MAC Scheduling Using Channel State Diversity for High-Throughput IEEE 802.11 Mesh Networks

Jian Zhang, Rutgers University
Yuanzhu Peter Chen, Memorial University of Newfoundland
Ivan Marsic, Rutgers University

ABSTRACT

The head-of-line blocking problem impairs the throughput of the *hotspot nodes* in multihop wireless mesh networks. FIFO scheduling in the current IEEE 802.11 MAC suffers from this problem when the network is highly loaded. The problem may keep the sender in backoff stage up to 70 percent of the time, leading to significant throughput degradation. One solution is to increase the RTS success rate by extending the RTS frame to multicast RTS so that multiple receivers can be checked simultaneously. We present an adaptive learning process that constructs the receiver list based on the dynamic channel-state diversity of candidate receivers. Our variable-length channel-state-based scheduling scheme outperforms the basic MRTS by 20–60 percent.

INTRODUCTION

Wireless mesh networking has seen great research interest recently. It is used to construct wireless community networks and in metropolitan area networks. Compared to single-hop access-point-based networks, multihop mesh networks provide extended coverage and greater flexibility for applications, since not all nodes are required to be directly within each other's transmission range. On the other hand, due to their decentralized self-organizing architecture, they present greater complexity relative to conventional single-hop wireless networks.

In wireless mesh networks some stations can be particularly overloaded. For example, a mesh network gateway needs to simultaneously deliver multiple up/downstream data flows between the Internet and many wireless stations; a mesh router may have to serve several neighbors by forwarding their packets along multihop paths. To maximize the performance of a mesh network, we should fully utilize the relaying capacity of such loaded stations. However, IEEE 802.11 [1], which is the dominant technology used in such networks, cannot achieve full relaying utilization in a highly loaded network. In particular, it is susceptible to the head-of-line (HOL) blocking problem. The HOL blocking problem occurs when the frame currently at the head of the queue in the sender's MAC layer cannot be transmitted successfully due to, say, the temporary unavailability of the receiver. In 802.11, each time a DATA or Request-to-Send (RTS) transmission times out, the sender doubles the contention window, to wait for a longer backoff time before retransmission, for the purpose of collision avoidance. The frame will not leave the queue until the transmission is acknowledged or until the maximal number of retries is reached. This frame has thus been blocking the subsequent frames from being transmitted although their receivers may be available at this time. Due to the exponentially-growing backoff time overhead, HOL blocking can lower greatly channel utilization and network capacity. Our simulation indicates that the fraction of backoff time at the sender's medium access control (MAC) layer may reach up to 70 percent. For a loaded mesh router or gateway, HOL blocking could result in a serious congestion problem. During the backoff process of a mesh gateway, more and more frames could arrive from a wired Internet connection and be blocked in the queue. With more frames arriving and the head frame blocking the queue, the router eventually gets overwhelmed, and the queue overflows and starts dropping packets. This may further trigger an upper layer (e.g., TCP) backoff, leading to further throughput degradation. Thus, in order to improve the performance of multihop mesh networks, the HOL blocking problem must be addressed.

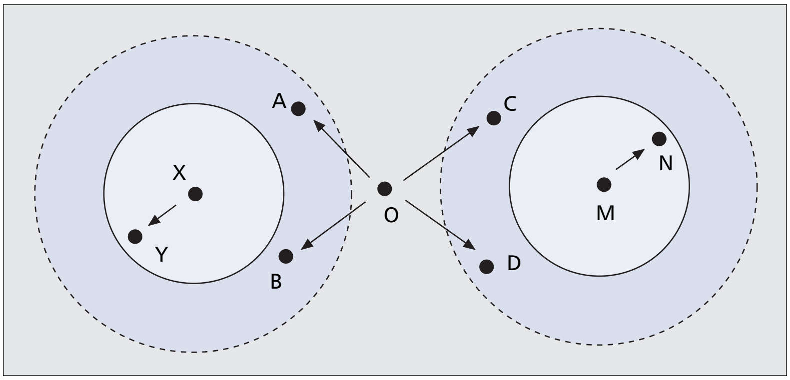
Most attempts at addressing the HOL blocking problem are based on a basic access scheme or unicast request to send (RTS) [2, 3]. Conversely, an innovative solution is to extend it to a multicast case (MRTS) [4, 5]. That is, the sender includes the addresses of multiple neighbors for whom it has data frames ready in the queue. By testing multiple neighbors for their availability, HOL blocking is alleviated. However, the extra

