

A Cooperative MAC Protocol for Wireless Local Area Networks

Pei Liu, Zhifeng Tao and Shivendra Panwar

The Department of Electrical and Computer Engineering, Polytechnic University
6 MetroTech Center, Brooklyn, NY 11201
Email: pliu@photon.poly.edu, jeffttao@photon.poly.edu, panwar@catt.poly.edu

Abstract—In this paper, a novel idea of user cooperation in wireless networks has been exploited to improve the performance of the IEEE 802.11 medium access control (MAC) protocol. The new MAC protocol leverages the multi-rate capability of IEEE 802.11b and allows the mobile stations (STA) far away from the access point (AP) to transmit at a higher rate by using an intermediate station as a relay. Two specific variations of the new MAC protocol, namely CoopMAC I and CoopMAC II, are introduced in the paper. Both are able to increase the throughput of the whole network and reduce the average packet delay. Moreover, CoopMAC II also maintains backward compatibility with the legacy 802.11 protocol. The performance improvement is further evaluated by analysis and extensive simulations.

I. INTRODUCTION

Signal loss in wireless networks due to distance and fading severely impairs system performance. In most implementations, transmissions received by stations other than the intended receiver will be discarded. To exploit the broadcast nature of wireless channels, recent work on cooperative coding [1]–[4] has shown that additional “cooperative” nodes, which overhear the transmission from the sender and then participate in additional transmission, can provide space diversity for the system. We adopt these ideas and design a new MAC protocol to increase the throughput of a wireless network and designed a new MAC protocol.

IEEE 802.11 [5] was the first international standard for indoor wireless local-area networks (WLAN). It provides physical layer rates of 1 and 2 Mbps. IEEE 802.11b [6] was introduced later in 1999. It uses three different modulation scheme (DBPSK, DQPSK and CCK) which provides a total of four different physical layer rates ranging from 1 to 11 Mbps. The basic Medium Access Control (MAC) of IEEE 802.11b is the distributed coordination function (DCF) which employs carrier sense multiple access with collision avoidance (CSMA/CA). Each station (STA) needs to sense the channel before data transmission. Virtual carrier sensing is also employed to avoid collisions, by the use of the Request To Send (RTS) and Clear To Send (CTS) frames. These two control packets are used to set the Network Allocation Vector (NAV), where the channel reservation information is stored, for all the stations to avoid the hidden terminal problem. After successfully exchanging the control packets, a data packet will be sent and the destination station will send back an acknowledgment (ACK) if the packet has been received without error.

The multi-rate modulation employed by 802.11b leads to fairness problems [7]. If all the stations have uniform traffic to/from the access point (AP), the low data rate stations will use much more channel time than the high data rate stations. This has two negative effects: not only does the low data rate stations get poor service, they also reduce the bandwidth of high data rate stations. This reduces the effective throughput of the network. In our previous work [8], we presented two advantages of a multi-hop extension to the IEEE 802.11 infrastructure mode; one in terms of increased throughput and another in providing a more uniform coverage throughout a 802.11 cell by improvements in the signal-to-interference ratio experienced by nodes at the edges of the cells. Here we introduce a MAC protocol to implement such an extension and present detailed analysis of the protocol.

In this paper, we first present a new MAC scheme (Section III) to allow the low data rate stations to transmit the data packet first to an intermediate station and then to the destination. Then we analyze the performance by deriving expressions for the throughput (Section IV). We also validate the protocol by means of a simulation (Section V). The results show that cooperation not only increases the throughput, but also reduces the service delays for each packet transmission.

II. SYSTEM OVERVIEW

In IEEE 802.11b, the Direct Sequence Spread Spectrum (DSSS) physical layer operates in the 2.4 GHz ISM band, and three different modulation schemes are used. They are: Differential Binary Phase Shift Keying (DBPSK) for the 1 Mbps data rate, Differential Quaternary Phase Shift Keying (DQPSK) for the 2 Mbps data rate, and Complementary Code Keying (CCK) for the 5.5 Mbps and 11 Mbps data rates. The control packets and header part of the data packets are always modulated using DBPSK at 1 Mbps. The modulation scheme of the data frame is indicated in the physical layer header of the data packet.

