802.11ec: Collision Avoidance Without Control Messages

Eugenio Magistretti, *Student Member, IEEE*, Omer Gurewitz, *Member, IEEE*, and Edward W. Knightly, *Fellow, IEEE*

***Abstract*—In this paper, we design, implement, and evaluate 802.11ec (Encoded Control), an 802.11-based protocol *without* control messages: Instead, 802.11ec employs correlatable symbol sequences that, together with the timing the codes are transmitted, encode all control information and change the fundamental design properties of the MAC. The use of correlatable symbol sequences provides two key advantages: 1) efficiency, as it permits a near order of magnitude reduction of the control time; 2) robustness, because codes are short and easily detectable even at low signal-to-interference-plus-noise ratio (SINR) and even while a neighbor is transmitting data. We implement 802.11ec on a field programmable gate array (FPGA)-based software defined radio. We perform a large number of experiments and show that, compared to 802.11 (with and without RTS/CTS), 802.11ec achieves a vast efficiency gain in conveying control information and resolves key throughput and fairness problems in the presence of hidden terminals, asymmetric topologies, and general multihop topologies.**

***Index Terms*—Channel allocation, collision avoidance, communication system signaling, correlation, IEEE 802.11 standards, wireless LAN.**

I. Introduction

MAC CONTROL messages are essential: For example, ACKs convey correctly received data, and RTS/CTS exchange can significantly mitigate hidden terminal collisions. However, even though the information conveyed in MAC control messages is small, their duration can be quite long, as in addition to the control information, they also need to include source/destination address, message type, etc., all of which are transmitted at base rate to improve the likelihood that they can be successfully decoded. For example, an ACK message is 14 B plus physical-layer encapsulation, but contains only one bit of relevant information (that DATA was successfully received). Likewise, RTS/CTS is rarely used in practice precisely due to excessive overhead despite its important role in mitigating collisions.[1]

In this paper, we design, implement, and evaluate 802.11ec (Encoded Control) as a control-message-free MAC. Instead of control messages, 11ec employs correlatable symbol sequences (CSSs) that, together with their transmission timing, convey all control information and change the fundamental design properties of the MAC. For example, 11ec replaces an 802.11 ACK *message* with a predefined ACK CSS that can be correlated instead of decoded, thereby vastly reducing its duration and dramatically improving its robustness by enabling its reception at low signal-to-interference-plus-noise ratio (SINR).

Control information can be classified along two dimensions: first, as to whether or not the information in the message can be represented from a small dictionary or codebook. For example, a small dictionary can encode the three different control messages used in 802.11 for data exchange (RTS, CTS, and ACK). Likewise, while the space of all MAC addresses is large (seemingly precluding a small dictionary), each node communicates with only a limited number of addresses at a time. Thus, both MAC addresses and control message type can be encoded from a small dictionary. Second, control information can further be classified according to whether they are necessarily public or can be private. For example, for correctness of the protocol, all nodes must know that a CTS should cause them to defer, i.e., this message must be public; on the other hand, only a data sender need know that its data was correctly received, i.e., its ACK may be private.

802.11ec's key techniques are twofold: First, we use a dictionary of correlatable symbol sequences to convey control information that can be represented by a limited dictionary. For example, instead of CTS that contains physical-layer preamble, frame control sequence, type field, frame check-sum, destination address, duration field (as well as it incurs a T_{SIFS} delay), we transmit a short (e.g., 127 symbols) CSS from a small dictionary to convey that it is a CTS. For an 802.11a physical layer, we show that this reduces the time to convey the control information by nearly an order of magnitude, from 60 to 6.35 μs. Second, we show that the information that cannot be represented by a limited dictionary can be conveyed via CSS timing. For example, nodes overhearing the 802.11 CTS message need to defer for an amount of time as specified by the CTS duration

[1] We define a message at the MAC layer to be a MAC-layer frame, whether a control frame such as RTS or a data frame preceded by a header.

Manuscript received November 25, 2012; revised May 31, 2013; accepted September 29, 2013; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. Javidi. Date of publication November 13, 2013; date of current version December 15, 2014. This work was supported by the NSF under Grants CNS-1126478 and CNS-1012831 and Cisco Systems, Inc., under a grant. The work of O. Gurewitz was supported by in part the EU FP7-FLAVIA project under Contract No. 257263.

E. Magistretti was with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: emagistretti@rice.edu).

O. Gurewitz is with the Department of Communication Systems Engineering, Ben Gurion University of the Negev, Beer Sheva 84105, Israel (e-mail: gurewitz@cse.bgu.ac.il).

E. Knightly is with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: knightly@ece.rice.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2013.2288365

field contained in the message. We show that 11ec nodes can instead simply defer until a channel-clear CSS is transmitted by the receiver (or until a timeout).

802.11ec's second technique is to distinguish between public and private information. Namely, 11ec only uses public CSSs for information that is required to be public, such as conveying channel reservation and channel clear. On the other hand, address fields need not be public, as the identity of the sender and receiver need not be known by other nodes. 11ec ensures that private control information, including addresses and ACKs, is not correlated by other nodes. This has the potential to thwart eavesdroppers not only from decoding data (as data can be encrypted), but even from knowing which nodes are communicating with each other; we show how all private control information, including addresses can only be correlated by the intended receiver.[2]

802.11ec enhances robustness in two ways. First, control information is more likely to be received in 11ec because control information is conveyed in short CSSs that are correlatable even at low SINR. For example, because 802.11ec replaces an ACK message with a CSS, 11ec ACKs are more robust and can be received even in the presence of transmitting interferers. Second, 802.11 is "fragile" to topological factors in that while 802.11 DCF without RTS/CTS yields high performance in fully connected wireless LANs [5], hidden terminals, asymmetric topologies, and general multihop topologies can yield severe throughput degradation and unfairness [6]. These latter topologies are becoming increasingly common because of device power asymmetries, e.g., between access points (APs), laptops, and popular smartphones, and of the wider coverage achievable with the adoption of subgigahertz frequencies, including TV white spaces [17], [20]. While use of RTS/CTS can significantly improve throughput in such challenged topologies, the additional overhead of RTS/CTS can sometimes overwhelm this improvement. Moreover, in fully connected topologies, RTS/CTS degrades throughput due to its unnecessary overhead. In contrast, 11ec overcomes these limitations through robust and short-duration control signals, i.e., 11ec minimally penalizes station throughput thus allowing to enable channel reservation independently of the network topology. Consequently, 11ec stations have vastly increased opportunities to obtain channel access thereby dramatically improving the network's fairness in throughput distribution.

We implement correlatable symbol sequences in a software defined radio, perform a large set of experiments, and study issues that have not been experimentally investigated previously. We find a correlatable symbol sequence *length* that simultaneously: 1) provides sufficient physical-layer robustness; 2) limits communication overhead; and 3) supports large networks. Specifically, we first investigate the tradeoffs between sequence length and physical-layer robustness and show that even short sequences, e.g., 127-symbol or 6.35 μs long, can be detected at -6 dB SINR with only 5% false negatives. We demonstrate that our encoded sequences can be detected at an SINR 10 dB lower than 802.11 control messages. Second, we show that 127-symbol code lengths can support more than 50 co-located nodes, with minimal penalty on detection errors.

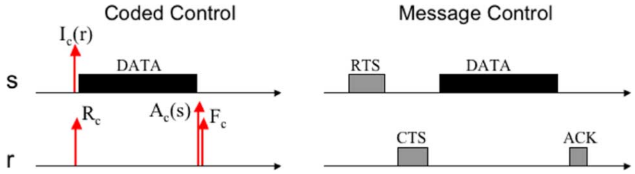


Fig. 1. Timeline of a packet exchange with Coded Control versus Message Control.

Finally, we implement 11ec in a measurement-driven emulator, whose inputs are channel measurements collected in a real deployment and real card performance parameters (e.g., bit error rate (BER) and multiple supported modulations). We compare 11ec's performance to 802.11 with and without RTS/CTS. We examine a wide set of basic topologies that are at the origin of throughput losses and/or imbalances in 802.11-based networks in order to provide an insight in understanding the performance of larger networks. Our finding is that 11ec can dramatically reduce throughput imbalances by improving Jain's index [13] by up to 90%. Moreover, while such a fairness improvement can often decrease total utilization, 11ec increases channel utilization by more than 10% via the use of short encoded control that simultaneously decreases vulnerability intervals and control overhead. We also study a larger topology and show that 11ec can improve the throughput of an under-served flow by a factor of over 22 folds. Over all flows, we improve Jain's index by up to 173% while also improving the channel utilization by up to 46%. Finally, we simulate larger 20-node topologies and show that 11ec highly benefits the lowest throughput links, by at least doubling the throughput of more than 49% of them.

The remainder of the paper is organized as follows. Section II discusses coded control CSSs and the design of 802.11ec. Section III includes a thorough experimental evaluation of CSSs using a software defined radio platform. Section IV investigates the benefits of 802.11ec in a measurement-driven emulator. Finally, Sections V and VI overview related works and conclude the paper.

II. MAC Protocol Design

11ec collision avoidance realizes and improves on the collision avoidance mechanism of IEEE 802.11 DCF with RTS/CTS, reduces the overhead by nearly an order of magnitude, and practically eliminates collisions, even in hidden terminal topologies. Specifically, 11ec retains the four-way handshake suggested by 802.11, where control messages are replaced by very short correlatable symbol sequences (see Fig. 1). It is important to note that: 1) the duration of each correlatable symbol sequence is nearly zero; 2) the duration between the start of the reservation signal until the data transmission is negligible, hence practically invulnerable to collisions.

In this section, we define CSSs and explain how control messages can be turned into CSSs. Furthermore, we show that CSSs are a key element for the realization of 11ec efficiency and robustness. The second part of the section provides a detailed description of 11ec including protocol primitives and an analysis

[2] While development of a complete privacy protocol is beyond the scope of this work, 11ec provides important mechanisms to design such a protocol.

of hidden terminal vulnerability leading to novel collision reduction opportunities.

