

CoopMAC: A Cooperative MAC for Wireless LANs

Pei Liu, Zhifeng Tao, Sathya Narayanan, Thanasis Korakis, Shivendra Panwar

TR2007-018 February 2007

Abstract

Due to the broadcast nature of wireless signals, a wireless transmission intended for a particular destination station can be overheard by other neighboring stations. A focus of recent research activities in cooperative communications is to achieve spatial diversity gains by requiring these neighboring stations to retransmit the overheard information to the final destination. In this paper we demonstrate that such cooperation among stations in a wireless LAN (WLAN) can achieve both higher throughput and lower interference. We present the design for a medium access control protocol called CoopMAC, in which high data rate stations assist low data rate stations in their transmission by forwarding their traffic. In our proposed protocol, using the overheard transmissions, each low data rate node maintains a table, called a CoopTable, of potential helper nodes that can assist in its transmission. During transmission, each low data rate node selects either direct transmission or transmission through a helper node in order to minimize the total transmission time. Using analysis, simulation and testbed experimentation, we quantify the increase in the total network throughput, and the reduction in delay, if such cooperative transmissions are utilized. The CoopMAC protocol is simple and backward compatible with the legacy 802.11 system. In this paper, we also demonstrate a reduction in the signal-to-interference ratio in a dense deployment of 802.11 access points, which in some cases is a more important consequence of cooperation.

IEEE Journal on Selected Areas in Communication (JSAC)

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

CoopMAC: A Cooperative MAC for Wireless LANs

Pei Liu, Zhifeng Tao, Sathya Narayanan, Thanasis Korakis, and Shivendra S. Panwar

Abstract—Due to the broadcast nature of wireless signals, a wireless transmission intended for a particular destination station can be overheard by other neighboring stations. A focus of recent research activities in cooperative communications is to achieve spatial diversity gains by requiring these neighboring stations to retransmit the overheard information to the final destination. In this paper we demonstrate that such cooperation among stations in a wireless LAN (WLAN) can achieve both higher throughput and lower interference. We present the design for a medium access control protocol called CoopMAC, in which high data rate stations assist low data rate stations in their transmission by forwarding their traffic. In our proposed protocol, using the overheard transmissions, each low data rate node maintains a table, called a CoopTable, of potential helper nodes that can assist in its transmissions. During transmission, each low data rate node selects either direct transmission or transmission through a helper node in order to minimize the total transmission time. Using analysis, simulation and testbed experimentation, we quantify the increase in the total network throughput, and the reduction in delay, if such cooperative transmissions are utilized. The CoopMAC protocol is simple and backward compatible with the legacy 802.11 system. In this paper, we also demonstrate a reduction in the signal-to-interference ratio in a dense deployment of 802.11 access points, which in some cases is a more important consequence of cooperation.

Index Terms—IEEE 802.11, medium access control, protocol design and analysis, multi-rate, rate adaptation, cooperative networking, cross-layer design, bridges.

I. INTRODUCTION

A. Cooperative Communication: A New Paradigm

THE BURGEONING demand for mobile data networks has highlighted the bandwidth constraint of wireless networks. Wireless links have always had orders of magnitude less bandwidth than their wireline counterparts. Interference and signal loss due to distance and fading severely reduce the total throughput achieved in wireless networks. While newer technologies such as multiple-input multiple-output systems (MIMO) increase the number of bits per hertz of bandwidth, it is not possible to integrate several antennas on handheld mobile devices because of their size and weight limitations. A

Manuscript received February 1, 2006; revised June 20, 2006. This work is supported by National Science Foundation under award 0520054, and in part by the New York State Center for Advanced Technology in Telecommunications (CATT) and the Wireless Internet Center for Advanced Technology (WICAT), an NSF Industry/University Cooperative Research Center. Some results contained in this paper have been previously presented at NYMAN'05 [1], WCNC'05 [2], ICC'05 [3] and ICC'06 [4], respectively.

Pei Liu, Thanasis Korakis, and Shivendra S. Panwar are with the Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, NY 11201 (e-mail: pliu01@utopia.poly.edu, korakis@poly.edu, panwar@catt.poly.edu).

Zhifeng Tao is with Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139 (e-mail: tao@merl.com).

Sathya Narayanan is with Panasonic Princeton Laboratory, Two Research Way, 3rd Floor, Princeton, NJ 08540 (e-mail: sathya@research.panasonic.com).

Digital Object Identifier 10.1109/JSAC.2007.070210.

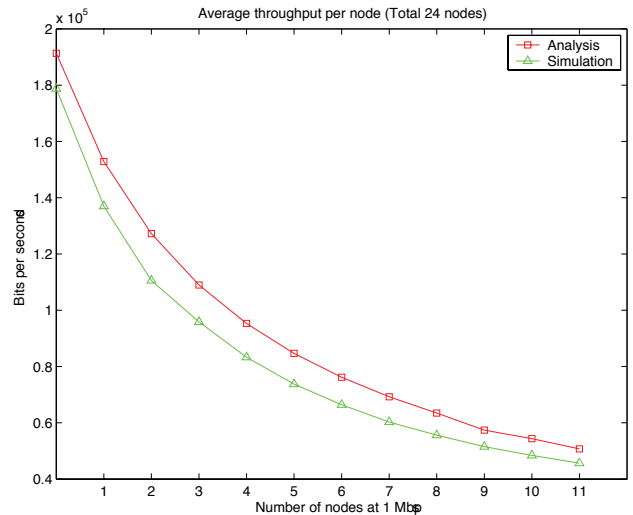


Fig. 1. Effect of slow stations.

significant amount of research on cooperative communication techniques [5]–[8] is being developed to allow stations to cooperate in their transmissions in order to improve the overall performance of the network. Since a transmission in the wireless channel is overheard by neighboring stations, these neighboring stations can process these signals and re-transmit them in order to facilitate better reception. The destination combines the signals received from the source and the helper, thus creating spatial diversity and robustness against channel variations due to fading.

Inspired by this work, we suggest introducing cooperation at the MAC layer. A few examples are provided below to illustrate the opportunities for performance improvement enabled by such MAC layer cooperation.

- Due to the shadowing and fading effects in a wireless LAN environment, the direct link between the access point (AP) and the mobile station is not always usable or is of low quality. This is especially true in an indoor environment, where most of the mobile stations do not have line-of-sight connections to the AP. Under such circumstances, it may be possible to have a third station act as a virtual antenna and opportunistically forward packets for the source station. However, a modification to the existing MAC layer design is needed to enable such cooperation.
- Heusse et al. [9] and Sadeghi et al. [10] demonstrated that the presence of a few low data rate stations will have an adverse effect on the overall throughput of the network. In [1], we discussed the potential benefit by enabling a bridging-like multihop transmission to mitigate the negative effects of slow stations. We show the negative effect of stations operating at a lower data rate on the

average throughput per node for an 802.11b system in Fig. 1, for a network with 24 nodes (Readers are referred to the Section III for the detailed analysis). As can be seen from Fig. 1, the presence of stations at 1 Mbps reduces the average throughput of all the stations in the network. This is because a 1 Mbps station takes roughly 11 times more transmission time than an 11 Mbps station to transmit the same number of bits. One of the aims of the proposed CoopMAC protocol is that it allows the high rate stations to help stations that can only sustain a low data rate to mitigate this effect. However, modifications are needed to existing MAC protocols, like the most widely used IEEE 802.11 DCF mechanism, to enable such cooperation.

To fully leverage the benefits of cooperation at MAC layer, in this paper we propose a new protocol based upon the existing IEEE 802.11 Distributed Coordination Function (DCF). The new protocol, which is called *CoopMAC*, enables cooperation in 802.11 networks. The details of CoopMAC will be further elaborated in section II. An analytical model has also been developed for CoopMAC, which is presented in section III. To validate the analysis, we performed extensive simulation studies, the results of which are presented in section IV. Along with improvement in throughput and media access delay in a saturated network, we demonstrate an increase in the signal-to-interference (SIR) ratio in proximal 802.11 cells. For a given network throughput, cooperative transmissions reduce the total transmission time leading to a decrease in the interference experienced in proximal cells. This reduction in inter-cell interference is explained in section V.

There are two potential costs for the helper stations introduced by the CoopMAC protocol. An opportunity cost, which is due to the transmission opportunities a helper may have to lose in order to forward data for another station, and an actual cost, which is the energy expense involved in transmitting data for other stations. It will become obvious that there is *no opportunity cost* because the helper nodes are allowed to access the channel without contention when forwarding data for other stations. We investigated the energy expense for helper nodes in participating in cooperative MAC protocols [11] and surprisingly found that there are *potential energy savings* for the helper nodes. This counter-intuitive result is due to the idle energy savings at the helper as cooperation reduces the amount of time a helper has to idle before gaining access to the channel for its own transmissions. A brief description of the energy model used and the results are discussed in section VI. Apart from the detailed analytical and simulation work, we also implemented the basic features of CoopMAC in a small scale testbed. Using our initial implementation, we conducted a series of experiments to measure the benefits of cooperation at the transport layer using file transfer tools [4]. In section VII, we have presented a brief description of the implementation effort, experiments and their results.

Our simulation study on the potential throughput improvement from using more than two-hop transmissions demonstrate that there is only a marginal potential for improvement over the two-hop case. Due to the complexity of designing a practical protocol to support more than two-hop transmissions we chose to investigate only two-hop transmissions.

Before we delve into the protocol details of CoopMAC, the related work is first reviewed in section I-B. A summary of pertinent features of the IEEE 802.11 protocol is provided in section I-C, for the sake of completeness.

