SNDR: A New Medium Access Control for Multi-channel Ad Hoc Networks

Zhijun Cai, Mi Lu
Department of Electrical Engineering
Texas A&M University
College Station, TX 77843
email: {mlu, czjhh}@ee.tamu.edu

Abstract - A new multi-channel, CDMA (Code Division Multiple Access) (or FDMA (Frequency Division Multiple Access)) and TDMA (Time Division Multiple Access) combined, contention free MAC (Medium Access Control), termed the Sequenced Neighbor Double Reservation (SNDR), is presented for mobile ad hoc networks. The SNDR uses the receiver-based data transmission strategy, based on which two methods are proposed. One is contention-based and the other is contention-free. In this paper, we put emphasis on the contention-free type (SNDR). On the other hand, the contention-based MAC, which needs further research, is also briefly discussed. The SNDR does not need any handshake process (such as Request to Send/Clear to Send (RTS/CTS) handshake) or any carrier sensing technology. It uses the neighbor sequenced method to avoid contentions and the double reservation method to improve the total throughput of ad hoc networks. No hidden or exposed terminal problem will exist in the SNDR. No collision will occur and no time slot will be wasted in the SNDR MAC frame. The protocol can be efficiently applied to the multichannel ad hoc networks. The performance of the SNDR is analyzed carefully. Some future work and applications are also discussed.

I Introduction and Backgrounds

Much work have been done on the MAC layer of ad hoc networks lately [1-15]. In MACA [1], Karn originally proposed the use of short control packets, the Request-To-Send (RTS) and Clear-To-Send (CTS) packets, for collision avoidance on the shared channel. Bharghava [2] suggested the use of RTS-CTS-DS-DATA-ACK message exchange for a data packet transmission in the MACAW protocol. Talucci and Gerla [3] introduced MACA-BI protocol, which is a simplified version of the well known MACA based on the RTS/CTS handshake. Zhu and Corson [4] proposed Five-Phase Reservation Protocol (FPRP) which is a single channel, TDMA based broadcast scheduling protocol. The FPRP is free of the “hidden terminal” problem and is designed so that reservations can be made quickly and efficiently with negligible probability of conflict. Garcia-Luna-Aceves and Fullmer [5] analyzed the performance of the FAMA-NCS (Floor Acquisition Multiple Access with Non-persistent Carrier Sensing) protocol which guarantees that a single sender is able to send data packets free of collisions to a given receiver at any given time. All these protocols are for single channel ad hoc networks. Despite the work done on the single channel MAC, not much research have been focused on the multi-channel ad hoc networks.

In this paper, a new multi-channel, CDMA (Code Division Multiple Access) (or FDMA (Frequency Division Multiple Access)) and TDMA (Time Division Multiple Access) combined, contention free MAC (Medium Access Control), termed the Sequenced Neighbor Double Reservation (SNDR), is presented for mobile ad hoc networks. The SNDR uses the receiver-based data transmission strategy, based on which two methods are proposed. One is contention-based and the other is contention-free. In this paper, we put emphasis on the contention-free type (SNDR). On the other hand, the contention-based MAC, which needs further research, is also briefly discussed. The SNDR does not need any handshake process (such as Request to Send/Clear to Send (RTS/CTS) handshake) or any carrier sensing technology. It uses the neighbor sequenced method to avoid contentions and the double reservation method to improve the total throughput of ad hoc networks. No hidden or exposed terminal problem will exist in the SNDR. No collision will occur and no time slot will be wasted in the SNDR MAC frame. The protocol can be efficiently applied to the multichannel ad hoc networks.

This paper is organized as follows. Section 2 describes the SNDR in detail. Section 3 gives the performance analysis. Section 4 gives conclusions and future directions of this research.

