Adaptive MAC Scheduling Using Channel State Diversity for Wireless Networks

Jian Zhang
ECE Department
Rutgers Univeristy
jianz@caip.rutgers.edu

Yuanzhu Peter Chen
Department of Computer Science
Memorial University of Newfoundland
yzchen@cs.mun.ca

Ivan Marsic
ECE Department
Rutgers Univeristy
marsic@caip.rutgers.edu

Abstract

Head-of-line blocking problem compromises the throughput of multi-hop wireless networks. FIFO scheduling in the current IEEE802.11 MAC causes this problem when the network is highly loaded. One solution is to increase the RTS success rate by extending the RTS frame to MRTS (multicast RTS), so that multiple receivers could be checked simultaneously. There is a tradeoff for the length of the MRTS-frame receiver list, since longer lists increase transmission success rate, but shorter lists impose lower control overhead. We present an adaptive learning process that observes the dynamic channel-state diversity among the candidate receivers. By maximizing the receiver diversity, we can achieve high transmission success rate using short receiver lists. This is supported by our simulation results.

1. Introduction

Multi-hop wireless networking has seen great research interest recently. It provides greater flexibility in applications, since in such networks not all nodes are required to be directly within each other's transmission range. On the other hand, they present greater complexity relative to conventional single-hop wireless networks, due to the more complex medium access control, a need for efficient and robust message forwarding, a need for smart congestion control in the presence of unreliable links, etc. Mobile ad hoc networks and mesh networks are examples of such a networking paradigm.

To maximize the throughput of a multi-hop wireless network, we should fully utilize the nodes' relaying capacity. However, IEEE 802.11 [3], which is the dominant technology used in such networks, cannot achieve full utilization of relaying in a highly loaded network. In particular, it is susceptible to the head-of-the-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 unavailability of the receiver. This frame is then blocking the subsequent frames from being transmitted although their receivers may be available at this time. This happens most often around- or at gateways in a mesh network. Thus, in order to improve the performance of multi-hop wireless networks, the HOL blocking problem must be addressed.

A number of methods have been proposed to solve the HOL blocking problem. Bhagwat *et al.* [1] propose channel-state-dependent packet scheduling (CSDPS) to exploit such channel-state diversity, observing that different spatially-distributed receivers may at once have different wireless channel states. CSDPS defers the transmission and retransmission of a lost packet when it suspects a bad link state. Fragouli *et al.* [2] enhance CSDPS by monitoring the history of RTS/CTS attempts, and use it to limit the maximum number of RTS retries, so that the sender does not persist with a receiver is in a bad channel state. Some recent work [6][4] addressing the HOL blocking problem proposed an extension to the medium access mechanism of 802.11, called multicast RTS (MRTS). In [6][4], MRTS frames are transmitted with a list of receiver addresses. Due to channel-state diversity among these receivers, the likelihood is low that none of the MRTS receivers would reply. This improves the throughput. However, the MRTS frame becomes larger to carry multiple receiver addresses. This can incur longer RTS transmission time and higher chance of collision, especially when the network is highly loaded.

Here we propose to schedule packets adaptively based on receiver's channel-state diversity with short receiver lists in MRTS frames. Our scheme groups the receivers with similar channel states and constructs a short list of receivers with mutually diverse channel states, which minimizes the overhead of larger MRTS frames without reducing the effectiveness. The rest of the paper is organized as follows. Section 2 provides an overview of the MRTS mechanism and a brief analysis of its overhead. We then present our adaptive channel-state-based scheduling in Section 3. The simulations are presented in Section 4. Section 5 concludes the paper.