0163-6804/07/$20.00 © 2007 IEEE

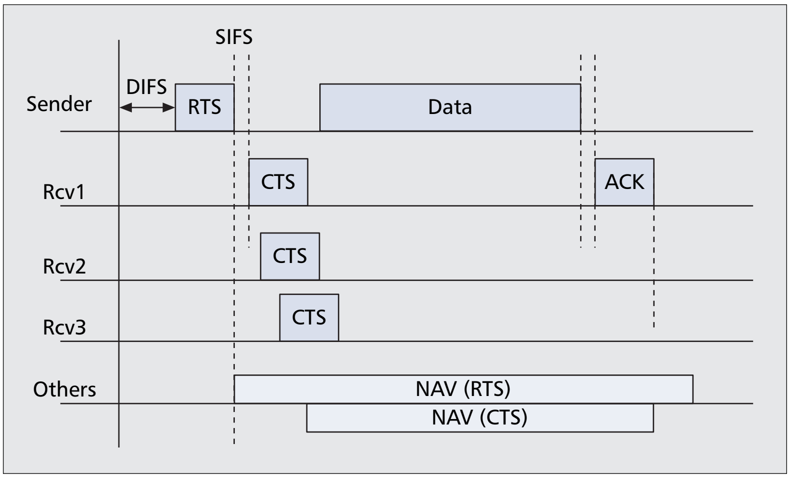
addresses contained in the MRTS add to the control overhead, particularly if a gateway serves a large number of mobile nodes within or beyond its immediate transmission range. A relatively small number of relay nodes can be used to serve both the nodes beyond the gateway's transmission range and some end nodes within range. However, relaying is not appropriate for all nodes within the gateway's range since it also introduces overhead. Hence, relays plus end nodes within range may add up to a considerable number of potential receivers. Thus, efforts are imperative to reduce such overhead while simultaneously maintaining its effectiveness.

We observe that geographically proximal stations are likely to share similar channel states. In this article the term *channel state* refers to MAC layer condition rather than physical channel condition. This means that a receiver's channel state is *good* if it is idle (no concurrent transmissions). If high correlation of channel states is observed for two candidate receivers, this implies low diversity between them; thus, it is unnecessary to include both of them in the MRTS list. For example, suppose that node *O* in Fig. 1 is delivering packets in four flows through its neighbors. Receivers *A* and *B* are both in the carrier-sensing range of station *X* (i.e., their channel states are synchronized to *X*'s behavior). When *X* is transmitting, both *A* and *B* are in "bad" state and are unable to reply *O*'s request. For sender *O*, the probability that *A* and *B* are both in good channel state is the same as for one of them to be in a good state; likewise for nodes *C* and *D*. Thus, we can achieve the same level of effectiveness as the MRTS that includes all nodes by using a shorter node list, and thus smaller overhead, by selecting the nodes with diverse channel state patterns in the list (e.g., {*A*, *C*} or {*B*, *D*}).

Here, we propose to schedule packets adaptively based on the correlation of channel states of receivers and to adjust adaptively the receiver list length in MRTS frames based on dynamic network conditions. Our scheme constructs a list of receivers with mutually diverse channel states, which minimizes the length of MRTS frames without losing effectiveness. The rest of the article is organized as follows. We provide an overview of the MRTS mechanism to motivate the need to intelligently construct the MRTS receiver list. We then present our adaptive channel-state-based scheduling. Simulation results are presented to demonstrate the benefits of the proposed protocol. We then conclude the article.