II Descriptions of the SNDR

As we know, for an ad hoc network, Mobile Nodes (MNs) can randomly move at any given time. If there are \( M \) MNs, \( N \) channels in an ad hoc network, how to design a MAC to maximize the total throughput of the ad hoc networks is a key problem. This is the goal of paper.
2.1 Assumptions

Due to the dynamic network topology, for a given node A, its neighbor set N(A) (The neighbor set of a given node means the set which includes all the neighbors of that node) is also dynamic. Neighbor sets should be updated after they are changed. This is the neighbor-finding problem in ad hoc networks which we will not focus on at this moment. We assume that a neighbor set can be correctly updated at any given time and it will keep unchanged during the transmission of one MAC frame (which means that the moving speed of the MNs is not very high). An identical transmission radius is assumed for all MNs which means that if A is the neighbor of B, B must be the neighbor of A. One antenna and half-duplex operation for every node is also assumed which means that every MN can not receive the data while sending the data or send the data while receiving the data even if receiving and sending are in the different channel. High SNR (Signal to Noise Ratio) for all receivers is assumed which means that data transmission between two nodes can be regarded reliable (This is not the case in reality. We have considered the high bit error rate for the key part of SNDR such as sequenced number structure). No global timing is needed. Based on these assumptions, we will describe the SNDR in detail.

2.2 The Channel Allocation (CA) and the Data Transmission Strategy

First we should define the neighborhood in ad hoc networks. For a given node A, A's neighborhood includes A and all of A's neighbors. Every MN has its own neighborhood. A is termed neighborhood center. Suppose there are M MNs in an ad hoc network. The ad hoc network is composed of M overlapped neighborhoods. In the SNDR, every neighborhood center is assigned a channel (the code sequence for CDMA or the frequency for FDMA and we will use code sequence to represent a channel in the following discussions). The restrictions is that: If two neighborhood centers are neighbors, they should not have the same code sequence. In other words, every MN will assign a code sequence so that inside one neighborhood, no two nodes can have the same code sequence. In fact, the CA problem (with minimum channels) is a NP problem. Furthermore, since the neighborhoods are dynamic, the CA should be adjusted to fit the restrictions. This problem is discussed in detail in our technical report. Here, we assume that the code sequences have been successfully preassigned in the ad hoc network. Note, we suppose the number of available channels in the ad hoc network is sufficient to satisfy the above requirements; If sufficient that is not the case, the SNDR should be slightly modified before used. Further research will be conducted for the case that the number of channels is not sufficient.

The following is the descriptions of the data transmission strategy. For a given node A which is A's neighborhood center, if one of its neighbor, say B, wants to send the data to A, B must use the same code sequence preassigned to A to send the data. In other words, A only receives the data packets modulated by the code sequence preassigned to it. TDMA will be used among the neighbors of A when they all want to send data to A. In a given neighborhood with center A, A mainly receives the data from its neighbors (we will see later, A should also send a few control packets and a frame start signal to its neighbors). In other words, the neighbors of A mainly send data to A. The control information and start signal sent by center A to its neighbors use the same channel assigned to A. In this data transmission process, the neighbors use the code sequence assigned to the center to send data to it while the center uses the same code sequence to send the control information and a start signal (rather than data) to its neighbors. Inside one neighborhood with center A, there are two data transmission strategies of A's neighbors that want to send data to A. One is contention-based, the other is contention free. We will first briefly describe the contention-based method.

2.3 The Contention-Based MAC

In the contention-based MAC, given a node A, suppose A's neighbor B wants to send data to A. First, it should switch its channel to A's channel. Then it will use CSMA/CA to send data. Here, the hidden terminal problem will occur and the Busy Tone Method can be used to avoid the hidden terminal problem. The performance of this MAC depends on the average number of neighbors of one node. The other issue is that for the one antenna per node case, center A can not be in the state of receive the data sent by its neighbors all the time. A will send its data to other nodes using other nodes' code sequences. A only receives the data from its neighbors in a certain time period. For that, A sends a receiving start signal to all its neighbors notifying that it will receive the data in the following time period T. When receiving such signal, the neighbors which want to send data to center A will utilize CSMA/CA in T. The value of T is determined by the overload of the ad hoc network and workload of the center. Further research will be conducted on the contention-based MAC, investigating the value of T, for example.

2.4 The Contention-Free MAC: SNDR

The main idea of the SNDR is to index the neighbors of a given node (from I to K), which, in fact, can be completed together with the neighbor-finding process. A simple method to maintain the sequence number structure is given below.

