

MAC-SCC: Medium Access Control with a Separate Control Channel for Multihop Wireless Networks

Yijun Li, Hongyi Wu, Dmitri Perkins, Nian-Feng Tzeng, and Magdy Bayoumi
The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Email: {yxl4444,wu,perkins,tzeng,mab}@cacs.louisiana.edu

Abstract

The recent research has shown that the basic RTS/CTS approach is not efficient in multihop wireless networks. In this paper, we propose a novel medium access control protocol with a separate control channel (MAC-SCC). A distinct feature of MAC-SCC (compared with other multichannel MAC protocols) is to use two Network Allocation Vectors (NAVs) for the data channel and the control channel, respectively. The transmission of the next data frame can be pre-determined during the current transmission via the separate control channel, and thus the frame collision probability and bandwidth wasted during backoff can be reduced. The system throughput is quantified via simulation. The results show that MAC-SCC yields a throughput gain up to 60% under high traffic load and has a significant lower link failure probability.

1. Introduction

Due to the limited transmission range of wireless interfaces, multiple hops may be needed to exchange data between nodes in a packet radio system, while each node acts both as a router and as a host. An ad hoc network is one example of such a system, including a collection of self-configurable nodes which communicate with each other via wireless links without the intervention of a centralized controller [1]. The ad hoc network has been extensively studied in the last few years, with the main focus on routing protocols and medium access control (MAC) protocols. It has been shown in [2] that the choice of MAC protocols can significantly affect the performance of routing protocols, accordingly dictating the overall performance of the multihop wireless networks.

The MAC protocol is used to resolve access contention for the shared medium. To alleviate the hidden station problem, the *Request To Send (RTS)/Clear To Send (CTS)*

mechanism has been widely used in wireless local area networks and standardized in IEEE 802.11 Distributed Coordination Function (DCF) [3]. However, it has been shown that the basic RTS/CTS approach may result in significant performance degradation in multihop environment (see Section 2.1 for details). Various MAC protocols have been proposed to improve MAC performance by using power control, directional antennas, multiple channels, etc.

In this paper, we propose a novel MAC protocol using a separate control channel (MAC-SCC), in which the total bandwidth is partitioned into two sub-channels: one is primarily used for data transmission and the other for signaling. The transmission of the next data frame can be pre-determined during the current frame transmission. Accordingly, the frame collision probability and the bandwidth wasted during backoff can be reduced. The rest of this paper is organized as follows. In Section 2, we introduce the background, motivations, and related work. In Section 3, we present the proposed MAC-SCC protocol. The simulation results and discussions are presented in Section 4. Finally, Section 5 concludes the paper.

2. Background, Motivations, Related Work

2.1. Background and Motivations

Here, we briefly describe the RTS/CTS scheme defined in the IEEE 802.11 standard [3], where its detailed description can be found. As shown in Figure 1 (a), when source $S1$ has data to send to node $D1$, it first senses the medium. If the medium is idle for an interval of DIFS (DCF Inter-Frame 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 and waiting for another SIFS interval, $S1$ sends the data frame, while $D1$ will accordingly response with an ACK. If the medium is busy, the source (e.g., $S2$ in Figure 1 (a)) has to defer its RTS transmission. More specifically, $S2$ starts

a *backoff timer*, which decreases only after the medium becomes idle for an interval of DIFS and sends RTS when the timer expires. Similarly, if the source node cannot receive the CTS frame from the destination, it assumes there was a collision and also starts the *backoff timer*. The initial value of the *backoff timer* is uniformly distributed from zero to a maximum value that is exponentially increased with the number of times of retry. In addition, there is a *duration* field in RTS and CTS frames to indicate the amount of time to be reserved for data transmission. On receiving the RTS or CTS frames, other nodes (which are not the source or the destination) update their Network Allocation Vectors (NAVs), each of which is a timer used to indicate the duration of channel occupancy. A node cannot send any frame when its NAV is larger than zero.

