MAC-SCC: A Medium Access Control Protocol with Separate Control Channel for Reconfigurable Multi-hop Wireless Networks

Yijun Li, Hongyi Wu, Nian-Feng Tzeng, Dmitri Perkins, and Magdy Bayoumi

***Abstract*— In this paper, we propose a novel Medium Access Control protocol with a Separate Control Channel (*MAC-SCC*) to increase the channel efficiency and address the unfairness and instability problems of IEEE 802.11 MAC protocol. In MAC-SCC, the available bandwidth is partitioned into two channels: a data channel and a control channel, each associated with a network allocation vector (NAV). To reduce hardware complexity, the station transmits or receives on one channel only at any given time. In the network employing MAC-SCC, the next data frame can be pre-scheduled during the current data transmission via the separate control channel, and thus reducing the frame collision probability and the bandwidth wasted during backoff. Moreover the use of the separate control channel helps to achieve fair medium access and solve the instability problem resulted from frequent link failures. The optimal bandwidth partitioning between the two channels is analyzed via a statistical model, which shows 10% bandwidth for the control channel and 90% bandwidth for the data channel. The performance of MAC-SCC is quantified via extensive simulations in both a stand-alone simulator developed by using PARSEC and a comprehensive network simulator called QualNet with whole protocol stack. Our results show that MAC-SCC can effectively reduce the link failure probability, achieve fair medium access when running multiple TCP sessions, and yield a throughput gain up to 60% under high traffic load.**

***Index Terms*— Bandwidth partitioning, channel efficiency, MAC-SCC, medium access control, mobile ad hoc network, network allocation vector, performance evaluation.**

I. INTRODUCTION

A RECONFIGURABLE multi-hop wireless network consists of a collection of self-configured nodes that act both as hosts and as routers, and communicate with each other via wireless links without the intervention of a centralized controller [1]. With the unmatched flexibility to support the communication of mobile users, the reconfigurable multi-hop wireless network has become increasingly popular in the past few years. At the same time, however, several challenges, such as Quality of Service (QoS), efficient routing and medium access control (MAC), need to be addressed in order to enable efficient communication in such reconfigurable multi-hop wireless networks. In this paper, we focus on the MAC protocol that are employed to resolve contentions for accessing the shared medium. It has been shown in [2] that MAC protocols can significantly affect the overall performance of the multi-hop wireless networks. The hidden station [3] is a main problem of the medium access control in multi-hop wireless networks. A hidden station is a node that is close to the receiver but outside the transmission range of the sender. Such node can not detect the ongoing transmission, and thus it may send out control or data frames and interfere the reception at the receiver. To alleviate the hidden station problem, the *Request To Send (RTS)/Clear To Send (CTS)* mechanism has been developed and standardized in IEEE 802.11 Distributed Coordination Function (DCF) [4]. Fig. 1 (a) illustrates a simple network topology composed of 4 nodes. As shown in Fig. 1 (b), when source $S1$ has data to send to node $D1$, it first senses the medium. If the medium is idle during an interval of DIFS (DCF InterFrame Space), $S1$ sends an RTS frame. Upon receiving RTS successfully, $D1$ responds with a CTS after an interval of SIFS (Short InterFrame Space). On receiving the CTS, $S1$ sends the data frame, and $D1$ will accordingly respond with an ACK. There is a *duration* field in any valid frames to indicate the period reserved for data transmission. On receiving a valid frame, a node updates its Network Allocation Vector (NAV), which is a timer used to indicate the duration of channel occupancy. If the source (e.g., $S2$ in Fig. 1 (b)) detects the channel busy based on either physical medium sensing or NAV, it has to defer its RTS transmission. More specifically, $S2$ starts its *backoff timer*, which decreases only after the medium becomes idle for an interval of DIFS, and sends RTS when the timer expires. The initial value of the *backoff timer* is uniformly distributed in the contention window whose size is exponentially increased with the number of retransmissions.

While IEEE 802.11 MAC protocol is employed widely in wireless local area network and ad hoc networks, it has been shown that the basic RTS/CTS approach may result in several problems as listed below.

1) Channel efficiency: The channel efficiency of the IEEE 802.11 MAC protocol may be reduced dramatically when the traffic load is high. As can be seen from Fig. 1 (b), a certain amount of time is used for signaling only, and in particular, the

Manuscript received August 5, 2004; revised April 16, 2005; accepted June 26, 2005. The associate editor coordinating the review of this paper and approving it for publication was Z. Zhang. The authors acknowledge the support of National Science Foundation CAREER Award under Award Number CNS-0347686, National Science Foundation under Award Number INF 6-001-006, U.S. Department of Energy under Award Number DE-FG02-04ER46136, Louisiana Board of Regents under Contract Numbers LEQSF(2003-06)-RD-A-37 and DOE/LEQSF(2004-07)-ULL, EETAPP program DE97ER12220, and the Governor's Information Technology Initiative.

The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, P.O. Box 44330, Lafayette, LA 70504. (e-mail: wu@cacs.louisiana.edu).

Digital Object Identifier 10.1109/TWC.2006.04523.

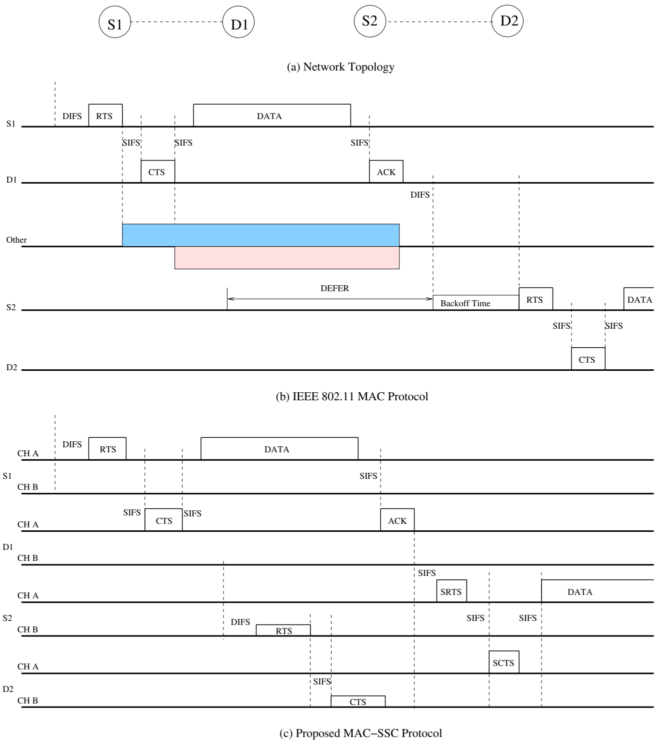


Fig. 1. Comparison of MAC-SCC and IEEE 802.11 MAC protocols.

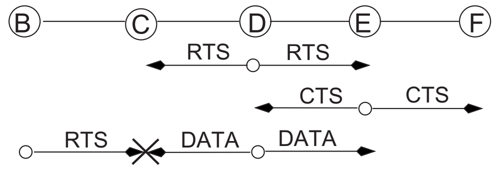


Fig. 2. Instability and unfairness problems in IEEE 802.11.

entire bandwidth is wasted during backoff. In wireless local area networks, where all nodes are within the transmission range of each other, the RTS/CTS frames are assumed not to collide with data frames, because every node maintains a NAV such that one node does not send RTS/CTS during the data transmission of another node. But this assumption is not valid in reconfigurable multi-hop environments due to the hidden station problem, and accordingly a large number of signaling frames may experience multiple collisions. Under high traffic load, multiple retransmissions may fail, leading to exponentially increased backoff time and further decreased channel efficiency.

2) Link failure and instability problem: Worse yet, the node may even falsely report a link failure. For example, in a simple network with the string topology shown in Fig. 2, when node D has data to send to node E, it sends an RTS. Assuming an omnidirectional antenna for transmission, then both nodes C and E will receive the RTS frame. As a result, node C will defer its transmission (if any) and node E replies with a CTS. After receiving CTS, node D can send the data frame. However, since node B cannot receive the RTS or CTS frames, it assumes that there is no on-going transmission. If node B has data to send to node C, it will send out an RTS. But node C cannot receive this RTS correctly because of the interference from node D, and thus will not reply to node B. After timeout, node B will backoff for a period of time before retransmitting the RTS frame. If node D is sending several long back-to-back TCP data packets, the channel will be kept busy, and thus multiple consecutive attempts to transmit an RTS frame from node B may fail. When failing seven times (a default value defined in IEEE 802.11 standard) to receive the CTS from node C, node B will quit and falsely report a link failure. In such a case, upper layers will initiate certain recovery schemes (e.g., re-routing), likely resulting in a long end-to-end delay. During route discovery, TCP throughput is about zero, leading to the instability problem as reported in [5].

3) Unfairness problem: Due to the binary backoff algorithm, IEEE 802.11 MAC protocol also results in unfairness problem among multiple TCP sessions [5], [6]. Assuming that the transmission range of a node only covers the immediate neighbor nodes and there are two TCP sessions in the network shown in Fig. 2: Session 1 with one hop from node C to node B and Session 2 with two hops from node D to node F. Clearly, node C and node D will contend for the channel. In IEEE 802.11, however, node C has a much higher probability to win contention for channel access, because of the following two reasons. First, since session 2 involves two hops, node D can transmit for no more than 50% of the session period (due to the contention with node E). Thus, only about 50% data frames of Session 1 experience channel contention. On the other hand, every data frame of Session 2 has to contend for the channel. Second, the IEEE 802.11 backoff algorithm favors the last successful transmission. More specifically, the station (e.g. node C) that just completes a successful transmission always uses the minimum contention window, leading to a short backoff time and high probability of winning channel contention. As a result, Session 2 is shut down with a throughput about zero.