The bit error probability for these modulation schemes in the AWGN channel can be found in [9]. The performance curves of BER versus SNR for different modulation schemes are shown in Figure 1.

If the quality-of-service parameters (e.g., BER, etc.) are given, we can always find the most suitable modulation scheme based on the received SNR.

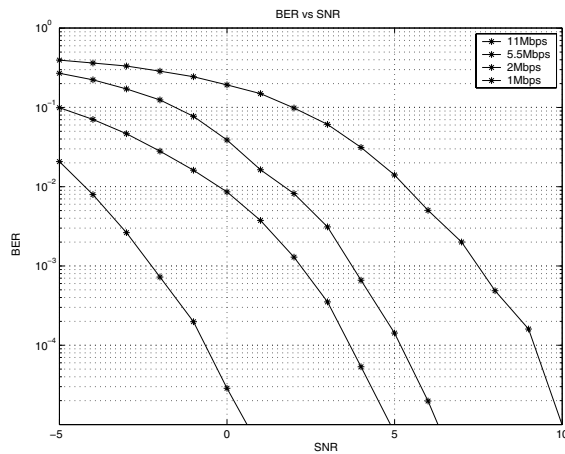


Fig. 1. BER vs SNR for different PHY mode

III. THE PROPOSED COOPERATIVE MAC PROTOCOL

The proposed cooperative MAC protocol is based on the Distributed Coordination Function (DCF) of IEEE 802.11. It is assumed that the transmission power for all the stations are fixed. RTS and CTS packets can be overheard by other stations besides the transmitter and the intended receiver. Transmitting stations choose the best modulation scheme based on the received signal-to-noise ratio (SNR). The threshold SNR for each modulation type is pre-defined and stored in a *physical mode table*. Since uplink (from mobile stations to AP) and downlink (from AP to mobile stations) traffic use the same frequency, the channels are symmetric. We will introduce the first variant of CoopMAC (a.k.a. CoopMAC I) next.

A. The CoopMAC I protocol

- 1) Once a station is associated with the AP in the BSS (Basic Service Set), it listens for RTS, CTS, Data Packets and ACKs sent out by other stations. This is essential in 802.11 networks because all stations need to set the NAV correctly and avoid the hidden node problem. In our new scheme, each station also maintains a table of all the possible *helpers* around itself. Each row of this helper table corresponds to a potential helper and has four fields. The first field is the *ID* (MAC Address) of the potential helper followed by the *time* that the last packet from that station was heard. The third field is the direct transmission data rate from that station to the AP, denoted by R_{hd} . The last field is used to record the data rate that can be used to send data packets to that potential helper from the current station, which is denoted by R_{sh} . This table is updated in the following way:

When any transmission between other stations is heard by the station (S_s) under consideration, it will check if the transmitting station is already in the table. If not, a new row is added for the sender and is identified by the sender's *ID*.

Then S_s estimates the relative channel condition (e.g. path loss) between the sender of that packet, which is denoted by S_h , and itself by measuring the received power level (in *dB*). Path loss can be calculated by subtracting the transmission power (in *dB*), which is typically fixed for all stations, and the received power. The availability of such information is supported by the IEEE 802.11k protocol. Since the channel is symmetric, the reverse channel has the same condition. By checking the physical mode table, we can find the data rate between this pair of nodes and use this value to update the rate R_{sh} for S_s . If any data packet between the AP and another node is overheard, S_s will be able to detect the transmission rate used because this information is included in the physical header, which is always transmitted at the base rate of 1 Mbps. This value is used to update the R_{hd} field. The *time* field will be updated each time a packet from S_s is heard.