A. Coded Control Information Versus Message Control Information

CSSs: Correlatable symbol sequences are predefined pseudo-noise binary codewords; namely, while codewords are deterministically generated, they retain the statistical properties of a sampled white noise. For this reason, the cross correlation of any such sequence with a matching copy obtains spike values, while it appears random to a listener without prior knowledge of the codeword. An example of a CSS is the 802.11 preamble used for packet detection, symbol synchronization, and radio parameter tuning.[3]

The CSS detection process via cross correlation enjoys three key advantages over data decoding. First, cross correlation obtains a large processing gain even for small codewords (e.g., the 802.11 preamble used for detection is 64 symbols), which permits reliable detection even at low SINR. Second, differently from decoding, detection is highly robust to imperfect radio parameter tuning, and thus a codeword does not need to be preceded by a preamble. For these reasons, a CSS can be short; for example, in our implementation, 11ec utilizes 127-symbol codewords that can be transmitted in 6.35 μs. Third, detection is almost instantaneous as no decoding is needed. For example, 802.11 intercontrol message time is at least $T_{\text{SIFS}} = 16\ \mu$s, including about 14 μs of data processing, while in 11ec no control message processing is required; hence, 11ec reduces substantially the short inter-CSS time.

Encoded Control: While consuming a significant amount of airtime, 802.11 control messages usually convey little information. For example, an ACK occupies the medium for up to 60 μs, i.e., an airtime sufficient to transmit 3240 bits at 54 Mb/s, while it contains a single bit of relevant information. 11ec replaces control messages with CSSs, which permit to shorten the transmission duration of nearly an order of magnitude to 6.35 μs, while retaining the information content.

According to the 802.11 standard [1], RTS, CTS, and ACK control messages may include up to four information fields: destination address, sender address, duration, and frame control (a fifth field is the frame check-sum that protects the other four). In particular, the frame control field is a 2-B-long sequence of bits representing specific control parameters. The values of most control bits are fixed for control messages; only frame subtype (4 bits) and station power management flag (1 bit) can assume different values. However, the latter does not convey novel information when used in control messages.

In order to represent the information content of the control messages as described above, 11ec considers the size of the dictionary needed to represent such information. Information that can be expressed by a small dictionary is conveyed using CSSs, while information that needs large dictionaries is conveyed with timing codes. First, in 802.11 data exchange, the type field that distinguishes the control messages may assume only three values, i.e., RTS, CTS, and ACK; 11ec conveys the type by associating each message with distinct CSSs. Second, 802.11 control messages include addresses (sender and/or receiver). Since the number of nodes a station communicates with at a time is generally small, i.e., can be represented by a small dictionary, 11ec integrates the addresses in CSSs, i.e., a single CSS may represent the combination of a control type and a specific address. For example, in 802.11 the RTS includes the address of the intended receiver; accordingly, in 11ec, RTS addressed to different receivers are represented by distinct CSSs. Third, some control messages include a duration field that cannot be represented by a small dictionary. For this information, 11ec utilizes a combination of time codes and new control types. For example, 11ec nodes reserve the channel for the duration of a data reception, by transmitting two CSSs corresponding to channel reservation (immediately before the reception) and release (immediately after). The potential interferers do not access the channel during the interval between the two CSSs (or before a timeout expires), i.e., effectively implement a form of virtual carrier sensing.[4]

Public CSSs Versus Private CSSs: A second dimension of control messages, and of the corresponding CSSs, is whether they can be private, i.e., carry information relevant only to a specific destination (e.g., acknowledgments), or are necessarily public, i.e., meant to be heard by all neighbors of a node (e.g., channel reservation/release). Accordingly, in the case of a private CSS, only the intended receiver possesses a copy of the correlatable symbol sequence, and thus can correctly detect it; conversely, all nodes possess copies of the public CSSs and can detect them. For example, only the data sender needs to cross-correlate its private acknowledgment CSS, while all nodes must cross-correlate public channel reservation/release CSSs.

Control During Data: Because correlatable symbol sequences can be detected even at subnoise SINR (e.g., 11ec 127-symbol CSSs can be detected at -6 dB with high reliability), 11ec nodes attempt detection even while receiving data. This technique provides a signaling mechanism effective even in cases in which the receiver is subject to long periods of noisy channel, or undesired data overhearing. In 802.11, if a node receives an RTS while also overhearing data, the node cannot decode the RTS and therefore cannot respond. In contrast, 11ec uniquely enables a node to receive a CSS signaling that another node is requesting to communicate. Therefore, the receiver can send a Request for RTS (RRTS) CSS to reserve the medium when it becomes free and initiate a data exchange [4]. Likewise, because ACKs are also encoded, they can be correctly received even if an interfering terminal is simultaneously transmitting data.

B. 11ec Channel Reservation Primitives

Wireless MAC protocols perform collision avoidance by silencing the medium in the vicinity of a transmitting link via channel reservation. Channel reservation fundamentally hinges on three key mechanisms: 1) *initiation*, performed by the node that initiates the exchange to request the cooperation of the other endpoint to reserve the channel; 2) *reservation*, performed to inform nodes potentially hindering the exchange; and 3) *deferral*, performed by the surrounding terminals in order to avoid

[3] A detailed discussion of signal correlation can be found in literature [11], [15], [23]. A short introduction is in the Appendix.

[4] Another alternative is to discretize the duration and associate different CSSs to each discrete value.

disturbing ongoing transmissions. 802.11 implements the three mechanisms via: a) RTS; b) CTS and data packet—the latter realizes channel reservation in the vicinity of the sender; and c) NAV and carrier sensing. When RTS/CTS is disabled, during the data transmission the medium is reserved exclusively in the vicinity of the sender. In the following, we show how 11ec implements these three mechanisms via CSSs and timing codes.

A key concept of 11ec channel reservation is very short channel reservation negotiation for near immunity to interruptions, e.g., collisions and capturing by other nodes. Specifically, 11ec channel reservation is based on three basic primitives (the subscript c indicates CSSs).

Initiation: $I_c(r)$*:* In 11ec, a sender wishing to start a data exchange performs virtual, and optionally physical, carrier sensing. If the medium is free, the sender waits for a backoff interval similar to 802.11 and then transmits a sender side channel request primitive, in short $I_c(r)$, to request the receiver r to reserve the channel. $I_c(r)$ need only be detected by r and not necessarily by the neighboring nodes; thus, we implement it as a private CSS. In order to convey the identity of the receiver r (as 11ec does not transmit the traditional MAC address), 11ec implements $I_c(r)$ via several CSSs and associates a distinct CSS with each receiver; i.e., when a sender needs to contact a receiver, it uses the receiver's $I_c(r)$. Nodes in the vicinity of the sender do not detect the initiation $I_c(r)$.

Reservation: R_c*:* A node r receiving an $I_c(r)$ checks if other nodes are communicating in its vicinity and may hinder its reception. If that is not the case, r immediately transmits a channel reservation primitive R_c to notify potential interferers. In order to realize the reservation, R_c should be detected by all nodes in the vicinity of the receiver r, and it is therefore transmitted via a public CSS.

In addition to channel reservation that forces neighbors to defer, R_c implicitly communicates to the transmitter that the channel is available and that data can be transmitted. Instead of providing a distinct CSS to convey the sender address, as in the previous case of the $I_c(r)$, we employ a simple temporal code: Since a receiver r transmits R_c immediately after $I_c(r)$, the sender, in contrast to the other neighboring nodes, interprets the reception of a R_c as an authorization to begin a data transmission.

Deferral: $R_c \rightarrow F_c$*:* 11ec implements the deferral and conveys its duration via a combination of CSSs and a simple time code. Specifically, after data reception and acknowledgment, the receiver explicitly releases the channel with a channel-free primitive F_c. Thus, nodes receiving R_c need to wait to receive an F_c (or wait a predefined timeout) before accessing the channel; practically, this procedure represents a form of virtual carrier sensing. Because all neighboring nodes need to receive F_c, 11ec implements F_c as a public CSS.

11ec further increases the robustness of R_c/F_c messages by pairing them. Our technique is based on associating a small number of CSS pairs to distinct R_c^i/F_c^i pairs; a receiver randomly picks and transmits any such pair to reserve and free the channel. This feature is particularly useful for a node located in the neighborhood of several receivers that may be active simultaneously, i.e., the receivers may send R_c and F_c that denote overlapping intervals. In that case, the node can still correctly determine the state of the medium in each moment by associating each overheard reservation R_c^i to its corresponding F_c^i.

Finally, we define an acknowledgment primitive, $A_c(s)$ (see Fig. 1), and we implement it as a private CSS associated to each sender s. Few recently proposed packet forwarding schemes leverage the ACK exchange for additional purposes other than data acknowledgment, e.g., network coding [8] and routing [9]. In such cases, 11ec can revert to 802.11 ACKs with small overhead penalty (the major gain of 11ec both in throughput and robustness derives from the novel channel reservation scheme). Note that CSMA/CN [23] uses signatures for acknowledgment, which are similar to our CSSs.

CSS Association: The Initiation and Acknowledgment primitives $I_c(r)$ and $A_c(s)$ require association of unique CSSs to specific nodes. While a detailed investigation of the issue is beyond the scope of this paper, we suggest two simple mechanisms that can be used for this purpose. The first leverages the solution already existing in the 802.11 standard, in which the AP assigns an identifier (AID) upon station association [1]; AIDs can be easily mapped to CSSs. In order to initiate the association, the stations may use a reserved CSS. The second mechanism utilizes multiple hash functions based on the station MAC address. In case of assignment conflicts, which can be easily detected using the address fields in the header of the data frames, the stations can switch the hash function utilized.