B. Related Work

The research community has just started to explore the new opportunities introduced by cooperation. Several other papers also address cooperation in 802.11 systems. For example, [12] looks at cooperation from a different perspective and proposes a cooperative MAC protocol for dense wireless networks. During the collision resolution process, the stations in this new MAC protocol cooperate with each other to choose proper backoff windows so that short-term fairness can be achieved with no compromise in throughput performance. Another way to cooperate is exemplified in a new path selection algorithm called *Divert* [13], which places multiple access points in the same cell so that mobile stations can adaptively switch to an alternate access point in real time, if the channel to the current access point degrades to an unacceptable level. Thus, by letting a *secondary* AP cooperate with a *primary* AP, adverse channel conditions can be circumvented. The authors of [14] proposed the *r*PCF protocol, which enables multi-hopping in the PCF mode. However, the PCF mode is seldom used and has limited applications. For the DCF protocol, an approach similar to ours, called *r*DCF, was independently proposed in [15]. The *r*DCF protocol enables packet relaying in the ad hoc mode of 802.11 systems by requesting each station to broadcast the rate information between stations explicitly. Each station chooses a rate as described in [16]. The CoopMAC protocol proposed in this paper differs from *r*DCF primarily in that CoopMAC is backward compatible with the 802.11 protocol. We also study the impact of cooperation on inter-cell interference, which has not been considered in any previous work. A relay based adaptive auto rate (RAAR) protocol proposed in [17] suggests the use of central control at the AP to select relay nodes. The RAAR protocol also allows for transmission of multiple back-to-back packets by the high data rate relay nodes and hence affects the long term channel access fairness of the MAC. Our proposal is a distributed mechanism similar to DCF and does not change the long term channel access fairness quality of the underlying 802.11 MAC. Investigations on the advantages of relaying in HiperLAN/2 and its protocol extensions can be found in [18]–[23]. All of these papers take advantage of the TDMA nature of HiperLAN/2 and use a centralized scheduling mechanism. Since CoopMAC is based on the 802.11 MAC, it is based on carrier sensing multiple access (CSMA) and is therefore a completely distributed media access protocol.

C. The IEEE 802.11 Protocol

The IEEE 802.11 protocol employs carrier sense multiple access with collision avoidance (CSMA/CA) as its medium access protocol for the distributed coordination function (DCF) mode. In this mode, each station (STA) can initiate a data transmission by itself. Channel sensing before packet transmission is essential to avoid collisions. If one station has data packet to send, it will first sense the channel to make sure the

channel is clear before the actual transmission starts. Since not all stations can hear each other, even if the channel is sensed to be free, a collision may occur. Thus virtual carrier sensing is also employed with the use of the Request To Send (RTS) and Clear To Send (CTS) frames to reserve channel time for the transmitting stations. These two control frames broadcast the channel reservation information to the whole network. Any station will be able to hear at least one of these control packets and use them to calculate the time needed for the data packet transmission. A Network Allocation Vector (NAV) is used by all the stations to discover the time for which the channel is going to be free.

The rate adaptation algorithm, which is used to choose the optimum rate, is not prescribed by the IEEE standard and left to the implementation. Very few algorithms have been published in this area. The first published algorithm is Auto Rate Fallback (ARF) [24], which switches to a higher rate if a fixed number of successful packets have been sent and falls back to a lower rate after several consecutive packet losses. Receiver Based Auto Rate (RBAR) [16] measures the received signal-to-noise ratio (SNR) of the RTS frame at the receiver and compares the SNR with a set of pre-defined thresholds to choose the best modulation scheme. Since this scheme measures the channel quality at the receiver at the time instant just before the data packet transmission, it is more accurate. However, it is not compatible with 802.11 because both the control and data packet format have to be modified.

II. COOPERATIVE MEDIA ACCESS CONTROL

We propose a set of new features both in the data plane and the control plane of the 802.11 MAC layer, while maintaining backward compatibility to the current MAC. A major component of CoopMAC control plane design is the mechanism for each station to learn about candidate helper stations, and the corresponding data structures used to store the information related to those identified candidates. In the data plane, a station can choose a helper from this list of potential helpers to use at the time of its transmissions, depending on the possibility of reducing the transmission time for the packet in hand. We first present the learning process and the corresponding data structure in section II-A, and then explain the cooperative operation in the data plane in section II-B.

A. Helper Detection

Each station in a basic service set (BSS) should maintain a table, referred to as the *CoopTable*, of potential helpers that can be used for assistance during transmission. Note that in the infrastructure mode, the AP has to maintain one *CoopTable* for each possible destination address, while mobile stations only need to keep one such table for the AP with which they are associated. The creation and updating of the *CoopTable* can be done by passively listening to all ongoing transmissions. As each station in an 802.11 network is required to check the packet header of all the packets it receives in order to pick up the packets intended for itself, this requirement does not require additional hardware. These stations are also required to decode the entire Request-To-Send (RTS), Clear-To-Send (CTS) and acknowledgment (ACK) frames to get the channel

reservation information to avoid the hidden node problem. The control frames and headers of data frames are always modulated at the base rate (e.g., 1 Mbps for 802.11b and 6 Mbps for 802.11a and 802.11g), so that all stations within the transmission range will be able to receive this information successfully.

When a transmission from a station (denoted by S_h) is overheard, a CoopMAC station S_s estimates the channel condition (e.g. path loss) between the sender of that packet and itself by measuring the received signal strength. Since all stations use the same frequency band for transmission and reception, the channel between any two stations is assumed to be symmetric. Path loss can be calculated by subtracting the transmission power (in dB), which is typically fixed for all stations, from the received signal power (in dB). The availability of such information is supported by the IEEE 802.11k [25] protocol. By checking the threshold value, which is pre-calculated and guarantees a certain bit error rate for each modulation scheme, we can find the corresponding data rate between S_h and S_s , denoted by R_{sh} . When station S_s overhears a data packet transmission between a pair of other stations (from S_h to S_d), it will identify the data rate used for this transmission from the Physical Layer Convergence Procedure (PLCP) header. This rate will be referred to as R_{hd} .

The fields contained in the *CoopTable* are shown in Fig. 2. Entries are ordered by the timestamp values, based on the last time a packet from that station is overheard. A helper station is stored in the *CoopTable* by S_s if it satisfies

$$\frac{1}{R_{sh}} + \frac{1}{R_{hd}} > \frac{1}{R_{sd}}, \quad (1)$$

where R_{sd} is the rate for direct transmission between S_s and S_d . The first column in Fig. 2, namely the *ID field*, stores the MAC address of the potential helpers learned from the RTS frames transmitted by the helper. The *Time* field stores the time of the last frame transmission heard from this helper. As described above, R_{hd} and R_{sh} store the data rate from the helper station to the destination S_d , and from the source S_s to the helper station, respectively. The last field in the table, *NumOfFailures* tracks the number of sequential failures associated with the particular helper station. When this number exceeds a predefined threshold value, which we recommend to be 3 in our protocol, the corresponding entry is removed from the *CoopTable*. The value of *NumOfFailures* is incremented after every failed transmission attempt through the helper station, and this value is reset to zero after a successful transmission through the particular helper station. Each of these entries is updated to reflect the current channel conditions and status. *CoopTable* entries can also be populated using information gained from cooperative transmissions received by a station.

B. Transmission algorithm

When a source station S_s has data of length L octets to send, it checks each entry in the *CoopTable* to decide whether to transmit through a particular helper. The transmission time for such a two hop transmission is $8L/R_{sh} + 8L/R_{hd}$, ignoring the overhead. The helper through which the minimum

ID (48bits)	Time (8bits)	R_{hd} (8bits)	R_{sd} (8bits)	NumOfFailures
MAC address of helper 1	Time the last packet heard from helper 1	Transmission rate between helper 1 and the destination	Transmission rate between the source and helper 1	Count of sequential transmission failures
.....
MAC address of helper N	Time the last packet heard from helper N	Transmission rate between helper N and the destination	Transmission rate between the source and helper N	Count of sequential Transmission failures

Fig. 2. Format of the CoopTable.

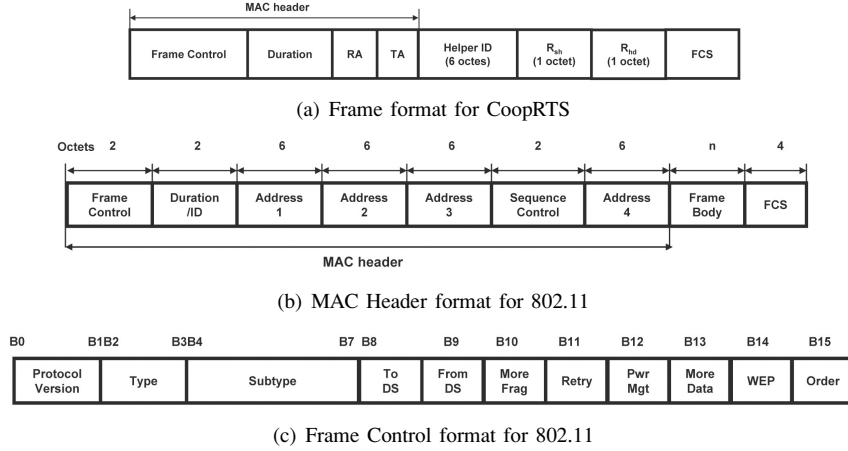


Fig. 3. Frame format.

transmission time can be achieved will be chosen as the candidate helper. If multiple stations have the same value, we choose the one with the most recent *Time* value.

As in the existing standard, the mode selection is based on a configurable RTS threshold. If the packet length is over this threshold, the RTS/CTS mode is chosen. If transmission through the chosen helper is more time efficient than a direct transmission, we will start a cooperative transmission. For the RTS/CTS mode, the condition for a cooperative transmission can be expressed as

$$\frac{8L}{R_{sh}} + \frac{8L}{R_{hd}} + T_{PLCP} + T_{HTS} + 2T_{SIFS} < \frac{8L}{R_{direct}}, \quad (2)$$

where R_{direct} is the sustainable data rate for a direct transmission from S_s to the destination S_d and T_{PLCP} , T_{HTS} and T_{SIFS} are the additional time associated with a helper-aided transmission for the physical layer overhead, HTS and SIFS, respectively. The HTS is a new message introduced to facilitate the cooperation, and will be explained in the following protocol description.

For the base mode, where the data packets are not preceded by RTS/CTS, the condition would be

$$\frac{8L}{R_{sh}} + \frac{8L}{R_{hd}} + T_{PLCP} + T_{SIFS} < \frac{8L}{R_{direct}}. \quad (3)$$

If the condition is not satisfied for any of the entries in the CoopTable, the data frame is transmitted directly to S_d .