- 2) When station S_s has L octets of data to send to a destination station S_d , it first checks the helper table and calculates the time needed to transmit via each potential helper. Since the transmission will be in two steps, first from the source to the potential helper and then from the potential helper to the destination, the transmission time is $8L/R_{sh} + 8L/R_{hd}$, ignoring contention time and overhead. After checking all the potential helpers, we can find the one with the minimum transmission time. If more than one potential helper have the same minimum transmission time, we use the one with the latest *time* field. This chosen helper is denoted by S_h . If the direct transmission rate is R between S_s and S_d , $8L/R$ is the time needed for direct transmission. If $8L/R_{sh} + 8L/R_{hd} < 8L/R$, two hop transmission via the helper will be more efficient. If $8L/R_{sh} + 8L/R_{hd} \geq 8L/R$, the packet will be delivered directly, the same as in the IEEE 802.11b standard.
- 3) S_s senses the channel first. If the channel is idle for a DIFS time and S_s has completed the required backoff procedure, a RTS frame will be sent, reserving the channel for a NAV duration. In the RTS frame, we will reserve the channel for the time needed for direct transmission (see Figure 2). The MAC address of the selected helper will also be included in the RTS frame. Three new fields are appended to the RTS frame, which are the *ID* of S_h , R_{sh} and R_{hd} . If the channel is busy, S_s should wait until the channel is idle plus a DIFS period, and then send the RTS frame. All stations should update their NAV values according to the rule specified in section III-B.
- 4) If the Station which has the same MAC address as indicated by the *ID* field in the RTS frame can successfully decode the RTS frame, it will check if it can sustain both R_{sh} and R_{hd} . If so, a helper ready to send (HTS) packet will be sent out in a SIFS time after the RTS is received. The format of HTS packet is the same as CTS packet. This packet will be heard both by S_s and S_d .

If the helper can't sustain the rate or the RTS packet is corrupted, the helper just stays idle.

S_d will be expecting the HTS packet after receiving the RTS packet. If the HTS packet is received, the CTS packet is sent to reserve the channel for the time needed for two hop transmission via the S_h . If HTS is not heard after two SIFS periods, S_d will still send the CTS packet. But this time, it will reserve the channel for the time needed for direct transmission.

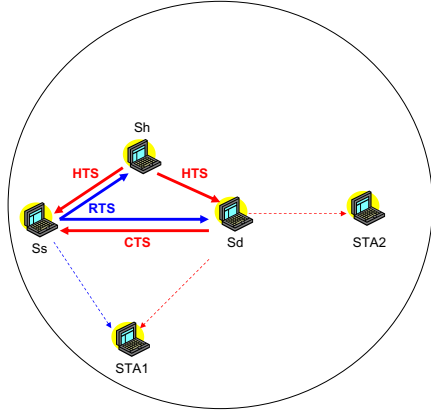


Fig. 2. Exchange of control packets for Cooperative MAC

An illustration of the exchange of control packets for the cooperative MAC protocol is shown in Figure 2. $STA1$ can hear both the RTS and CTS packets; $STA2$ can only hear the CTS packet, thus it is a hidden terminal (i.e. a terminal which can't hear the ongoing transmission, but within the receiving range of the receiver).

- 5) Once S_s receives the CTS packet from S_d , the data packet starts transmission (see Figure 3). If the HTS packet has been received, S_s sends the data packet to S_h using rate R_{sh} . S_h checks the CRC field of the captured data packet and after a SIFS time forwards the packet to S_d , if it is not corrupted, using rate R_{hd} . It is also possible that S_s has not received the HTS packet before the CTS packet is received. This might occur because the channel condition changed due to mobility. In that case, S_s deletes S_h from the potential helper table and transmits the data packet directly to S_d , using rate R .
- 6) After S_d receives the data packet, an ACK packet is sent. Otherwise S_d stays idle. In the latter case, the source station will notice the failure of transmission after a timeout period and starts the exponential backoff, which is the same as in the standard IEEE 802.11b MAC. After a successful transmission is finished via S_h , S_s updates the corresponding *time* field in its potential helper table.