As can be seen from Figure 1 (a), a certain amount of time is used for signaling only, and in particular, the entire bandwidth is wasted during backoff. Under high traffic load, multiple retries may fail, leading to exponentially increased backoff time and further decreased channel efficiency. Moreover, when using this protocol in ad hoc networks, some nodes may experience serious instability and unfairness problems as reported in [7]. In wireless local area networks, 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 multihop environment, and accordingly a large number of signaling frames may experience multiple collisions and long backoff time. Worse yet, the node may even falsely report a link failure. For example, in a simple network with the string topology shown in Figure 2, when node *D* has data to send to node *E*, it sends an RTS. Assuming an omnidirectional antenna for transmission, 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, while the contention window (as well as the average backoff time) will double after each retry. 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

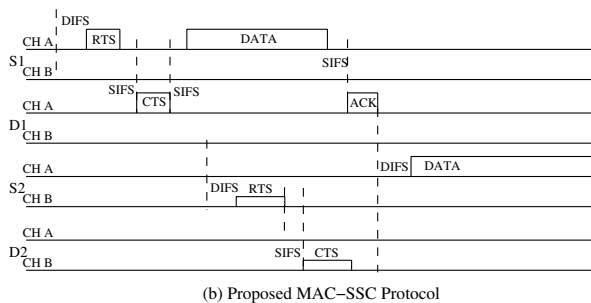
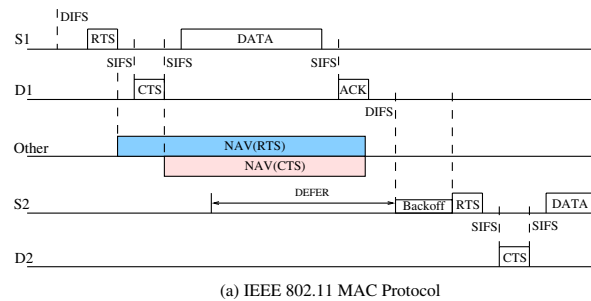


Figure 1. Using MAC-SSC protocol.

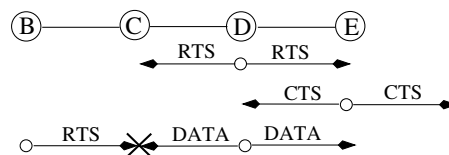


Figure 2. A problem in IEEE 802.11.

a long end-to-end delay.

2.2. Related Work

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

The dual busy tone multiple access (DBTMA) has been proposed in [4], where a single common channel is split into two sub-channels: a data channel and a control channel. two busy tones are assigned to the control channel, one to indicate that the node is transmitting, while the other to indicate that the node is receiving. The carrier sense is done by detecting the busy tones. [8] enhances DBTMA by combining the concept of power control with the RTS/CTS-based and the busy tone-based protocol to increase channel efficiency.

[9] proposes a multi-channel carrier sense multiple ac-

cess (CSMA) protocol, where the total available bandwidth is divided into N narrow-band sub-channels. An idle sub-channel is selected randomly for data transmission. [10] further improves the multi-channel protocol by maintaining a channel quality list of free sub-channels at each node.

Similar to [9, 10], the total available bandwidth is divided into a number of narrow-band sub-channels in [5]. But one sub-channel is reserved for control. Each node senses the medium and builds a free-channel list in the RTS frame of the transmitting node. The receiving node can choose the channel which is free for both of them. A similar idea is used in [6] and further enhanced in [11] with power control.

Note that, although the multi-channel scheme allows multiple concurrent transmissions, the peak rate of each node is decreased to $1/N$, because one node usually uses one sub-channel (among N sub-channels) to transmit. In addition, any existing multi-channel MAC protocol with a control channel [4, 5, 6] maintains only one NAV. The control channel is used by the nodes to compete for the current transmission alone. In our proposed MAC-SCC protocol, however, two NAVs are maintained respectively for two subchannels, as will be discussed in Section 3. Thus the control channel can be used to schedule not only the current data transmission but also the next data transmission.

3. Proposed MAC-SCC Protocol

The basic idea of our proposed MAC-SCC protocol is to divide the entire bandwidth into two channels (e.g., by using two codes with different spreading factors in Direct Sequence Spread Spectrum), i.e., $CH a$ which is primarily used for data transmission and $CH b$ which is used for signaling, with bandwidth W_d and W_s ($W_d \gg W_s$), respectively. But note that the nodes are not required to receive and transmit at the same time. We consider data transmissions that follow the four way handshake RTS-CTS-DATA-ACK, which is similar to that defined in IEEE 802.11 standard. All frames can be transmitted on $CH a$, while $CH b$ may only be used to transmit RTS or CTS frames. Note that, the same frame needs different time to be transmitted on these two channels. Hereafter, we denote T_{data}^a , T_{ack}^a , T_{rts}^a , and T_{cts}^a to be the time to transmit a data frame, an ACK frame, an RTS frame, and a CTS frame on $CH a$, respectively, while denote T_{rts}^b and T_{cts}^b to be the time to transmit an RTS and a CTS frame on $CH b$, respectively.