In this paper, we propose a novel MAC protocol using a separate control channel (*MAC-SCC*) to address the above problems in IEEE 802.11. In MAC-SCC, the available bandwidth is partitioned into two channels: a data channel and a control channel, each associated with a network allocation vector (NAV). To reduce hardware complexity, the station transmits or receives on one channel only at any given time. In the network employing MAC-SCC, the next data frame can be pre-scheduled during the current data transmission via the separate control channel, and thus reducing the frame collision probability and the bandwidth wasted during backoff. Moreover the use of the separate control channel helps to achieve fair medium access and solve the instability problem resulted from frequent link failures. The optimal bandwidth partitioning between the two channels is analyzed via a statistical model, which shows 10% bandwidth for the control channel and 90% bandwidth for the data channel. The performance of MAC-SCC is quantified via extensive simulations in both a stand-alone simulator developed by using PARSEC and a comprehensive network simulator called QualNet with entire protocol stack. Our results show that MAC-SCC can effectively reduce the link failure probability, achieve fair medium access when running multiple TCP sessions, and yield a throughput gain up to 60% under high traffic load.

The rest of this paper is organized as follows. The related work is discussed in Section II. The proposed MAC-SCC protocol is described in Section III. The simulation results are presented in Section IV. Finally, Section V concludes the paper.

II. Related Work

Various MAC protocols have been proposed to improve the performance of the basic RTS/CTS scheme. MACAW [6], [7] uses an RTS-CTS-DS-DATA-ACK handshake sequence and a different backoff algorithm to achieve high channel utilization and fair channel access. As discussed in [6], however, MACAW cannot solve the unfairness problem completely, especially in the situation as shown in Fig. 2. Efficient MAC protocol with power control are proposed in [8], [9] in order to reduce the interference and improve system throughput. In [10]–[12], the authors propose MAC protocols for the wireless network deploying directional antennas to improve channel efficiency via spacial reuse. [13] proposes a dynamic channel assignment scheme, where the channels are assigned to the nodes dynamically so that the same channels are not used by the nodes which are within two hops. [14] proposes to combine different MAC protocols in order to adapt to dynamic network conditions.

In this paper, we focus on the use of multiple channels, which is an effective way to enhance MAC protocol efficiency and may be applied together with other techniques (such as, dynamic channel assignment scheme, the use of directional antennas and/or power control) as well. Several existing multichannel MAC protocols for multihop wireless networks are discussed as follows.

The dual busy tone multiple access (DBTMA) is proposed in [15]. To further alleviate the hidden station problem and reduce the collision probability, extra hardware is deployed to support two outband busy tones: one indicating data transmission and the other indicating data reception. The carrier sense is done by detecting these two busy tones. DBTMA without MAC layer acknowledgement uses RTS-CTS-DATA handshake sequence. The error recovery relies on the high layer protocol, such as TCP. [16] enhances DBTMA by combining power control with RTS/CTS-based and busy tone-based approach to increase channel efficiency.

[17] proposes a multi-channel carrier sense multiple access (CSMA) protocol, where the total available bandwidth is divided into N narrow-band sub-channels. The RTS/CTS mechanism is not used. Instead, an idle sub-channel is selected randomly for data transmission. [18] further improves the multi-channel protocol by maintaining a list of free sub-channels at each node. The available sub-channel with the highest channel quality is chosen for data transmission. A receiver-based multi-channel MAC protocol is proposed in [19], where each mobile node is assigned with a channel to receive data packets. The maximal matching algorithm is employed to improve the channel efficiency and network throughput. [20] presents a receiver-initiated channel-hopping with dual polling (RICH-DP) protocol, where the total bandwidth is divided into a number of subchannels and the nodes hop among the subchannels by following the same sequence. If a node is involved in data transmission or reception, it stays in the currently used subchannel until the data transmission is finished. Otherwise it hops to the next channel after a short dwell time.

Similar to [17]–[20], the total available bandwidth is divided into a number of narrow-band sub-channels in [21]. But one sub-channel is reserved for control (i.e., transmitting RTS/CTS messages). Reserving control channel can significantly reduce the collision probability of data channel, because control packets are usually very short. Each node senses the medium and builds a free-channel list. This list is embedded in the RTS frame of the transmitting node. The receiving node can accordingly choose the channel which is free for both of them. Clearly, the RTS/CTS frames need the extra field/bytes to accommodate the free-channel list. In addition, it is assumed in [21] that multiple packets on different channels can be received simultaneously, which may increase hardware cost. A similar idea is used in [22]–[24] and further enhanced in [25] with power control. [26] proposes to employ two control channels, one for RTS/CTS and the other for ACK, respectively, in order to improve data channel reuse ratio.

Several approaches have been proposed recently to apply the multiple channel MAC schemes on existing IEEE 802.11 channels. In [27], the authors propose to reserve one IEEE 802.11 channel for signaling and use the other channels for data transmission. Five control frames (two RTS frames, two CTS frames, and a Probe frame) are involved in channel reservation for each data frame. [28] proposes to apply multi-channel MAC protocol on three non-overlapped IEEE 802.11b channels and focuses on solving the multi-channel hidden station problem. In order to further improve channel efficiency, [29] proposes to employ three half-duplex transceivers that can reduce the channel switching delay.

Note that, all of the above multi-channel MAC protocols [17]- [26] share several problems compared with our proposed MAC-SCC protocol. First, to implement multi-channel protocols, the wireless network interface card (NIC) needs to scan all channels. The more channels are employed, the more complex and expensive is the protocol implementation. If multiple NICs are used, the cost is even higher. In contrast, MAC-SCC only maintains two channels (no matter how much bandwidth is used by the network), thus significantly reducing the hardware complexity and cost. Second, the multi-channel protocols have a low peak transmission rate and may potentially waste the channel bandwidth. More specifically, each pair of wireless terminals can be assigned with only one channel for communication, even when there are many other idle channels. As a result, the communication delay is long, compared with MAC-SCC where each data frame is transmitted over the entire data channel, whose bandwidth may be (theoretically) equivalent to the sum of all data channels in the multi-channel protocols. Third, in most multi-channel protocols, all channels have the same bandwidth. One channel is used for control while the other channels are for data transmission. It is not clear whether the control channel will be overloaded or its bandwidth may be wasted. In this paper, we present a solid traffic analytic model. The bandwidth partitioning in MAC-SCC is based on our analytic results, which allocates appropriate amount of bandwidth to the control channel and the data channel, respectively.

III. Proposed MAC-SCC Protocol

In this section, we first introduce the proposed MAC-SCC protocol. Then, we discuss the optimal bandwidth partitioning between the control channel and the data channel. Finally, we discuss and compare MAC-SCC with several related protocols.

A. Protocol Description

1) Protocol Overview: The basic idea of our proposed MAC-SCC protocol is described as follows while the details will be discussed later in this section. In MAC-SCC, the entire bandwidth into two subchannels (e.g., by using two frequence bands or two time slots or two orthogonal codes): *CH a* that is primarily used for data transmission and *CH b* that is used for signaling, with bandwidth W_d and W_s ($W_d \gg W_s$), respectively. When a source node initializes data transmission, it senses *CH a* first. If *CH a* is idle, similar to that in IEEE 802.11, a RTS-CTS-DATA-ACK handshake sequence is applied on *CH a* for channel reservation. If *CH a* is busy, *CH b* is used to schedule the next data transmission. As shown in Fig. 1 (c), nodes $S2$ and $D2$ can exchange the RTS and CTS frames on *CH b*, even when node $S1$ is sending a data frame to node $D1$ on *CH a*. In other words, the contention of next channel access is resolved during the transmission of the current data frame. As soon as the current data transmission finishes, the next data frame (from $S2$ to $D2$) can be transmitted by following an SRTS-SCTS-DATA-ACK handshake sequence, where SRTS and SCTS are simplified RTS and simplified CTS, respectively. As we can see, the channel reservation in MAC-SCC is in sharp contrast to IEEE 802.11 MAC protocol, where $S2$ has to defer sending RTS until the medium becomes idle for interval equal to the sum of DIFS and a random backoff time, during which the entire bandwidth is wasted.

2) Frames and Channels: In addition to RTS, CTS, DATA and ACK, two extra frames, SRTS and SCTS, are employed in MAC-SCC. The SRTS and SCTS frames include only three fields: $frametype$, $duration$ and $checksum$. They are employed to address the potential hidden station problem due to the use of two channels. More specifically, the RTS and CTS transmitted on *CH b* (e.g. by S2 and D2 in Fig. 1) may not be received by the nodes (e.g. S1 and D1 in Fig. 1) that are currently sending or receiving on *CH a* because of the hardware limitation. As a result, S1 and D1 don't update their NAV^a and may become hidden stations when S2 sends data frame to D2. In order to avoid such hidden station problem (i.e. to block any possible transmission from S1 to D1), SRTS and SCTS are transmitted on *CH a* before the transmission of data frame.