After node A finds a new neighbor, it will assign a sequence number K+I to it (K is the largest sequence number in the current neighbor set of node A). When A recognizes that one of its neighbor has moved away, it should send a
Delete packet including the sequence number of the moving neighbor to all of its other neighbors. Every neighbor of A will check this packet to see if that number is less than the sequence number of itself. If YES, it will decrease its sequence number by one. Otherwise, it will leave its sequence number unchanged.

For a given node A, every neighbor of A will know its sequence number to A. In other words, for a given node B, B will know its sequence numbers to all of B's neighbors (every neighbor of B will assign B a sequence number). Based on this, a complete neighbor table includes a sequence number and a channel number with respect to each neighbor, as shown in Table 1.

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>Channel Number of the Neighbor</th>
<th>Sequenced Number to the Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 1: the Format of the Neighbor Table

We should point out that, for a given node A, it is unnecessary to store the sequence numbers of its neighbors while it is necessary to know its sequence numbers to its different neighbors.

If the mobility of an ad hoc network is high, the neighbor set will be updated very often. In this case, the sequence number structure is inefficient. Another issue is that if the Delete packet sent by center A is not received correctly by some of its neighbors, the sequence structure will be incorrect. The center will observe that and send the Delete packet again to repair the sequence structure. Because for a given node, the number of its neighbors is not large, keeping the sequenced structure is pretty easy in the case that the mobility of the ad hoc network is moderate.

Based on the above, we know that for any given node A, a code sequence is given and all its neighbors have been sequenced. The major problem now is how to schedule the neighbors of A when they want to send data to A (all neighbors of A will use A's code sequence). We separate the channel in one neighborhood into two parts using TDMA. One is from the neighbors to the center, termed "upstream channel" and the other is from the center to its neighbors, termed "downstream channel". The upstream channel plays a major role while the downstream channel the minor role in data transmission (only the control packets from the center and the frame start signal are included).

2.5 The Double Reserved Method

Consider one neighborhood and suppose the center is A and the neighbor set is N(A).

In the SNDR, Time Division Duplex (TDD) is assumed for the upstream and the downstream channel. Since the neighbors are sequenced, it is possible to reserve one time slot for each neighbor in each upstream frame (a frame is divided into certain number of time slots). However, this method is very inefficient, since normally not all neighbors want to send the data to the neighborhood center. Many slots in the upstream frame will be wasted due to the pre-reservation.

We propose the Double Reserved (DR) Method to avoid the waste of slots while retain the collision-free properties. In the DR method, no time slot will be wasted as the length of one upstream frame is variable. Also, no collisions will occur. This method improve the throughput of the neighborhood, thus it will improve the performance of the total ad hoc network. We will describe the DR method in detail below. First, let's look at the structure of the frame including downstream and upstream which is shown in Figure 1. It is of variable length based on how many data packets the neighbors will actually send to the center.

1. Part A is the Upstream reserve field which reserves one bit for each neighbor, from 1 to N. This part is of fixed length (N bits). N is the maximum number of neighbors that a node can have in the ad hoc network. A neighbor having sequence number m, is assigned the mth bit of the Upstream reserve field. Normally each neighbor has only one reservation bit, but exceptions are allowed. When a neighbor has data to send, it will set its reservation bit to 1. If no data is to be sent, no operation is needed since the original value was 0.

2. Part B is the Downstream reserve reply field (the same format as Upstream reserve field) which is replicated from the Upstream reserve field. Based on such scheme, the hidden terminal problem can be easily avoided. Every neighbor can obtain the reservation information (from the Downstream reserve reply field) about the current Upstream sub-frame, through which one can calculate which slot it should occupy.