C. CoopMAC - RTS/CTS mode

The RTS/CTS mode defined by 802.11 is extended to include an HTS (Helper ready To Send) for the helper station

to acknowledge its participation. The HTS packet has the same format as CTS in the 802.11 standard and hence the legacy stations can successfully decode this packet. The source station S_s selects one of the potential helpers S_h from the CoopTable and specifies the helper in the modified CoopRTS message. The format for CoopRTS message is shown in Fig. 3. The exchange of control messages in CoopMAC and the corresponding NAV settings are shown in Fig. 4 and 5, respectively. The flow charts at S_s , S_h and S_d are depicted in Fig. 6 and 7. A detailed description is provided below:

• Source station S_s

- 1) Whenever there is at least one packet buffered in the queue, S_s should search for a helper candidate in the CoopTable. If a helper entry is successfully found, S_s sends a CoopRTS message with the *helper ID* in the *Address 1* field to specify the helper being selected. Besides, R_{sh} and R_{hd} also should be included in the corresponding fields of CoopRTS, indicating the expected data rates between S_s and S_h , and between S_h and S_d , respectively. If the table lookup yields a failure, the regular IEEE 802.11 MAC procedure for data transmission should be followed. We denote the time duration for a RTS, CTS and ACK frame is T_{RTS} , T_{CTS} and T_{ACK} , respectively. The *duration* field in the CoopRTS is given by

$$Duration_{CoopRTS} = 4T_{SIFS} + T_{CTS} + \frac{8L}{R_{direct}} + T_{PLCP} + T_{ACK}. \quad (4)$$

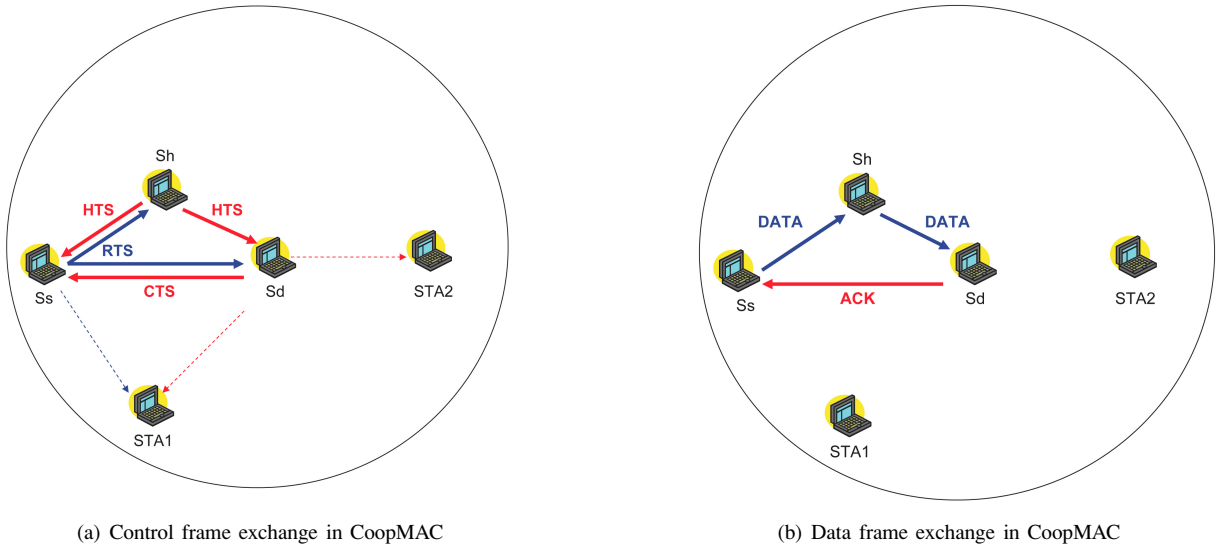
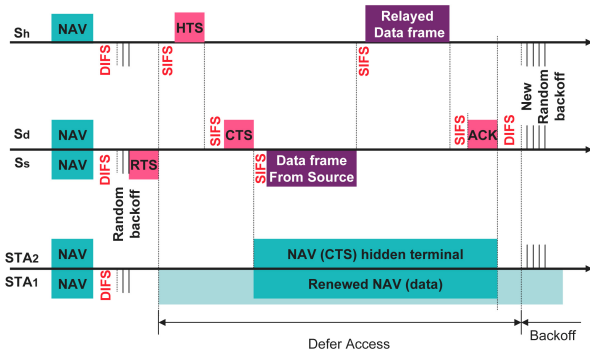
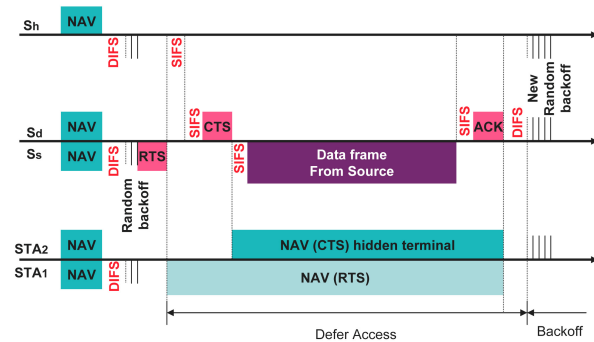


Fig. 4. Message flow in CoopMAC.



(a) NAV for CoopMAC with Helper



(b) NAV for CoopMAC without Helper

Fig. 5. NAV settings in CoopMAC.

- 2) If neither an HTS from S_h nor a CTS from S_d is heard after $2SIFS + CTS$ time, or a CTS is lost after an HTS was sent by the helper, S_s should perform regular random backoff, as if it encountered a collision.
- 3) If S_s does not receive any HTS message from S_h , but does hear a CTS from S_d , it should then send the data, using a direction transmission. To update the CoopTable, S_s should increment

$NumOfFailures$, if the HTS message is not received after a $SIFS$ time. If the value of $NumOfFailures$ is greater than the threshold (i.e., 3), S_s should remove the entry from CoopTable.

- 4) If both HTS and CTS messages are received, S_s sends the data to S_h at the rate of R_{sh} , and set the ACK timeout as follows:

$$ACKTimeout_H = 2T_{SIFS} + \frac{8L}{R_{hd}} + T_{PLCP} + T_{ACK}. \quad (5)$$

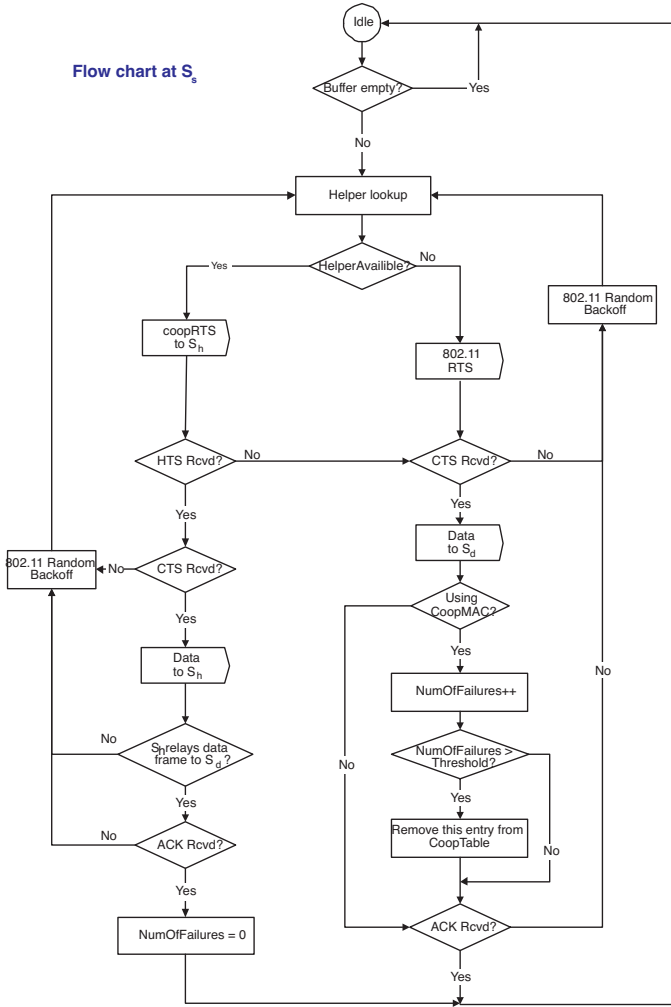
S_s should also reset $NumOfFailures$ in the CoopTable for the corresponding entry.

- 5) If an acknowledgment (ACK) is not received after an ACK timeout, S_s should perform random backoff, following the legacy 802.11 protocol. Otherwise, S_s should declare a success, and handle the next packet in its queue.

• Helper station S_h

- 1) If S_h receives a CoopRTS message, whose *Address I* field contains its MAC address, then S_h should verify whether the rate R_{sh} between itself and S_s , and a rate R_{hd} between itself and S_d suggested in the CoopRTS message are sustainable. If yes, it then sends an HTS message back to S_s , after a $SIFS$ time, with the duration field calculated by Eq. 6 (next page).
- 2) After sending the HTS to S_s , S_h should run a timer of value $T_{SIFS} + T_{CTS}$, and expect a CTS from S_d . If S_h does receive such a CTS, it then should wait for the data packet from S_s to arrive $SIFS$ time after the CTS message. If S_h does not receive either the CTS message or the data packet as expected, it should assume that the data transmission was aborted, and revert to the initial state. When the data packet to be forwarded arrives, S_h should forward the data packet to S_d at the rate R_{hd} , a $SIFS$ interval after the completion of the reception.

$$Duration_{HTS} = 4T_{SIFS} + T_{CTS} + \frac{8L}{R_{sh}} + \frac{8L}{R_{hd}} + 2T_{PLCP} + T_{ACK} \quad (6)$$

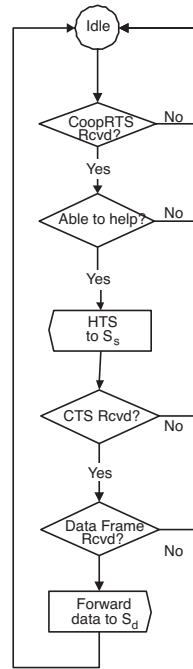
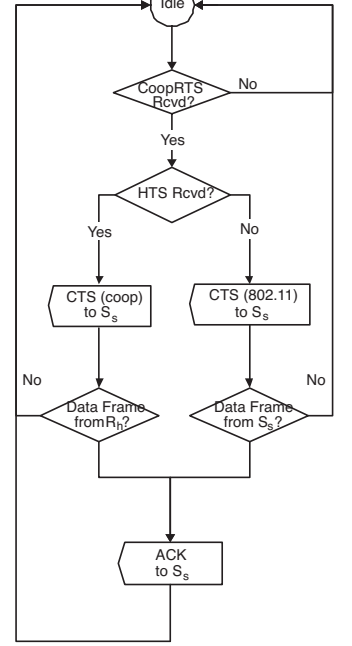
Fig. 6. Flow chart at the source station S_s .