Since *CH a* and *CH b* have different bandwidth, the same frame needs different time to be transmitted on these two channels. To facilitate our discussion, we denote T^a_{data}, T^a_{ack}, T^a_{srts}, T^a_{scts}, T^a_{rts}, and T^a_{cts} to be the time to transmit a data frame, an ACK frame, an SRTS frame, an SCTS frame, an RTS frame, and a CTS frame on *CH a*, respectively, while denote T^b_{rts} and T^b_{cts} to be the time to transmit an RTS and a CTS frame on *CH b*, respectively. Each node maintains two NAVs, NAV^a and NAV^b for *CH a* and *CH b*, respectively. The values of NAV^a and NAV^b are always decreasing if they are larger than zero. Updating the NAVs is similar to that in IEEE 802.11. More specifically, the frame header includes a *duration* field with a value $t_{duration}$ to indicate the time interval that *CH a* is to be reserved for data transmission. When sending an RTS on *CH a*, $t_{duration} = T^a_{data} + T^a_{ack} + T^a_{cts} + 3SIFS$; When sending an SRTS on *a*, $t_{duration} = T^a_{data} + T^a_{ack} + T^a_{scts} + 3SIFS$; When sending a CTS or an SCTS on *CH a*, $t_{duration} = T^a_{data} + T^a_{ack} + 2SIFS$; When sending an RTS or a CTS on *CH b*, $t_{duration} = T^a_{srts} + T^b_{scts} + T^a_{data} + T^a_{ack} + 4SIFS$[1]. Upon receiving RTS or CTS on *CH a* or *CH b*, a node updates its NAV^a or NAV^b as to be discussed next in Section III-A.3. The RTS and CTS frames also include a *defer time* field to indicate the amount of time (t_{defer}) taken before the node can send or receive data on *CH a*. In addition, similar to that in the IEEE 802.11 standard, when the medium is busy or collision occurs, a node starts a *backoff timer*, which has an initial random value and decreases when the channel becomes idle for an interval of DIFS.

TABLE I
Listening of Two Channels

NAV values	Channels
$NAV^a = 0$ $NAV^b = 0$	*CH a* and *CH b*
$NAV^a > 0$ $NAV^b = 0$	*CH b*
$NAV^a > 0$ $NAV^b > 0$	*CH a* and *CH b*

TABLE II
Usage of Two Channels

Frame Types	Channel
RTS-CTS	*CH a* or *CH b*
SRTS-SCTS	*CH a*
DATA-ACK	*CH a*

Note that, although two channels are used, a node is not required to transmit and/or receive on *CH a* and *CH b* simultaneously. More specifically, when a node is transmitting on one channel, it doesn't receive any frame at the same time. When a node is not transmitting, it listens to different channels depending on its NAVs (as summarized in Table I). When both NAV^a and NAV^b are greater than zero or $NAV^a = NAV^b = 0$, the node monitors both channels. However, it only receives on one channel upon a frame arrival. When $NAV^a > 0$ and $NAV^b = 0$, the node listens to (and receives frames on) *CH b* only. Table II summarizes the frames transmitted on the two channels. The RTS and CTS frames can be transmitted on either *CH a* or *CH b*, while SRTS, SCTS, DATA, and ACK frames can be transmitted on *CH a* only.

3) Detailed Description: The MAC-SCC protocol is discussed in details as follows, with the major steps summarized in Fig. 3. In Step 1, the source S first senses the medium. If both *CH a* and *CH b* are idle, it means there is no on-going data transmission. In order to reduce the delay for signaling, the RTS frame is sent on *CH a*, which has higher bandwidth. If *CH a* is busy while *CH b* is idle, S constructs an RTS frame with $t_{defer} = NAV^a$ to indicate that S cannot send

[1]To simplify the protocol description, we assume *CH a* and *CH b* use the same interfame spaces (DIFS and SIFS). But different values can be used for *CH a* and *CH b* with minor modifications.

1. *Source S sends RTS*:
When a node S has data to send to a node D, it first senses *CH a* and *CH b*.
If both *CH a* and *CH b* are idle for a period of DIFS
  S sets $t_{defer} = 0$;
  $t_{duration} = T^a_{cts} + T^a_{data} + T^a_{ack} + 3SIFS$
  S sends *RTS* on *CH a*;
Else if only *CH b* is idle for a period of DIFS
  S sets $t_{defer} = NAV^a$;
  $t_{duration} = T^a_{srts} + T^a_{scts} + T^a_{data} + T^a_{ack} + 4SIFS$
  S sends *RTS* on *CH b*;
Else
  S starts a *backoff timer* and sends *RTS* on *CH b* when it times out.
Endif

2. *Destination D receives RTS and sends CTS*:
If D receives RTS successfully
  If D receives RTS on *CH a*
    D sets $t_{defer} = 0$;
    $t_{duration} = t^{rts}_{duration} - T^a_{cts} - SIFS$
    D replies with CTS on *CH a* after SIFS;
  Else
    If $NAV^b == 0$ (let t' = t_{defer} obtained from RTS)
      D sets $t_{defer} = max\{t' - SIFS - T^b_{rts}, NAV^a\}$;
      $t_{duration} = t^{rts}_{duration}$
      D replies with CTS on *CH b* after SIFS;
    Endif
  Endif
Endif

3. *Source S receives CTS and sends data*:
If S receives CTS (let t' = t_{defer} obtained from CTS)
  If S receives CTS on *CH a*
    $t_{duration} = t^{cts}_{duration} - T^a_{data} - SIFS$
    S sends data frame after SIFS;
  Else
    $t_{duration} = T^a_{scts} + T^a_{data} + T^a_{ack} + 3SIFS$
    S sends SRTS frame after $t' - T^b_{cts} + SIFS$;
  Endif
Else
  S starts a *backoff timer* and sends *RTS* on *CH b* when it times out.
Endif

4. *Other nodes update their NAV*:
If receiving a frame on *CH a*
  If $NAV^a < t_{duration}$
    $NAV^a = t_{duration}$
  Endif
Else If receiving a frame on *CH b*
  If $NAV^b < t_{duration} + t_{defer}$
    $NAV^b = t_{duration} + t_{defer}$
  Endif
Endif

5. *Destination D receives SRTS and sends SCTS*:
If D receives SRTS on *CH a*
  $t_{duration} = t^{srts}_{duration} - T^a_{scts} - SIFS$
  D sends SCTS frame after SIFS;
Endif

6. *Source S receives SCTS and sends DATA*:
If S receives SCTS on *CH a*
  $t_{duration} = t^{scts}_{duration} - T^a_{data} - SIFS$
  S sends data frame after SIFS;
Endif

7. *Destination D receives DATA and sends ACK*:
When D receives DATA successfully, it sends an ACK to S on *CH a*.

8. *For other nodes, Converting NAV^b to NAV^a*:
If $NAV^a == 0$ and $NAV^b > 0$
  $NAV^a = NAV^b$; $NAV^b = 0$;
Endif

Fig. 3. The major steps in MAC-SCC. (Note that, we use $t^{frame}_{duration}$ to represent the $duration$ field in the received frame, where $frame$ could be RTS, CTS, SRTS, SCTS, DATA or ACK. Similarly, t^{frame}_{defer} is used to represent the $defer$ field in the received frame, where $frame$ could be RTS or CTS.)

data frame during the interval, and sends RTS on *CH b*. If both *CH a* and *CH b* are busy, S backs off for a random time and then sends RTS on *CH b*.

In Step 2, if destination D receives RTS on *CH a* successfully, it sets $t_{duration}$ of CTS to be the duration for completing the data transmission (i.e., $T^a_{data}+T^a_{ack}+2SIFS$) and responds with a CTS on *CH a* after SIFS. Otherwise, if D receives RTS on *CH b* and its NAV^b is 0, it sets t_{defer} to be the interval that either S cannot send or receive data frames (i.e., $t^{rts}_{defer} - SIFS - T^b_{rts}$) or D cannot send or receive data frames (i.e., NAV^a). Then D sets $t_{duration}$ of CTS to be the time to complete the next data transmission ($T^a_{data} + T^a_{ack} + T^a_{srts} + T^a_{scts} + 4SIFS$). Finally, D sends CTS on *CH b*.

In Step 3, when S receives CTS on *CH a*, it updates $t_{duration}$, and sends data frame. When S receives CTS on *CH b*, it sends SRTS frame after the deferred time. If S cannot receive CTS, it assumes collision occurred and backs off for a random time before sending another RTS on *CH b*.

In Step 4, upon receiving a valid frame, other nodes, which are neither the source nor the destination, update their NAV^a or NAV^b based on the *duration* field. Note that, after setting NAV^b, the signaling on *CH b* is forbidden for a period until the current data transmission is finished (as shown in Step 8). In other words, *CH b* is used to schedule the next data frame only (but not the third data frames or beyond in the future), because to schedule additional data frames (other than the next one) increases the complexity but does not help improve channel efficiency noticeably.

In Step 5, destination D responds with SCTS after it receives SRTS. In step 6, source S sends the data frame after it receives SCTS. As shown in Step 7, D responds with an ACK frame after it receives the data frame successfully. Finally, in Step 8, when NAV^a becomes 0 and $NAV^b > 0$, it means the current data transmission is over, and the next data frame is ready to be sent. The node sets $NAV^a = NAV^b$, which is the duration for the next data transmission, and sets $NAV^b = 0$ to free up *CH b* for sending control frames.

B. Optimal Bandwidth Partitioning

One important design issue in MAC-SCC is bandwidth partitioning. More specifically, one needs to decide the minimum amount of bandwidth (W_s) to be allocated for the control channel such that there is a high probability ($1 - \varepsilon$ where ε is a small value) that at least one RTS/CTS pair for the next data frame succeeds during the transmission of the current frame. The bandwidth for *CH a* ($W_d = W - W_s$ where W is the total bandwidth), on the other hand, should be as large as possible to achieve high peak data rate.