3. Part C is the control field which is used for the center to send the control information and start signal to its neighbors.
4. Part D is the *Upstream data field*. The length of the *Upstream data* is variable. Based on the reservation information in the *Downstream reserve reply field*, every neighbor of the center can easily calculate which data slot of the *Upstream data field* it should occupy to send data. In this scheme, no data slot will be wasted at any time and no hidden terminal problem will occur. For example, when neighbors 1, 3, 4, 6 are sending data to the center (Suppose \( N = 8 \)), the first, third, fourth and sixth bit of the *Upstream reserve field* will be set to 1 correspondingly and the *Upstream reserve field* will hold 10110100. The *Downstream reserve reply* will also hold 10110100. Upon receiving the *Downstream reserve reply* signals, every neighbor will know the reservation information in the current *upstream data field*. Based on that, each neighbor can independently and parallelly calculate which slot it should occupy to send the data. Neighbor 1 will occupy the first data slot in *Upstream data field*. Neighbor 2 will never send data. Neighbor 3 will sum all the 1’s ahead of it based on the received *Downstream reserve reply* signals. The total number of 1’s is 1 in this example and the data will be sent in slot (1 + 1). For neighbor 4, the sum is 2 and it will know to send data in the data slot (2 + 1). For neighbor 6, the sum is 3 and it will know to send data in data slot (3 + 1). Then, the current *Upstream data* ends. The center will know the length of the *Upstream data* based on received *Upstream reserve* signals. It will then decide when to send the next frame start signal to receive the next frame of data from its neighbors. Note, in the one antenna per node case, the center can not be always in the state of receiving data sent from its neighbors. It will also send its data to other nodes using other nodes’ code sequences. Due to this reason, the center should receive the data and send the data in turn. The time periods of receiving and sending is determined by the load of an ad hoc network and the workload of the center. Normally they should approximately be the same. The tradeoff can be identified to obtain the maximum throughput.

In this scheme, no data slot will be wasted and no collision will occur. Moreover, the hidden terminal problem will be totally avoided. The throughput of the system will be improved.

5. For Part D, every data slot is followed by a tag of one bit. If the tag of a certain slot is 1, it means that the corresponding neighbor has not finished its current data transmission. Otherwise, the data transmission is completed. The center will decide whether it should enter the receiving state or the sending state partially based on these tags.

2.6 The Basic Process of the SNDR

We will describe how the SNDR works below (Under the assumption of one antenna per node):

First, we should point out that the MAC frame for the center is not continuous, because while the center (supposedly A) sends the data to other nodes, it can not receive the data at the same time. In the SNDR, there are two states for every MN. One is the Receiving State (RS) and the other is the Sending State (SS). Given a node A, if A is in the SS state (suppose A wants to send its data to B), it will change the code sequence to that of B. While upon receiving the frame start signal from B, A will send the data in B’s frame. If A enters into RS, A will send its frame start signal using its own code sequence to get the data from its neighbors. For the neighbors that want to send the data to A, they will detect the frame start signal and send their data to A.

Which state to enter is determined by the node itself based on the load of the ad hoc networks and the workload of the node. In the SNDR, when one node has data to send, it will enter the SS state. However, the node can not be in the SS state very long. Normally, repeated SS state should be followed by an RS, because if all nodes are in SS state, the whole system will be in deadlock. Normally, when the system is stationary, the number of SS state for a node = the number of RS state for a node \( * N \) (\( N \) is the average number of neighbors a node can have in the network). A node can also decide which state it should enter partially based on the tag information. If the tag information informs a certain node that there are still a large amount of data waiting to be sent to the node by its neighbors, the node will decide to enter the RS state with high probability. If a node has no data to send, it remains in RS state. If two nodes want to send data to each other and they are all in SS state, each of them waits for the frame start signal from the other node. No one can send or receive data in this case. We give a timeout \( t \) for the length to stay SS. If after \( t \), the data is not sent yet, then the node will transfer to RS state.

When a node is in RS, it collects the data from its neighbors. The node should send the frame start signal to start one SNDR MAC frame. When a node is in SS, it sends data to one of its neighbors. The node should wait for the frame start signal. Suppose A is in RS, first, it should send its frame start signal with its own code sequence. Then A will receive Part A of its frame. Then it will broadcast Part B and Part C using its own code sequence. After that, A will receive Part D modulated by its code sequence. It is clear that for the nodes in RS, they need not change the code sequence, but they should change the antenna state (receiving or sending). If A is in SS (A wants to send data to B), first A should wait for the frame start signal using the code sequence of B. When A detects it, it will mark its reservation bit of the *Upstream reserve field*. Then It will wait for Part B and Part C. Upon receiving Part B and C, it will send its data in Part D of B’s frame. It is obvious that the communication within one neighborhood will use the same code sequence (data trans-
mission from neighbors to the center). In the same state (RS or SS), the node need not change its code sequence but only change its antenna state (receiving or sending).