3) If S_h cannot support rates R_{sh} and R_{hd} , S_h simply goes back to the initial state.

• Destination station S_d

- 1) If S_d receives a CoopRTS, whose RA field is set to the MAC address of S_d , S_d should wait for the corresponding HTS message from S_h .
- 2) If S_d hears an HTS message from S_h , it transmits a CTS message back to S_s after a SIFS interval, with the duration field calculated using (7) below. A data timer $SIFS + 8L/R_{sh} + T_{PLCP} + SIFS$ corresponding to the expected time of the arrival of data packet is initiated. If the data frame does not arrive before this timer expires, it assumes that the data transmission was aborted and goes back to the initial state.

$$Duration_{CTS-H} = 3T_{SIFS} + \frac{8L}{R_{sh}} + \frac{8L}{R_{hd}} + 2T_{PLCP} + T_{ACK}. \quad (7)$$

Flow chart at S_h Flow chart at S_d Fig. 7. Flow chart at S_h and S_d .

- 3) If the HTS message is not received by S_d after a SIFS interval, S_d follows the standard 802.11 approach and transmits the CTS message after another SIFS interval, with the duration field defined by

$$Duration_{CTS-NH} = 2T_{SIFS} + \frac{8L}{R_{direct}} + T_{PLCP} + T_{ACK}. \quad (8)$$

D. CoopMAC - Data transmission

Each CoopMAC station should be able to discriminate whether a packet is for itself or is to be forwarded to another station. In a RTS/CTS protected data transmission, each station will be able to do so. However, in the base mode operation of 802.11 MAC, CoopMAC allows the nodes to transmit a data frame directly to one of the potential helper nodes without going through the RTS/CTS procedure. Thus we need to have a unique CoopMAC data frame.

The Address 4 field in the IEEE 802.11 frame format (Fig. 3(b)) is never used for data frames, except when the data frame is exchanged between APs, where the *toDS* and *fromDS* subfields within the *frame control* field are both set to 1. We propose the following frame format for data transmission both in the base mode and in the CoopRTS-HTS-CTS mode. We retain the same functionalities for *toDS* and *fromDS* while utilizing the reserved data frame *Subtype* value of 1000 for CoopMAC data frames. In the first hop, source station S_s puts the helper S_h address in the *Address 1* field of the MAC header and the final destination address S_d in *Address 4*.

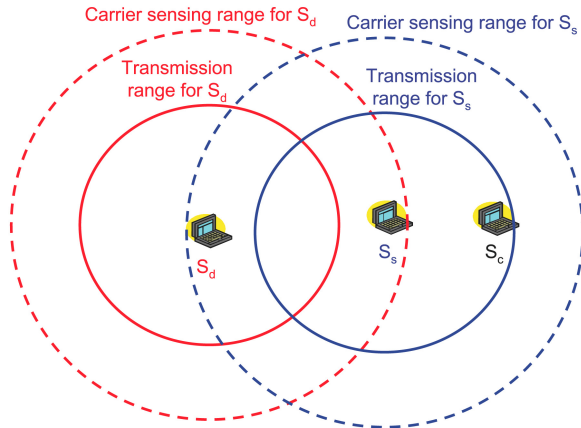


Fig. 8. Network scenario.

When the packet arrives at the helper, the helper will move the address of S_d in *Address 4* to *Address 1*, recalculate the frame check sequence (FCS) and forward the data frame to the final destination S_d after a SIFS interval. S_d sends an ACK message directly to S_s . The data flow is shown in Fig. 4. In the case where an ACK message is not received by S_s , it must increment the *NumOfFailures* and remove the potential helper station from its *CoopTable*, if *NumOfFailures* $>$ *Threshold*.

III. ANALYSIS OF COOPMAC

In this section, we will derive an expression for the saturated throughput of the CoopMAC protocol based on 802.11b physical layer specifications. This result can be easily extended to 802.11a/g and later physical layer extensions.

A. Cooperative Regions

Assume the maximum transmission range for 11 Mbps, 5.5 Mbps, 2 Mbps, 1 Mbps are r_{11} , $r_{5.5}$, r_2 , r_1 meters respectively and n mobile stations are uniformly distributed in the coverage area. If a helper is within r_x meters of the source station and r_y meters of the destination, it can help the transmission in a two hop manner using rates x and y Mbps.

The overlap area for two circles, given that the distance between the center of the two circles is l , and their radii are r_1 and r_2 respectively, is

$$S_{r_1, r_2}(l) = r_1^2 \arcsin h/r_1 + r_2^2 \arcsin h/r_2 - hl, \quad (9)$$

where $h = \sqrt{2r_1^2 r_2^2 + 2(r_1^2 + r_2^2)l^2 - (r_1^4 + r_2^4) - l^4}/2l$.

Fig. 9 shows the cooperative region for our proposed scheme. If a helper exists in region $A_{x,y}$, station S_s can transmit its packets in two hops using rates x Mbps and y Mbps to the AP. Since the stations are uniformly distributed in the coverage area, the probability that a station is located in one of these areas is listed in Eq. 10 (next page), where r is the distance from S_s to the AP.

When choosing the helper, we prefer the one that will use the least transmission time, i.e., the one with minimum $1/R_{sh} + 1/R_{hd}$. For example, we prefer two hops, both at

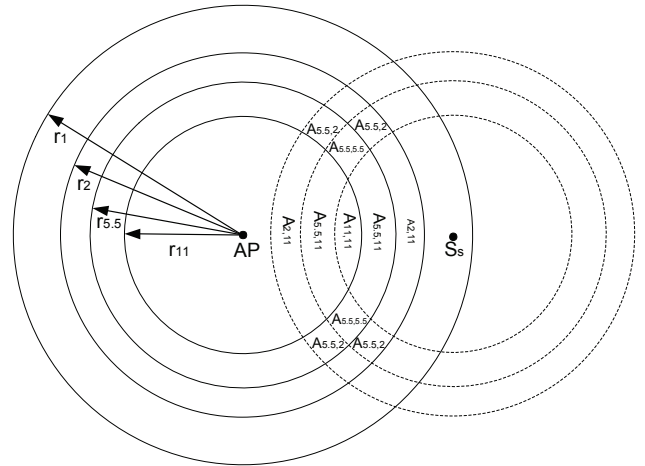


Fig. 9. Cooperative regions for CoopMAC.

rate 11 Mbps, over one hop at 11 Mbps and the other at 5.5 Mbps. Thus the probability that the optimum transmission scheme for a station with a 2 Mbps direct transmission rate is through a two hop transmission with rate R_x and R_y is given by Eq. 11 (next page).

The probability that no helper is present is $1 - P_{11,11} - P_{5.5,11} - P_{5.5,5.5}$. Similarly, we can derive these probabilities for stations with a 1 Mbps direct transmission rate, which are typically at the edge of the coverage area. There is no need for helpers for stations that can sustain 5.5 or 11 Mbps to the AP.

B. Saturated Throughput

The IEEE 802.11 MAC performs an exponential backoff if a station is involved in a collision. Specifically, when there is a collision, each station picks a random number X from $[0, CW]$, where CW is the current size of the congestion window, and retransmits after X time slots. If the retransmission is successful, CW is set to CW_{min} , otherwise $CW = \min(2 \times CW, CW_{max})$. The number of retransmission stages is m . The expected time spent in the contention procedure for each station increases with the number of stations in the network.

In our analysis, we assume that the network is heavily loaded. This implies that there is always at least one packet awaiting transmission at each station. An analytical approach to compute the saturated throughput is given in [26] and [27]. We extended their analysis to the CoopMAC scheme.

From the perspective of logical medium activity, three events can take place on the wireless channel at any randomly chosen time slot. The channel may be idle for a backoff, or busy either due to a successful transmission or due to a collision. If we assume each station has an equal expected transmission probability of τ in the considered slot time, then the probability that there is at least one station transmitting in that time slot will be

$$P_{tr} = 1 - (1 - \tau)^n, \quad (12)$$

where τ can be calculated by solving the following nonlinear

$$\begin{aligned}
p_{11,11}(r) &= \frac{S_{r_{11},r_{11}}(r)}{\pi r_1^2}, \\
p_{5.5,11}(r) &= \frac{2(S_{r_{5.5},r_{11}}(r) - S_{r_{11},r_{11}}(r))}{\pi r_1^2}, \\
p_{5.5,5.5}(r) &= \frac{S_{r_{5.5},r_{5.5}}(r) + S_{r_{11},r_{11}}(r) - 2S_{r_{5.5},r_{11}}(r)}{\pi r_1^2}, \\
p_{2,11}(r) &= \frac{2(S_{r_2,r_{11}}(r) - S_{r_{5.5},r_{11}}(r))}{\pi r_1^2}, \\
p_{5.5,2}(r) &= \frac{2(S_{r_2,r_{5.5}}(r) + S_{r_{5.5},r_{11}}(r))}{\pi r_1^2} - \frac{2(S_{r_2,r_{11}}(r) + S_{r_{5.5},r_{5.5}}(r))}{\pi r_1^2}
\end{aligned} \tag{10}$$

$$\begin{aligned}
P_{11,11} &= 1 - \int_{r_{5.5}}^{r_2} \frac{2r(1 - p_{11,11}(r))^{n-1}}{r_2^2 - r_{5.5}^2} dr, \\
P_{5.5,11} &= 1 - P_{11,11} - \int_{r_{5.5}}^{r_2} \frac{2r(1 - p_{11,11}(r) - p_{5.5,11}(r))^{n-1}}{r_2^2 - r_{5.5}^2} dr, \\
P_{5.5,5.5} &= 1 - P_{11,11} - P_{5.5,11} - \int_{r_{5.5}}^{r_2} \frac{2r(1 - p_{11,11}(r) - p_{5.5,11}(r) - p_{5.5,5.5}(r))^{n-1}}{r_2^2 - r_{5.5}^2} dr
\end{aligned} \tag{11}$$

equations, which is given in [27].

$$p = 1 - (1 - \tau)^{n-1}, \tag{13}$$

$$\tau = \frac{2(1 - 2p)}{(1 - 2p)(CW_{min} + 1) + pCW_{min}(1 - (2p)^m)}. \tag{14}$$

A transmission is successful if only one station transmits in the time slot. Let P_s be the probability that a successful transmission occurs conditioned on the fact that at least one station transmits. Then

$$P_s = \frac{n\tau(1 - \tau)^{n-1}}{P_{tr}}. \tag{15}$$

Let the transmission time for one packet, if the direct transmission rate is x Mbps, be represented by T_x . Using the same method as in [28], we have

$$\begin{aligned}
T_{11} &= T_{overhead} + \frac{8L}{R_{11}}, \\
T_{5.5} &= T_{overhead} + \frac{8L}{R_{5.5}},
\end{aligned}$$

where $T_{overhead} = T_{PLCP} + T_{DIFS} + T_{RTS} + T_{CTS} + 3T_{SIFS} + T_{ACK}$.