2. Overview of the MRTS protocol

The IEEE 802.11 MAC layer specifies a CSMA/CA-based protocol, enhanced with an RTS/CTS handshake for virtual carrier sensing. An MRTS, in contrast to a unicast RTS in conventional RTS/CTS, 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 queued. Each element of the list contains the receiver's address and the duration or the network allocation vector (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 sooner a receiver's address appears on the MRTS list, the
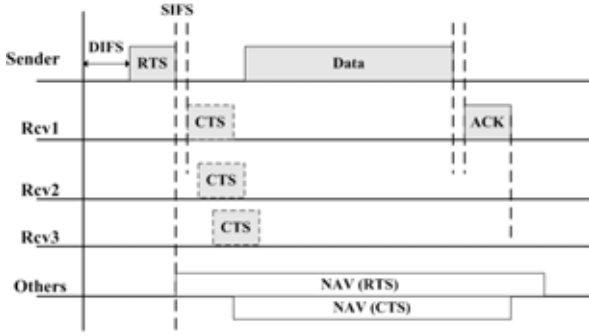
Fig. 1 MRTS protocol timeline

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 remained silent, it has a right to reply with a CTS (Fig. 1). Such a right-to-reply is implicitly propagated down the chain until a non-blocked receiver sends a CTS or all receivers remain silent and the sender times out. The sender finds the correct 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.

Consider the option that 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, longer lists imply greater overhead. For example, suppose that senders use a 4-node list, as in [6]. For each receiver in the list, 8 bytes are added to the RTS frame (6 for the address and 2 for the duration). A 4-node list adds 24 bytes (or 192 bits), i.e., an additional $192\mu s$ to the RTS transmission time in 802.11b, where RTS/CTS is transmitted at the basic rate of 1Mbps. This added overhead grows if the DATA frame is transmitted with higher-rate modulation, e.g., the duration of an RTS/CTS/DATA/ACK cycle with a 4-node list MRTS is 13.8% longer than with legacy 802.11 for 500-byte data frames at a rate of 11Mbps. In addition, the longer an RTS is, the higher the chance of collision. To avoid collisions, an exponential backoff mechanism is used in 802.11. A station ready to transmit RTS has to wait for a DIFS and a random amount of time between 0 and the contention window time, $T_{CW} = (CW) \times SlotTime$. CW is set to 31 for the first attempt and is approximately doubled for each unsuccessful subsequent attempt. Hence, if two stations attempt to access medium, the probability of no RTS collision is

$$P = \Pr\left(|X - Y| > T_{RTS}\right)$$
$$= 2 \cdot \int_{y=T_{RTS}}^{T_{CW}} \int_{x=0}^{y-T_{RTS}} \frac{1}{T_{CW}^2} dxdy = \left(1 - \frac{T_{RTS}}{T_{CW}}\right)^2 \quad (1)$$

where X and Y are the random variables between $[0, T_{CW}]$ picked by two stations respectively and T_{RTS} is the transmission time of RTS. For the first attempt, the probabilities of no collision using unicast RTS vs. a 4-node MRTS are 0.187 and 0.015, respectively. These probabilities drop sharply for longer MRTS lists.

A more thoughtful construction of the MRTS receiver list is needed to reduce the overhead. In particular, if the receivers are chosen so that they are likely to have diverse channel states, then a short list can achieve the same effectiveness in HOL blocking avoidance as longer lists. The knowledge used for such a construction can be acquired adaptively from previous transmissions.

3. Adaptive channel-state-based scheduling for MRTS

We design the adaptive channel-state-based scheduling by observing that geographically-proximal stations are likely to share similar channel states. If high correlation of channel states is observed for two candidate receivers, this implies low diversity, and thus it is unnecessary to include both of them in the MRTS list. For example, suppose that two receivers A and B of sender O are both in the carrier-sensing range of station C, i.e., their channel states are synchronized to the C's behavior. For sender O, the probability that A and B are both in good channel state is the same as that of one of them being in a good state. Thus, we can shorten the MRTS receiver list without mitigating the diversity of MRTS. Our method consists of two components, an estimator of channel-state diversity and the scheduler that uses such information.