B. NEW NAV MECHANISM

The original RTS and CTS frame formats defined in IEEE 802.11b have a *duration* field. This field in RTS contains the time value, in microseconds, required to transmit the

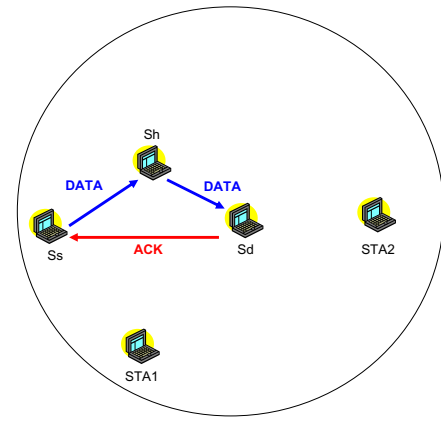


Fig. 3. Exchange of data and ACK packet for Cooperative MAC

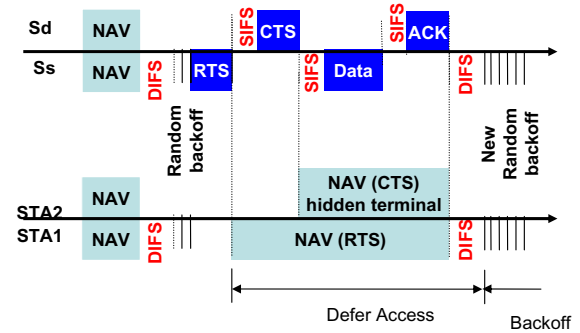


Fig. 4. NAV mechanism defined in IEEE 802.11b

pending frame, plus one CTS frame, one ACK frame, three SIFS intervals, while that in CTS has a time value equal to the *duration* value in the corresponding RTS less one CTS time and one SIFS interval. This is depicted in Figure 4. To convey the additional information of the ID of S_h and the data rates R_{sh} and R_{hd} , we introduced another 3 fields before the *FCS* (frame check sequence) field in the current RTS frame. According to the legacy IEEE 802.11b, the NAV at each station should be set to the value of the duration field in the received frame. Note that this requires a change in the IEEE 802.11 standard. We will present a version of CoopMAC which does not need such changes in section III-C.

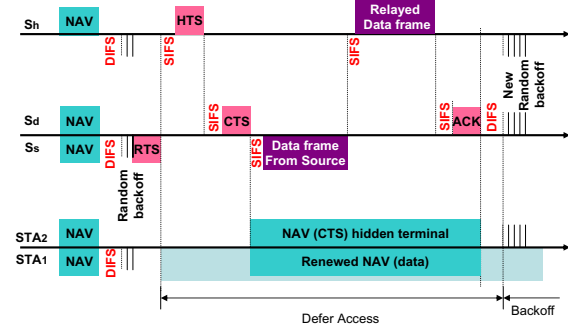


Fig. 5. New NAV mechanism for Cooperative MAC

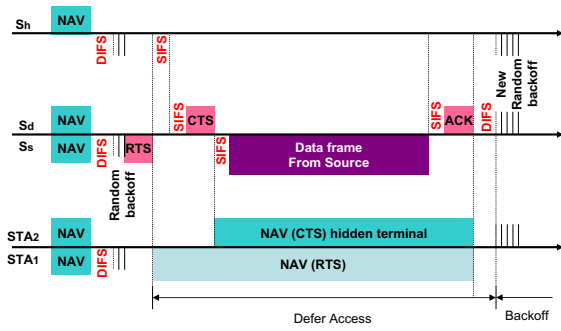


Fig. 6. New NAV mechanism when helper cannot help

In our protocol, the NAV will be calculated differently, as illustrated in Figures 5 and 6. The *duration* carried in RTS should be, the time in microseconds required to transmit the pending data frame through direct transmission from S_s to S_d , plus one CTS frame, one ACK frame and three SIFS intervals.

$$Duration_{RTS} = 3 \times T_{SIFS} + T_{CTS} + 8L/R + T_{ACK} \quad (1)$$

This guarantees that even if the helper is not present, we can send the packet through the network by direct transmission.