For analytical tractability, we ignore the hidden station problem in the following discussion. As a result, the probability of a successful RTS (P_s) equals the probability that no arrivals occur during the propagation and medium sense delay (τ) [30]. In addition, since the SRTS and SCTS frames are free of collision, they are considered as part of data frame in our analysis. Assume that RTS arrival is Poisson distributed with an average rate of λ, the probability that an RTS can be received correctly without collision is:

$$P_s = e^{-\tau\lambda}. \tag{1}$$

Let k be the number of retry needed to achieve a success probability of $1 - \varepsilon$, i.e., $(1 - e^{-\tau\lambda})^k < \varepsilon$, then,

$$k > \frac{log(\varepsilon)}{log(1 - e^{-\tau\lambda})}. \tag{2}$$

Denote l_{data}, l_{ack}, l_{rts} and l_{cts} to be the length for data frame, ACK frame, RTS frame and CTS frame, respectively. The time for data transmission on *CH a* is

$$T_A = \frac{l_{data} + l_{ack}}{W_d} + SIFS. \tag{3}$$

Assume the RTS frames have an exponential increase of backoff time after each collision. More specifically, the backoff time after the ith collision is $2^i \frac{l_{rts}}{W_s}$. Accordingly, the time for successfully exchanging RTS/CTS frames after k attempts on *CH b* is

$$\begin{aligned} T_B &= \sum_{i=0}^{k-1} 2^i \frac{l_{rts}}{W_s} + \frac{l_{cts}}{W_s} + SIFS \\ &= (2^k - 1)\frac{l_{rts}}{W_s} + SIFS + \frac{l_{cts}}{W_s}. \end{aligned} \tag{4}$$

The optimal bandwidth partitioning should allow RTS/CTS on *CH b* to complete before the end of the current data transmission on CHa. Let $T_A = T_B$, we can obtain the optimal value of the division of bandwidth (D),

$$D = \frac{W_d}{W_s} = \frac{l_{data} + l_{ack}}{2^{k-1}(l_{rts} + l_{cts})}. \tag{5}$$

Plugging typical values for the parameters in the above equations, we can have an observation on the bandwidth partitioning. As discussed in [4], we assume $l_{rst} = 20\ bytes$, $l_{ack} = l_{cts} = 16\ bytes$, and $\tau = 10\ \mu s$. In addition, we consider IP traffic with an average size of $l_{data} \approx 1500\ bytes$, $\lambda = 800/sec$ and $\varepsilon = 10^{-5}$. According to Equations (1)-(5), we arrive at $D \approx 9.99$ or $W_s \approx 0.09W$. In other words, *CH b* needs about 9% of the total bandwidth to ensure an RTS success probability of $1 - 10^{-5}$.

C. Further Discussion

1) Comparison with Single Channel MAC Protocols: Compared with typical MAC protocols with a single channel, the use of a separate control channel in MAC-SCC may increase the implementation cost. However, our study shows that such cost increase is very limited. We address this issue in two folds. First, we discuss it from the point of view of bandwidth cost and efficiency. Note that, while MAC-SCC employs two subchannels, it doesn't require additional bandwidth. In our performance evaluation, we assume that the same bandwidth is available for MAC-SCC and IEEE 802.11 networks. The latter uses all bandwidth as a single channel, while the former divides the total available bandwidth into two subchannels. Our results show that, with the same total bandwidth, MAC-SCC achieves higher channel efficiency and fairness. Second, we discuss this problem from the point of view of implementation cost. In MAC-SCC, the transceiver needs to operate on two channels. However such tunable transceiver shall have a complexity similar to the current IEEE 802.11 devices, which is already tunable with a limited switching delay (usually about 100s) [4]. Note that, such delay has no significant impact on our MAC-SCC protocol. More specifically, if the data channel (i.e., CH A) is idle, all frames (RTS, CTS, DATA, and ACK) are transmitted on CH A only. Thus there is no channel switching at all. Only when CH A is busy, the stations need to transmit the control frames on CH B first, and then switch to CH A for data transmission, which results in about 100s delay. Note that however, since ongoing transmission on CH A is very likely to last for much longer than 100s, the channel switching delay will not lead to the waste of bandwidth on CH A. Therefore, we believe that the tunable transceiver can be developed for MAC-SCC with a similar cost compared with the ones currently used in IEEE 802.11 devices. Other possible increase in software complexity, e.g., due to the maintenance of two NAVs, shall be very limited as we have observed in our simulation.

2) Comparison with Other Multi-Channel MAC Protocols: Table III summarizes the features of our proposed MAC-SCC protocol and compares it with several representative multi-channel MAC protocols, such as DBTMA [15] and multi-channel MAC [21], as well as the popular IEEE 802.11 MAC protocol [4]. DBTMA achieves excellent channel efficiency. At the same time, however, it needs extra hardware to enable the busy tones. Additionally, DBTMA is not efficient at error recovery, since it has no ACK in MAC layer and completely relies on transport layer protocol to recovery errors.

Both MAC-SCC and multi-channel MAC protocol can achieve good channel efficiency under high traffic load. But multi-channel protocol requires simultaneous transmission/reception, resulting in complex and costly hardware implementation. Although the multi-channel scheme with N subchannels allows multiple concurrent transmissions, the peak rate of each node is decreased to $1/N$, because one node usually uses one sub-channel for data transmission. The performance of multi-channel protocols depends on node density. If the number of nodes in the transmission range is less than N, some channel bandwidth is wasted.

In addition, any existing multi-channel MAC protocol with

TABLE III

COMPARISON OF MAC-SCC, DBTMA, MULTI-CHANNEL PROTOCOL AND IEEE 802.11

Features	MAC-SCC	DBTMA	Multi-channel protocol	IEEE 802.11
Channels	2	2 + 2 busy tones	> 2	1
Optimal channel partitioning	Optimized	No	Studied in simulation	N/A
Simultaneous transmission/reception	Not required	Yes	Yes	Not required
NAV	2 NAV	No	1 NAV	1 NAV
Pipelined channel reservation	Yes	No	No	No
MAC ACK	Yes	No	Yes	Yes
Handshake sequence	RTS-CTS-(SRTS-SCTS)-DATA-ACK	RTS-CTS-DATA	RTS-CTS-DATA-ACK	RTS-CTS-DATA-ACK
Fairness	Yes	Not investigated	Not investigated	No
Stability	High	Not investigated	Not investigated	Low
Channel efficiency	High at high traffic load	High	High at high traffic load	Low at high traffic load

TABLE IV

SIMULATION PARAMETERS (I)

RTS Length	20 bytes
CTS Length	14 bytes
SRTS Length	8 bytes
SCTS Length	8 bytes
ACK Length	14 bytes
Average Data Frame Payload Length	1500 bytes
DIFS	50 μs
SIFS	10 μs
Propagation Delay	10 μs
Total Bandwidth	11 Mbps

a control channel [15], [21], [22] maintains only one NAV. The control channel is used by the nodes to compete for the current transmission alone. In contrast, our proposed MAC-SCC protocol maintains two NAVs respectively for the data channel and the control channel, as discussed in Section III. The transmission of the next data frame can be pre-scheduled during the current data transmission via the separate control channel, and thus reducing the frame collision probability and the bandwidth wasted during backoff. Moreover, the usage of the separate control channel helps to achieve fair medium access and solve the instability problem resulted from frequent link failures.

IV. SIMULATION AND DISCUSSION

We have done extensive simulations to evaluate the performance of the proposed MAC-SCC protocol, in terms of throughput, link failure probability, and fairness. We first implement MAC-SCC as a stand-alone protocol using PARSEC [31] to verify its correctness and effectiveness. Then, we implement MAC-SCC in QualNet [32] to study its performance in the whole protocol stack.

A. Stand-alone Simulation in PARSEC

We have implemented MAC-SCC by using PARSEC and studied it as a stand-alone protocol without interaction with other network layer protocols. We simulate 25 and 50 static nodes uniformly distributed in a circle with a diameter of $500\ m$. Each node has a default transmission range of $250\ m$. The parameters defined in the IEEE 802.11 standard, such as the length of SIFS, DIFS, propagation delay, and control frames, are adopted in this simulation (as listed in Table IV). We consider Internet traffic with an average packet length of 1500 bytes, and the data frame length can be calculated

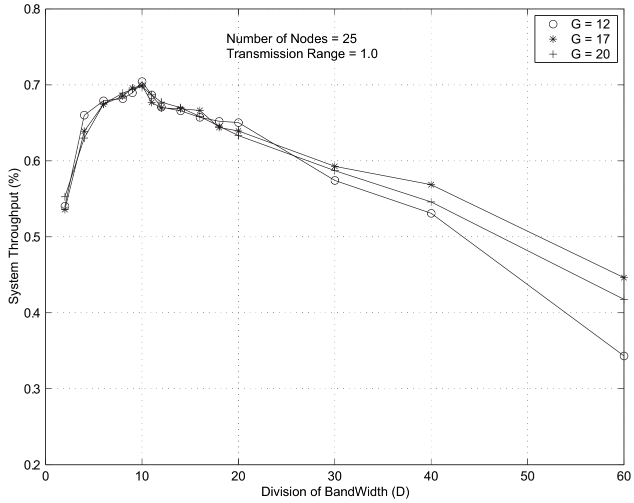


Fig. 4. Optimal Bandwidth Partitioning.

accordingly. The packet arrival is a Poisson process with a variable average rate. The traffic load (G) is defined to be the number of frames per frame time [33]. The total available bandwidth is assumed to be 11 Mbps.

Note that the transmission times of control/data frames on different channels are different. Let L be the frame transmission time in the single channel scheme (i.e., in the IEEE 802.11). In MAC-SCC, the time for transmitting the frame on CHa is $(D+1)/D \times L$, while the time for transmitting the frame on CHb is $(D+1) \times L$. In this simulation, the main performance metric is the system throughput, which is defined as the amount of successful data transmission measured by the percentage of the maximal bandwidth (11 Mbps). We also consider the link failure probability as discussed in Section II and study the optimal bandwidth partitioning.