■ **Figure 1.** *Scenario 1.*



■ **Figure 2.** *MRTS protocol timeline.*

OVERVIEW OF MRTS

The IEEE 802.11 MAC layer specifies a carrier sense multiple access with collision avoidance (CSMA/CA)-based protocol, enhanced with an RTS/clear to send (CTS)/DATA/acknowledgment (ACK) handshake for virtual carrier sensing. The RTS/CTS dialog is used to reserve channel on both the sending and receiving sides. Originally, the RTS frame is addressed to a unique receiver. An MRTS, in contrast, is directed to a list of receivers. That is, an MRTS frame contains a list of next-hop receivers for which the sender has DATA packets currently queued. Each element of the list contains the receiver's address and the NAV of its corresponding packet. The priority among different receivers is decided by the order in which the receivers are arranged in the MRTS frame. That is, the earlier a receiver's address appears on the MRTS list, the sooner this receiver can return a CTS. The top candidate receiver that successfully receives MRTS replies with a CTS, unless it is blocked by an ongoing transmission in its neighborhood. If a lower-priority candidate detects that all higher-priority candidates remain silent for a defined period of time, it has the right to reply with a CTS (Fig. 2). Such a right to reply is implicitly propagated down the chain until a nonblocked receiver sends a CTS or all receivers remain silent and the sender times out. The sender finds the responding receiver's address from the received CTS frame. Then the sender retrieves the corresponding packet from its queue and transmits it to that receiver. The dialog ends with an ACK from the receiver if the transmission is successful.

Including more receivers in the MRTS list helps with the success ratio and thus the throughput. Consider the case where an MRTS fails (i.e., all of the receivers in its list remain silent). The likelihood of this is lower for longer lists of receivers. However, there is a cost associated with the MRTS extension. The longer control frame causes higher transmission overhead and increases the likelihood of collision. Thus, a more careful construction of the MRTS receiver list is needed

The margin of such a higher success rate will decrease and may even become negative when factoring in the overhead of having a long MAC header with a large number of neighbors included. Our goal is to use a variable-length, but short, list in the MRTS to achieve high network performance in general settings.

to reduce the overhead. In particular, if the receivers are chosen not randomly but based on the knowledge of their historical channel states, so they are likely to have diverse channel states, a short list can achieve the same effectiveness in HOL blocking avoidance as longer lists. The knowledge used in this process can be acquired adaptively from the receivers' historical responses to MRTSs. In addition, a mechanism of list length adjustment is required to find automatically the appropriate list lengths for various network- and channel conditions.

ADAPTIVE CHANNEL-STATE-BASED SCHEDULING FOR MRTS

Earlier work on MRTS includes a subset of a transmitting node's neighbors in the address list of the frame in order to increase the transmission success rate of the RTS. The determination of such a subset, however, is somewhat arbitrary. The effectiveness of the MRTS could be significantly improved if nodes make the decision with more judgment. Indeed, with the information of channel states and status of its neighboring receivers, more intelligent decisions can be made by the transmitting node. The multicast characteristic of MRTS measures the channel conditions of multiple receivers simultaneously. Based on the observed responses of MRTSs, the sender can estimate the neighbors' channel states and their correlations. Furthermore, such information can be used not only to affect the selection of the receivers, but also to adjust the length of the receiver list in the MRTS to adapt to the network conditions. It measures the degree of negative correlation between two receivers' channel conditions. The effectiveness of introducing such a notion is supported by a set of experiments that indicate an intelligent inclusion of two nodes provides higher throughput than randomly including two nodes. It can be speculated that when more neighbors are included in the MRTS, the likelihood that all of them are busy is lower. However, the margin of such a higher success rate will decrease and may even become negative when factoring in the overhead of having a long MAC header with a large number of neighbors included. Here, our goal is to use a variable-length but short list in the MRTS to achieve high network performance in general settings.

We first present how channel diversity information can be recorded using counters. Second, we show how to select a subset of neighbors for the MRTS using the above recorded information. Furthermore, we enable each node to make its own decision on how long this list should be, depending on the current network condition.

CHANNEL DIVERSITY ESTIMATION

Here, we assume that a node can include up to L neighbors in its MRTS. We use a 2D table to record the necessary information for later decision making. The outcome of an MRTS of length L is that only the rth neighbor replies with a CTS, where $1 \le r \le L$. That is, L neighbors of the sender were polled in the order specified by the list contained in the MRTS. The sender can tell that none of the first $r - 1$ neighbors in the list was able to reply (i.e., all are in bad channel state), while the rth neighbor is in a good state and is able to reply with a CTS. Note that we use the case $r = L + 1$ as notation for the special case when no node in the MRTS list replies. We denote by r the rank of the MRTS. For an MRTS of rank r, the relevant information is that the rth neighbor is in a different (i.e., better) channel condition than every node i $(1 \le i < r)$ and that nodes i and j $(i \neq j,\ 1 \le i < r$ and $1 \le j < r$ are in the same bad channel state.