The corresponding *duration* field in HTS will be set according to the following equation

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

To set the *duration* in the CTS, we have to check out if the HTS packet have been received. If so, we can do a faster transmission via the S_h and the value of *duration* is

$$Duration_{CTS} = 3 \times T_{SIFS} + 8L/R_{sh} + 8L/R_{hd} + T_{ACK} \quad (3)$$

If HTS is not present, the *duration* in the CTS should equal that of the corresponding RTS frame, less one CTS and one SIFS interval, as defined below:

$$Duration_{CTS} = 2 \times T_{SIFS} + 8L/R + T_{ACK} \quad (4)$$

For those stations that can receive both the RTS and CTS frames (e.g. $STA1$), they need to set their NAV according to the duration in the RTS packet first. Once the HTS or CTS packet is captured, they might need to reset the NAV according to this new message. The *duration* fields in the new packets may reserve less time if the helper can forward packet from the source station. The NAVs at those stations should end right after the ACK packet. The hidden terminals, e.g. $STA2$, can only set their NAV by the *duration* field in the CTS, which is always accurate.

If the data packet is successfully delivered to S_d , every station then proceeds to start a new cycle. On the other hand, if the data packet is lost, S_s will start an exponential backoff since it has not received the ACK packet from the destination station.

C. The CoopMAC II protocol

In the CoopMAC I protocol, we introduced three new fields in the RTS frame. In order to implement this protocol, hardware modification is required. We therefore introduce CoopMAC II as an alternative to CoopMAC I. For CoopMAC II, we will use the fact that the *Address 4* field in the IEEE 802.11 frame format (Figure 7) is never used for data frame, except when the data frame is exchanged between APs, in which case the *toDS* and *fromDS* subfields within the *frame control* field are both set to 1.

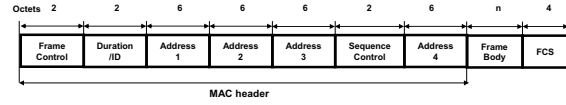


Fig. 7. MAC Header format for 802.11

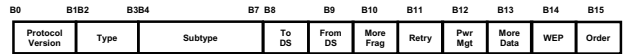


Fig. 8. Frame Control format for 802.11

Therefore, it is feasible for cooperative MAC to use address 4 field to indicate the MAC address of the selected helper. Since the subtype values from 1000-1111 for *Subtype* field in *frame control* are reserved, we can use one of these values to distinguish our packets from the regular data packets.

This allows us to design another MAC protocol, using the *Subtype* to denote that the data packet needs a relay, and the *Address 4* field to address the helper. Since there is no need to add the MAC address of the helper and rate in the RTS frame, we will use the same RTS frame as in the legacy system.

In this protocol, when a station has a packet to send, it will pick a helper in the same manner as in CoopMAC I. Then S_s senses the channel and a RTS frame will be sent, reserving the channel for a NAV duration corresponding to the two hop data transmission. S_d will also send the CTS packet after the RTS is received. Once S_s receives the CTS packet from S_d , the data packet starts transmission (see Figure 3). The first packet is still sent from S_s to S_h at rate R_{sh} , but with the modification to the MAC header of the data frame mentioned above. Any node will be able to understand that this is a data packet which should be forwarded to the destination. Only the one which has the same MAC address as in *Address 4*, i.e., S_h , relays the packet to the S_d . An ACK will be sent from the S_d if there is no error.

The corresponding *duration* carried in the RTS and CTS frames for CoopMAC II should be $Duration_{RTS} = 4 \times T_{SIFS} + T_{CTS} + 8L/R_{sh} + 8L/R_{hd} + T_{ACK}$ and $Duration_{CTS} = 3 \times T_{SIFS} + 8L/R_{sh} + 8L/R_{hd} + T_{ACK}$.

D. Comparison

Since CoopMAC II does not have an HTS packet, it incurs less overhead than CoopMAC I and results in a simpler implementation. Moreover, this also guarantees the backward compatibility with the legacy 802.11 system. On the other

hand, CoopMAC II is more vulnerable to a change in the availability of a helping station caused by mobility. More specifically, since a two-way handshake is employed in CoopMAC II, the original source S_s is unable to learn whether the helper it selects is available or not for this particular transmission before S_s actually sends the packet.

CoopMAC I does not entail the issues discussed above, but does need to define a new HTS message and introduces three new fields in the RTS, which may affect the normal operation of coexisting legacy 802.11 stations. Since this impact depends on the detailed implementation of the legacy node state machine, we do not claim that CoopMAC I is backward compatible with the legacy system.