The multicast characteristic of MRTS measures the channel conditions of multiple receivers simultaneously. For an MRTS with a 2-nodes list, its outcome falls into three categories: (*i*) the first-in-the-list receiver replies with CTS; (*ii*) the second-in-the-list replies; or, (*iii*) none of them replies. The first case provides incomplete information. The second case tells the sender that at the moment the two receivers are in different channel states, i.e., high diversity. The last case causes an RTS timeout which means that both of receivers are in bad state, i.e., high correlation. We define a parameter called diversity weight by the following formula to represent how uncorrelated receivers i and j are in their channel states:

$$W_{ij} = W_{ji} = \frac{S_{ij} + S_{ji} + 1}{S_{ij} + N_{ij} + S_{ji} + N_{ji} + 1} \quad (2)$$

In Eq. (2), S_{ij} and S_{ji} denote the numbers of occurrences in historical records where both i and j are receivers listed in the given order, but only the second receiver in the list replies. N_{ij} and N_{ji} denote the numbers of timeout occurrences, when none of the receivers replied. We add one to both the numerator and denominator for initialization. The diversity weights are updated every time when a new observation is made. In Eq (2), when the total number of observations grows large, a new observation makes no 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 weights are calculated by counting 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 of the stations. The sliding window size is set to 20 in our simulations.

With estimated diversity weights for each pair of receivers, we make a 2-node list and schedule the packets by as follows. At the sender, the packets are classified by

the addresses of their next-hop receivers, so that different queues correspond to different next-hop receivers. First, the scheduler randomly picks a non-empty queue. The head packet in this queue is scheduled and its receiver is put first in the MRTS list. Second, the scheduler compares the diversity weights of the remaining candidates against the first selected one. For each pair, it generates a random variable between 0 and their corresponding diversity weight. The receiver with the highest random value is scheduled as the second receiver. An MRTS receiver list is made for each packet transmission. The diversity weighting in our method maximizes the diversity of channel states of the two receivers on the list. In addition, the randomization ensures that no combination of receiver pairs is completely excluded even if a pair is highly correlated at that moment. This is important for future updates when channel states are changed. Further, such randomization promotes fairness.

With channel-state-based scheduling, the same level of diversity of receiver channel-states can be preserved while the length of the MRTS frames is minimized.

4. Simulation results

We test and compare our channel-state-based scheme with the random scheduling schemes using ns-2. In the first scenario, we show that the channel-state-based scheme outperforms the random schemes with list lengths of 2 and 4, in terms of throughput. The second scenario demonstrates that our scheme adapts scheduling to changes in channel diversity due to mobility.

### 4.1. Scenario 1: 2D topology with stationary stations

Fig. 2 shows a scenario with 9 nodes and 6 UDP flows with fixed packet size of 500 bytes. We evaluate the performance of channel-state-based scheduling on node O under the influence of two other interfering flows XY and MN. The rates of flows OA, OB, OC and OD are all set at
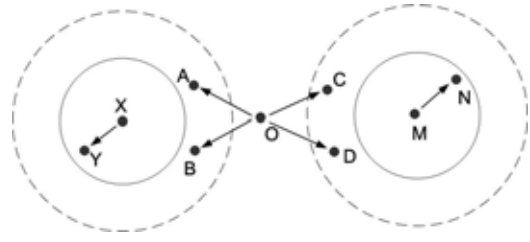

Fig. 2 Scenario 1

650 pkts/sec so that 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 pkts/sec. For each rate, we perform 50-second simulations with 2-node list random scheme, 4-node list random scheme and the channel-state-based scheme, respectively. We also show the performance of 802.11 regular unicast RTS scheme for reference. Here, nodes X and O are hidden from each other. The transmission by X can interfere with the receptions and cause collisions on nodes A and B. On the other hand, 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(a) we only show the sum of throughputs OA, OB, OC and OD, i.e., the aggregate throughput of node O.