Primitive Extensions: As was briefly mentioned, 11ec can support a signaling mechanism to alleviate starvation similar to RRTS [4] via control during data and can highly reduce the occurrence of exposed terminals via robust CSS acknowledgment. For reason of space, these cases are not covered in this paper.

C. Contending Flows and Vulnerability Interval

The shortness and robustness of correlatable symbol sequences dramatically reduces vulnerability to collisions. The vulnerability interval of a packet exchange is twice the time delay from the beginning of a transmission until all potential interferers are prevented from corrupting the exchange, i.e., it includes the whole interval, before and after the beginning of the intended exchange, during which interfering transmissions may start and corrupt the exchange. In the case of hidden terminals in *802.11* with RTS/CTS, the vulnerability interval is twice the delay from the moment a node sends an RTS to the detection of CTS by the hidden terminals. Considering the case of 802.11a/g and 6-Mb/s control packets, the total duration of the vulnerability interval can be computed as follows. First, node s transmits the RTS, for a total duration of 52 μs including preambles. Second, the RTS propagates to the receiver for up to 1 μs. Third, the receiving node r waits $T_{\text{SIFS}} = 16$ μs before sending the CTS, due to practical communication issues such as RTS decoding. Fourth, the CTS propagates to hidden terminals for up to 1 μs. Fifth, the hidden terminals need up to 4 μs to detect the packet. Last, additional 2 μs account for potential radio turnaround (all temporal indications are taken from section 17 of the 2007 version of the standard, for 20 MHz bandwidth [1]). The total amounts to 152 μs, i.e., twice 76 μs. In 802.11 without RTS/CTS, the vulnerability interval can be considerably larger, spanning twice the data transmission duration and as large as 4 ms.

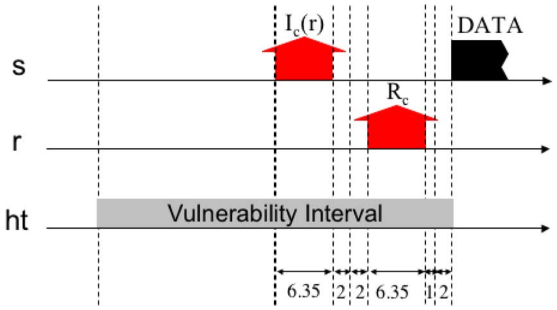


Fig. 2. Timeline of the vulnerability interval of 802.11ec (time indications are in microseconds).

802.11ec's CSSs shorten the vulnerability interval and practically reduce the set of potential hidden terminals. The vulnerability interval of 11ec can be computed as follows (see Fig. 2, where the intervals below are denoted by numerical time indications). First, node s transmits $I_c(r)$ for 6.35 μs. Second, the receiving node r waits for up to 2 μs, in order to detect potential overlapping hidden terminal transmissions as explained below (this interval is a design choice and includes the propagation delay from the transmitter). Third, r needs a turnaround time to prepare its radio for transmission, i.e., 2 μs (according to 802.11 standard). Fourth, r transmits R_c for 6.35 μs. Finally, the hidden terminals need a propagation delay of up to 1 μs to receive the R_c and an additional 2 μs to account for potential radio turnaround. 11ec vulnerable interval is twice 19.7 μs, i.e., 39.4 μs or about 25.9% of the vulnerability interval of 802.11.

CSSs also nearly eliminate the collisions of nodes starting in the same slot when the SINR allows it.[5] In fact, the receiver r can detect simultaneous and overlapping transmissions of multiple $I_c(r)$ because of the CSS processing gain and request retransmission. Specifically, r waits for a round-trip propagation time in order to detect potentially overlapping transmissions and, in that case, sends a negative acknowledgment that prompts the contending nodes to undergo a quick backoff repetition. However, in order to take advantage of this technique, we need to enlarge the slot size to encompass the half vulnerability duration, i.e., 20 μs (note that this is sufficient, as the F_c of the receiver synchronizes all of its transmitters). Also in cases of no hidden terminals, this choice induces only minor throughput penalties at high data rates, as we show in Section IV-B.

Coexistence With Legacy 802.11: For backward compatibility with 802.11, 11ec exactly follows the standard [1] except for the techniques described in this section. For example, a key element for coexistence is the arbitration of the medium, which leverages carrier sensing based on the correlation of the data preambles and backoff mechanism. Accordingly, 11ec uses the same data preamble format as 802.11 and sets the contention window size following the same binary exponential backoff scheme. A more complete discussion of coexistence is beyond the scope of this paper.

III. Experimental Evaluation of CSS

In this section, we present an experimental evaluation of correlatable symbol sequences using software defined radios. Specifically, our evaluation covers the following issues, *none of which has been previously experimentally studied in the literature*.

1) We explore the tradeoff between length of the sequences, i.e., overhead, and processing gain, i.e., robustness.
2) We contrast the performance of CSS detection with control message decoding.
3) We determine the codebook size that 11ec can support, i.e., the number of distinct CSSs that can be practically used, by studying the cross correlation between different CSSs and its effect on the probability of false positives.

The experimental results in this section show that the design selection of 127-symbol CSSs via Gold codes, which can support more than 50 co-located nodes, provides a good tradeoff between overhead and robustness without any penalty on false positives.

A. Experimental Setup

1) Tools: WARP and WARPLab: Our reference software defined radio is the WARP platform [3]. WARP is a field programmable gate array (FPGA)-based platform, including custom designed radios based on the MAX2829 chipset. WARPLab is a programming environment that permits to drive WARP from a host computer. Relevantly to our experiments, WARPLab supports the execution of micro experiments, each one of approximately 400 μs duration [2^{14} samples at the 40-MHz frequency of digital-to-analog converter/analog-to-digital converter (DAC/ADC)], and accesses analog sample send/receive buffers and RSSI recordings collected during each experiment. RSSI is measured by the MAX2829 circuit and digitized by a dedicated 10-bit ADC.

Azimuth Channel Emulator: In order to perform experiments under controllable and repeatable conditions, we used an Azimuth ACE MX channel emulator.[6] The channel emulator permits creation of different network topologies by tuning the attenuation along each path independently and predictably.

2) Implementation: We implement CSS transmission/detection and OFDM packet transmission/decoding on WARPLab. Specifically, CSSs are BPSK sequences filtered, upsampled, and transmitted via standard wideband methods. This solution enjoys a practical advantage over alternative solutions, e.g., OFDM-modulated BPSK sequences, due to the lower peak-to-average-power ratio [27]. Finally, in order to reproduce 802.11 as closely as possible, we implement all types of OFDM 802.11a/g modulation and convolutional code pairs in WARPLab.

B. Channel Emulator Validation

The results in this section are obtained using a channel emulator. In order to validate the emulator setting, our methodology includes a preliminary validation contrasting results of a cross-correlation experiment performed over the air, with an identical experiment conducted with the emulator. Specifically,

[5] Nodes may use power adaptation techniques exclusively to transmit control CSSs while transmitting data at full power, i.e., without modulation rate adaptation requirement or throughput penalty. While this may improve the performance of 11ec, we defer its investigation to future work.

[6] Azimuth Systems Inc., http://www.azimuthsystems.com/

our experiment consists of exchanging CSSs between two WARP nodes a and b, under the interference generated by random OFDM transmissions of a third interfering node c. In the first part of the experiment, we deploy the three nodes inside an office building. In an over-the-air setting, it is difficult to control SINR given interference from 802.11 networks operating in the building. For this reason, we perform the experiment late at night and measure the SINR on links $a-b$ and $c-b$ a few seconds before and after the experiment. Then, we repeat the experiment under controllable and repeatable conditions using the channel emulator, where we can control the SINR with high accuracy. We repeat both experiments several times and for different SINR and show a representative sample result.

In both experiments, node a transmits seven repetitions of a 127-symbol CSS. For each newly acquired sample (i.e., at 40 MHz frequency of the WARP platform ADC), node b computes the signal correlation with a local copy of the transmitted CSS. Fig. 3(a) and (b) shows a representative outcome of a realization in which the SINR on link $a-b$ carrying the CSS is -6 dB. The x-axis is the temporal progression of collected samples, while the y-axis is the correlated value. The thick crossing line in the plots represents a possible choice of the detection threshold; such a threshold is strategic in determining the robustness of CSSs by balancing false positives and false negatives. In the experiments we conduct in this section, the threshold is chosen in order to obtain a false positive probability of 10^{-8}. While we defer more details on how to tune the threshold to Section III-F, here we observe that because of the threshold design, the correlation value on the y-axis is normalized according to the magnitude of correlated I/Q samples. In the figures, correlation spikes are clearly identifiable in coincidence with the reception of each single CSS as all and only marks exceeding the threshold. Thus, the detection of 127-symbol CSSs is possible at -6 dB with few errors. *By comparing the two plots, we conclude that controllable emulator experiments and over-the-air experiments provide similar results for identical SINR values*. However, because of the difficulty to constantly control the SINR in over-the-air settings, we perform the remaining experiments in this section using the channel emulator, thereby also ensuring their repeatability.

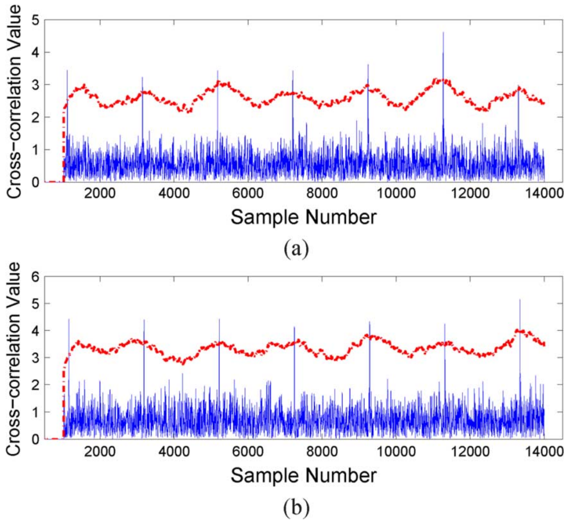


Fig. 3. Example of 127-symbol CSS correlation at -6 dB. (a) Over the air. (b) Channel emulation.