For stations where the direct transmission rate is 2 Mbps, we have to average over the transmission time when a helper is available and when not. The total time needed is shown in Eq. 16 on following page, where $T_{CoopOH} = 2T_{PLCP} + T_{DIFS} + 5T_{SIFS} + T_{RTS} + 2T_{CTS} + T_{ACK}$ and R_x is x Mbps. We can get the average transmission time T_1 for stations with a direct rate of 1 Mbps in the same manner.

Because of the long term channel access fairness guaranteed by the CSMA/CA protocol, each station in the network will have an equal expected number of packet transmissions over a long period of time. This is assuming that all stations have a backlog of packets for transmission in the buffer. Let us denote

the fraction of stations of rate x Mbps by f_x . The average transmission time for one packet can then be calculated by

$$T_s = f_{11}T_{11} + f_{5.5}T_{5.5} + f_2T_2 + f_1T_1, \tag{17}$$

where $f_{11} = r_{11}^2/r_1^2$, $f_{5.5} = (r_{5.5}^2 - r_{11}^2)/r_1^2$, $f_2 = (r_2^2 - r_{5.5}^2)/r_1^2$ and $f_1 = (r_2^2 - r_1^2)/r_1^2$.

The saturated throughput for the network can be expressed as:

$$S = \frac{P_s P_{tr} L}{(1 - P_{tr})SLOT + P_s P_{tr} T_s + P_{tr}(1 - P_s)T_c}, \tag{18}$$

where $T_c = T_{RTS} + T_{DIFS} + SLOT$, and $SLOT$ is the time period of one time slot.

IV. SIMULATION RESULTS

To validate the performance improvement of the CoopMAC protocol, we have developed a custom event-driven simulator using the C programming language. The simulator generates events faithfully following every state transition of the 802.11 MAC including the head of line arrival, backoff count down, DIFS time, individual transmissions, RTS/CTS transmissions for both legacy and CoopMAC protocol. The accuracy of the baseline 802.11/802.11e simulator was also validated in our previous research work. As shown in Table I, the set of core parameters used in the simulation assume the default values specified in IEEE 802.11b standard.

In the simulation, mobile stations are placed randomly in a circle with a radius of 100 meters, while the access point is located at the center of the circle. Rayleigh fading with unit mean is used to model the wireless channel in a typical indoor environment. Packets are transmitted at different rates, depending on the distance between the AP and the stations. The relation between the rates and the ranges is shown in Table II. The network operates under a heavy load condition

$$T_2 = (P_{11,11} + P_{5.5,11} + P_{5.5,5.5})T_{CoopOH} + \frac{16P_{11,11}L}{R_{11}} + \frac{8P_{5.5,11}L}{R_{11}} + \frac{8P_{5.5,11}L}{R_{5.5}} + \frac{16P_{5.5,5.5}L}{R_{5.5}} + (1 - P_{11,11} - P_{5.5,11} - P_{5.5,5.5})(T_{overhead} + \frac{8L}{R_2}) \quad (16)$$

TABLE I
PARAMETERS USED IN SIMULATION.

MAC header	272 bits
PHY header	192 bits
RTS	352 bits
CTS	304 bits
ACK	304 bits
Data rate for MAC and PHY header	1 Mbps
Slot time	20 μ s
SIFS	10 μ s
DIFS	50 μ s
aCWMin	31 slots
aCWMax	1023 slots
retryLimit	6

TABLE II
PHYSICAL MODE TABLE (PATH LOSS EXPONENT (PLE) = 3).

Data Rate	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
Range ($BER \geq 10^{-5}$)	48.2 m	67.1 m	74.7 m	100 m

with three different MSDU lengths have been plotted in this figure. As expected, the saturation throughput for legacy MAC increases as the packet size grows. This is due to the fact that a lower percentage of channel time is occupied by the transmission of PHY and MAC overheads. CoopMAC performs better than the legacy 802.11b system, when the MSDU length is above a certain threshold, as explained below.

A closer examination of the relation between payload size and throughput gain is provided in Fig. 10(d), which shows results with RTS/HTS/CTS enabled for all packet lengths. When the payload size is below a certain threshold, the time saved by using two hop transmission at higher rates is canceled by the overhead entailed by the new HTS message, as well as additional PHY header and the SIFS time. For short payload sizes, the legacy system outperforms CoopMAC in terms of the aggregate throughput. When the frame size increases, the throughput gain that CoopMAC can achieve over legacy 802.11b becomes more and more significant. Fig. 10(d) also reveals that a larger number of stations usually results in a higher performance improvement, which again agrees with the similar observation made from Fig. 10(c). It is worthwhile to note that the aforementioned threshold value is usually fairly small (e.g., approximately 120 bytes in the 12-station case) for most of the network scenarios. CoopMAC therefore helps mainly to boost the delivery of long packets typically generated during file transfers. The above observation applies for the CoopMAC protocol with the RTS/HTS/CTS enabled. The threshold discussed will be lower (e.g., less than 100 bytes) for the CoopMAC operating in the base mode, where the RTS/CTS mechanism is not employed. In that case, the threshold is low enough such that applications using short frame sizes (e.g., VoIP) can also benefit from cooperation, as is further illustrated in [29].

The improvement in aggregate throughput also translates into a better performance in packet transmission delay, which is shown in Fig. 11. Service delay, which is also known as channel access delay, is defined as the duration from the time when a packet reaches the head-of-line (HOL) position, to the time it is successfully delivered. The cumulative service delay distribution for successful transmissions depicted in Fig. 11 corresponds to the network scenario where there are 8 mobile stations in a heavily loaded system. As revealed by the figure, service delay for CoopMAC is substantially lower than that for the legacy 802.11b protocol.

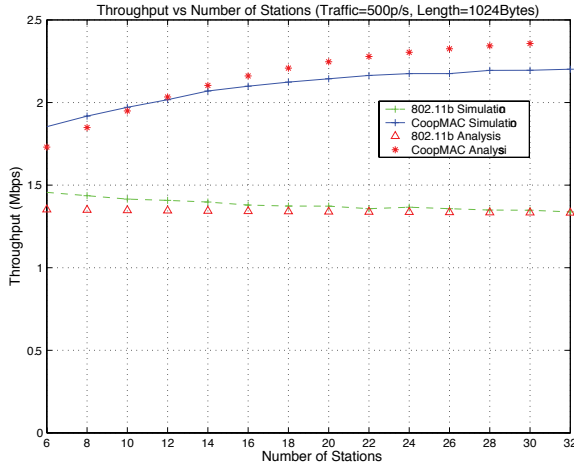
The simulation results for throughput and delay presented above are for networks with stationary nodes only. Using

and traffic is evenly distributed across all the stations in the network. Packets with fixed length arrive at the network according to a Poisson distribution. For the sake of a fair comparison, both the legacy MAC and the CoopMAC share the same minimum and maximum window sizes and retry limit.

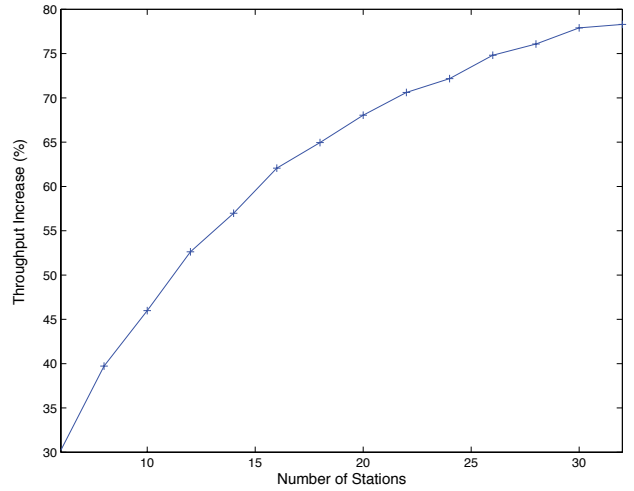
Fig. 10(a) compares the saturation throughput achieved by both legacy 802.11b MAC and CoopMAC, when the total packet arrival rate is high enough to saturate the network (500 packets/sec with 1024 bytes MSDU). As expected, the simulation results demonstrate that the throughput for both MAC protocols are significantly lower than 11 Mbps, because only a portion of the stations are able to transmit at 11 Mbps. Moreover, PHY/MAC layer overhead and collisions among transmissions also contribute to the reduced throughput.

Under most circumstances, CoopMAC can achieve higher throughput than the legacy system, as is confirmed by both simulation and analysis. More specifically, as the number of mobile stations increases, the throughput for legacy 802.11b network decreases, due to excessive collisions. On the other hand, the throughput achieved by CoopMAC increases from approximately 1.8 Mbps to 2.2 Mbps as the number of mobile stations in the network increases. At that point, the CoopMAC throughput plateaus at approximately 2.2 Mbps. This highly desirable feature of CoopMAC is attributed to the fact that as the number of stations increases, the likelihood of a low rate station finding a high rate two-hop path also increases. The growing availability of helping stations not only cancels the throughput degradation caused by increasing collisions, but also results in a substantial net increase in the aggregate network throughput. Based upon Fig. 10(a), we also plot the relative throughput increase expressed as a percentage versus the number of stations in Fig. 10(b), which demonstrates the benefits of cooperation.