1) Optimal bandwidth partitioning: Fig. 4 shows the system throughput with different bandwidth partitionings under various traffic loads. With an increase in bandwidth for *CH b*, it takes less time to transmit RTS/CTS on *CH b* and accordingly has a higher probability to succeed. But since the bandwidth of *CH a* is decreased, data transmission then takes a longer time. This tradeoff has been studied by simulation. As we can see, the optimal value of D is about 10, verifing the analytic results discussed in Section III-B. The control channel (or the data channel) becomes the bottleneck when $D \gg 10$ (or $D \ll 10$). In the following discussion, $D = 10$ is used as

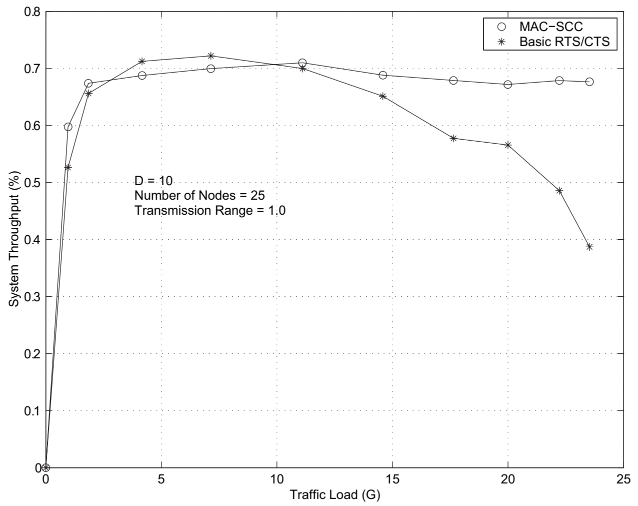


Fig. 5. Throughput comparison.

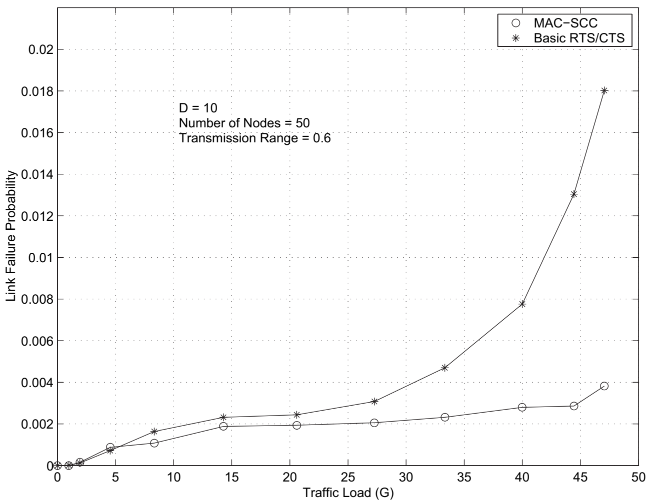


Fig. 6. Link failure probability.

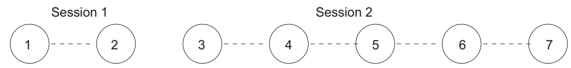


Fig. 7. Two TCP sessions running in a network with a line topology.

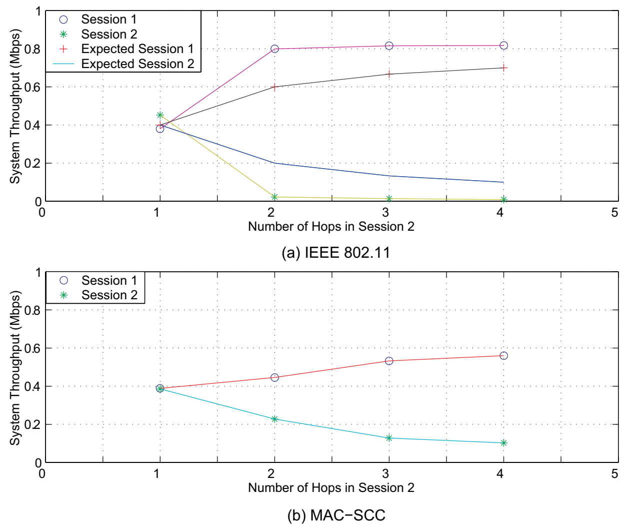


Fig. 8. Fairness in TCP.

a default value.

2) Throughput: The throughput of MAC-SCC and the basic RTS/CTS scheme for a network with 25 nodes are depicted in Fig. 5. In the basic RTS/CTS scheme, the system throughput decreases rapidly with the increase of G, because under a heavy traffic load, the collision probability is high, and the long backoff time results in low channel efficiency. In MAC-SCC, since the next data transmission can be scheduled on *CH b* even when *CH a* is busy, the collision probability and the backoff time are reduced. The proposed MAC-SCC is more robust under a heavy traffic load and can maintain a high throughput with up to 60% gain, when compared with the basic RTS/CTS approach. Similar experimental results are obtained when we simulated 50 nodes, each with a transmission range of 150 m.

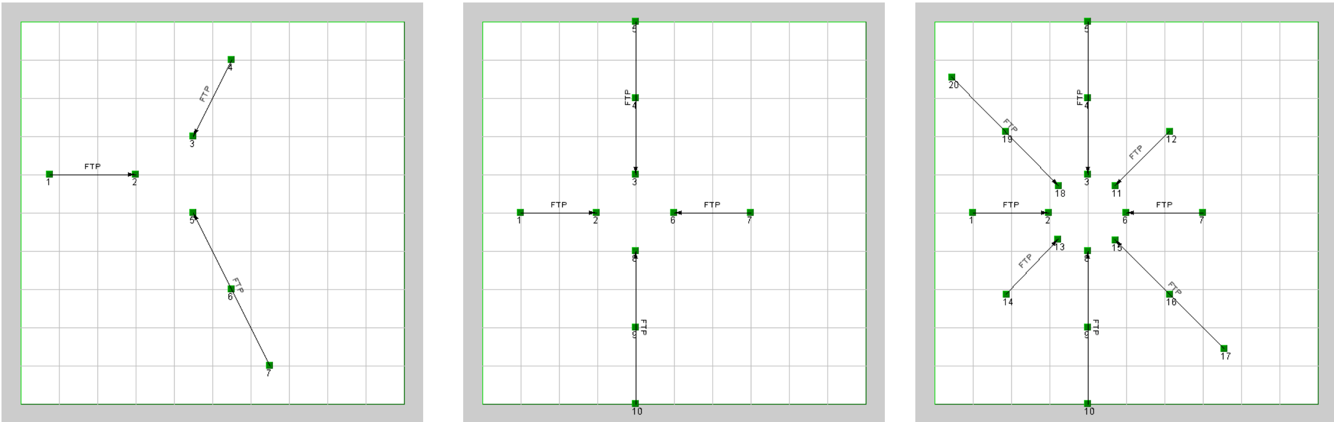
3) Link failure probability: Fig. 6 shows the link failure rate (reported to the upper layer) as a function of traffic load. As discussed in Section II, the IEEE 802.11 MAC protocol reports a link failure after seven consecutive RTS attempts fail, to the upper layer, which in turn starts a recovery procedure, e.g., to discover a new route. Clearly, frequent link failures may result in a large amount of recovery overhead. As shown in Fig. 6, MAC-SCC significantly reduces the link failure rate, because when *CH a* is busy, the node can still send control frames on *CH b* and avoid multiple RTS retransmissions due to the lack of responses from the receiving node.

B. Comprehensive Simulation in QualNet

QualNet [32] is a commercial network simulator with the whole protocol stack (from physical layer to application layer). In order to implement MAC-SCC in QualNet, we have modified the IEEE 802.11b physical layer model to enable two subchannels. The new MAC-SCC protocol is added into the protocol stack, following our discussion in Section III. The results obtained from our simulation are discussed as follows.

1) Fairness among TCP sessions: In order to study the fairness problem among TCP sessions, we first adopt a string topology similar to that used in [5] (see Fig. 7). The nodes are equally spaced. The distance between two adjacent nodes equals to the nodal transmission range. Thus, an end-to-end connection usually spans over multiple hops. For example, if a packet is sent from node 1 to node 5, it must go through 4 hops to reach its destination. We simulated two FTP sessions: Session 1 with one hop from node 1 to node 2 and Session 2 with multiple hops. The destination of Session 2 is always node 3, while the source varies from nodes 4, 5, 6 to 7, resulting in 1, 2, 3 or 4 hops, respectively. Each FTP session starts at the first second of our simulation and transmits 2000 data frames that are generated by using *tcplib*, a library for realistic TCP/IP network traffic generation [34]. Fig. 8 (a) and (b) show the throughput of these two FTP sessions running on IEEE 802.11 and MAC-SCC, respectively.

Before examining the results, let us first discuss the "ideal" fairness between two TCP sessions: Session 1 with one hop and Session 2 with k hops (as shown in Fig. 7). If the two sessions run individually, their throughputs are S and S/k,



(a) 3 TCP sessions. (b) 4 TCP sessions. (c) 8 TCP sessions.

Fig. 9. Network configuration with multiple TCP sessions.