Each node maintains a table with two counters S_{ij} and N_{ij}, for each pair of its neighbors i and j to record the number of occurrences of the above difference among the outcomes. S_{ij} denotes the numbers of occurrences in historical records where i and j are receivers included in an MRTS and i appears before j in the list, but only j, the latter, was able to reply. S_{ij} indicates how diverse i and j's channel states are. Similarly, N_{ij} counts occurrences when both i and j are included in an MRTS frame but neither was able to reply, which means that they are simultaneously in a bad state for N_{ij} times. These counters will be used to calculate a weight and are updated every time a new observation is made. When the total number of observations grows large, a new observation makes an insignificant difference in the estimated weight. Therefore, a sliding window is used to increase the agility of adaptation. The window keeps only M most recent observations for each pair of receivers, where M is the size of the window. The counters record only the observations in the window. The size of the window can be adjusted to match the factors affecting the channel state, such as average session lifetime and the movement pattern of the stations. The sliding window size is set to 20 in our simulations.

To maintain the diversity counters to reflect the channel diversity among the neighbors of a transmitting node, this node updates the entries of the table according to the observation that it has made based on the rank r of the latest MRTS frame $(1 \le r < L)$. Specifically, we increase the counter S_{ir} by one for every i $(1 \le i < r)$. In addition, we increase the counter N_{ij} by one for every i and j $(1 \le i < j < r)$.

ADAPTIVE SCHEDULING

To utilize the above history information, we define a value, called *diversity weight* W_{ij}, for each pair of a sender's neighbors i and j to represent how uncorrelated receivers i and j are in their channel states:

$$W_{ij} = W_{ji} = \frac{S_{ij} + S_{ji} + 1}{S_{ij} + N_{ij} + S_{ji} + N_{ji} + 1}$$

The sum of counters S_{ij} and S_{ji} represents how many times i's and j's historical channel states are different. The denominator is the size of the whole sample space including the counters of instances N_{ij} and N_{ji}, when both receivers' states may be influenced and synchronized by the same or similar traffic pattern. Thus, W_{ij} indicates normalized state diversity between i and j. We add one to both the numerator and denominator for initialization when the counters are zero.

When a node has a packet to send, it constructs an MRTS of length L. The neighbors to be included in the MRTS list are selected as follows. It first selects a neighbor, uniformly at random, denoted by N_1, for which it has packets in the queue. To include the second neighbor, it calculates the diversity weight between node N_1 and every other neighbor for which it has packets queued. It then selects a node among these neighbors with likelihood proportional to the diversity weights calculated. That is, the neighbor with a packet available and with highest diversity weight relative to node N_1 has the highest probability to be selected. Denote it N_2. To include the third neighbor, it calculates the *combined diversity weight* for each neighbor j for which it has packets available, relative to nodes N_1 and N_2. The combined diversity weight is defined as $W_{jN_1} \times W_{jN_2}$. Again, it then selects a node among these neighbors with likelihood proportional to this combined weight. The likelihood of a neighbor appearing early on in the list is proportional to its contribution to the diversity measure. We denote such a selected node N_3. Generally, to include the ith neighbor in the MRTS ($2 < i \leq L$), it calculates the combined diversity weight for each node j for which it has packets available, relative to the $i - 1$ neighbors already included in the list. This weight is defined as

$$\prod_{m=1}^{i-1} W_{jN_m}.$$

It then selects the ith neighbor to be included in the MRTS randomly with likelihood proportional to the weight. Thus, the node completes the construction of the list of neighbors to be included in the MRTS.

Note that the diversity weighting described above attempts to maximize the diversity of channel states of the receivers included in the list. In addition, the randomization ensures that no combination of receivers is completely excluded even if a combination is highly correlated at the moment. This is important for future updates when channel states are changed. Furthermore, such randomization promotes fairness by avoiding starvation of any flow.