C. CSS Length Versus Robustness Tradeoff

The first issue is the trade-off between the CSS length L and robustness under different SINR; we quantify robustness in terms of the probability of false positives and false negatives. The outcome of this assessment is important, as robustness to SINR is one of the two key elements in the choice of CSS length (the other element guiding this choice is the number of CSSs, discussed in Section III-E). In this section, we determine a length that can tolerate significant interference without high communication overhead. In this experiment, we deploy the three-node topology above and use the channel emulator to vary the SINR on link $a-b$. Specifically, the link between node a transmitting the CSS, and the receiving node b is maintained fixed to -82 dBm, while the attenuation on the interfering link $c-b$ is set in order to obtain the desired SINR. We iterate the experiment for different combinations of SINR and CSS lengths. Each experiment consists in the detection of at least 100 CSSs of lengths L ranging from 63 to 511 symbols. We vary the SINR between 0 and -10 dB.

Fig. 4 shows the probability of false negatives as a function of SINR and CSS length. Specifically, the x-axis denotes the SINR on link $a-b$, while the y-axis denotes the probability of false negatives. The different curves correspond to different CSS lengths. The figure shows that longer CSSs are more robust due to the processing gain, e.g., at -8 dB, CSSs of length 63 can be detected only 4% of the time (96% of false negatives in the figure), while CSSs of lengths 127, 255, 511, can be detected approximately 30%, 99%, and 100% of the time, respectively. However, increasing the CSS length involves an overhead penalty; in fact, while a 63-symbol CSS can be delivered in about 3.15 μs, 127-, 255-, and 511-symbol CSSs require 6.35, 12.75, and 25.55 μs, respectively. With regard to the probability of false positives, we never obtained more than a single occurrence (out of hundreds of thousands of tests performed) for all the experiments related to a fixed SINR and length combination. Finally, it is relevant to notice that our results show only minor degradation with respect to theoretical performance in additive white Gaussian noise (AWGN) channel. For instance, considering the probability of false positives and false negatives at -8 dB, the length of sequences with similar performance in AWGN would be 47, 81, and 198 for the cases of 63, 127, and 255 actual lengths. *We conclude that 127-symbol CSSs provide a good compromise between overhead (6.35 μs), and resilience as they can be detected at -6 dB with 5.7% false negatives and no false positives.*

D. CSS Detection Versus Control Message Decoding

Our second experiment aims to show that control CSSs are more robust to noise than 802.11 control messages, i.e., that CSSs can be reliably detected at considerably lower SINR than control messages. The metrics we use are false positives for the case of CSS detection, and packet error rate for control message decoding. We consider 127-symbol CSSs versus 160-bit

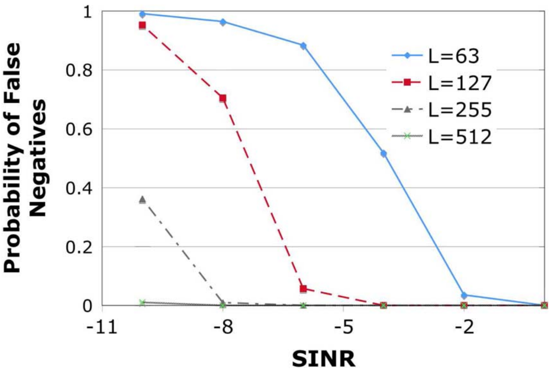


Fig. 4. Robustness versuss length tradeoff for different CSS lengths.

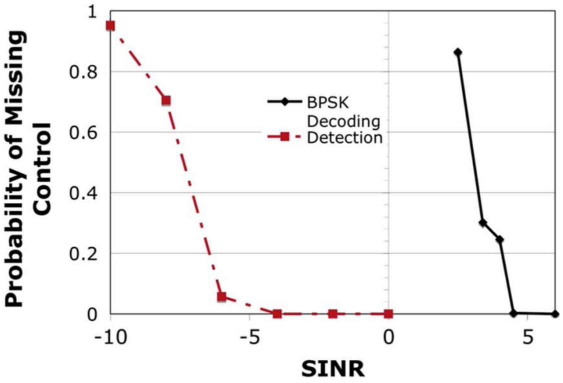


Fig. 5. Probability of missing CSS detection versus missing message decoding.

messages transmitted via BPSK modulation, with 1/2 rate convolutional coding, corresponding to an RTS packet transmitted at base rate of 6 Mb/s in 802.11 OFDM. In order to create the interference scenarios, we follow a methodology identical to the previous experiment. For the case of BPSK modulation, our experiment directly measures the BER out of at least 100 000 bit transmissions. Then, we convert the obtained value to packet error rate by considering a random and independent distribution of the bit errors among the packets (i.e., $1-(1-\mathrm{BER})^{\mathrm{PL}}$, where PL is the packet length in bits). Note that the adoption of burst error models, such as Gilbert–Elliot [10], with expected burst length of 6 bits [28] may vary the results by about 1 to 1.5 dB.

Fig. 5 shows two curves corresponding to CSS detection and control messages decoding. The x-axis denotes the SINR, while the y-axis denotes the *probability of missing control*, i.e., the probability of false negatives (resp. of packet decoding error) for CSSs (resp. for control messages). The plot shows that control CSSs are substantially more robust than control messages since their probability of false negatives is much less than the error probability of control packets for any SINR. Furthermore, similar *probabilities of missing control* are obtained for the two control mechanisms, for SINR values separated by about 10 dB. For example, CSSs obtain probability of false detection of 5.7% at -6 dB, while control messages achieve 24% packet error rate at $+4$ dB, and 0.2% at $+4.5$ dB. *We conclude that due to the improved robustness of CSS detection with respect to packet decoding, control CSSs are about 10 dB more resilient to noise than 802.11 control messages.*

E. 11ec Codebook Size

In 11ec, nodes use multiple CSSs and need to be able to reliably detect and discern all of them. In this experiment, we investigate whether cross correlation between CSSs affects detection accuracy, and we explore the number of distinct CSSs that can be practically used. Specifically, we assess the probability of falsely detecting CSS A when CSS B is transmitted instead. For a given CSS length, a tradeoff exists between the number of CSSs that 11ec uses and the magnitude of the cross correlation between any CSS pair, which in turn influences the probability of false positives. For this error probability to be small, we use well-known sparse binary sequences, with optimal cross-correlation properties. Instances of such sequences have been studied for lengths corresponding to powers of 2, e.g., in cellular communications [22]. Different families of sequences provide larger (resp. smaller) sets of codewords, with larger (resp. smaller) cross correlation between any codeword pair. Our design implements Gold codes, which provide 127 CSSs for our 127-symbol length, with a theoretical cross correlation on the order of 12%. The choice of Gold codes permits us to support more than 50 co-located nodes by assigning distinct CSSs pairs to each node representing $I_c(r)$ and $A_c(s)$, while saving several CSSs for F_c^i/R_c^i pairs. In case a larger number of nodes needs to be supported, 11ec can switch to 255-symbol Kasami large codes for example, which allow more than 2000 nodes with 4011 CSSs. Note that in this case the CSS overhead would roughly double, while the CSS robustness would increase as shown in Fig. 4. Other code alternatives include Zadoff–Chu sequences, which are used in 3GPP LTE [7].

To verify our choice, we emulated a situation in which a CSS A is sent, and 10 nodes try to detect CSSs different from A within the same samples. We repeated the experiment for 100 detection attempts, for 127-symbol Gold codes and SINR from 0 to -10 dB. The goal of this experiment is to assess the probability that the other nodes obtain false positives of their own CSS when the signature A is sent. The number of false negatives is immaterial in this experiment. For each SINR experiment, we obtained at most one false positive more than the case of a single CSS detection. Fig. 6 shows an example outcome for -6 dB SINR (where the axes have the same meaning as in the experiment in Fig. 3). The overlapping plots show the cross-correlation values obtained by 11 different nodes (including the one that expects to detect the transmitted CSS). While the number of spikes is unchanged, the noise looks visually denser due to overlapping plots. *We conclude that by using Gold codes, 11ec can support more than 50 co-located nodes without significant incidence of false positives.*

F. Discussion on Signal Correlation

Practical Detection Threshold Selection: The choice of the detection threshold is strategic in balancing the tradeoff between false positives and false negatives. For example, as mentioned above, in Fig. 3(a) and (b) the threshold is denoted by the thick crossing line; corresponding to the chosen threshold, the figures show 0 false positives and 0 false negatives. In general, a higher value of the threshold decreases the probability of false positives at the expense of a large probability of false negatives; vice versa, a lower threshold increases the occurrence of false

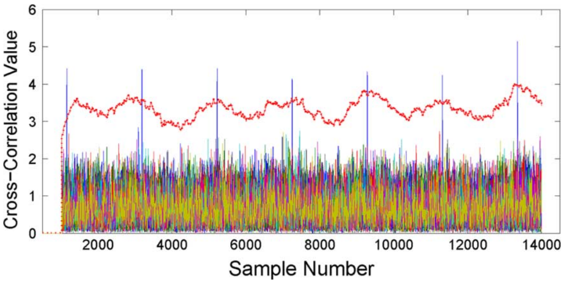


Fig. 6. Low cross correlation of CSSs different from the one transmitted.

positives. The theory of correlation in Gaussian noise provides an optimal threshold for a target detection SINR, minimizing the probability of false negatives for a given probability of false positives [15]. Specifically

$$T = \sqrt{\frac{L \cdot \mathcal{E} \cdot \mathcal{N}}{2}} * Q^{-1}(P_{\mathrm{FA}}) \tag{1}$$

where Q is the tail probability of the standard normal function. The formula shows that the optimal T depends on noise power $\mathcal{N}$, CSS power $\mathcal{E}$, CSS length L, and on the target probability of false positives P_{FA} that we fix to 10^{-8}. We remark that: 1) generally, the power of the noise (which may be due to interfering transmissions) varies in time, thus the detection threshold should also change; 2) practically, it is difficult to estimate in advance the power of the noise and the power of the signal. In order to address the latter concern, we establish a lower bound on the SINR of the signal that we aim to detect (in our experiments -6 dB), and we tune T correspondingly (i.e., $T = \sqrt{\frac{L \cdot \mathrm{SINR} \cdot \mathcal{N}^2}{2}} * Q^{-1}(P_{\mathrm{FA}})$). Unfortunately, this solution is not sufficient because of the difficulty to estimate $\mathcal{N}$. In fact, the receiver can only measure the total power of the incoming signal, which may or may not contain the target CSS. Thus, we conservatively choose to replace $\mathcal{N}$ with the total signal power received, as if the incoming signal did not contain the CSS; practically, when the CSS is actually present, this choice has the effect of tuning T to a higher value than desired, i.e., it increases the occurrence of false negatives. Figs. 3 and 6 show that the value of the threshold increases when the signal is present, and decreases otherwise.