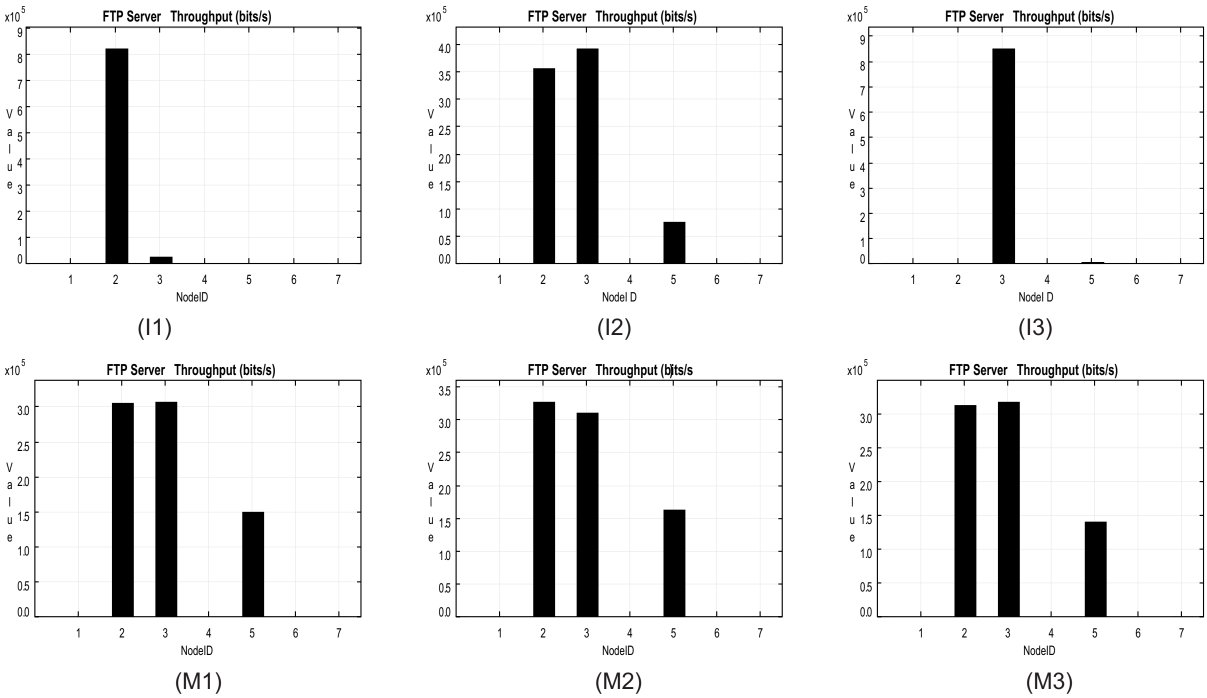


Fig. 10. FTP Server Throughput Comparison. Three sessions with nodes 3, 4, and 5 as servers. M1, M2, and M3 stand for MAC-SCC with seed 1, 2, and 3, respectively. I1, I2, and I3 stand for IEEE 802.11 with seed 1, 2, and 3, respectively.

respectively[2]. When they run simultaneously and the MAC protocol is perfectly fair, their throughput should be $(k-1+0.5)\times S/k$ and $0.5\times S/k$, respectively, because Session 2 has k hops and interferes with Session 1 only at the last hop. More specifically, $(k-1)/k$ of total traffic in Session 1 will not be interfered by Session 2. For remaining $1/k$ of total traffic, we can assume half of it will be transmitted successfully if the MAC protocol is fair. On the other hand, all traffic of Session 2 will experience contention. The ideal throughput results are depicted in Fig. 8(a) (shown as "Expected Session 1" and "Expected Session 2"). From Fig. 8(a), we clearly observe the unfairness problem of IEEE 802.11, in which Session 2 with more than one hop is almost shut down completely. When Session 2 has one hop only, although the average throughput of Session 2 is about the same as that of Session 1, our simulation exhibits that the channel might be dominated randomly by either Session 1 or Session 2. In a sharp contrast, when MAC-SCC is employed, the throughput of the two sessions (given in Fig. 8(b)) shows almost ideal fairness and is very close to the expected results of the perfectly fair protocol (as illustrated in Fig. 8(a)), signifying that MAC-SCC can effectively relieve the unfairness problem of IEEE 802.11.

We have also studied the scenarios with more than two TCP sessions. Specifically, the results obtained from three typical scenarios (see Figs. 9(a)-9(c)) are presented here. Scenario 1 contains three TCP sessions: two 1-hop sessions and one 2-hop session; Scenario 2 contains four TCP sessions: two 1-hop sessions and two 2-hop sessions; Scenario 3 contains eight TCP sessions, four 1-hop sessions and four 2-hop sessions. For each scenario, three different seeds are used in simulation. The

[2]Though not shown here, this has been verified via our simulation results.

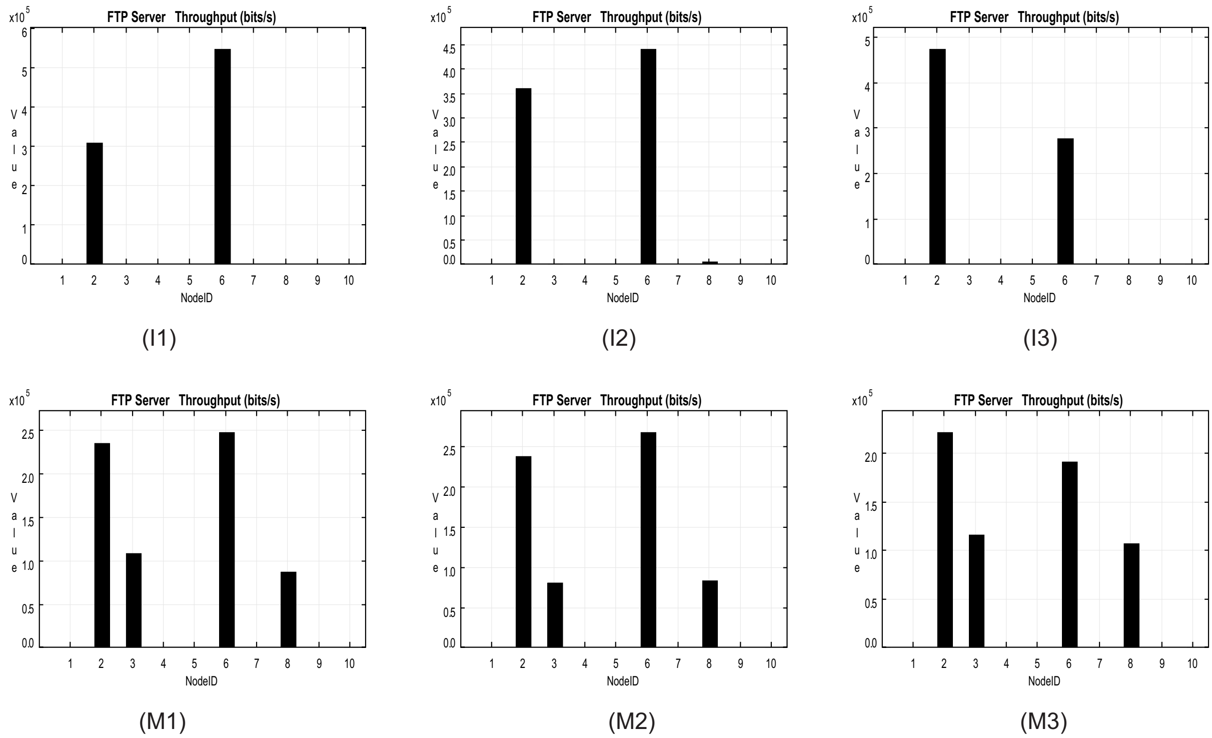


Fig. 11. FTP Server Throughput Comparison. Four sessions with nodes 2, 3, 6, and 8 as servers. M1, M2, and M3 stand for MAC-SCC with seed 1, 2, and 3, respectively. I1, I2, and I3 stand for IEEE 802.11 with seed 1, 2, and 3, respectively.

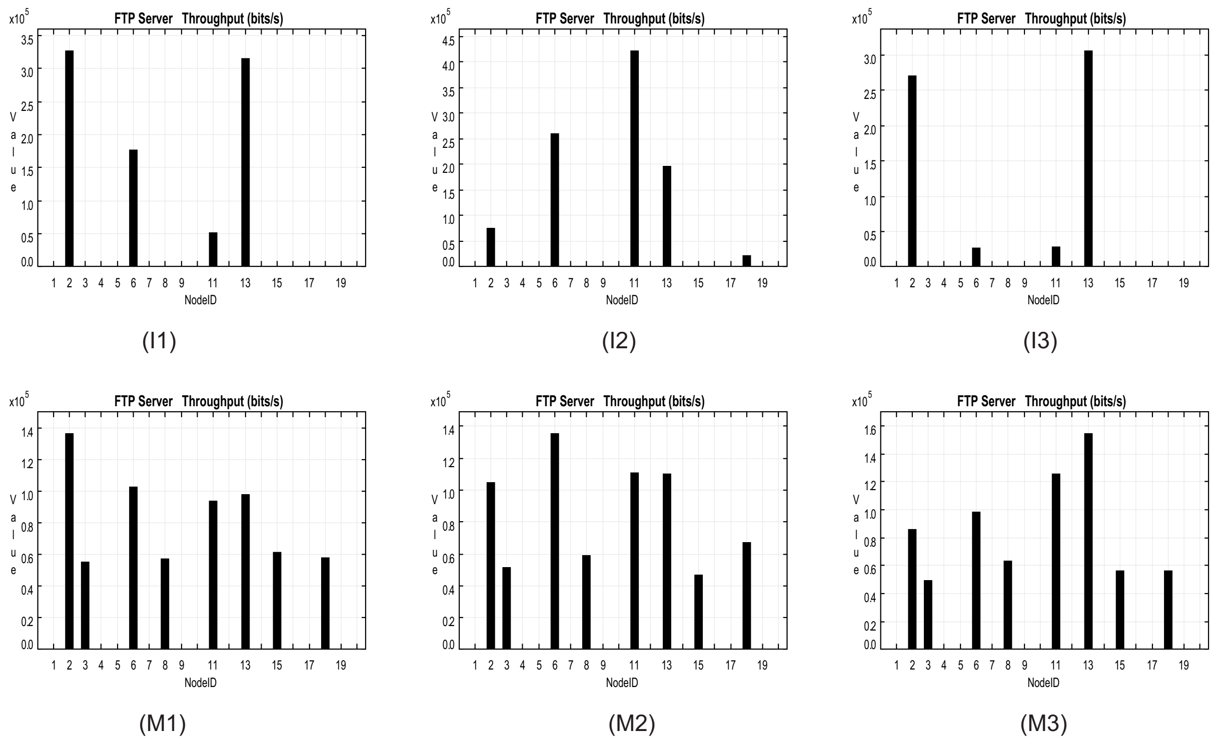


Fig. 12. FTP Server Throughput Comparison. Eight sessions with nodes 2, 3, 6, 8, 11, 13, 15, and 18 as servers. M1, M2, and M3 stand for MAC-SCC with seed 1, 2, and 3, respectively. I1, I2, and I3 stand for IEEE 802.11 with seed 1, 2, and 3, respectively.