IV. THROUGHPUT ANALYSIS

In this section, we will derive an expression for the saturated throughput of our proposed MAC protocol. All stations are uniformly distributed in the coverage area and are assumed to be stationary.

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. Since only mobile stations which are more than $r_{5.5}$ meters away from the AP can reduce their transmission time using two hop transmission, we will focus on these stations. If a third node is within r_x meters to the source node and r_y meters to the destination, it can help the transmission in a two hop manner using rate x Mbps and y Mbps. The probability that a node exists in such region will be denoted by $P_{x,y}$. The probability that no helper is present is $1 - P_{11,11} - P_{5.5,11} - P_{5.5,5.5} - P_{5.5,2}$.

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 [10], we have:

$$T_{11} = T_{cont}(n) + T_{overhead} + \frac{8L}{R_{11}}$$

$$T_{5.5} = T_{cont}(n) + T_{overhead} + \frac{8L}{R_{5.5}}$$

where $T_{overhead} = T_{PLCP} + T_{DIFS} + T_{RTS} + T_{CTS} + 3T_{SIFS} + T_{ACK}$ and $T_{cont}(n)$ accounts for the time spent in the contention period.

For stations which have a direct transmission rate of 2 Mbps, we have to average over the transmission time where a helper is present and the time needed if the helper is not available.

$$T_2 = T_{cont}(n) + (P_{11,11} + P_{5.5,11} + P_{5.5,5.5})T_{CoopOH}$$

$$+ \frac{16P_{11,11}L}{R_{11}} + \frac{8P_{5.5,11}L}{R_{11}} + \frac{8P_{5.5,11}L}{R_{5.5}} + \frac{16P_{5.5,5.5}L}{R_{5.5}}$$

$$+ (1 - P_{11,11} - P_{5.5,11} - P_{5.5,5.5})(T_{overhead} + \frac{8L}{R_2})$$

where $T_{CoopOH} = 2T_{PLCP} + T_{DIFS} + 5T_{SIFS} + T_{RTS} + 2T_{CTS} + T_{ACK}$ for CoopMAC I and R_x is x Mbps. Similarly we can derive the average transmission time T_1 for stations with a direct rate of 1 Mbps.

Because of the long term channel access fairness guaranteed by the CSMA/CA protocol, each station in the network 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. 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 as follows:

$$T = f_{11}T_{11} + f_{5.5}T_{5.5} + f_2T_2 + f_1T_1 \quad (5)$$

and

$$f_{11} = r_{11}^2/r_1^2$$

$$f_{5.5} = (r_{5.5}^2 - r_{11}^2)/r_1^2$$

$$f_2 = (r_2^2 - r_{5.5}^2)/r_1^2$$

$$f_1 = (r_1^2 - r_2^2)/r_1^2$$

A. Probability of Collision

If multiple hosts attempt to transmit at the same time, the transmissions will collide and the sending stations will start exponential backoff. Each station picks a random number X from $[0, CW]$, where CW is the size of congestion window, and retransmit after X time slots. If the retransmission is successful, CW is set to CW_{min} , otherwise $CW = \min(2 \times CW, CW_{max})$.

It is obvious that the time spent in the contention procedure for each station increases with the number of stations in the network. An analytical approach to compute $T_{cont}(n)$ is given in [11], [12]. Here we use a simple approximation proposed in [7]:

$$T_{cont}(n) \approx SLOT \times \frac{1 + P_c(n)}{2n} \times \frac{CW_{min}}{2} \quad (6)$$

where $P_c(n) = 1 - (1 - 1/CW_{min})^{n-1}$ and $SLOT$ is the time period corresponding to one time slot.

The throughput of the network, in bps, is given by the following equation:

$$S = 8L/T \quad (7)$$

V. SIMULATION

To validate the proposed MAC scheme, we developed an event-driven custom simulator using the C programming language. The transmission ranges are shown in Table I.