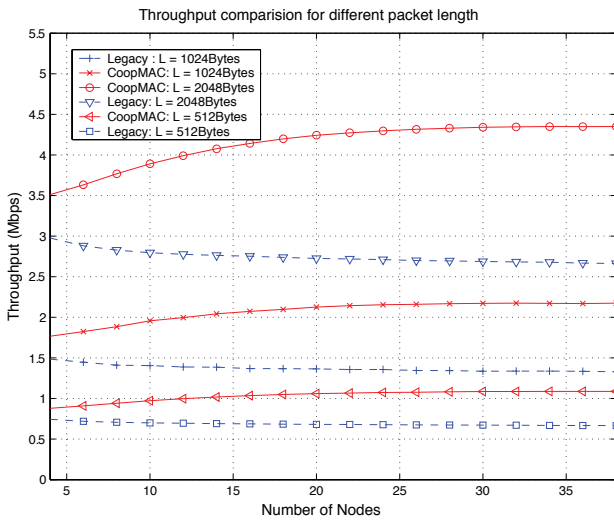
As is well known, the size of the MSDU has a major impact on the efficiency of any MAC protocol. Fig. 10(c) studies this effect on both legacy 802.11b and CoopMAC using simulation results. The throughput curves of two MAC protocols obtained



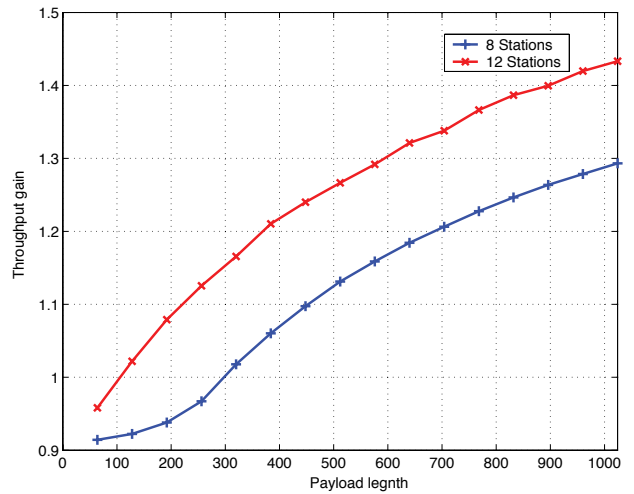
(a) Throughput for 802.11b and CoopMAC vs number of stations



(b) Throughput increase vs Number of Stations



(c) Throughput vs number of Stations under different packet length



(d) Relation between the payload size and throughput gain

Fig. 10. Throughput performance.

mobile nodes instead of stationary nodes is expected to impact the performance of CoopMAC negatively. Particularly, once stations start to move around, the rate information contained in the CoopTable of each station may become out-of-date at the time it is actually used by CoopMAC to select a helper. To characterize the effect of mobility on cooperation-related aspect of system performance, we incorporated mobility based on the *random waypoint* model described in [30] and [31] in our simulation. We present the simulation results for an eighteen node network in Figure 12. In the random waypoint model, each station begins the simulation by remaining stationary for *pause time* seconds. It then selects a random destination within the coverage area of the network and moves to that destination at a speed distributed uniformly between 0 and *max speed*. Upon reaching the destination, the station pauses again for *pause time* seconds, selects another destination, and proceeds there as previously described, repeating this behavior for the duration of the simulation.

Note that in a legacy 802.11 network, mobility would reduce the effectiveness of rate adaptation algorithm, and ultimately lead to a degraded throughput performance. Since link adaptation by itself is a challenging research undertaking,

we need to isolate the impact of mobility on our cooperative MAC from that on the link adaptation algorithm. Therefore, we use an indirect comparison approach as described below.

As shown in Fig.12, when the *pause time* increases, the throughput achieved by CoopMAC increases in a network with mobile nodes. More importantly, for a wide range of *pause time* values, CoopMAC performs better, even with mobility, than the stationary legacy 802.11 network. Since the throughput of a mobile 802.11 network is expected to be lower than that of a stationary 802.11 network, it is reasonable to conclude that the performance improvement of CoopMAC will be much better compared with legacy 802.11 protocol under a typical indoor office mobility setting. In order to understand the effect of an increase in the number of nodes in a mobile environment, we ran the same simulation with 40 nodes and found the result to be not significantly different from that shown in Fig.12.

V. DENSE DEPLOYMENT OF 802.11 NETWORKS

The IEEE 802.11b and IEEE 802.11g specifications operate in the 2.4 GHz unlicensed band, where there are 11 channels

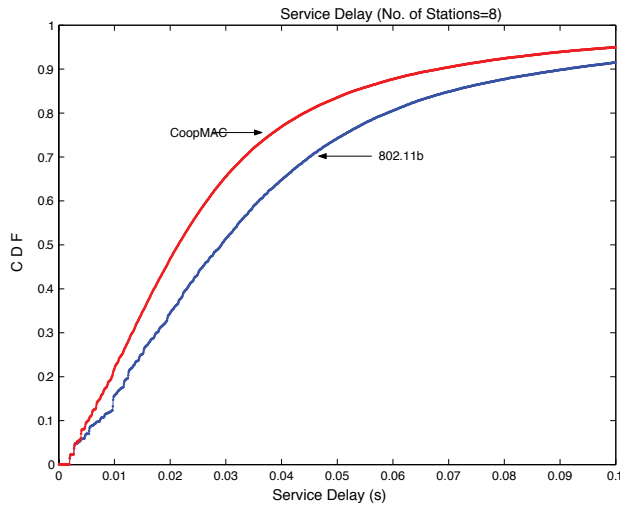


Fig. 11. Cumulative distribution for service delays.

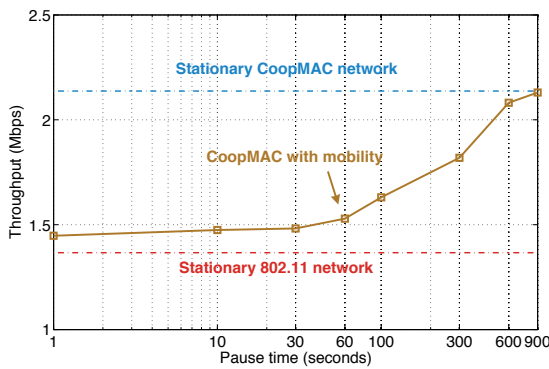


Fig. 12. Impact of mobility: max speed = 1 m/s, 18 stations, heavy load with MSDU size of 1024 bytes.

defined for operation in the US. These channels, numbered from 1 to 11, are 22 MHz wide and spaced at 5 MHz intervals. Thus if two APs use adjacent channels, there will be a very big overlap in the spectrum which will cause a lot of interference. Among all the available channels, only three channels do not overlap, i.e., channel(s) 1, 6 and 11. If we only use these three channels, as is normally done in practice, the frequency reuse factor would be 3. In a multicell environment where access points are placed close to each other, an issue is that the received packets have a higher probability of physical layer error due to interference. This effect can be alleviated by placing the APs far apart from each other. However, this is not compatible with meeting coverage or traffic load requirements. It is therefore worthwhile to evaluate network performance when it is deployed in a cellular topology designed to reuse a limited bandwidth.

Fig. 13 and 14 compare the interference for CoopMAC and 802.11b MAC in a multicell environment with a frequency reuse factor of 3. Both systems are under the same traffic load in all cells. From these figures, another advantage of cooperation becomes apparent. CoopMAC incurs about 35% less interference. Since CoopMAC is more efficient in term of throughput, the transmission time for the same amount of traffic using the CoopMAC protocol is less than that of

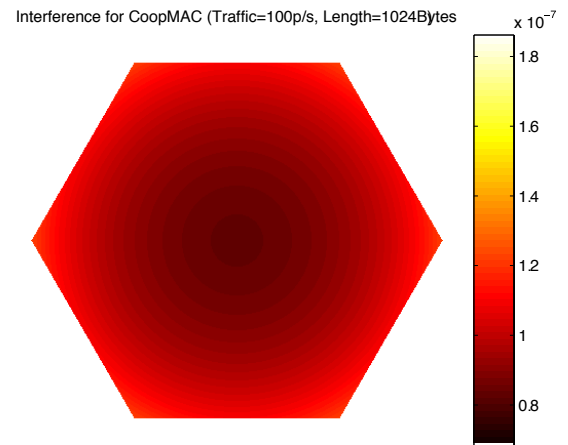


Fig. 13. Interference for CoopMAC.

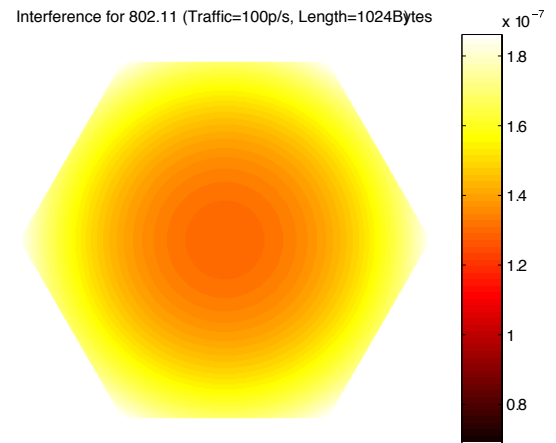


Fig. 14. Interference for 802.11b.

the legacy system, therefore reducing total energy radiated to the network. Thanks to the lower background interference, the sustainable regions for all four rates supported by IEEE 802.11b are extended.

The reduction in interference for CoopMAC system can further improve the system performance when compared with 802.11b. We calculated the achievable rate for various locations within the cell. As we can see in Fig. 15, it is obvious that the sustainable regions for all four rates are extended because of lower background interference. More stations are within the high rate regions as compared with 802.11b. The rate regions for 2 Mbps and 1 Mbps are also extended, which directly increases the coverage area. Also, the expected rate, which takes into the consideration of the probability that one or more helper exists, is much higher than the legacy systems. These results are illustrated in Fig. 15.