results of Scenario 1 are shown in Fig. 10, where “I1”, “I2”, and “I3” stand for IEEE 802.11 with seed 1, 2, and 3, respectively while “M1”, “M2”, and “M3” stand for MAC-SCC with seed 1, 2, and 3, respectively. As can be seen, the 2-hop session is almost shut down completely when IEEE 802.11 protocol is employed. The throughput is dominated randomly by one of the two 1-hop sessions. In contrast, MAC-SCC shows perfect fairness (as we have discussed earlier), where the 1-hop sessions have almost the same system throughput, while the 2-hop session has about half of the throughput of 1-hop sessions. Scenarios 2 and 3 (see Fig. 11 and Fig. 12) also show a similar trend. With more TCP sessions, the interference

TABLE V
SIMULATION PARAMETERS (II)

Session	Start(s)	End(s)	Interval(s)
1	70	100	0.01
2	70	100	0.05
3	82.49	199	0.025
4	91.39	248	0.008
5	107.8	274	0.011
6	107.8	274	0.011

TABLE VI
THROUGHPUT OF UDP WITH MAC-SCC OR 802.11

Scalar	Traffic Load	MAC-SCC (Mbps)	802.11(Mbps)
1	0.92	1.2012	1.4367
0.5	1.82	1.2777	1.3830
0.4	2.28	1.3322	1.3766
0.3	3.04	1.3518	1.2789
0.2	4.56	1.2492	0.9305
0.1	9.12	1.2379	0.8497

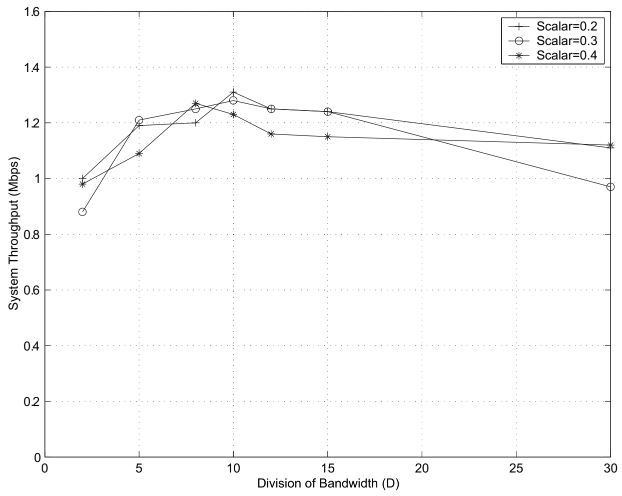


Fig. 13. Bandwidth Partitioning under VBR traffic.

among the nodes become more significant. As a result, the above simple analysis on perfect fairness no longer holds. As we can see from the figures, however, MAC-SCC do improve the fairness effectively compared with IEEE 802.11 protocol.

2) UDP throughput: In order to study the throughput of UDP running over MAC-SCC, we simulate 50 nodes that are randomly distributed in a square area of $250m \times 250m$. The transmission range of a node is $250\ m$ and thus most nodes are within the transmission range of each other. The packet length is exponentially distributed with an average length of 512 bytes. We simulated 6 TRAFFIC-GEN sessions in the application layer. The start time, the end time, and the default packet interval time of each session are shown in Table V. We vary the traffic load by multiplying the default packet interval time of all sessions with a scalar equal to 1, 0.5, 0.4, 0.3, 0.2, 0.1, respectively. We define the normalized traffic load to be total traffic loads of six sessions divided by total available bandwidth. The normalized traffic loads (that are corresponding to different scalar values) are shown in the second column of Table VI. The throughput outcomes of MAC-SCC and IEEE 802.11 are listed in the third and the fourth column of Table VI, respectively. As can be seen, MAC-SCC exhibits roughly 84% throughput of that under IEEE 802.11 when the traffic load is low. When the traffic load increases beyond 3, MAC-SCC outperforms IEEE 802.11 and keeps a very stable throughput. When traffic load reaches 9.12, for example, MAC-SCC has a throughput gain of 46%, compared with IEEE 802.11. The gap is expected to grow with traffic load as we discussed earlier in Section IV-A. When the traffic load is extremely high (such as 99.1), 60% or higher throughput gain can be achieved.

3) Bandwidth Partitioning: In our earlier analysis and stand-alone simulations, we have assume that the packet arrival at MAC layer is a Poisson process. This however, may not be true due to various applications and the cross-layer interactions, which may consequently affect the channel partitioning. To evaluate the impact of traffic on channel partitioning, we simulate 50 nodes that are randomly distributed in a square area of $250m \times 250m$. The nodes have a transmission range of $250m$. Six variable bit rate (VBR) applications are simulated in the application layer. The start time, the end time, and the default packet interval time of each session are shown in Table V. The packets are with average length of 1500 bytes and sent at variable rate which is a function of $interval \times scalar$. The system throughput with different bandwidth partitioning are shown in Fig. 13. As we can see, the system with bandwidth partition of 10 always yield optimal or near-optima results. We have also carried out simulations under other traffic patterns, such at FTP and TRAFFIC-GEN traffic. We observe that the system always reaches the highest throughput when the division of bandwidth (D) is between 8 and 12.

V. CONCLUSION

We have proposed a novel Medium Access Control protocol with a Separate Control Channel (*MAC-SCC*). In MAC-SCC, the available bandwidth is partitioned into two channels: a data channel and a control channel, each associated with a network allocation vector (NAV). To reduce hardware complexity, the station transmits or receives on one channel only at any given time. The performance of MAC-SCC is quantified via extensive simulations in both a stand-alone simulator developed by using PARSEC and a comprehensive network simulator called QualNet with whole protocol stack. Our results show that MAC-SCC can effectively reduce the link failure probability, achieve fair medium access when running multiple TCP sessions, and yield a throughput gain up to 60% under high traffic load, when compared with the basic RTS/CTS scheme.

REFERENCES

[1] C. Perkins, *Ad Hoc Networking*. Addison Wesley, 2001.
[2] E. M. Royer, S. J. Lee, and C. E. Perkins, "The effects of MAC protocols on ad hoc network communications," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'00)*, pp. 543–548.
[3] H. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol," *Wireless Networks, The Journal of Mobile Communication, Computation and Information*, vol. 3, no. 3, pp. 217–234, 1997.
[4] IEEE, *IEEE std 802.11–Wireless LAN media access control (MAC) and physical layer (PHY) specifications*, 1997.

[5] S. Xu and T. Saadawi, "Revealing the problems with 802.11 MAC protocol in multi-hop wireless ad hoc networks," *Journal of Computer Networks*, vol. 38, no. 4, pp. 531–548, Mar. 2002.

[6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," in *Proc. ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM'94)*, pp. 212–225.

[7] M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multi-hop networks," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications - WMSCA '99*, pp. 41–50.

[8] E. S. Jung and N. H. Vaidya, "A power control MAC protocol for ad hoc networks," in *Proc. ACM/IEEE the 8th Annual International Conference on Mobile Computing and Networking (MobiCom'02)*, pp. 36–47.

[9] J. Monks, V. Bharghavan, and W. M. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proc. IEEE Conference on Computer Communication (INFOCOM'01)*, pp. 219–228.

[10] L. Bao and J. J. Garcia-Luna-Aceves, "Transmission scheduling in ad hoc networks with directional antennas," in *Proc. ACM/IEEE the 8th Annual International Conference on Mobile Computing and Networking (MobiCom'02)*, pp. 48–58.

[11] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, "Using directional antennas for medium access control in ad hoc networks," in *Proc. ACM/IEEE the 8th Annual International Conference on Mobile Computing and Networking (MobiCom'02)*, pp. 59–70.

[12] Y. B. Ko, V. Shankarkumar, and N. H. Vaidya, "Medium access control protocols using directional antennas in ad hoc networks," in *Proc. IEEE Conference on Computer Communication (INFOCOM'00)*, pp. 13–21.

[13] J. J. Garcia-Luna-Aceves and J. Raju, "Distributed assignment of codes for multihop packet-radio networks," in *Proc. IEEE Military Communications Conference (MILCOM'00)*, pp. 595–602.

[14] A. Farago, A. D. Myers, V. R. Syrotiuk, and G. Zaruba, "Meta-MAC protocols: Automatic combination of MAC protocols to optimize performance for unknown conditions," *IEEE J. Select. Areas Commun., Analysis and Synthesis of MAC Protocols*, vol. 18, no. 9, pp. 1670–1681, Sept. 2000.

[15] J. Deng and Z. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," in *Proc. IEEE Internaltional Conference on Universal Personal Communications (ICUPC'98)*, pp. 973–977.