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

TABLE I
PHYSICAL MODE TABLE

Other basic parameters used in the simulation are shown in Table II. The mobile stations are randomly distributed in a circle with a radius of 100 meters and the access point is located in the center of this circle. The network is under a heavy load condition and traffic is evenly distributed across all the nodes in the network. Packets arrive the network

MAC header	272 bits
PHY header	192 bits
RTS	352 bits
CTS, HTS	304 bits
ACK	304 bits
Data rate for header	1 Mbps
Slot time	20 μ S
SIFS	10 μ S

TABLE II
PARAMETERS USED IN SIMULATION

according to the Poisson distribution and the simulation stops after 1 million packets. The minimum congestion window size (CWMin) is 16 slots, maximum congestion window size (CWMax) equals 1024 slots and the the maximum number of backoff stages is 6, after which the packet is dropped.

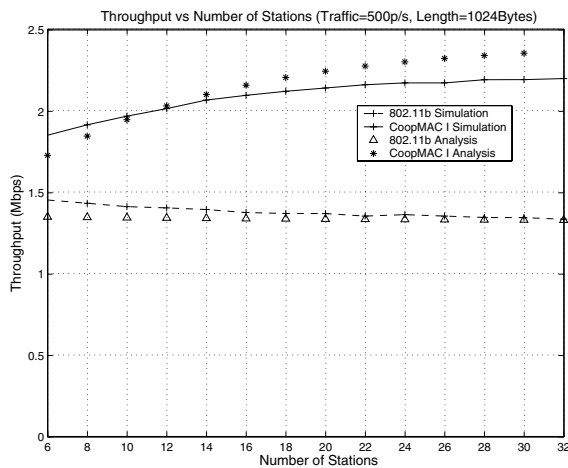


Fig. 9. Throughput for 802.11b and CoopMAC I vs number of stations

Figure 9 shows the saturated throughput for 802.11b MAC and our proposed CoopMAC I scheme. In this simulation, packets arrive to the system at a rate of 500 packets/sec and each MSDU packet is 1024 bytes in length. From the figure, we can see that the throughput for both MAC schemes are much less than 11 Mb/s. This is because not all the mobile stations can transmit at the highest data rate. MAC overhead and collisions reduce the throughput further. As we can see, both in simulation and analysis results, the CoopMAC I scheme has much higher throughput than the legacy system. When the number of stations increases, throughput for the standard 802.11b network decreases since more collisions will occur. In our MAC scheme, the throughput starts from 1.8 Mb/s and increases to 2.2 Mb/s when we have 20 nodes. After that, the throughput is almost flat. This is because, initially, the more stations in the network, the higher possibility that a station can find a helper and transmit at a higher rate. This effect not only offsets the throughput decrease caused by collisions, but leads to a net increase in the total throughput. The percentage throughput increase is shown in Figure 10, using the simulation results.