The randomization ensures that no combination of receivers is completely excluded even if a combination is highly correlated at the moment. This is important for future updates when channel states are changed. Further, such randomization promotes fairness by avoiding starvation of any flow.

DETERMINATION OF LIST LENGTH

In order for the nodes to adapt to the channel conditions, we further enable each node to decide on how many nodes to include in the MRTS depending on the updated network conditions. To do that, each node maintains a running average of the ranks of the MRTS frames it has attempted, denoted $\bar{r}$. When the rank r of a new MRTS is recorded, $\bar{r}$ is updated with $0.875\,\bar{r} + 0.125\,r$. Using this average rank, the MRTS frame a node constructs always has a length $l = round(\bar{r}) + 1$. Such length adjustment can be seen as a feedback control mechanism. That is, when the average rank of the MRTS outcomes becomes large as the network condition changes, the list will be extended to maintain a high level of diversity. The small constant (i.e., 1 in the above definition) added to the average rank enables the adaptive list growth. If the network condition changes in the opposite direction so that some receivers in the current list do not contribute to the diversity value, those receivers will be scheduled later than the others since our channel-state-based scheduling puts ahead the receivers with the highest combined diversity weight. Therefore, the average rank will decrease. By observing such decreases, the mechanism shortens the MRTS list automatically.

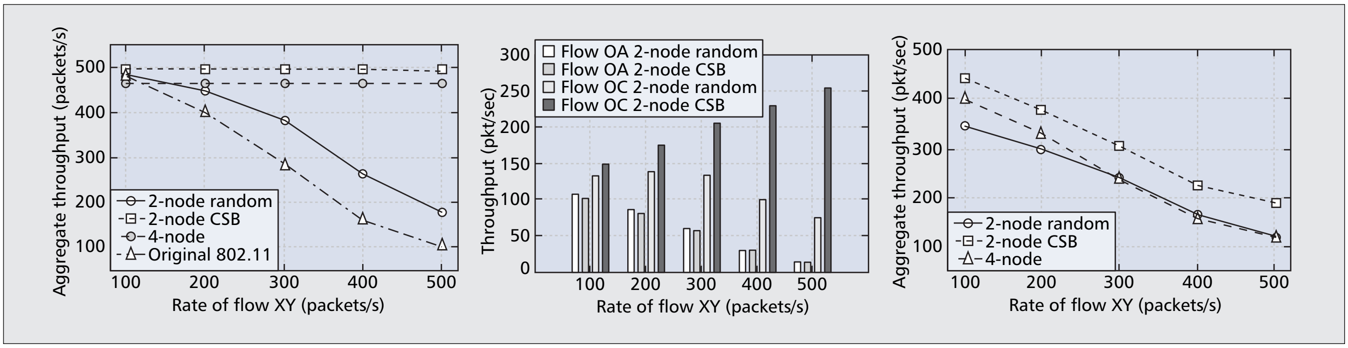
SIMULATION RESULTS

We test and compare our fixed-length and variable-length channel-state-based schemes against the random scheduling schemes in the basic MRTS using ns-2 with default PHY layer settings. Two scenarios are designed below for this purpose. We show in scenario 1 that the channel-state-based (CSB) scheme with a short fixed-length MRTS receiver list can outperform the random scheme with an equal or even longer list. In scenario 2, we show that an appropriate list length is adjusted adaptively to the dynamic traffic situations in the network. The variable-length CSB (VL-CSB) scheme achieves the highest throughput compared to the fixed-length random and *include-all* schemes.

