B
T_{BO}	67.5 μs	L_{XORhdr}	40 B
T_{pre}	16 μs	L_{FCS}	4 B
T_{sig}	4 μs	P_t	1.65 W
T_{sym}	4 μs	P_r	1.4 W
T_{sigEx}	6 μs	P_i	1.15 W
T_h	10 ms	P_s	0.045 W

The expression to compute the transmission time of each packet using the ERP-OFDM PHY layer is given in [3] as

$$T_x{=}T_{pre}{+}T_{sig}{+}T_{sym}\left\lceil \frac{L_{serv}{+}8 \cdot L_x{+}L_{tail}}{N_{DBPS}} \right\rceil{+}T_{sigEx} \quad (3)$$

where x is the packet type and all the variables and their values are provided in Table I. The MAC packet length is referred to as L_x. A data packet includes the MSDU together with a MAC header (L_{MAChdr}) and a Frame Check Sequence (FCS), L_{FCS}. Also, an XOR header (L_{XORhdr}) is added after the MAC header in coded data packets, as specified in [2]. For instance, for an MSDU of 1500 bytes and PHY RTS/data and CTS/ACK transmission rates of 54 and 24 Mbps, respectively, T_{RTS}, T_{CTS}, T_{DATA}, $T_{XORDATA}$, and T_{ACK} are obtained by (3) as 30, 34, 254, 262, and 34 μs, respectively.

In Table I, we also provide other variables that are computed as follows. T_{DIFS} is calculated as $T_{DIFS}{=}T_{SIFS}{+}2T_{slot}$ and T_{EIFS} as $T_{EIFS}{=}T_{DIFS}{+}T_{SIFS}{+}T_{ACK}\,(6Mbps)$. Since we consider no collisions, T_{BO} is obtained by the minimum CW size (CW_{min}) and the slot time (T_{slot}) as $T_{BO}{=}\left(\frac{CW_{min}}{2}\right)T_{slot}$ and has only been used to plot the analytical results. For the same reason, the EIFS interval and the maximum CW size (CW_{max}), which are related to collisions and retransmissions, has only been used to plot the simulation results. This is also the case for the holding time (T_h) used in the DCF+NC and GreenCode protocols. The values of power consumption have been taken from [5]–[7].

Regarding the awake/sleep radio transitions, we make the following observations, based on the works in [5]–[7]: *i)* $T_{i\to s}$ has shown to be similar to $T_{s\to i}$, *ii)* $P_{i\to s}$ has shown to be much lower than P_s, and *iii)* $P_{s\to i}$ has shown to be much higher than P_i. Therefore, we consider that: *i)* $T_{i\to s}$ is equal to $T_{s\to i}$, *ii)* $P_{i\to s}$ is equal to P_s, and *iii)* $P_{s\to i}$ is modeled as αP_i, where α is defined as the wakeup radio transition coefficient between idle and sleep states and $\alpha > 1$.

C. Results

The results presented in the figures have been plotted for an MSDU length of 1500 bytes, PHY control and data rates of 54 and 24 Mbps, wakeup radio transition coefficient of α=1.5 ($P_{s\to i}$=1.725 W), and awake/sleep radio transition time of 250 μs each radio transition, when each parameter has been fixed.

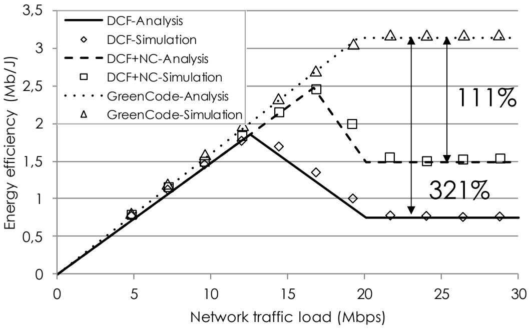

Fig. 3. Energy efficiency versus total offered traffic load.