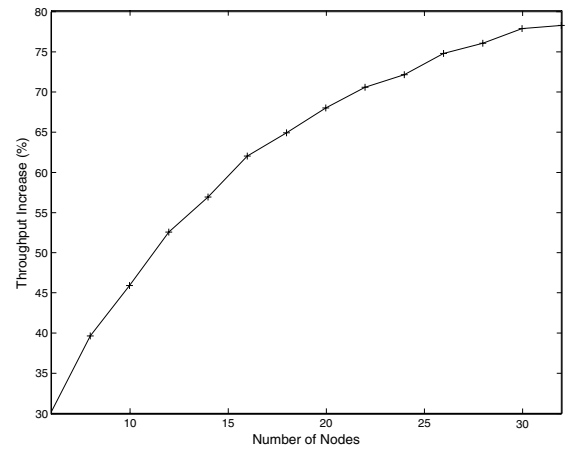


Fig. 10. Throughput increase vs Number of Stations for CoopMAC I

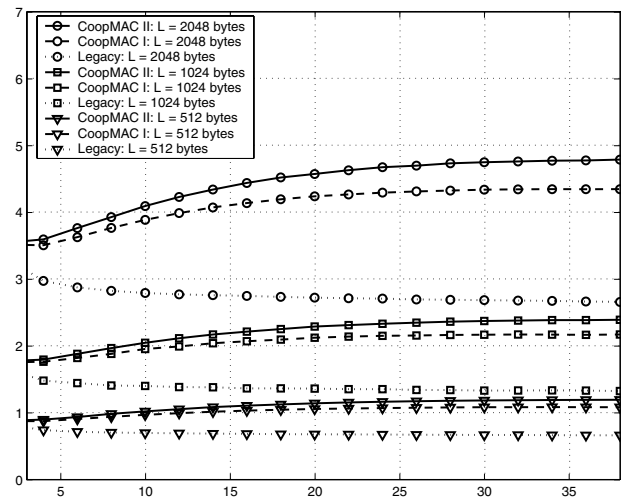


Fig. 11. Throughput vs number of Stations for different packet lengths

Figure 11 compares the throughput improvement under three different MSDU packet lengths. The saturated throughput for 802.11b system increases with packet size. This is due to a lower percentage of the overhead time devoted to MAC and PHY layer header. Both of our proposed MAC schemes achieves higher throughput than the 802.11b systems. CoopMAC II achieves higher throughputs than CoopMAC I because of less control packet overhead.

Figure 12 shows the cumulative service delay distribution for data packets when there are 8 nodes in the system and the system is heavily loaded. Service delay is the time duration from when the packet becomes the Head-of-Line (HOL) packet in the buffer till it leaves the system. We can see the delay in our scheme is substantially lower than that of the traditional 802.11b.

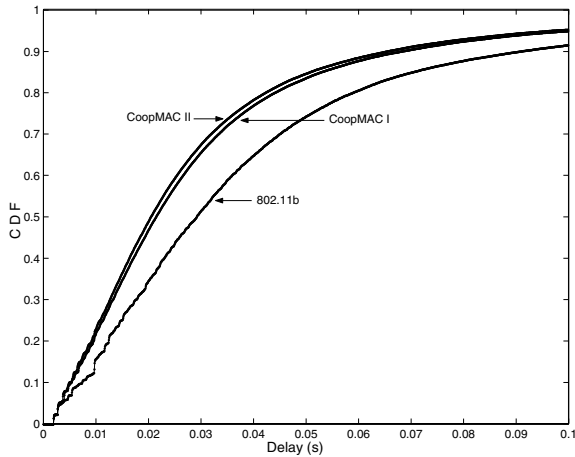


Fig. 12. Cumulative distribution for service delays

VI. CONCLUSIONS

In this paper, we have proposed a new MAC scheme for the 802.11 wireless LAN. This scheme is fully compatible with the legacy systems and can be extended to higher PHY rate systems, e.g. 802.11a/g. Cooperation between mobile stations utilizes space diversity and makes better use of the multiple transmission rates offered by the PHY layer. The MAC algorithm is then evaluated via simulation and the result shows a consistent throughput increase, using the same physical layer as in 802.11b. It also helps to reduce the delay experienced by each packet. In our scheme the low data rate stations, which are usually at the edge of the coverage area, use less transmission time than their 802.11 counterparts. This also helps to relieve the fairness problem in multiple rate 802.11

system. For the future work, we plan to study the interference reduction effect of cooperative MAC, as initially suggested in [8]. We also will extend both versions of CoopMAC to an ad hoc environment.

REFERENCES

- [1] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity-Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, November 2003.
- [2] J. N. Laneman, D. Tse, and G. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 3062–3080, December 2004.
- [3] M. C. Valenti and N. Correal, "Exploiting Macrodiversity in Dense Multihop Networks and Relay channels," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, vol. 3, March 2003, pp. 1877–1882.
- [4] 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.
- [5] "IEEE Std. 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.
- [6] "IEEE Std. 802.11b-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer Extension in the 2.4GHz Band," 1999.
- [7] 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.
- [8] 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.
- [9] J. G. Proakis, *Digital Communications (Fourth edition)*. McGraw Hill, 2001.
- [10] 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, no. 4, pp. 278–292, October-December 2002.
- [11] 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.
- [12] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select. Areas of Commun.*, vol. 18, no. 3, pp. 535–547, March 2000.