Wireless Communications Issues: It is important to note that two issues may affect the performance of correlation, both due to the fact that the transmitter and receiver radio generate independent clocks [11], [23]. First, the clock phases at the transmitter and receiver are in general not aligned; this produces a *phase offset* between the two radios, which causes a fixed rotation of the received symbols of an angle γ. In order to compensate for this effect, we compute the magnitude of the correlation, with a theoretical penalty on the processing gain of about 0.5 dB [21]. Second, while the nominal frequencies of transmitter and receiver clocks are identical, they practically differ by a small Δf; this problem is known as *carrier frequency offset*. Carrier frequency offset produces a continuous rotation of the received symbols. Practically, Δf is sufficiently small (e.g., $\sim$1–4 kHz [23]), so that its effect is negligible over the CSS lengths/durations considered in this paper.

Hardware Implementation: The hardware implementation of CSS transmission and detection only requires the replication of components that are already present in off-the-shelf 802.11 chipsets, and specifically of filters and correlators. The basic implementation of 11ec uses four correlators; in particular, each node r needs to be able to detect $I_c(r)$, R_c, F_c, and $A_c(r)$ (additional correlators may help increase channel reservations robustness as per Section II-B). Because the correlated BPSK sequences at most require a sign flip on the received I/Q samples[7] and several summations (with respect to expensive multiplications required to implement 802.11 floating-point correlators), CSS correlators occupy a very limited amount of resources. Finally, note that $I_c(r)$ and $A_c(r)$ sequences are not necessarily known *a priori* by node r and may be determined only during network operation. Hence, the updating function should be implemented via the software register control.

IV. Experimental Evaluation of 11ec

In this section, we present an experimental validation of 11ec using a measurement-based emulator that we design and implement. We perform the following experiments.

1) We investigate the performance of 11ec in basic topologies that typically incur loss or imbalance for 802.11.
2) We investigate the performance of 11ec in a larger 5-flow network topology.
3) We complete the evaluation of 11ec simulating its performance in a 20-node network topology.

The results in this section show that 11ec dramatically *improves network fairness*; furthermore, while such improvement can often decrease total utilization, 11ec remarkably *increases channel utilization*. Unlike 802.11, 11ec gives equal opportunities to weak links characterized by low data rates and strong high data rate links; for this reason, 11ec may sometimes achieve lower cumulative network throughput.

A. Measurement-Driven Network Emulator

Our measurement-driven emulator is based on the GloMoSim simulator [29].[8] For the sake of realism, we modify GloMoSim in three ways: 1) we implement the support for multiple 802.11a/g modulations, i.e., BPSK 1/2, QPSK 1/2, 16QAM 1/2, and 64QAM 3/4 (corresponding to 6, 12, 24, and 54 Mb/s respectively); 2) we implement a new propagation model that calculates links attenuation using our channel measurements; 3) we implement a new traffic generator that draws the packet lengths from a distribution obtained experimentally. Specifically, with regard to the former, we perform a set of measurements at the channel emulator using the same transmitter/receiver/interferer setup described in Section III, and we measure the BER as a function of the SINR. With regard to the second issue, we deploy up to eight WARP nodes simultaneously in an office building (see Fig. 7) and measure the signal strength between any pair, i.e., for any run of the experiment, a single node transmits 400 μs packets, and all others record

[7] The magnitude of the actual samples is scaled by a gain control component.

[8] Even though the last version of GloMoSim dates to late 2001, the basic operations of the 802.11 MAC layer are consistent with the latest standard. The physical layer includes features such as noise accumulation, which make it preferable to alternative simulators.

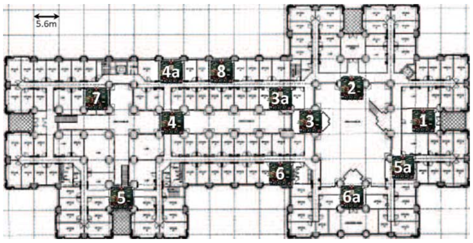


Fig. 7. Layout of our office building deployment.

the received power. As a result of these two measurements, we manually select for each link in the network emulator the highest data rate that its channel SINR can support with negligible packet error probability. Finally, we integrate all results into our measurement-driven emulator. Next, we collect packet lengths of a user session running a mix of applications including Web browsing, video downloading, audio streaming, and VoIP. The characteristics of the traffic are as follows: 773 different packet lengths, with a median value of 143 B, and average of 596 B. We implement a GloMoSim traffic generator that randomly determines the lengths of the transmitted packets following the distribution obtained experimentally.

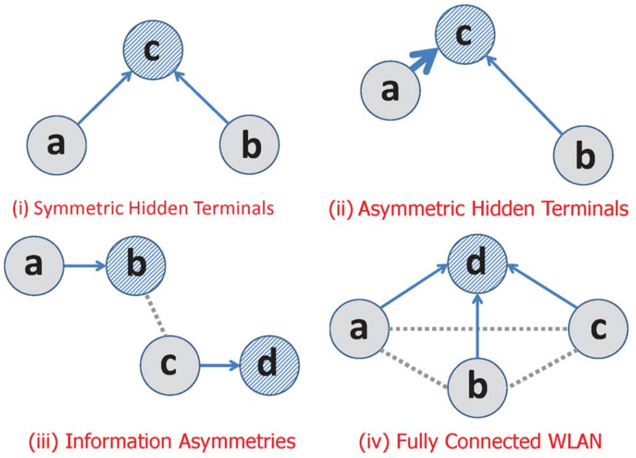
CSS Implementation: We implement CSS reception and detection as an autonomous physical layer component, independent of the packet detection architecture of GloMoSim, i.e., CSSs are not simulated via small packets. Specifically, nodes store incoming CSSs, and schedule their evaluation after a delay corresponding to CSSs length, i.e., 6.35 μs for a 127-symbol CSS. For any stored CSS, the emulator keeps track of the variation of the background interference. At the moment of the evaluation, the average SINR of the CSS is computed, and CSS detection is triggered if the SINR exceeds a threshold tuned to -6 dB for 127-symbol CSS (see Section III-C). Our implementation permits each node to simultaneously store, evaluate, and potentially detect multiple CSSs overlapping with other CSSs or incoming packets. Note that the original GloMoSim implementation of packet decoding does not support any of the features above, i.e., delayed evaluation and simultaneous multisignal reception.

Finally, we implemented 11ec's MAC-layer-state machine by building on GloMoSim's 802.11. In particular, the design integrates the novel procedures corresponding, e.g., to deferral and timeout management.

B. Basic Topologies

In this set of experiments, we evaluate the performance of 11ec in a few basic topologies (mostly including two flows) that are characterized by symmetries or asymmetries in link signal strength differences and carrier-sensing relationships. This study is important because these topologies are at the origin of throughput losses and/or imbalances in 802.11-based networks [6]. We show that 11ec largely overcomes the problems of 802.11 with and without RTS/CTS.

In our study, we classify the basic topologies into four main groups according to the prevalence of one of the two links with respect to the other (e.g., due to higher SINR), and to the carrier-sensing relationships between the transmitters. Specifically, the four topologies are the following (see Fig. 8)



(i) Symmetric Hidden Terminals

(ii) Asymmetric Hidden Terminals

(iii) Information Asymmetries

(iv) Fully Connected WLAN

Fig. 8. Basic topologies at the origin of throughput losses and/or imbalances.

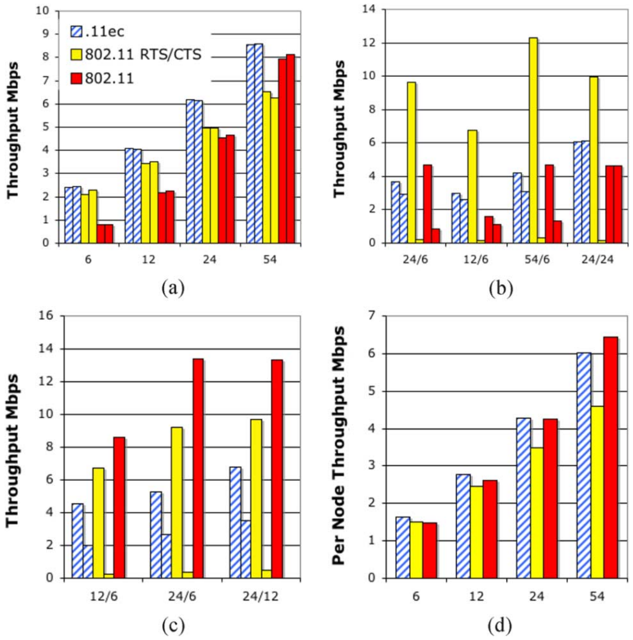
i) *Symmetric Hidden Terminals* include topologies in which a link is formed between each of two transmitters that are not able to carrier-sense each other and a common receiver; moreover, the links have similar reception power at the receiver. Specifically, these topologies include links with signal strengths at the receiver differing by no more than 4 dB; we choose this threshold to exclude capture at any 802.11 modulation.

ii) *Asymmetric Hidden Terminals* include topologies in which two non-carrier-sensing transmitters share a common receiver, and one of the formed links has a significant power advantage. Specifically, these topologies include links with signal strengths at the receiver differing by more than 5 dB; this threshold is chosen to permit one of the two links to capture over the other at BPSK modulation.

iii) *Information Asymmetries* include link pairs *a*-*b* and *c*-*d*, with transmitters *a* and *c* not carrier-sensing each other, and with different receivers; moreover, one of the two links *c*-*d* interferes with the other link *a*-*b*, but not vice versa.

iv) *Fully Connected WLANs* include topologies in which all nodes carrier-sense each other and transmit to a common receiver.