In Fig. 3, we evaluate the impact of variable traffic load from low to saturation on the energy efficiencies of the protocols. We can see that GreenCode outperforms the DCF and DCF+NC for medium to high traffic loads by a maximum gain of up to 321% and 111%, respectively, under saturation. One reason for this outstanding gain is related to the long-term MAC fairness for all contending nodes. In DCF and DCF+NC, the relay node can only access the wireless channel for a portion of time that is not sufficient to send out packets with the same rate as they arrive, when the source nodes transmit at high rates. For this reason, their energy efficiencies increase up to a maximum value before reaching saturation and then significantly decrease down to a minimum stable value under saturation. In contrast, this behavior is not present in GreenCode because the relay node can obtain a much higher share of the wireless channel through reactive RD transmissions involving both coded and non-coded packets, hence matching incoming and outgoing packet rates. Another reason is that, unlike DCF and DCF+NC, GreenCode allows some overhearing nodes not involved in transmission to sleep during bidirectional coded data transmissions, hence saving energy and increasing the energy efficiency.

Figs. 4 and 5 show the effects of the MSDU length and PHY data rate on the saturation energy efficiencies of the protocols. It can be seen that the protocols increase their energy efficiencies for longer MSDU lengths or faster rates since the amount of information conveyed in each data transmission increases or the time to transmit a data packet decreases. GreenCode achieves the highest energy efficiency for all MSDU lengths and data rates. The results of Fig. 4 indicate that the highest gain of GreenCode (e.g., 331% for 250 bytes) is achieved for small packets because the data transmission time has a lower impact on the overall transmission time. Then, the gain decreases as the MSDU length increases until the packet length is sufficiently long to enable the sleep period. This corresponds to an MSDU length of 1250 bytes, where the achieved gain is 319%. For longer MSDU lengths, the gain increases up to 325% for 2250 bytes since the packet transmission time increases and nodes can sleep longer, hence saving more energy. In addition, the results of Fig. 5 reveal that GreenCode can achieve the highest gains for lower data rates (e.g., 338% for 9 Mbps) because the data transmission time becomes longer and the sleep period increases. Then, the gain (e.g., 326% for 36 Mbps) decreases as the data rate increases.

To conclude, we investigate the influence of the duration and power consumption of the awake/sleep (on/off) radio transitions on the saturation energy efficiency of GreenCode

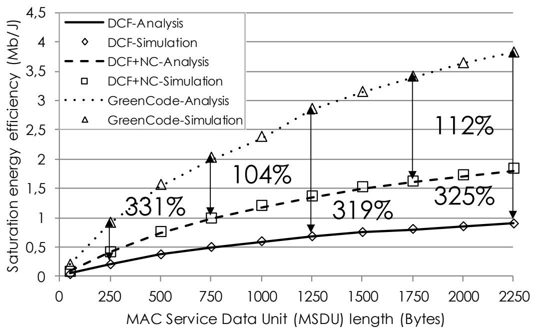

Fig. 4. Saturation energy efficiency versus MSDU length.

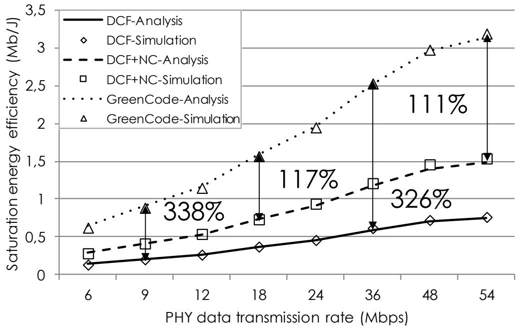

Fig. 5. Saturation energy efficiency versus PHY data transmission rate.

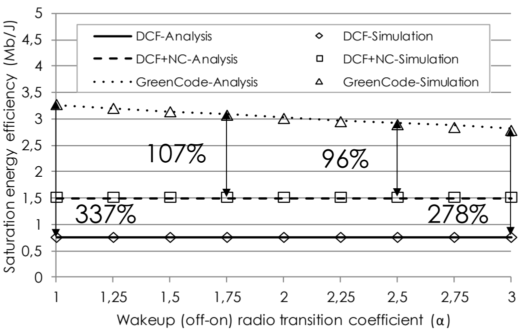

Fig. 6. Saturation energy efficiency versus wakeup radio transition coefficient.