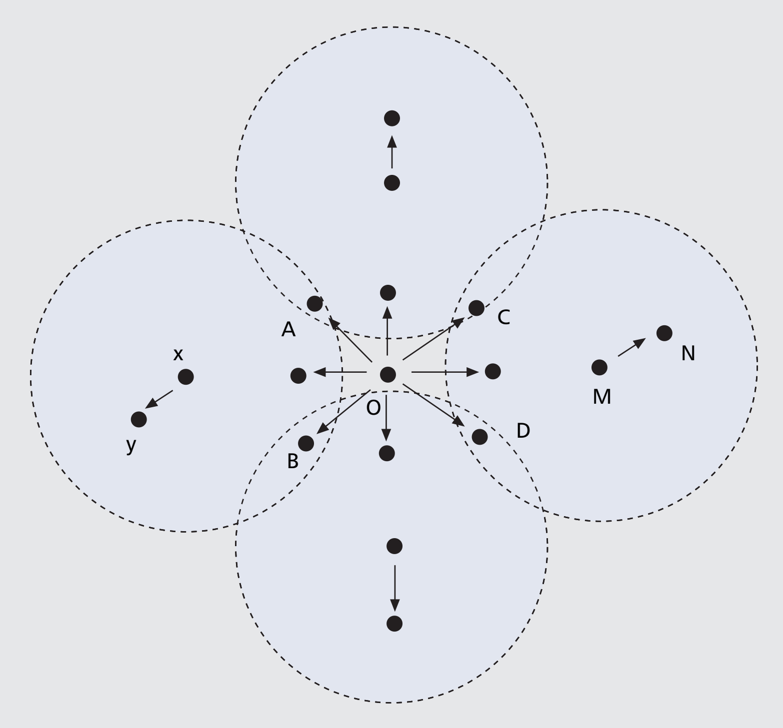
SCENARIO 1

Figure 1 shows a scenario with nine nodes and six UDP flows (OA, OB, OC, OD, XY, and MN) with fixed packet size of 500 bytes. We evaluate the performance of CSB scheduling on node O under two other interfering flows XY and MN. The rates of flows OA, OB, OC, and OD are all set to 650 packets/s, so node O always has packets in the queue for each flow. We set the rate of flow MN to 0 at first. Then we vary the sending rates of flow XY from 100 to 500 packets/s. For each rate, we perform 50 s of simulation with the two-node random scheme, four-node random scheme, and 2-node CSB scheme, respectively. We also show the performance of the 802.11 regular unicast RTS scheme as baseline. Here, nodes X and O are hidden from each other. The transmission by X can interfere with reception and cause collisions on nodes A and B. Conversely, node O is outside the interference range of receiver Y. Due to this asymmetry, flows OA and OB are influenced by flow XY, but not vice versa. Thus, in Fig. 3 we only show the throughputs of OA, OB, OC, and OD, and their sum (i.e., the aggregate throughput of node O).

In Fig. 3a the two-node CSB scheme outperforms both the two- and four-node random schemes for all rates of flow XY. The aggregate throughput of the two-node random scheme declines as the rate of flow XY grows, while the throughputs of the CSB and four-node random schemes remain constant. In this scenario node O's receivers are grouped into two subsets $\{A, B\}$ and $\{C, D\}$. Nodes A and B are within the carrier-sensing range of node X, and their channel state is the same most of the time. For the two-node random scheme, there is a 1/3 chance that two nodes from the same subset are selected, which leads to low channel state diversity in MRTS frames. In contrast, the two-node CSB and four-node schemes achieve higher MRTS success rates and thus lower backoff overhead by maintaining high channel state diversity in the receiver list. The two-node CSB and four-node



■ **Figure 3.** *a) Aggregate throughput when rate of flow MN is 0; b) individual throughputs; and c) aggregate throughput when rate of flow MN is 300.*



■ **Figure 4.** *Scenario 2.*

random schemes achieve the same level of diversity because the success rate of an MRTS with two nodes, each selected from different subsets, is the same as that of a four-node MRTS. The constant gap between the two-node CSB and four-node schemes in Fig 3a can be attributed to the higher overhead in transmission time of four-node MRTSs. Figure 3b shows the throughputs of individual flows *OA* and *OC* with the two-node random and CSB schemes. With the latter, the throughput of flow *OA* remains almost the same as with two-node random, but the throughput of flow *OC* is greatly improved. This confirms that avoiding selecting both *A* and *B* in the MRTS list reduces backoff and retransmission overhead and alleviates the HOL blocking problem. It indicates that the local network capacity can be significantly increased by appropriate construction of the MRTS list.