Most of the experiments in this section are performed by reproducing the topologies in our measurement-driven network emulator, with fully backlogged traffic and packet lengths determined according to the experimental distribution described above. Each figure includes the bar graph of the throughputs of the flows for the three protocols we compare, namely 11ec, 802.11 with RTS/CTS, and 802.11 without RTS/CTS. Where utilized, RTS/CTS are transmitted at the OFDM 6-Mb/s base rate. 11ec implementation includes CSS acknowledgments, but does not support RRTS mechanisms where not differently specified. The experiments in Fig. 9(a)–(c) involve flow pairs; accordingly, the figures contain groups of six bars that correspond to the throughput of each flow, as achieved by the three protocols. The x-axes of the graphs indicate the transmission data



(a) (b) (c) (d)

Fig. 9. Throughput of 11ec, 802.11 with/without RTS/CTS in basic topologies. (a) Symmetric Hidden Terminals. (b) Asymmetric Hidden Terminals. (c) Information Asymmetry. (d) Fully Connected WLANs.

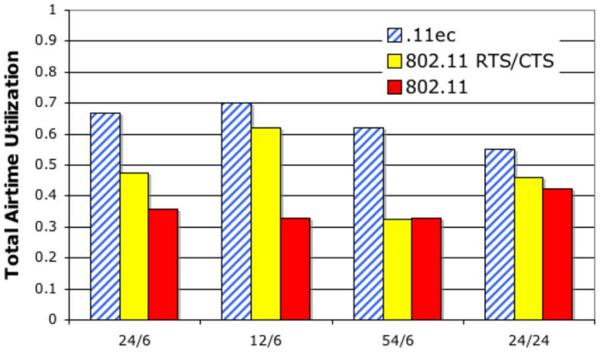


Fig. 10. Total airtime utilization in the case of Asymmetric Hidden Terminals.

rates of the flows (when two different values are used, they match the order of the bars in the pairs), while the y-axes are in megabits per second (Mb/s). In Fig. 10, we introduce a metric termed *total airtime utilization*, which denotes the time share during which successful data packets are transmitted. Finally, we evaluate fairness according to two well-known indicators, namely Jain's index [13] and proportional fairness [16]. Jain's index assumes values in the interval [0, 1]; for both indicators, higher values correspond to higher fairness.

Symmetric Hidden Terminals: We consider four instances of symmetric hidden terminals: Three of them are selected from our deployed network and use modulation rates corresponding to 6/12/24 Mb/s. The fourth case (54 Mb/s) is artificially generated in order to explore the effect of using higher modulation schemes. Fig. 9(a) shows that all solutions assign similar throughputs to both flows. However, 11ec and 802.11 with RTS/CTS achieve considerably higher total throughput than 802.11 for most rates. Furthermore, 11ec outperforms 802.11 with RTS/CTS due to the smaller duration of CSSs with respect to control messages (11ec control CSS exchange lasts 19.7 μs, with respect to 128 μs of 802.11 control messages). Besides reducing the control overhead, this entails the decrease of the collision probability (see Section II-C) from 13% for 802.11 with RTS/CTS to 6.5% for 11ec regardless of the rate used to transmit the actual data. The effect of the overhead reduction becomes more and more evident as the packet data rate increases; at 54 Mb/s, the total throughput gain of 11ec over 802.11 with RTS/CTS is about 34%. Finally, at 54 Mb/s, the data packets are sufficiently short to permit low collision probability to 802.11 without RTS/CTS (19% as opposed to 66% obtained at 6 Mb/s); nonetheless, 11ec still shows 5% throughput gain.

Asymmetric Hidden Terminals: We consider four instances of asymmetric hidden terminals, all based on actual link power measurements. In these topologies, packets sent by the sender with high signal-to-noise ratio (SNR) at base rate, e.g., control packets, capture over packets sent by the sender with low SNR. However, packets sent by the sender with high SNR at data (i.e., higher) modulation rate get corrupted when overlapping with packets sent by the sender with low SNR. This is because we choose the data modulation rates of the links based on the SNR in the absence of interference. Fig. 9(b) shows that the capture effect has disastrous consequences for the flow with low SNR in 802.11 with and without RTS/CTS, while it has no effect on 11ec. For example, in the first instance presented (i.e., modulation rates 24/6), 11ec improves the throughput of the underserved flow with respect to 802.11 with (without) RTS/CTS by about 13-fold (2.5-fold). In 802.11 with RTS/CTS, the imbalance is due to the fact that in the case of overlapping RTS, the RTS of the stronger link is correctly decoded, while the other is ignored. Thus, while the stronger link enjoys a high successful probability, the RTS collision probability of the weaker link is over 50% in all instances. A similar consideration also applies to 802.11ec; in fact, link SNRs can be sufficiently different to permit the CSSs from one of the links to capture over the CSSs from the other in case of overlapping. However, the probability of CSSs overlapping is so low for 802.11ec that even in such cases (not shown in the figure), the resulting throughput imbalance remains confined within about 15%. Since 802.11 without RTS/CTS does not use the base rate, but transmits all packets at data rate, the throughput imbalance originates only from the shorter duration of the data packet of the dominant link, which permits it to enjoy higher success probability. The packet collision probability of the weaker link is again over 49% for all cases of heterogeneous rates. The result for the fourth instance (i.e., last two bars in the graph) supports this claim: When both links operate at 24 Mb/s, their throughputs do not depend on any SNR imbalance, and the collision probability is reduced to 30%. Finally, because of the greater number of data packet collisions (that have a longer duration than RTS or CSS collisions), 802.11 without RTS/CTS has a lower total throughput. In contrast to 802.11, 11ec reduces the imbalance by reducing the packet loss to about 11%.

In terms of fairness, considering for example the first instance, 11ec improves Jain's index from approximately 0.52 and 0.68 in 802.11 with and without RTS/CTS to 0.99; similarly, in terms of proportional fairness, 11ec improves the sum of the logs of the share of the rates from -1.67 (-0.87) of 802.11 with (without) RTS/CTS to -0.6. The aggregate throughput of 11ec is lower than the competitors. However, this is misleading, and is due to the fact that 11ec improves the throughput of flows with lower data rate. Fig. 10 clarifies this aspect by showing the total airtime utilization, i.e., the fraction of time used for successful transmissions, for all instances represented in Fig. 9(b). 11ec obtains up to 90% higher airtime utilization than the other 802.11 versions.

Information Asymmetry: In all three instances of these topologies, which as before are based on our channel measurements, the interfering link always succeeds, while the interfered may become severely underserved due to high number of collisions. In fact, the sender of the underserved link cannot perceive the state of the channel (busy or free) at its receiver and randomly selects transmission instants; in the likely case of collision, the sender of the underserved link backs off, thus accessing the channel less and less frequently. Fig. 9(c) shows that the information asymmetry completely starves the underserved link in 802.11. 802.11 with RTS/CTS slightly outperforms 802.11 without due to the greater probability of the underserved flow to correctly transmit an entire RTS without being interrupted by the interferer. In fact, the collision probability decreases from almost 100% for packets transmitted by 802.11 without RTS/CTS to 80%–90% for RTS transmitted by 802.11 with RTS/CTS. It is important to note that several 802.11 flows in this experiment have a throughput close to zero (invisible in the figure). In contrast, even without the feature of control during data, 11ec manages to assign significant throughput (about 45% of the interfering link) to the underserved link. For example, in the last instance presented (i.e., the left-most six bars in the figure) 11ec improves the throughput of the underserved link with respect to 802.11 with (without) RTS/CTS by ninefold (500-fold). In terms of fairness, 11ec improves Jain's index from approximately 0.5 in 802.11 with and without RTS/CTS to 0.87, with a 65% gain, and proportional fairness from -1.47 and -3.34 of 802.11 with and without RTS/CTS, respectively, to -0.67. Similarly to the case of asymmetric hidden terminals, 11ec achieves a cumulative throughput lower than 802.11 but gains up to 21% (8.5%) in airtime utilization with respect to 802.11 with (without) RTS/CTS. Since in all cases the interfering link does not suffer from collisions, the 11ec gain is solely due to the small duration of control CSSs that have a high probability of being received during free channel intervals.

Fully Connected WLANs: Fig. 9(d) shows the average throughput obtained for three carrier-sensing links, and groups of three bars correspond to the three protocols compared. In this scenario, 802.11 without RTS/CTS generally performs the best due to low overhead and rare collisions. In the worst case of 54 Mb/s, 11ec obtains 6.9% less throughput than 802.11 without RTS/CTS, while 802.11 with RTS/CTS achieves 23% less throughput than 11ec due to the control message overhead. At low data rates, all protocols perform similarly due to the long packet durations. This result clearly shows that even in the absence of hidden terminals, control CSSs and larger slot size of 11ec do not incur significant throughput penalties.