Each node in MAC-SCC 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 trans-

mission. When sending an RTS on $CH a$, $t_{duration} = T_{data}^a + T_{ack}^a + T_{cts}^a + 3SIFS$; When sending a CTS on $CH a$, $t_{duration} = T_{data}^a + T_{ack}^a + 2SIFS$; When sending an RTS or a CTS on $CH b$, $t_{duration} = T_{data}^a + T_{ack}^a + SIFS$ ¹. Upon receiving RTS or CTS on $CH a$ or $CH b$, a node updates its NAV^a or NAV^b . The RTS and CTS frames also include a *defer time* field to indicate the amount of time (t_{defer}) 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.

The major steps of MAC-SCC are shown in Figure 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 time for signaling, the RTS frame is sent on $CH a$, which has a larger bandwidth. If $CH a$ is busy while $CH b$ is idle, S sets $t_{defer} = NAV^a$ to indicate the interval that S cannot send data frame, 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 the destination D receives RTS on $CH a$ successfully, it sets NAV^a to be the duration to complete the data transmission (i.e., $T_{cts}^a + T_{data}^a + T_{ack}^a + 2SIFS$) and responds with a CTS on $CH a$. 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., $t^l - SIFS - T_{rts}^b$) or D cannot send or receive data frames (i.e., NAV^a). Then D sets NAV^b to be the time to complete the current data transmission (t_{defer}) and the next data transmission ($T_{data}^a + T_{ack}^a + SIFS$) plus a *DIFS* interval between two transmissions. Finally, D sends CTS on $CH b$. Note that, after setting NAV^b , the signaling on $CH b$ is forbidden for a period until the current data transmission is finished (i.e., when $NAV^a = 0$ as shown in Step 6). In other words, $CH b$ is used to schedule the next data frame only (but not the third or more data frame in the future), because to schedule additional data frames (other than the next one) increases the complexity but does not help improve the channel efficiency. If D receives RTS on $CH b$ but NAV^b is greater than 0, it implies that S did not update its NAV^b correctly (e.g., due to link error or collision). In such a case, D sends an empty frame including its NAV^b value in the header to update the NAV^b of node S .

In Step 3, when S receives CTS, it updates NAV^a or NAV^b , and starts sending data frame after the deferred time. Note that NAV^a of the source and NAV^a of the destination are employed to indicate the channel occupancy duration and to set t_{defer} . In contrast to the basic RTS/CTS scheme in IEEE 802.11, MAC-SCC allows a

¹To simplify the protocol description, we assume $CH a$ and $CH b$ use the same interframe 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 $t_{defer} = 0$;
 S sends RTS on $CH a$;
 Else if only $CH b$ is idle for a period of DIFS
 S sets $t_{defer} = NAV^a$;
 S sends RTS on $CH b$;
 Else
 S starts a *backoff timer* and sends RTS on $CH b$ when it times out.
 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 $CH a$
 S sets $NAV^a = T_{data}^a + T_{ack}^a + SIFS$;
 S sends data frame after SIFS;
 Else
 S sets $NAV^b = t' - T_{cts}^b + DIFS + T_{data}^a + T_{ack}^a$;
 S sends data frame after $t' - T_{cts}^b + SIFS + DIFS$;
 Endif
 Else
 S starts a *backoff timer* and sends RTS on $CH b$ when it times out.
 Endif
5. *Destination D receives DATA and sends ACK:*
 When D receives DATA successfully, it sends an ACK to S on $CH a$.
2. *Destination D receives RTS and sends CTS:*
 If D receives RTS successfully
 If D receives RTS on $CH a$
 D sets $t_{defer} = 0$, replies with CTS on $CH a$;
 D sets $NAV^a = T_{cts}^a + T_{data}^a + T_{ack}^a + 2SIFS$;
 Else
 If $NAV^b == 0$ (let $t' = t_{defer}$ obtained from RTS)
 D sets $t_{defer} = \max\{t' - SIFS - T_{rts}^b, NAV^a\}$;
 D sets $NAV^b = T_{data}^a + T_{ack}^a + SIFS + t_{defer} + DIFS$;
 D replies with CTS on $CH b$;
 Else
 D responses with an empty frame including NAV^b ;
 Endif
 Endif
 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 $NAV^b < t_{duration}$
 $NAV^b = t_{duration}$
 Endif
 Endif
6. *For all nodes, Converting NAV^b to NAV^a :*
 If $NAV^a == 0$ and $NAV^b > 0$
 $NAV^a = NAV^b$;
 $NAV^b = \max\{0, T_{rts}^b + T_{cts}^b + 2SIFS - t'\}$;
 Endif

Figure 3. The major steps in MAC-SCC.

source/destination to send/receive data frames even though NAV^a is larger than zero. 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 not either the source or the destination, update their NAV^a or NAV^b based on the *duration* field. As shown in Step 5, D responds with an ACK frame after it receives the data frame successfully. Finally, in Step 6, when $NAV^a == 0$ 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 $NAV^b = \max\{0, T_{rts}^b + T_{cts}^b + 2SIFS - t'\}$, due to that sometime the transmission of RTS/CTS frame on $CH b$ terminates later than data transmission on $CH a$. When $NAV^b = 0$, $CH b$ is free free up for sending control frames.