In summary, by the SNDR, we can easily avoid the slot-wasting problem and retain the collision-free property. This method does not need any handshake process when the neighbors send data to the center. It will greatly improve the throughput of the ad hoc network.

### 2.7 Two Antennas per Node Case

The SNDR discussed above is based on the half-duplex operation (one antenna per node), which will make the hardware implementation simple. In our transmission strategy, multi-channel ad hoc networks, the channel in which a node sends data is assumed different from the channel in which it receives the data. It is possible for a node to send and receive data at the same time. Two antennas per node, full-duplex operations are needed in this case. The hardware implementation becomes more complex hereby, but the throughput of the total ad hoc networks will significantly increased. The following is the transmission strategy for full duplex operations, including contention-based mode and contention-free mode.

For the contention-based mode, given a node A, if A's neighbor B wants to send data to A, A need not send any receive start signal, while B changes its channel to A's channel. Then it uses CSMA/CA to send data. A will use one antenna to send data and the other antenna to receive data from its neighbors (in different channels). To avoid the hidden terminal problem, the Busy Tone Method can be used in the MAC. The performance of the MAC will depend on the average number of the neighbors per node.

For the contention-free mode, every MN can stay in RS and SS at the same time which means that one antenna is used to send data to other nodes while the other antenna is used to receive the data from its neighbors (in different channels). On the one hand, every node will send a frame start signal to its neighborhood right after its previous MAC frame was completed, which means that the MAC frames transmitted by one node will be continuous. In other words, every node can continue receiving data from its neighbors. On the other hand, every node can continue sending data to other nodes at the same time. No state transfer exists any more in such case with two antennas per node. Every node is receiving data from its neighborhood and sending data to other nodes at the same time. The total throughput of the system is greatly improved. With a little modification of on the MAC frame, QoS (Quality of Service) can be further assured in ad hoc networks.

### III Analysis of the SNDR

When A wants to send data to B, A should enter SS and change its code sequence to that of B. When A wants to receive data from its neighbors, it will enter RS and use its own preassigned code sequence. A waits for the frame start signal of B to send data to it. Receiver B will control at what time A sends data to B. As we mentioned above, the default state for every node is the RS state. If one node has data to send, it will enter the SS state. In the SNDR, it is the receiver, not the sender, plays the major role. The SS is an unstable state and the RS is a stable state. We should emphasize that no two nodes inside one neighborhood can have the same code sequence, based on the assumption of one antenna per node.

We discuss below the advantages of SNDR:

1. It is easy to implement. The neighbor-sequence can be realized together with the neighbor-finding procedure. Furthermore, we can use the control field to maintain the neighborhood-property.

2. It will improve the overall throughput of the ad hoc network. For SNDR, it needs no handshake process when sending or receiving data. The MAC guarantees no collision during the data transmission. The additional overhead for the SNDR transmitted in the ad hoc network can be ignored (N/4 bytes for reservation information overhead where N is the maximum number of neighbors that a node can have, and the Delete packets to maintain the neighbor sequenced property). It is obvious that the throughput of the ad hoc network can be improved. Under the assumption of two antennas per node, the performance can be even much better.

### IV Conclusion and Future Work

We have designed a new MAC named SNDR to support multi-channel ad hoc networks efficiently. This method can significantly improve the throughput of the ad hoc network. The collisions during data transmission can be completely avoided. For every MN, it can reserve the data slot without sending any request packets. It can greatly speedup the communication of the ad hoc network, which is very important for QoS communication, especially under the assumption of multiple antennas per node. Some variations such as one antenna per node vs. two antennas per node and contention-based vs. contention-free scheme are discussed. In a word, the SNDR will greatly support the communication based on multi-channel ad hoc networks. It is worth while to make continuous efforts in this and related research areas, including the contention-based MAC and a simple implementation of SNDR.
References