VI. ENERGY EXPENSE AT THE HELPER NODE

In order to look at the energy cost to a forwarding (helper) node associated with forwarding another nodes' data, we need to keep the following points in mind:

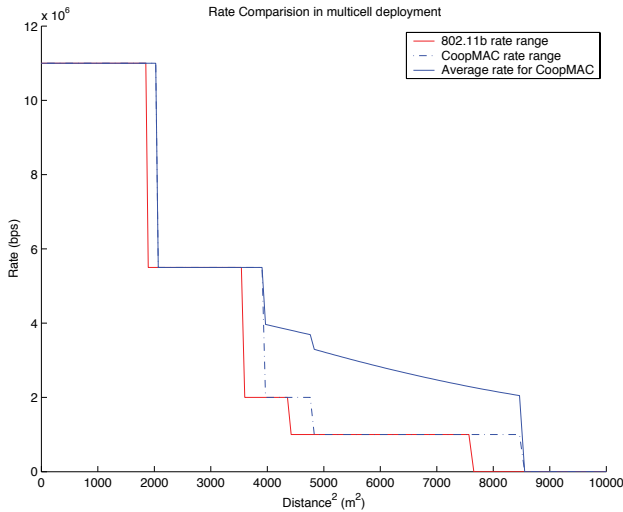


Fig. 15. Rate vs distance for 802.11b and CoopMAC in a multicell deployment.

- Similar to our discussion on throughput, we consider a saturated network with every node having a packet to transmit all the time.
- We analyze the energy expense for a given number of application bits (L) for the high-data rate node. The underlying assumption is that the node shuts down after it is done transmitting its application bits or constantly has bits to send. Obviously, if a node stays online transmitting data for other nodes without any of its own traffic to send, the bits-per-joule ratio for this node will be lower leading to higher energy cost.

Given that a node (referred to as $Node_A$) has to achieve data transfer of L bits and there are N nodes in the network, the energy used by $Node_A$ will be:

$$E = T_T P_T + T_R P_R + T_L * ((1 - F_R) P_I + F_R P_R) \quad (19)$$

where P_T , P_R and P_I are the power consumption rates during the transmission, reception and idle states of the WLAN card. T_T and T_R are the time $Node_A$ will spend transmitting frames and receiving frame respectively, while T_L is the time $Node_A$ will be listening to a packet transmission going on between two other nodes. When such a transmission between two other nodes is taking place, a fraction (F_R) of the packet will be received at $Node_A$ before it realizes the transmission is not meant for itself and switches to idle mode. This behavior is captured by the last component of the equation (19). Carvalho *et al.* [32] use a similar model for estimating the energy consumption, but does not differentiate between the states where the node is receiving a packet and where the node is idle.

Using (19), we calculated the bits-per-joule with and without forwarding for the high data rate node and compared it to the results from our simulation. As shown in Table III, we see that in a saturated network, a high data rate node can get more bits per joule if it is participating in two-hop forwarding schemes. Further details on the analysis and simulation study can be found in [11].

TABLE III
BITS-PER-JOULE (PKT LENGTH = 1024 BYTES).

11 Mbps node	W/o forw($\times 10^4$)b	With forw($\times 10^4$)b
Analysis	8.2845	8.8909
Simulation	7.8552	8.7389
5.5 Mbps node	W/o forw($\times 10^4$)b	With forw($\times 10^4$)b
Analysis	8.1544	8.2206
Simulation	7.7032	8.4592

VII. IMPLEMENTATION

A. Implementation Description

For the implementation of our protocol we used an open source Linux driver called *HostAP* [33]. HostAP is an 802.11b wireless driver, based on the *Intersil Prism 2, 2.5* or *3* chipset.

When it comes to system design, all the features specified in IEEE 802.11 MAC protocol are logically partitioned into two modules, according to the time-criticality of each task. The lower module usually operates on a wireless card as a part of firmware and fulfills the time-critical missions such as generation and exchange of RTS/CTS control messages, transmission of the acknowledgment (ACK) packet, execution of random backoff, etc. The other module, which normally assumes the form of the system driver, is responsible for more delay-tolerant control plane functions such as the management of the MAC layer queue(s), the formation of the MAC layer header, fragmentation, association, etc.

As the cooperative MAC protocol requires changes to both time-critical and delay-tolerant logics, the inaccessibility to firmware unfortunately causes additional complexity in implementation. Indeed, compromises had to be made and alternative approaches had to be pursued, due to this constraint. For illustrative purposes, the three main circumventions are outlined below.

- *Suspension of the 3-way Handshake*

As mentioned earlier, a 3-way handshake has been defined in the cooperative MAC protocol, which requires the selected helper to transmit a new control message called “Helper ready To Send” (HTS) between the RTS and CTS messages. Since the strict sequence of RTS and CTS packet has been hardwired in the firmware, an insertion of HTS becomes impossible at the driver level. As an alternative, it has then been determined that the 3-way handshake of the protocol would be entirely suspended.

- *Unnecessary Channel Contention for the Relayed Packet*

Once the channel access has been allocated to the source station, the helper should relay the packet a *SIFS* time after its reception, without any additional channel contention. Since the *SIFS* time is set to $10\mu\text{s}$ in IEEE 802.11b, any function demanding such a short delay must be implemented in the firmware. As a result, a compromise has been made in the implementation, where channel contention for the relayed packet on the second hop has to be attempted.

- *Duplicate ACK*

Each successful data exchange in the original cooperative MAC protocol involves only one acknowledgment message, which is sent from the destination to the source

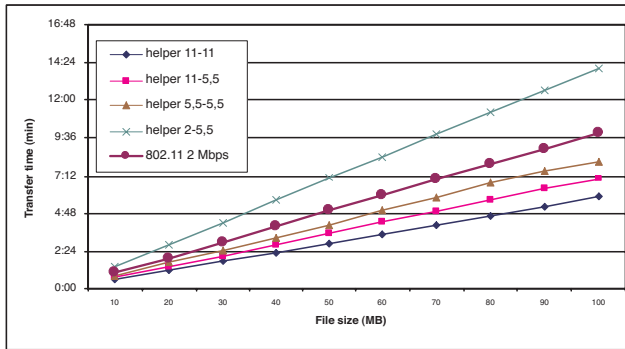


Fig. 16. File transfer times for CoopMAC vs 802.11 with a 2 Mbps direct link.

directly. Since the acknowledgment mechanism is an integral function of firmware, it is impossible to suppress the unnecessary ACK message generated by the relay station for the packet it will forward on behalf of the source. Therefore, the unwanted ACK from the relay has to be tolerated, instead of being blocked.

An alternative approach was to use broadcast transmissions for the data packets on the two hops, in order to suppress the undesirable ACKs. Unfortunately, this approach results in poor performance for the protocol as there was no acknowledgment in the MAC layer and thus no guarantee of reliable transmission. This fact forced the higher layer mechanisms (e.g., TCP) to take care of the lower layer reliability issues. Thus, we decided to follow the two ACKs approach, one from the destination to the helper and one from the helper to the transmitter, increasing the overhead of the protocol.

As an implication of the circumventions described above, a faithful implementation of cooperative MAC is anticipated to outperform the one demonstrated in this paper.

B. Experiments

In our experiments we use a basic setup of three stations, one source, one destination and a helper. We ran different experiments changing the position of the helper between the different regions in Fig.9 by forcing the data rates between the source, helper and destination nodes. For every position of the helper, large files were transferred from the source to the destination with and without cooperation from the helper node. For the file transmissions, two TCP based Unix file transfer applications, FTP and SCP were used. In this paper, we present the average transmission time from repeating each experiments 10 times.

Our first experiment was a file transfer time for a source that transmits directly to the destination at data rate of 2 Mbps was compared with the file transfer time if the same source received assistance from helper node at various higher transmission rates. The results from this experiment is shown in Fig.16. In this figure, *helper x-y* stands for a data rate x Mbps between the source and the helper and y Mbps between the helper and the destination. Potential positions of the *helper x-y* are in the area $A_{x,y}$ of Fig.9. As can be seen in Fig.16, CoopMAC performs better, resulting in shorter transmission

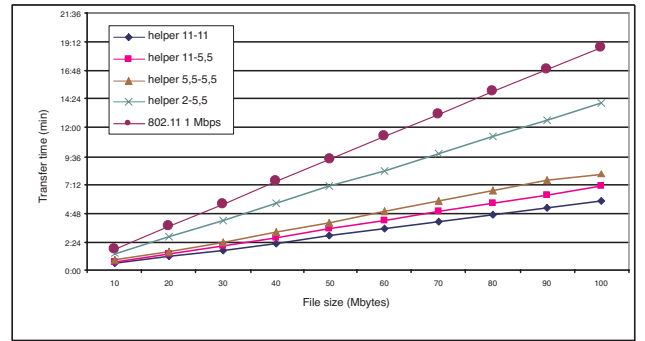


Fig. 17. File transfer time for CoopMAC vs 802.11 with a 1 Mbps direct link.

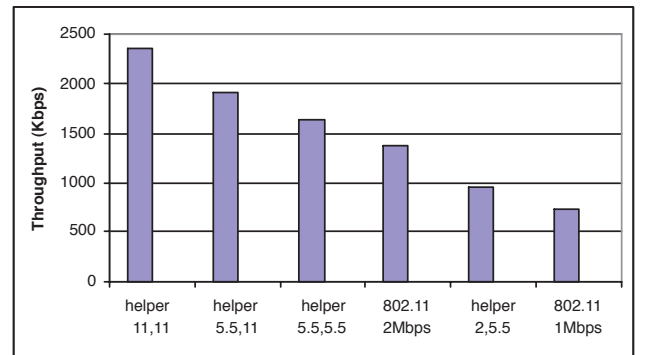


Fig. 18. End to end throughput for CoopMAC for several helpers options and 802.11 with a 1 Mbps direct link.

times for the files. The potential helpers in this case lie in the regions $A_{11,11}$, $A_{5,5,11}$, $A_{5,5,5,5}$ of Fig.9.

We repeat the same experiment, with a data rate of 1 Mbps data rate for direct transmission between the source and the destination. Results are shown in Fig.17. The cooperative MAC results in shorter transfer times, even in the case of a helper that lies in the region $A_{2,5,5}$ of Fig.9. Finally, we calculate the throughput that is achieved at the application layer. This throughput was calculated by dividing the amount of data transferred each time by the transfer time. The results shown in Fig.18 are the average values from the experiment. We must bear in mind that the calculated throughput is the application layer throughput after the TCP overhead and not the raw MAC layer throughput. Further details on the implementation effort can be found in [4].