Figure 1 (b) shows one example of the MAC-SCC protocol. By using MAC-SCC, 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 $CH a$. Accordingly, the next data frame (from $S2$ to $D2$) can be transmitted as soon

as the current data transmission finishes. This is in sharp contrast to IEEE 802.11, when $S2$ has data to transmit but the medium is busy due to the data transmission between $S1$ and $D1$, it has to defer sending RTS until the medium is idle for a duration of the sum of DIFS and a random backoff time, during which the entire bandwidth is wasted. MAC-SCC can pipeline the contention of data channel for the next data frame with the transmission of the current data frame. Accordingly, the bandwidth wasted during the backoff time can be significantly reduced, and a higher channel utilization and thus an enhanced average data rate can be achieved.

4. Simulation and Discussion

We have implemented the proposed MAC-SCC and the basic RTS/CTS schemes by using PARSEC [12] for performance evaluation and comparison. We simulate 25 or 50 static nodes uniformly distributed in a circle with a diameter of 2. 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 1). We consider Internet traffic with an aver-

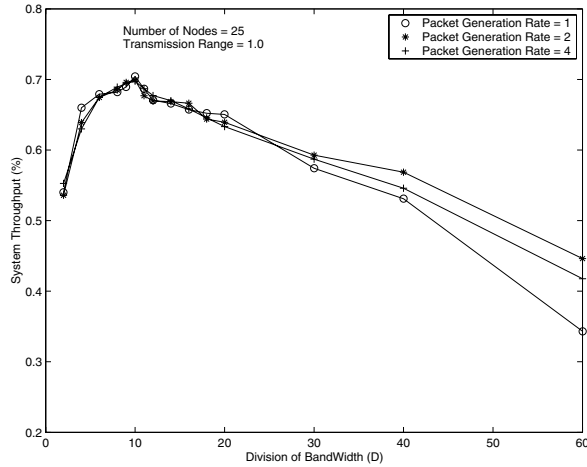


Figure 4. Optimal bandwidth partitioning.

age packet length of 1500 bytes, and the data frame length can be calculated accordingly. The packet arrival is a Poisson process with a variable average rate to simulate different traffic loads. The total bandwidth is assumed to be 11 Mbps.

Note that the transmission time 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 basic RTS/CTS scheme). 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 2 and study the optimal bandwidth partitioning.

Table 1. Simulation Parameters

| | |
|-----------------------------------|------------|
| RTS Length | 20 bytes |
| CTS Length | 14 bytes |
| ACK Length | 14 bytes |
| Average Data Frame Payload Length | 1500 bytes |
| DIFS | 50 μ s |
| SIFS | 10 μ s |
| Propagation Delay | 10 μ s |
| Total Bandwidth | 11 Mbps |

Optimal bandwidth partitioning

Figure 4 shows the system throughput with different bandwidth partitionings under the cases of different traffic

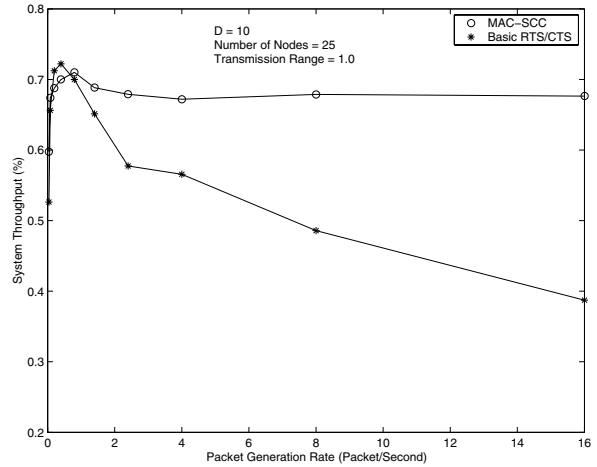


Figure 5. Throughput comparison

loads. With an increase in bandwidth for CHb , it takes less time to transmit RTS/CTS on CHb and accordingly has a higher probability to succeed. But since the bandwidth of CHa is decreased, data transmission then takes a longer time. The tradeoff has been studied in this simulation. As shown in Figure 4, the optimal value of D is about 10 for different packet generation rates. The control channel and the data channel become the bottleneck when $D \gg 10$ and $D \ll 10$, respectively. Thus $D = 10$ will be used as a default value in following discussion.