We then set the rate of another interfering flow *MN* at 300 packets/s to overload the network and repeat the previous tests (Fig. 3c). The network capacity now is inadequate to sustain all six flows, and the aggregate throughput on node *O* decreases for all schemes as the rate of flow *XY* increases. The aggregate throughput of node *O* with the CSB scheme is consistently higher than for the other two schemes. In high load cases, the throughput of the four-node scheme drops drastically as the overhead of a high collision rate due to longer MRTS frames becomes dominant.

SCENARIO 2

In this scenario (Fig. 4) we add four neighbor nodes of *O* and two more interfering flows. We start the four interfering flows at different times of the simulation, emulating a more realistic dynamic network environment. The rate of each interfering flow is fixed at 400 packets/s. The flow on the left is started first, followed by the other three in counterclockwise order, one in each stage of 50 s. So for stage *n*, there are *n* interfering flows running. In such a complex and dynamic scenario, fixing the length of the MRTS list for different situations permanently is apparently not a good solution. When few interfering flows are present, a long list may cause redundancy and lead to high overhead as the results of scenario 1 show. Conversely, a short list is insufficient to leverage the channel state diversity of the list as more interfering flows are started.

The experimental results (Fig. 5) show that by combining the list length adaptation with CSB scheduling, our VL-CSB scheme achieves the best throughput performance in all stages, compared to the fixed-length (four-node) random scheme and eight-node include-all scheme. The latter always includes all the active next-hop receivers of the sender and is used in the comparison to illustrate the inappropriateness of using excessively long MRTS lists. Figure 5a shows how the average rank, observed by the sender at the network center, grows as the number of interfering flows increases at every stage in the VL-CSB result. Every time an additional interfering flow starts, the acceptance rate of the MRTS drops and the rank increases. Accordingly, the VL-CSB grows the receiver list to maximize the MRTS acceptance probability. Without such a length adjustment mechanism, in Fig. 5b the include-all scheme's performance suffers in the first two stages when there is low diversity of