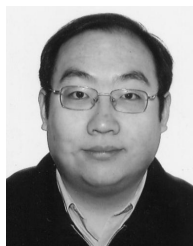
VIII. CONCLUSION

In this paper, we explore cooperation at the medium access control (MAC) layer and propose a new protocol called CoopMAC, which is based upon the existing IEEE 802.11 DCF mode. As verified by analysis and extensive simulation, CoopMAC can achieve substantial throughput and delay performance improvements, without incurring significant complexity overhead in system design. Moreover, for a given network throughput, the new protocol can reduce the interference experienced in proximal cells and thus can provide a more uniform coverage under a dense deployment. Since the helper station simply forwards the packet without looking into the contents of the MSDU, the confidentiality

of the MSDU can be maintained by encrypting the content. Access fairness is not compromised in the new MAC, since the relaying station is allowed to access the network without utilizing its own transmission opportunities. Note that the CoopMAC can be readily extended to other higher data rate extensions of 802.11, even though the current CoopMAC is evaluated for IEEE 802.11b. Through analysis and simulation [11], we have been able to demonstrate that the energy-per-bit experienced by the helper stations is decreased by participating in cooperation. This counter-intuitive result is due to the reduction in idle energy consumption incurred by the helper as it waits for its transmission opportunity while a slow node is occupying the channel. An initial implementation of the cooperative transmission has been completed and experimental results from the implementation were presented.

REFERENCES

- [1] S. Narayanan, P. Liu, and S. Panwar, "On the advantages of multi-hop extensions to IEEE 802.11 infrastructure mode," in *4th New York Metro Area Networking Workshop (NYMAN 2004)*, Sept 2004.
- [2] S. Narayanan, P. Liu, and S. Panwar, "On the advantages of multi-hop extensions to IEEE 802.11 infrastructure mode," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, (New Orleans, USA), March 2005.
- [3] P. Liu, Z. Tao, and S. Panwar, "A cooperative mac protocol for wireless local area networks," in *Proc. of IEEE ICC*, (Seoul, Korea), June 2005.
- [4] T. Korakis, S. Narayanan, A. Bagri, and S. S. Panwar, "Implementing a cooperative MAC protocol for wireless LANs," in *Proc. of IEEE ICC'06*, (Istanbul, Turkey), 2006.
- [5] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity-Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, pp. 1927-1938, November 2003.
- [6] J. N. Laneman, D. Tse, and G. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Trans. Inform. Theory*, vol. 50, pp. 3062-3080, December 2004.
- [7] M. Valenti and N. Correal, "Exploiting Macrodiversity in Dense Multihop Networks and Relay channels," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, vol. 3, pp. 1877-1882, March 2003.
- [8] I. Hammerstroem, M. Kuhn, B. Rankov, and A. Wittneben, "Space-Time Processing for Cooperative Relay Networks," in *Proc. of IEEE Vehicular Technology Conf. (VTC)*, October 2003.
- [9] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. of IEEE INFOCOM*, (San Francisco, USA), March-April 2003.
- [10] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate Ad Hoc network," in *Proc. ACM MobiCom'02*, September 2002.
- [11] S. Narayanan and S. S. Panwar, "To forward or not to forward - that is the question," to appear in Special Issue on Cooperation in Wireless Networks, Springer - Wireless Personal Communications.
- [12] C. Cetinkaya and F. Orsun, "Cooperative medium access protocol for dense wireless networks," in *The Third Annual Mediterranean Ad Hoc Networking Workshop - Med Hoc Net 2004*, (Bodrum, Turkey), June 2004.
- [13] A. Miu, G. Tan, H. Balakrishnan, and J. G. Apostolopoulos, "Divert: Fine-grained path selection for wireless LANs," in *Proceedings of ACM MobiSys 2004*, (Boston, USA), June 2004.
- [14] H. Zhu and G. Cao, "On improving the performance of IEEE 802.11 with relay-enabled PCF," *Mobile Networks and Applications*, vol. 9, pp. 423-434, August 2004.
- [15] H. Zhu and G. Cao, "rDCF: A Relay-enabled Medium Access Control Protocol for Wireless Ad Hoc Networks," in *Proc. IEEE Infocom*, March 2005.
- [16] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in *Proc. of ACM MOBICOM*, July 2001.
- [17] C.-H. R. L. J.-S. Liu, Y.-Chang Wong, "A relay-based multi-rate protocol in infrastructure wireless lans," in *Proceedings of 12th IEEE International Conference on Networks (ICON)*, November 2004.
- [18] H. Karl and S. Mengesha, "Analysing capacity improvements in wireless networks by relaying," in *Proc. of IEEE Intl. Conf. On Wireless LANs and Home Networks*, pp. 339-348, December 2001.
- [19] H.-F. Geerdes and H. Karl, "The potential of relaying in cellular networks," in *Proc. of Intl. Network Optimization Conf. (INOC)*, October 2003.
- [20] H. Kwak and S. Yoshida, "Efficient two-hop wireless channel access protocol," in *Proceedings of Vehicular Technology Conference*, pp. 2266-2270, April 2003.
- [21] S. Mengesha, H. Karl, and A. Wolisz, "Capacity increase of multi-hop cellular WLANs exploiting data rate adaptation and frequency recycling," in *Proc. of MedHocNet 2004*, June 2004.
- [22] S. Mengesha, H. Karl, and A. Wolisz, "Improving goodput by relaying in transmission-power-limited wireless systems," in *Proc. Of Informatik*, pp. 537-544, September 2001.
- [23] D. Hollos and H. Karl, "A protocol extension to HIPERLAN/2 to support single-relay networks," in *Proc. of 1st German Workshop on Ad-Hoc Networks*, pp. 91-108, March 2001.
- [24] A. Kamerman and L. Monteban, "WaveLAN-II: A High-performance wireless LAN for the unlicensed band," *Bell Lab Technical Journal*, pp. 118-133, Summer 1997.
- [25] "IEEE P802.11k/D1.0, Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications, Amendment 7: Radio Resource Measurement," July 2004.
- [26] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement," in *Proc. IEEE INFOCOM*, (San Francisco, USA), March-April 1998.
- [27] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select. Areas of Commun.*, vol. 18, pp. 535-547, March 2000.
- [28] D. Qiao, S. Choi, and K. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs," *IEEE Trans. on Mobile Computing*, vol. 1, pp. 278-292, October-December 2002.
- [29] S. Narayanan and S. S. Panwar, "When two-hop meets VoFi," in *Proc. of IEEE CCNC'06*, January 2006.
- [30] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. <http://monarch.cs.rice.edu/papers.html>, 1996.
- [31] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in *Proc. of MobiCom'98*, October 1998.
- [32] M. M. Carvalho, C. B. Margi, K. Obraczka, and J. Garcia-Luna-Aceves, "Modeling energy consumption in single-hop ieee 802.11 ad hoc networks," in *Proc. of ICCCN'04*, October 2004.
- [33] "Host AP driver for Intersil Prism2/2.5/3, <http://hostap.epitest.fi/>"



Pei Liu received his B.S. and M.S. degrees in electrical engineering from Xi'an Jiaotong University, China, in 1997 and 2000, respectively. Currently, he is a Ph.D. candidate at the Department of Electrical and Computer Engineering of Polytechnic University. His research interests are in mobile and wireless systems.



Zhifeng Tao obtained his B.S. degree in Communication and Information Engineering from Xi'an Jiaotong University, Xi'an, P. R. China, in 2000. He received the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Polytechnic University at Brooklyn, NY, in 2002 and 2006, respectively. He joined Mitsubishi Electric Research Laboratories at Cambridge, MA in fall 2006, where he is currently a member of technical staff. His present research interests include wireless networking, medium access control, quality of service, and cooperative communications.



Sathya Narayanan received his M.C.A from College of Engineering, Guindy, India, in 1994, M.S. in Computer Science from Temple University, Philadelphia, in 1998 and the Ph.D. degree in Computer Science from Polytechnic University, Brooklyn, NY, in 2006. He has over ten years of industrial experience in both research and development of networking technologies. He was involved in the development and maintenance of different communication products at Hughes Software Systems, New Delhi, India and Technical Communications

Corporation, Concord, MA, as a senior software engineer during 1994 to 1997.

Since 1999, he has been employed by Panasonic Princeton Laboratory at Princeton, NJ, where he currently holds a senior scientist position. His research interests include 802.11 MAC, cooperative communications, IPv6 mobility, Voice over IP and peer-to-peer networking. He is active in IETF working groups related to IPv6 mobility, SIP (Session Initiation Protocol) and has co-authored multiple internet-drafts. He was a contributing engineer to CableLabs[®] and has written acceptance test plans for DOCSIS1.1 standard.



Thanasis Korakis was born on December 4, 1972 in Ioannina, Greece. He obtained the Bachelor and the M.S. degree in Informatics and Telecommunications from the University of Athens, Greece, in 1994 and 1997 respectively. He obtained the Ph.D. degree in Computer and Communication Engineering from the University of Thessaly, Greece in 2005. In the summer of 2004 he was a visiting researcher at the Computer Science & Engineering Department at the University of California, Riverside.

Dr. Korakis is currently a research scientist associated with the New York State Center for Advanced Technologies in Telecommunications (CATT) at Polytechnic University. His research interests are in the field of wireless networks with an emphasis on access layer protocols, cooperative networks, directional antennas, quality of service provisioning and network management.



Shivendra S. Panwar is a Professor in the Electrical and Computer Engineering Department at Polytechnic University. He received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1981, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, Amherst, in 1983 and 1986, respectively. He joined the Department of Electrical Engineering at the Polytechnic Institute of New York, Brooklyn (now Polytechnic University) in 1985. He is currently the Director of

the New York State Center for Advanced Technology in Telecommunications (CATT). He spent the summer of 1987 as a Visiting Scientist at the IBM T.J. Watson Research Center, Yorktown Heights, NY, and has been a Consultant to AT&T Bell Laboratories, Holmdel, NJ. His research interests include performance analysis and design of networks. Current work includes video systems over peer-to-peer networks, switch performance and wireless networks.

Dr. Panwar has served as the Secretary of the Technical Affairs Council of the IEEE Communications Society ('92-'93) and is a member of the Technical Committee on Computer Communications. He is a co-editor of two books, *Network Management and Control, Vol. II*, and *Multimedia Communications and Video Coding*, 1994 and 1996, respectively, both published by Plenum, and co-authored a textbook, *TCP/IP Essentials: A Lab-Based Approach*, Cambridge University Press, 2004. He is a co-recipient of the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.