In Fig. 3(a), the channel-state-based scheme outperforms both the 2-node and 4-node random schemes for all rates of flow XY. The aggregate throughput of the 2-node random scheme declines as the rate of flow XY grows, while the ones of channel-state-based scheme and 4-node random scheme remain constant. In this scenario, the receivers of O are grouped into two subsets $\{A, B\}$ and $\{C, D\}$. Nodes A and B are within the carrier-sensing range of node X, and they have the same channel state most of the time. For 2-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
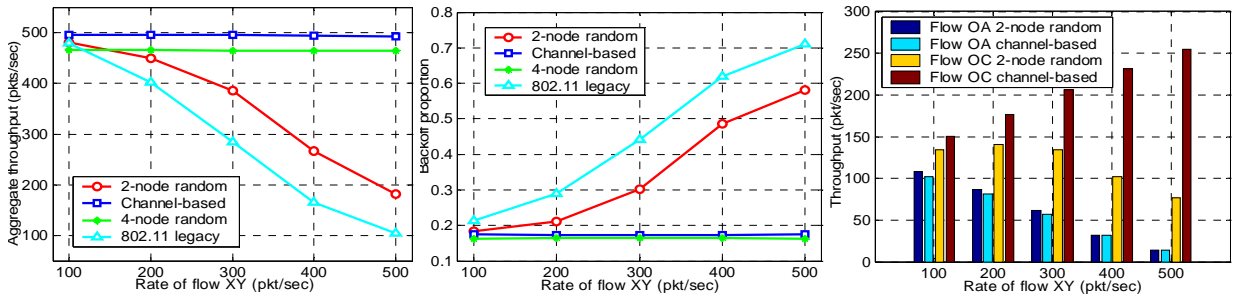
Fig. 3 (a) aggregated throughput; (b) backoff time proportion; (c) Individual throughputs; Rate of MN is 0
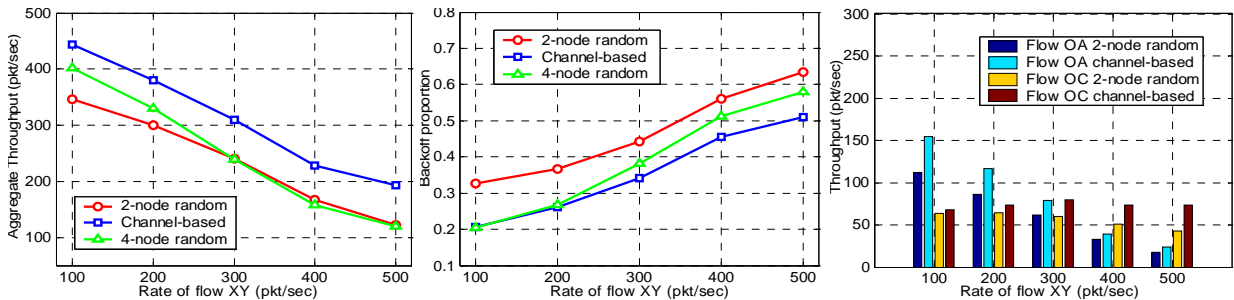
Fig. 4 (a) aggregated throughput; (b) backoff time proportion; (c) Individual throughputs; Rate of MN is 300

MRTS frames. In contrast, the channel-state-based scheme and the 4-node scheme achieve higher MRTS success rate and thus less backoff overhead by maintaining the high channel-state diversity in the receiver list. Fig. 3(b) shows that the backoff time fractions on *O* for channel-state-based scheme and 4-node scheme are much lower than for the 2-node random scheme, especially when the rate of *XY* flow is high.

Both channel-state-based scheme and the 4-node random scheme 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 4-node MRTS. The constant gap between the channel-state-based and 4-node schemes in Fig 3(a) and (b) can be attributed to the extra overhead in transmission time of 4-node MRTS's. Fig. 3(c) shows the throughputs of individual flows *OA* and *OC* with the 2-node random scheme and the channel-state-based scheme. With channel-state-based scheme, the throughput of flow *OA* is slightly lower but the throughput of flow *OC* is greatly improved. This confirms that avoiding selecting both *A* and *B* in MRTS list reduces backoff- and retransmission overhead and alleviates the HOL-blocking problem, especially when the rate of flow *XY* is high. Fig. 3 indicates that the local network capacity can be significantly increased by an appropriate construction of the MRTS list.