Throughput

The throughput of MAC-SCC and the basic RTS/CTS scheme are depicted in Figures 5. In Figure 5, we simulate 25 nodes, each with a transmission range of 1. In the basic RTS/CTS scheme, the system throughput decreases rapidly with an increase in the packet generation rate, because under a high 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 CHb even when CHa is busy, the collision probability and the backoff time is 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. The same experimental results are got when we simulate 50 nodes, each with a transmission range of 0.6.

Link failure probability

Figure 6 shows the link failure rate (reported to the upper layer). As discussed in Section 2, in IEEE 802.11 MAC protocol, after seven consecutive RTS attempts fail, a link failure is reported to the upper layer, which in turn starts a recovery procedure, e.g., to discover a new route. Clearly, a frequent link failure may result in a large amount of recovery overhead. As can be seen in Figure 6, MAC-SCC can

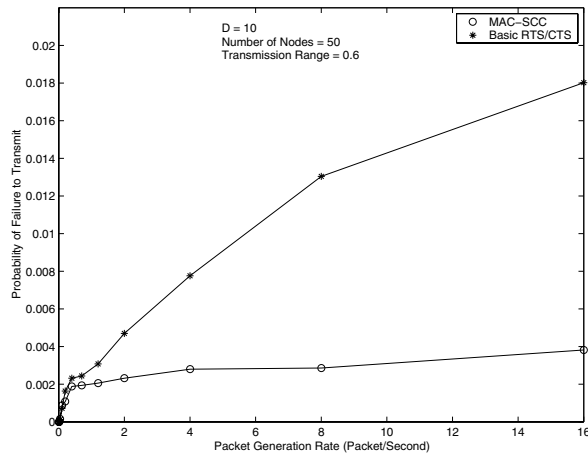


Figure 6. Link failure probability.

significantly reduce the link failure rate reported, 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. We also notice that, with the use of the defer time (t_{defer}) in scheduling the next frame, the nodes in MAC-SCC automatically decrease their packet generation rate under high traffic load, thus avoiding serious congestion.

5. Conclusion and Future Work

We have proposed a novel medium access control protocol with a separate control channel (MAC-SCC), which employs two Network Allocation Vectors (NAVs) for the data channel and the control channel, respectively. As shown by our analysis and simulation, 10% of the total available bandwidth needs to be reserved for the control channel to achieve the best performance. The simulation results have shown that MAC-SCC may achieve a throughput gain of up to 60% and lead to a far lower link failure probability, when compared with the basic RTS/CTS scheme. In our future work, we will perform theoretical analysis on the optimal bandwidth partitioning, study QoS support in MAC layer using the separate control channel, and evaluate the effect of higher layers (e.g., TCP) on the proposed MAC-SCC protocol.

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 *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'00)*, pp. 543–548, September 2000.
- [3] IEEE, IEEE std 802.11–Wireless LAN media access control (MAC) and physical layer (PHY) specifications, 1997.
- [4] J. Deng and Z. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," in *IEEE International Conference on Universal Personal Communications (ICUPC'98)*, pp. 973–977, October 1998.
- [5] S. R. D. N. Jain and A. Nasipuri, "A multichannel MAC protocol with receiver-based channel selection for multihop wireless networks," in *IEEE 9th International Conference on Computer Communications and Networks (IC3N'01)*, pp. 432–439, October 2001.
- [6] S. Wu, C. Lin, Y. Tseng, and J. Sheu, "A new multichannel MAC protocol with on-demand channel assignment for mobile ad hoc networks," in *International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 232–237, December 2000.
- [7] 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, March 2002.
- [8] S.-L. Wu and Y.-C. Tseng, "Intelligent medium access for mobile ad hoc networks with busy tones and power control," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1647–1657, 2000.
- [9] A. Nasipuri, J. Zhuang, and S. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *IEEE Wireless Communications and Networking Conference (WCNC'99)*, September 1999.
- [10] A. Nasipuri and S. Das, "Multichannel CSMA with signal power-based channel selection for multihop wireless networks," in *IEEE Fall Vehicular Technology Conference (VTC Fall 2000)*, pp. 211–218, September 2000.
- [11] 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.
- [12] 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.