Final Remarks: First, in the experiments above, we consider UDP traffic. We note that the throughput imbalances we observed for 802.11 would negatively affect TCP dynamics, e.g., by magnifying the penalties due to small congestion windows, longer timeouts etc. By inspecting the traces, we also observe that even in cases of balanced throughput (such as *symmetric hidden terminals*), 802.11 (both with and without RTS/CTS) alternately serves one of the two links for long periods of time by almost starving the other [4]; this is extremely detrimental to TCP performance. Second, our experiments do not implement rate adaptation, but manually select the best rate achievable based on the links SNR. We observe that rate adaptation does not produce any benefit to 802.11 with RTS/CTS since control messages need still to be transmitted at base rate and data packets rarely collide. On the other hand, 802.11 without RTS/CTS may benefit in case of hidden terminals, but typically at the price of higher unfairness even in fully connected WLANs due to capture effect [19]. Finally, it is remarkable to notice that in contrast to common tenets of related literature, RTS/CTS at 6 Mb/s does produce a significant performance improvement over 802.11 without RTS/CTS and only slightly penalizes the throughput in the absence of hidden terminals.

C. Network-Wide Emulation

Here, we investigate larger topologies in order to demonstrate the fairness gains of 11ec in case of multiple flow interactions. We consider a five-flow topology based on the channel measurements we performed; specifically, the flows operate at 24, 24, 24, 54, and 6 Mb/s (in the order of flows depicted in Fig. 11), and each one conveys fully backlogged traffic, with packet length distribution as discussed above. Fig. 11 shows the detailed bar graph of the throughput of all flows for 11ec and 802.11 with/without RTS/CTS; the flows on the x-axis match the node locations in Fig. 7. The figure shows that 802.11 with RTS/CTS and 11ec achieve higher fairness than 802.11 without RTS/CTS. In addition, while 802.11 with RTS/CTS almost starves flow $4a \rightarrow 3a$ by assigning 67 kb/s, 11ec manages to assign it 1.536 Mb/s for a gain of 2292%; 802.11 without RTS/CTS only assigns 0.4 kb/s to that flow. Because the flow operates at 6 Mb/s, this has a significant effect on the airtime utilization, which increases from 0.41 and 0.48 of 802.11 with and without RTS/CTS to 0.60 of 11ec, for a gain of 46% and 25%, respectively. Finally, 11ec significantly increases throughput fairness; specifically, Jain's index is 0.57, 0.34, and 0.93 for 802.11 with and without RTS/CTS and 11ec, respectively. 11ec shows about 30% higher proportional fairness with respect to 802.11 with RTS/CTS.

D. Large Network Simulation

This experiment aims to evaluate the performance gain of 11ec in larger network topologies including 20 nodes. We generate 10 scenarios, each one obtained by randomly deploying 20 nodes within an area of 250×250 m^2. We model the channel according to a path loss with attenuation exponent of 3.5; the nodes use a transmission power of 100 mW. The resulting link SNRs dictate the data rate that can be utilized on the links and their carrier-sensing relationships; according to the latter, the network spans at most 5 hops. In this experiment, the link data rates are manually selected in order to achieve reliable decoding for an interference 3 dB stronger than noise, i.e., in order to maintain an SNR margin of

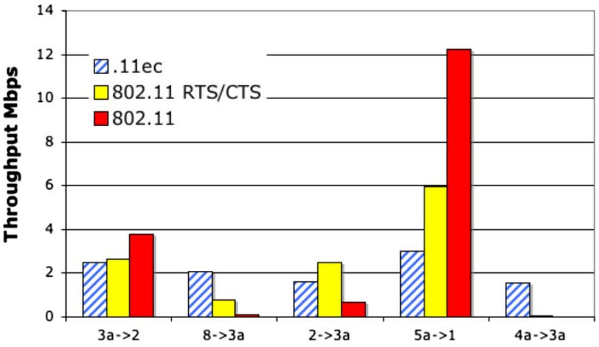
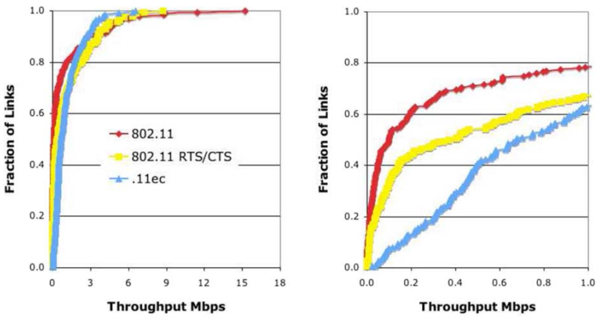


Fig. 11. Throughput distribution for a five-flow topology.



(a) (b)

Fig. 12. Throughput of 11ec, 802.11 with/without RTS/CTS in 20-node simulated topologies. (a) Entire CDF. (b) Zoom in.

3 dB. For each node, the data flow is determined by randomly choosing one of its neighbors as the destination; the traffic is fully backlogged, and the packet lengths are chosen according to the experimental distribution above. Fig. 12(a) shows the cumulative distribution function (CDF) of all link throughputs achieved by the three protocols through all 10 topologies of our experiment; Fig. 12(b) is a zoomed-in version. In the figures, the y-axes represent the fraction of flows, while the x-axes are throughput values (in Mb/s). Accordingly, the plotted values indicate the fraction of flows with a throughput smaller than or equal to the corresponding abscissa; e.g., the point at (0.4, 0.444) means that 40% of the flows have a throughput lower than or equal to 444 kb/s. Note that the flows are represented in the graphs according to the sorted values of their throughputs, i.e., neither the x-axis nor the y-axis are related to the specific flow identities.

The left plot represents the entire CDF, while the right plot magnifies the results for the flows that achieve less than 1 Mb/s. Clearly, the right plot shows that 11ec highly benefits the underserved links, i.e., the bottom 70.5% (85.5%) with respect to 802.11 with (without) RTS/CTS. Specifically, the average throughput of the 70% lowest-throughput flows is 192% (632%) greater for 11ec than for 802.11 with (without) RTS/CTS. Moreover, at the 20/40/60th percentiles, 11ec achieves a throughput value of 7.7×/3.5×/1.4× (25×/10×/4.8×) greater than 802.11 with (without) RTS/CTS. As discussed above, the throughput gain of weak links in 11ec occurs at the expense of the strong links; in fact, the left plot shows that the top 9%–29% flows achieve best performance with 802.11 with RTS/CTS, while the top 9% obtain highest throughput with 802.11 without RTS/CTS. The average link throughput is 1.09 Mb/s for 11ec and 1.15 (0.94) Mb/s for 802.11 with (without) RTS/CTS. 802.11 with RTS/CTS achieves higher average throughput than 11ec. However, this gain occurs at high expense of medium utilization and fairness. In fact, in terms of airtime 802.11 with (without) RTS/CTS achieves 0.61 (0.39), while 11ec achieves 0.76. Jain's index improves from 0.34 (0.27) for 802.11 with (without) RTS/CTS to 0.53.

E. Extensions: Control During Data Simulation

In Section II-A we introduce the possibility of conveying control information during data reception or overhearing, by leveraging the robustness of CSSs. This technique is particularly useful when the SINR on a weak link is -6 dB or more with respect to the hidden terminal interferers. Consider for instance the *information asymmetry* topology; the major issue is that the sender of the weak link cannot determine the state of the medium at its receiver. Consequently, the transmissions of the sender likely overlap with the interferer transmissions and cannot be decoded. Instead, in 11ecm the receiver can detect the initiation CSSs and be notified of the sender's intention to communicate even while the interferer is simultaneously active, as long as the SINR conditions allow it. When the transmission of the interferer is over, the receiver may contend for the medium in spite of the sender and transmit an invitation to send an RTS, similar to the RRTS suggested in [4], that prompts the sender to perform the data exchange while preventing the interferer from accessing the medium.

We implemented this mechanism as a variation of 11ec and verified its performance in information asymmetry topologies, designed to satisfy the SINR condition above. Fig. 13 contrasts the throughput on the two links for 11ec with and without the RRTS enhancement for the different data rate configurations. In the experiment, the weak link operates at 6 Mb/s (left bar in each pair), while the data rate of the interfering link varies in the range of 6, 12, 24, and 54 Mb/s in the different experiments (right bar in each pair). The figure shows that in this case RRTS highly improves the performance of the weak link with respect to the interfering, and even leads the weak link to obtain the higher throughput. This is due to the fact that as soon as the medium becomes interference-free, the receiver reserves it for the sender. In general, the results show a fairer throughput distribution between the links. Specifically, Jain's index improves from 0.84–0.87 to 0.97–0.998, while the proportional fairness increases by 14%. The increase in fairness is accompanied by an increase in airtime utilization ranging from 2% to 10%.