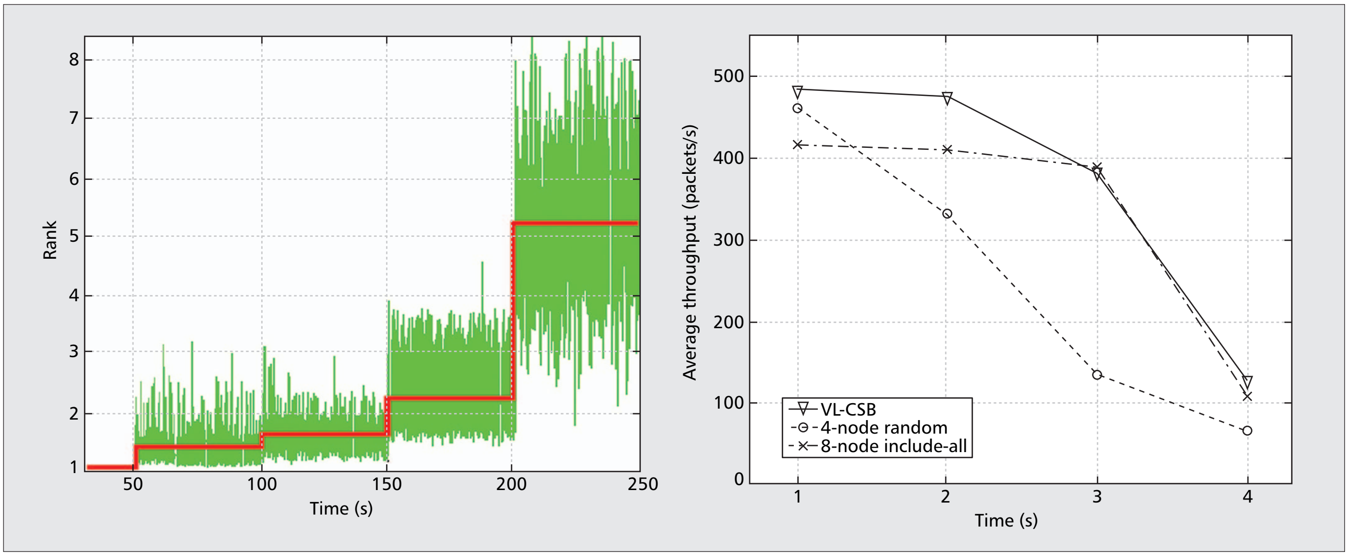


Figure 5. *a) Average rank; and b) aggregate throughput.*

channel states in the neighborhood. This is due to the overhead of transmitting its excessively long MRTS frames. In the last two stages its advantage of high diversity due to the long list overcomes the transmission overhead, and its performance approaches VL-CSB's. These results show that the length adjustment mechanism effectively lowers the overhead of MRTS. In addition, the VL-CSB outperforms the four-node random scheme significantly in stages 2 and 3. This means that random scheduling cannot achieve the same degree of channel state diversity in the list as our CSB scheduling or the include-all scheme. The four-node random scheme lowers the MRTS success ratio. In contrast, the CSB scheme makes it possible to use shorter lists to achieve higher diversity. According to Fig. 5a, the length of MRTS chosen by VL-CSB is less than 4 (i.e., mostly 2 or 3) in stages 2 and 3. The performance boost by VL-CSB against the four-node random scheme (Fig. 5b) suggests higher diversity in the lists constructed by CSB.

CONCLUSION

The IEEE 802.11 was designed to implement wireless LANs and is currently the most dominant technology to implement wireless mesh networks. It naturally has shortcomings in this new application, such as the HOL blocking problem at the MAC layer, for which we propose a promising solution, an MRTS extension to 802.11. To maximize the benefit of MRTS, we present an adaptive channel-state-based scheduling scheme that controls both the contents and length of the receiver list in the MRTS frame.

Our solution, dedicated to single-channel environments, can cooperate with other multiradio/multichannel MAC protocols to further enhance the capacity of WMNs. MRTS with its extension can mitigate the blocking time while operating on each individual channel. Moreover, as shown above, the spatial and channel proximity of receivers can be estimated by the sender through their historical responses to MRTS. Such information may be helpful for improving efficiency of a channel assignment scheme by assigning nonoverlapping channels to proximal stations to reduce mutual interference. This feature of MRTS combined with multichannel mesh networks represent our future work.

REFERENCES

[1] IEEE 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1997.

[2] P. Bhagwat *et al.*, "Enhancing Throughput over Wireless LANs using Channel State Dependent Packet Scheduling," *Proc. INFOCOM*, 1996.

[3] C. Fragouli, V. Sivaraman, and M. B. Srivastava, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State Dependent Packet Scheduling," *Proc. INFOCOM*, 1998.

[4] S. Jain and S. R. Das, "Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks," *Proc. 6th IEEE WoWMoM Symp.*, 2005.

[5] J. Wang, H. Zhai, and Y. Fang, "Opportunistic Packet Scheduling and Media Access Control for Wireless LANs and Multi-Hop Ad Hoc Networks," *Proc. WCNC*, 2004.

BIOGRAPHIES

JIAN ZHANG (jianz@caip.rutgers.edu) is a Ph.D. student in the Department of Electrical and Computer Engineering at Rutgers University, Piscataway, New Jersey. He received his B.E. in 1994 and M.S. in 1997 from Huazhong University of Science and Technology, China. His research interests include MAC and routing protocols and QoS in wireless mesh/ad hoc networks.

YUANZHU PETER CHEN is an assistant professor in the Department of Computer Science at Memorial University of Newfoundland, St. John's, Newfoundland. He received his Ph.D. from Simon Fraser University in 2004 and B.Sc. from Peking University in 1999. Between 2004 and 2005 he was a post-doctoral researcher at Simon Fraser University. His research interests include mobile ad hoc networking, wireless sensor networking, distributed computing, combinatorial optimization, and graph theory.

IVAN MARSIC received Dipl.Ing. and M.S. degrees in electrical and computer engineering from the University of Zagreb, Croatia, and a Ph.D. degree in biomedical engineering from Rutgers University, New Brunswick, New Jersey, in 1994. Currently he is an associate professor of electrical and computer engineering at Rutgers University. His research interests include software engineering, computer networks, distributed computing, and human-computer interfaces. His research focuses on engineering complex collaborative systems that include people and computers.