[16] S. L. Wu and Y. C. Tseng, "Intelligent medium access for mobile ad hoc networks with busy tones and power control," *IEEE J. Select. Areas Commun.*, vol. 18, no. 9, pp. 1647–1657, 2000.

[17] A. Nasipuri, J. Zhuang, and S. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'99)*, pp. 1402–1406.

[18] A. Nasipuri and S. Das, "Multichannel CSMA with signal power-based channel selection for multihop wireless networks," in *Proc. IEEE Fall Vehicular Technology Conference (VTC Fall 2000)*, pp. 211–218.

[19] S.-H. Hsu, C.-C. Hsu, S.-S. Lin, and F.-C. Lin, "A multi-channel mac protocol using maximal matching for ad hoc networks," in *Proc. 24th International Conference on Distributed Computing Systems Workshops 2004*, pp. 505–510.

[20] A. Tzamaloukas and J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *Proc. IEEE Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, pp. 189–198.

[21] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," in *Proc. IEEE 9th International Conference on Computer Communications and Networks (IC3N'01)*, pp. 432–439.

[22] S. Wu, C. Lin, Y. Tseng, and J. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for mobile ad hoc networks," in *Proc. International Symposium on Parallel Architectures, Algorithms and Networks 2000*, pp. 232–237.

[23] N. Choi, Y. Seok, and Y. Choi, "Multi-channel mac protocol for mobile ad hoc networks," in *Proc. IEEE 58th Vehicular Technology Conference 2003*, pp. 1379–1382.

[24] W. C. Hung, K. E. Law, and A. Leon-Garcia, "A dynamic multi-channel mac for ad-hoc lan," in *Proc. 21st Biennial Symposium on Communications 2002*, pp. 31–35.

[25] S. Wu, Y. Tseng, C. Lin, and J. Sheu, "A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks," *The Computer Journal*, vol. 45, no. 1, pp. 101–110, 2002.

[26] J. Zhang, Y. Wang, and J. Wang, "Dcc-mac: a new mac protocol for ad-hoc networks based on dual control channel," in *Proc. 14th IEEE Personal, Indoor and Mobile Radio Communications 2003*, pp. 1341–1345.

[27] F. Fitzek, D. Angelini, G. Mazzini, and M. Zorzi, "Design and performance of an enhanced ieee 802.11 mac protocol for multihop coverage extension," *IEEE Wireless Commun.*, vol. 10, no. 6, pp. 30–39, 2003.

[28] J. So and N. Vaidya, "Multi-channel mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC'04)*, pp. 222–233.

[29] J. Pathmasuntharam, A. Das, and A. Gupta, "Primary channel assignment based mac (pcam) - a multi-channel mac protocol for multi-hop wireless networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'04)*, pp. 1110–1115.

[30] J. J. Garcia-Luna-Aceves and C. L. Fullmer, "Floor acquisition multiple access (FAMA) in single-channel wireless networks," *ACM Mobile Networks & Applications Journal (MONET)*, vol. 4, no. 3, pp. 157–174, 1999.

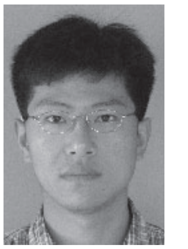
[31] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song, "PARSEC: A Parallel Simulation Environment for complex systems," *Computer*, pp. 77–85, October 1998.

[32] S. N. Technologies, *QualNet user manual*, available at *http://www.scalable-networks.com/*.

[33] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*. Prentice Hall, 2002.

[34] P. Danzig and S. Jamin, "Tcplib:a library of tcp internetwork traffic characteristics," 1991.

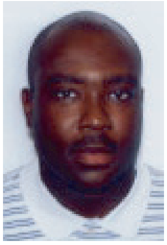
Yijun Li (S'05) Yijun Li received the BS degree in electrical engineering and the MEng degree in computer science and engineering from the Harbin Institute of Technology (HIT), China, in 1994 and 1997, respectively, and the MS degrees in computer engineering from the University of Louisiana at Lafayette, LA, U.S.A, in 2003. From 1997 to 2001, he was working on compiler design and raster image processing as a senior software engineer in Beijing University Founder Electronic Cor. China. He is now pursuing the Ph.D degree in Computer Engineering at the University of Louisiana at Lafayette, U.S.A. He is working on embedded system HW/SW co-design, MAC/PHY layer design for wireless telecommunication and VLSI design for JPEG2000 and H.264.



Hongyi Wu (M'02) received his Ph.D. degree in Computer Science and M.S. degree in Electrical Engineering from State University of New York (SUNY) at Buffalo in 2002 and 2000, respectively. He received his B.S. degree in Scientific Instruments from Zhejiang University in 1996. He is currently an Assistant Professor at the Center for Advanced Computer Studies (CACS), University of Louisiana (UL) at Lafayette. His research interests include wireless mobile ad hoc networks, wireless sensor networks, next generation cellular systems, and integrated heterogeneous wireless systems. He has served as chair and technical committee member of several IEEE conferences, and a guest editor of *ACM MONET* special issue on Integration of Heterogeneous Wireless Technologies. He has published more than two dozen technical papers in leading journals and conference proceedings. He received NSF CAREER Award in 2004.

Nian-Feng Tzeng (S'85-M'86-SM'92) received the Ph.D. degree in Computer Science from the University of Illinois at Urbana-Champaign. Since 1987, he has been with Center for Advanced Computer Studies, the University of Louisiana at Lafayette, where he is currently a professor. His current research interest is in the areas of computer communications and networks, high-performance computer systems, parallel and distributed processing and fault-tolerant computing. He was on the editorial board of the *IEEE Transactions on Parallel and Distributed Systems*, 1998-2001, and on the editorial board of the *IEEE Transactions on Computers*, 1994-1998. He served as a Distinguished Visitor of the IEEE Computer Society, 1994-1997, and was the Chair of Technical Committee on Distributed Processing, the IEEE Computer Society, from 1999 till 2002. He was on the technical program committees of various conferences and served as the Technical Program Chair of the 10th International Conference on Parallel and Distributed Systems, July 2004.

Dr. Tzeng is the recipient of the outstanding paper award of the 10th International Conference on Distributed Computing Systems, May 1990. He received the University Foundation Distinguished Professor Award in 1997.



Dmitri Perkins received his B.S. degree in Computer Science from Tuskegee University in 1995 and his M.S. and Ph.D. degrees in Computer Science and Engineering from Michigan State University in 1997 and 2002, respectively. Currently, Dr. Perkins is an Assistant Professor at the Center for Advanced Computer Studies at the University of Louisiana at Lafayette. His research interests include (1) the modeling, analysis, and performance evaluation of wireless networks; and (2) system-level design and development of mobile and wireless networking systems. He has published numerous journal and conference papers in these areas. In 2000, Dr. Perkins received the GE Faculty of the Future Fellowship and Grant. Dr. Perkins is also the recipient of a 2005 National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award.

Magdy Bayoumi (S'85-M'86-SM'92-F'99) received the B.Sc. and M.Sc. degrees in electrical engineering from Cairo University, Cairo, Egypt, in 1973 and 1977, the M.Sc. degree in computer engineering from Washington University in St. Louis, MO, in 1981, and the Ph.D. degree in electrical engineering from the University of Windsor, Windsor, ON, Canada, in 1984. Currently, he is the Director of the Center for Advanced Computer Studies (CACS), Department Head of the Computer Science Department, the Edmiston Professor of Computer Engineering, and the Lamson Professor of Computer Science at The Center for Advanced Computer Studies, University of Louisiana at Lafayette, where he has been a Faculty Member since 1985. He has edited and coedited three books in the area of VLSI Signal Processing. He has one patent pending. His research interests include VLSI design methods and architectures, low-power circuits and systems, digital signal processing architectures, parallel algorithm design, computer arithmetic, image and video signal processing, neural networks, and wide-band network architectures. Dr. Bayoumi received the University of Louisiana at Lafayette 1988 Researcher of the Year Award and the 1993 Distinguished Professor Award.

He was an Associate Editor of the *IEEE Circuits and Devices Magazine*, the *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, the *IEEE Transactions on Neural Networks*, and the *IEEE Transactions on Circuits and SystemsII: Analog and Digital Signal Processing*. He was an Associate Editor of the *Circuits and Devices Magazine* and is currently an Associate Editor of *Integration*, the *VLSI Journal*, and the *Journal of VLSI Signal Processing Systems*. He is a Regional Editor for the *VLSI Design Journal* and on the Advisory Board of the *Journal on Microelectronics Systems Integration*. From 1991 to 1994, he served on the Distinguished Visitors Program for the IEEE Computer Society, and he is on the Distinguished Lecture Program of the Circuits and Systems Society. He was the Vice President for technical activities of the IEEE Circuits and Systems Society. He was the Cochairman of the Workshop on Computer Architecture for Machine Perception in 1993, and is a Member of the Steering Committee of this workshop. He was the General Chairman of the 1994 MWSCAS and is a Member of the Steering Committee of this symposium. He was the General Chairman for the 8th Great Lake Symposium on VLSI in 1998. He has been on the Technical Program Committee for ISCAS for several years and he was the Publication Chair for ISCAS99. He was also the General Chairman of the 2000 Workshop on Signal Processing Design and Implementation. He was a founding member of the VLSI Systems and Applications Technical Committee and was its Chairman. He is currently the Chairman of the Technical Committee on Circuits and Systems for Communication and the Technical Committee on Signal Processing Design and Implementation. He is a Member of the Neural Network and the Multimedia Technology Technical Committees. Currently, he is the faculty advisor for the IEEE Computer Student Chapter at the University of Louisiana at Lafayette.