V. Related Work

Many random access MAC protocols have been proposed to mitigate collisions and address hidden terminals, e.g., [4], [14], and [26]. In contrast to tones and messages used by such protocols, we design a new primitive of correlatable symbol sequences that, by virtue of being short and robust, increases throughput and fairness with minimal overhead. More recently, while several papers have addressed 802.11 throughput overhead reduction [19], [24], they completely neglect the case of hidden terminals and, because of their more aggressive contention mechanisms, suffer severe throughput penalties

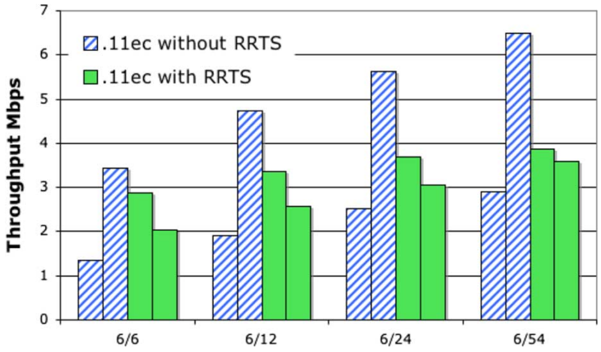


Fig. 13. Effects of the use of the feature of Control during Data in 11ec on Information Asymmetry topologies.

in their presence. Ongoing standardization efforts, namely 802.11ah [2], target overhead reduction for subgigahertz communications, with application to smart grids, surveillance systems, etc. The strategy adopted in [2] consists in removing or compressing some information fields of the control messages, for a total of few bytes; compared to 11ec, this has a minor effect on the control message overhead, e.g., due to the preservation of the cumbersome preamble structure. Other techniques have been proposed to improve 802.11 throughput, e.g., [11], [18], and [23], but address collision resolution rather than collision avoidance. For this reason, they are complementary to (and can be used in combination with) 11ec. Furthermore, our physical-layer model is significantly more simple than [11], [19], [23], and [24] since it relies only on the replication of components (correlators) that are already present in common 802.11 cards. Finally, other applications of signal correlation have been recently shown in [12], [25], and [30].

VI. Conclusion

In this paper we present 802.11ec, an 802.11-based protocol without control messages. 802.11ec introduces control correlatable symbol sequences that provide robustness and efficiency. Through a wide set of experiments on a software defined radio, we show that 6.35-μs correlatable symbol sequences can be detected at an SINR of -6 dB, i.e., 10 dB lower than 802.11 control messages. Finally, we implement 802.11ec on a measurement-based network emulator and show that it improves network fairness by up to 90%, channel utilization by up to 90%, and throughput of underserved flows by over 22 folds, with respect to 802.11.

Appendix

A CSS s is detected via cross correlation with a local copy, i.e., at the reception of a complex signal y that may contain s, y is cross-correlated with the complex conjugate of the target CSS $s*$. Formally, for a CSS of length L

$$\mathcal{C}(\Delta) = \sum_{0}^{L-1} s^*(k)y(k+\Delta) \tag{2}$$

where Δ represents the position of the correlation with respect to the input signal, i.e., the sample for which we perform the correlation. Note that: 1) if $[y(\Delta)\ldots y(\Delta+L-1)]$ does not contain exactly s, the value of $\mathcal{C}(\Delta)$ is nearly 0; 2) if $[y(\Delta)\ldots y(\Delta+L-1)]$ contains a copy of s with sufficient SINR, the $\mathcal{C}(\Delta)$ obtains a large value proportional to the energy of the signal.

We use cross correlation as a test statistic for the detection of a target CSS and repeat its computation at each new sample of the incoming signal. Specifically, detection is performed by setting a threshold T (see Section III-F); if $\mathcal{C}(\Delta) \geq T$ (resp. $\mathcal{C}(\Delta) < T$), the presence (resp. absence) of the CSS in $y(k+\Delta)$ is declared. The performance of cross correlation can be quantified in terms of false positives and false negatives. Specifically, after deciding on a threshold T, a false positive is declared when $C(\Delta) < T$ even though the CSS is present within $[y(\Delta)\ldots y(\Delta+L-1)]$, and a false negative is declared when $C(\Delta) \geq T$ even though the CSS is absent. Cross-correlation detection provides a processing gain, which is linear in the length of the correlated sequence [15]. This means that a sequence of length $2L$ obtains a value of $C(\Delta)$ twice as large a sequence of length L, i.e., it is considerably more likely to exceed T.

Acknowledgment

While the authors employ the name 802.11ec, their work does not represent an IEEE standard. They use this name for reasons that are apparent in the body of this paper.

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std 802.11-2007–Part 11, 2007 [Online]. Available: http://standards.ieee.org/about/get/802/802.11.html

[2] "IEEE P802.11 Sub 1 GHz Study Group," 2010 [Online]. Available: http://www.ieee802.org/11/Reports/tgah_update.htm

[3] Rice University, Houston, TX, USA, "Rice University WARP project," [Online]. Available: http://warp.rice.edu

[4] V. Barghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," in *Proc. ACM SIGCOMM*, 1994, pp. 212–225.

[5] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[6] J. Camp, E. Aryafar, and E. Knightly, "Coupled 802.11 flows in urban channels: Model and experimental evaluation," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[7] D. Chu, "Polyphase codes with good periodic correlation properties," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 531–532, Jul. 1972.

[8] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *Proc. IEEE INFOCOM*, 2008, pp. 1022–1030.

[9] D. S. J. De Couto, D. Aguayo, J. C. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. ACM MobiCom*, 2003, pp. 134–146.

[10] E. O. Elliot, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 1977–1997, Sep. 1963.

[11] S. Gollakota and D. Katabi, "Zig-zag decoding: Combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM*, 2008, pp. 159–170.

[12] S. Hong and S. Katti, "DOF: A local wireless information plane," in *Proc. ACM SIGCOMM*, 2011, pp. 230–241.

[13] R. K. Jain, D. M. W. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research, Tech. Rep. TR-301, 1984.

[14] P. Karn, "MACA—A new channel access method for packet radio," in *Proc. ARRL Comput. Netw. Conf.*, 1990, pp. 134–140.

[15] S. M. Kay, *Fundamentals of Statistical Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1998, vol. 2.

[16] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[17] K. LaCurts and H. Balakrishnan, "Measurement and analysis of real-world 802.11 mesh networks," in *Proc. ACM IMC*, 2010, pp. 123–136.

[18] T. Li, M. K. Han, A. Bhartia, L. Qiu, E. Rozner, and Y. Zhang, "CRMA: Collision-resistant multiple access," in *Proc. ACM MobiCom*, 2011, pp. 61–72.

[19] E. Magistretti, K. K. Chintalapudi, B. Radunovic, and R. Ramjee, "WiFi-Nano: Reclaiming WiFi efficiency through 800 ns slots," in *Proc. ACM MobiCom*, 2011, pp. 37–48.

[20] B. Radunovic, D. Gunawardena, P. Key, A. Proutiere, N. Singh, V. Balan, and G. Dejean, "Rethinking indoor wireless mesh design: Low power, low frequency, full-duplex," in *Proc. IEEE WiMesh*, 2010, pp. 1–6.

[21] M. Richards*, Fundamentals of Radar Signal Processing*. New York, NY, USA: McGraw-Hill, 2005.

[22] D. V. Sarwate and M. B. Pursley, "Cross-correlation properties of pseudo-random and related sequences," *Proc. IEEE*, vol. 68, no. 5, pp. 593–619, May 1980.

[23] S. Sen, R. R. Choudury, and S. Nelakuditi, "CSMA/CN: Carrier sense multiple access with collision notification," in *Proc. ACM MobiCom*, 2010, pp. 25–36.

[24] S. Sen, R. R. Choudury, and S. Nelakuditi, "No time to countdown: Migrating backoff to the frequency domain," in *Proc. ACM MobiCom*, 2011, pp. 241–252.

[25] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G. Voelker, "SAM: Enabling practical spatial multiple access in wireless LAN," in *Proc. ACM MobiCom*, 2009, pp. 49–60.

[26] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II—The hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1417–1433, Dec. 1975.

[27] R. Van Nee and R. Prasad*, OFDM for Wireless Multimedia Communications*. Norwood, MA, USA: Artech House, 2000.

[28] A. Willig, M. Kubish, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *IEEE Trans. Ind. Electron.*, vol. 49, no. 6, pp. 1265–1282, Dec. 2002.

[29] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A library for parallel simulation of large-scale wireless networks," in *Proc. IEEE PADS*, 1998, pp. 154–161.

[30] X. Zhang and K. Shin, "E-MiLi: Energy-minimizing idle listening in wireless networks," in *Proc. ACM MobiCom*, 2011, pp. 205–216.

Eugenio Magistretti (S'04) received the Laurea and Doctorate degrees from the University of Bologna, Bologna, Italy, in 2003 and 2007, respectively, and the Ph.D. degree from Rice University, Houston, TX, USA, in 2013, all in computer engineering.

His main research interests are in the area of wireless MAC protocols, with a focus on design, performance modeling, and evaluation.

Omer Gurewitz (S'00–M'05) received the B.Sc. degree in physics from Ben Gurion University, Beer Sheva, Israel, in 1991, and the M.Sc. and Ph.D. degrees in electrical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 2000 and 2005, respectively.

He is an Assistant Professor with the Department of Communication Systems Engineering, Ben Gurion University. Between 2005 and 2007, he was a Post-doctoral Researcher with the Electrical and Computer Engineering (ECE) Department, Rice University, Houston, TX, USA. His research interests are in the field of performance evaluation of wired and wireless communication networks. His current projects include cross-layer design and implementation of medium access protocols for 802.11 as well as 802.16 (WiMAX) standards.

Edward W. Knightly (S'91–M'96–SM'04–F'09) received the B.S. degree from Auburn University, Auburn, AL, USA, in 1991, and the M.S. and Ph.D. degrees from the University of California, Berkeley, CA, USA, in 1992 and 1996, respectively, all in electrical engineering.

He is a Professor of electrical and computer engineering with Rice University, Houston, TX, USA. His research interests are in the areas of mobile and wireless networks and high-performance and denial-of-service resilient protocol design.

Prof. Knightly is a Sloan Fellow. He served as Associate Editor of numerous journals and special issues including the IEEE/ACM TRANSACTIONS ON NETWORKING and IEEE JOURNAL ON SELECTED AREAS OF COMMUNICATIONS Special Issue on Multi-Hop Wireless Mesh Networks. He served as Technical Co-Chair of IEEE INFOCOM 2005 and General Chair of ACM MobiHoc 2009 and ACM MobiSys 2007, and served on the program committee for numerous networking conferences including ICNP, INFOCOM, MobiCom, and SIGMETRICS. He is a recipient of a National Science Foundation CAREER Award. He received the Best Paper Award from ACM MobiCom 2008.