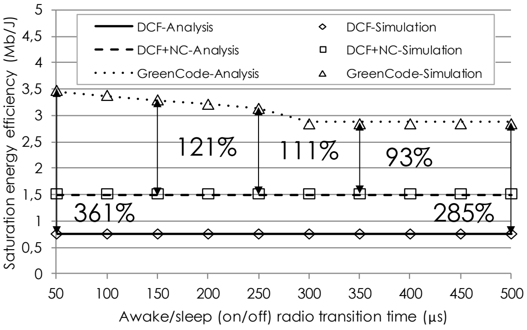

Fig. 7. Saturation energy efficiency versus awake/sleep radio transition time.

in Figs. 6 and 7. These values are predetermined by the radio hardware design and are critical for the proper operation of GreenCode. The stable energy efficiencies of DCF and DCF+NC under saturation are also shown in the figures for the purpose of comparison with the energy efficiency of GreenCode. Fig. 6 shows that GreenCode achieves a lower energy efficiency with gains from 337% down to 278% as the wakeup radio transition coefficient (α) increases from α=1 to α=3. The reason is that nodes consume more energy during the transitions from sleep to idle (awake). In Fig. 7, we observe that the energy efficiency of GreenCode decreases with gains from 361% down to 285% as the radio transition time between idle and sleep states increases from 50 μs to 250 μs. In this case, the reason is that the sleep period becomes shorter and nodes consume more energy. The critical value that makes the sleep period be equal or lower than zero is 300 μs. For radio transition times above this value, the high energy efficiency of GreenCode remains constant because none of the nodes can go to sleep. These results show that the impact of the awake/sleep radio transitions on the energy efficiency of GreenCode will depend on the MSDU length and PHY data rate used.

V. CONCLUSIONS

A new NC-aware energy-efficient MAC protocol, called GreenCode, has been proposed in this paper to increase the energy efficiency of wireless networks based on the IEEE 802.11 Standard. Its operation enables packet-based sleeping periods and receiver-initiated bidirectional transmissions being aware of the NC approach, based on the combination of the IEEE 802.11ac TXOP PSM and IEEE 802.11n RDP. The energy efficiency of GreenCode has been evaluated via analysis and computer-based simulations in a simple (cross) network topology composed of four source nodes and a relay node. The results presented in this paper have shown the high energy efficiency of GreenCode for all the evaluated parameters when compared to those of the DCF with and without NC. More specifically, GreenCode is particularly suited for medium-high traffic loads, short/long transmitted packets, and low data transmission rates. For instance, the maximum gains vary from 350% to 298% and from 319% to 325% as the packet length increases and from 340% to 321% as the data rate increases.

Motivated by the promising results presented in this paper, ongoing work is aimed at evaluating the energy efficiency of GreenCode in more complex network topologies.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," in *ACM SIGCOMM*, 2006, pp. 243–254.

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

[4] R. Chandanala and R. Stoleru, "Network Coding in Duty-Cycled Sensor Networks," in *IEEE INSS*, 2010, pp. 203–210.

[5] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997.

[6] P. J. M. Havinga and G. J. M. Smit, "Energy-efficient TDMA medium access control protocol scheduling," in *AMOC*, 2000, pp. 1–10.

[7] E.-S. Jung and N. H. Vaidya, "An Energy Efficient MAC Protocol for Wireless LANs," in *IEEE INFOCOM*, vol. 3, 2002, pp. 1756–1764.

[8] R. Palacios, H. Haile, J. Alonso-Zarate, and F. Granelli, "Analysis of a Network Coding-Aware MAC Protocol for IEEE 802.11 Wireless Networks with Reverse Direction Transmissions," in *IEEE GLOBECOM*, 2014, pp. 1218–1224.

[9] R. Palacios, D. Kliazovich, and F. Granelli, "Reverse Direction Transmissions and Network Coding for Energy-Efficient Wi-Fi Networks," in *IEEE CAMAD*, 2014, pp. 100–104.