We then set the rate of another interfering flow *MN* at 300 pkts/sec and repeat the previous tests. The results are shown in Fig. 4. The network capacity now is not enough to sustain all 6 flows and the aggregate throughput on node *O* slips for all schemes when rate of flow *XY* increases. The aggregate throughput of node *O* with channel-state-based scheme is consistently higher than for the other two schemes. In high load cases, the throughput of 4-node scheme drops drastically as the overhead of high collision rate due to longer MRTS frames becomes dominant. Also, notice that the throughputs of all individual flows in Fig. 4(c) are increased with the channel-state-based scheme. It means that, in such overloaded scenario, every flow benefits from the backoff overhead cut achieved by the channel-state-based scheme.

4.2. Scenario 2: Chain topology with moving stations

In wireless networks, the local channel conditions can be changed by movement of nodes. In Fig. 5, as node *Y* moves toward *O*, nodes *A*, *B* and *C* fall into *Y*'s interference zone, in order. In this scenario, we test the adaptability of our channel-state-based scheme.

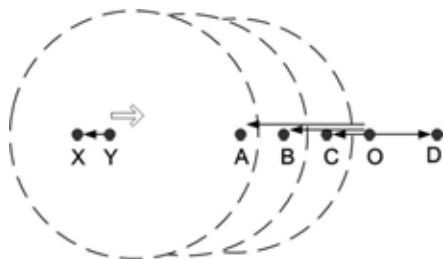The results are shown in Fig. 6. In stage 1, where only node *A* is under the interference of node *Y*, the throughput
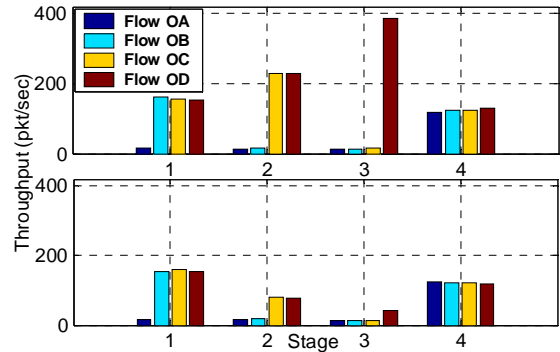

Fig. 6 (a) Channel state based; (b) 2-node random

of *OA* is low and receivers *B*, *C* and *D* share most of the local bandwidth. In stage 2, where *A* and *B* are in the interference zone, the HOL-blocking problem lowers the throughput of flows *OC* and *OD* for the 2-node random scheme. The channel-state-based scheme detects the increasing correlation of *A*'s and *B*'s states, and the probability of scheduling them in the same MRTS is reduced accordingly. Thus, flows *OC* and *OD* obtain greater fraction of serving time on *O*. Likewise, in stage 3, where *A*, *B* and *C* are under interference, frames of flow *OD* are more frequently scheduled by the channel-state-based scheme. This again alleviates the HOL-blocking problem. The throughput of flow *OD* (Fig. 6(a)) is 3 times higher in stage 2 and 7 times higher in stage 3 than for the random scheme (Fig. 6(b)). Lastly, as we stop flow *YX* in stage 4, both schemes assign the resource equally for all four receivers (*A*, *B*, *C* and *D*). The results show that the channel-state-based scheme is responsive to channel state changes and more efficient than the random scheme.

5. Conclusion

In this paper we present an adaptive, channel-state-based scheduling scheme for MRTS protocol in 802.11-based networks. It alleviates the HOL problem and avoids the overhead of long MRTS frames. Simulation results show that the performance of MRTS is significantly improved with channel-state-based scheduling.

6. References

[1] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Enhancing throughput over wireless LANs using channel state dependent packet scheduling", *Proc. of INFOCOM*, 1996.

[2] 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. of INFOCOM*, 1998.

[3] IEEE Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11, 1997

[4] S. Jain and S. R. Das, "Exploting path diversity in the link layer in wireless ad hoc networks", *Proc. of the 6th IEEE WoWMoM Symposium*, 2005.

[5] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia, "Exploting medium access diversity in rate adaptive wireless LANs", *Proc. of ACM MobiCom*, 2004.

[6] J. Wang, H. Zhai, and Y. Fang, "Opportunistic packet scheduling and media access control for wireless LANs and multi-hop ad hoc networks", *Proc. of WCNC*, 2004.

Fig. 5 Scenario 2