Enhanced Broadcasting and Code Assignment in Mobile Ad Hoc Networks

Jinfang Zhang, Zbigniew Dziong, Francois Gagnon and Michel Kadoch
Department of Electrical Engineering, Ecole de Technologie Superieure
Montreal, Quebec, Canada

ABSTRACT

A CDMA-based mobile Ad Hoc networks face two main design challenges. One is to periodically update connectivity information, namely, neighboring nodes and the codes used by neighboring nodes. The other is to guarantee that there is no code collision in two hops' distance. This paper proposes an enhanced time-spread broadcasting schedule for connectivity information update. Based on the connectivity information, a code assignment and potential code collision resolution scheme to solve hidden/exposed nodes problem is proposed. Simulation results demonstrate the efficiency and effectiveness of the proposed schemes.

Keywords: time-spread, broadcasting, connectivity update, code assignment, code collision resolution, Ad Hoc networks

I. INTRODUCTION

Most recent radio access schemes for Ad Hoc networks use multiple channel approaches for more efficient radio resource utilization[1][2]. CDMA is a promising multiple channel radio technique where a channel is defined by the use of a determined pseudo-random sequence. The basic idea of CDMA-based multichannel MAC design[3] is that the nodes within two hops' distance should adopt different codes, so that the code collision can be avoided. To do that, each node is required to periodically broadcast its code on a common code channel so that the neighboring nodes can have the information of what code is being used. Periodical broadcasting is also important for updating distributed databases, routing tables, etc.. As the wireless channel is inherently a broadcasting medium, it is important to have efficient algorithms for broadcasting schedule in order to avoid collisions. Aloha-based totally random algorithms can resolve reception conflict by random backoff, but they do not provide a delay bound. Time-Spread Multiple-Access (TSMA)-based algorithms schedule transmissions in deterministic time slots for a static or centralized network [4][5]. However, algorithms proposed in [4] depend critically on network topology and cannot efficiently support a highly mobile environment while schemes presented in [5] need the knowledge of the exact network size, which is practically a varying parameter. Additionally, all these broadcasting algorithms assume data collision resolution by retransmissions, and to our knowledge, there is no discussion on node identity and code assignment broadcasting in CDMA-based mobile Ad Hoc networks, neither is any consideration on code collision resolution. Code collision has the risk of hidden/exposed nodes problem, which results in either reception collision or reception error in transmitter-based or receiver-based data exchanges. Therefore, it is critical to detect and resolve the potential code collision immediately after each connectivity update.

An enhanced time-spread broadcasting scheme which takes advantage of deterministic time slot allocation and random time-spread properties to achieve high probability of successful broadcasting is proposed in this paper. In periodic broadcasting, the transmission order of each node is determined based on the pre-assigned code channel together with a random time-spread to greatly decrease probability of collisions. The second contribution of the paper is a distributed code assignment and potential code collision resolution scheme. Whenever a code collision is detected due to node mobility, the proposed collision resolution scheme resolves the hidden and exposed nodes problem. In this way, each node maintains a table with the updated neighboring nodes information and uses this table for data scheduling and transmission on dedicated code channels. The data scheduling and transmission is out of the scope of this paper and the interested reader is refered to [6][7][8].

The remainder of the paper is organized as follows. Section II presents the system model, followed by the proposed enhanced time-spread broadcasting scheme in Section III. Section IV discusses in detail the distributed code assignment and potential code collision resolution schemes. Simulation results that validate the proposed algorithms are given in Section VI, followed by conclusion remarks.

II. SYSTEM MODEL

The considered network works on a single frequency band. We assume a symmetric connection between nodes, which means that if node i can hear node j, then node j can hear node i as well. The code channels are divided into two groups. The common code channel is used by all nodes for broadcasting; the dedicated code channels are assigned to each node. The dedicated channel has two functions: one is to assist broadcasting scheduling, and the second one is for dedicated data communications. Each node is equipped with a half-duplex receiver so that it cannot transmit and receive at the same time, which is denoted as first class collision. Since broadcasting is executed on common code channel, if two or more transmissions arrive at a node simultaneously, then all the transmissions are destroyed, which is denoted as second class collision. When a node is broadcasting a message, for successful reception by all its neighbors, all the first hop neighbors are not allowed to transmit to avoid first class collision, and at the same time, all the neighbors of the first hop neighbors are not allowed to transmit to avoid second class collision. From the above constraint, it is observed that two nodes can broadcast at the same time without conflicts if and only if they are more than two hops away from each other.

All the nodes are synchronized at packet level. The transmission time is divided into periodic connectivity update phase and code assignment and collision resolution phase. In connectivity update phase, each node broadcasts its identity and code information to notify physical existence and to assist with hidden/exposed nodes avoidance. The new coming node is assigned a code channel and the potential code collision is resolved in the following code assignment and collision resolution phase.

III. ENHANCED TIME-SPREAD BROADCASTING

If we arrange the broadcasting in one slot in connectivity update phase, severe first class and second class collisions will happen. To avoid those collisions, simultaneous broadcasting nodes must be located at least two hops away from each other. Since the basic idea of multicode CDMA-based MAC design is that nodes within two hops' distance adopt different code channels for collision avoidance, we can schedule broadcasting by using the code channel information.

In an Ad Hoc network with average node degree (the number of first hop neighboring nodes that are identified by a node) of D, to satisfy the constraint that nodes within two hops' distance should be allocated with different codes, we assume that M number of codes are required. The detailed implementation of code assignment will be discussed in the following section. Logically, we can associate each pseudorandom code with a number. For instance, M number of codes can be indexed to code channel $1, 2, 3, \cdots, M$, respectively. For a node with code channel index i, $1 \leq i \leq M$, all the other nodes with the same code channel index i are located more than two hops away. Therefore, if we time-spread one broadcasting slot into M slots and arrange the broadcasting in the increasing order of code channel index, the two classes of collision can be effectively avoided. This is referred to as code-based time-spread in the following discussion. Fig. 1 demonstrates a simple network topology and the broadcasting order of a 6, N1, N2,$\cdots$, N6, nodes network. The connections between nodes mean that they are neighbors to each other, and the number above each node is code channel index. For the shown network topology, the node degree $D = 2$. With the code assignment of 1 to nodes N1 and N4, code 2 to nodes N2 and N5, and code 3 to nodes N3 and N6, we can achieve exclusive code assignment within two hops' distance. As shown in Fig. 1, if the broadcasting order is set such that nodes with code i broadcasts in time slot starting from $T + (i-1)T_x$, where T is the starting point of a connectivity update phase; T_x is the broadcasting slot duration which is determined by the transmission delay and propagation delay, then each broadcasting can be successfully received by all the neighboring nodes.

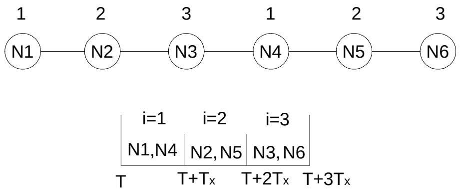

Fig. 1. Broadcasting order for static networks

Though the demonstrated broadcasting schedule is optimal for a static network topology, it is far from optimal when network topology changes with node mobility. When two or more nodes with the same assigned code move into the neighborhood, the constraint of no code assignment duplication in two hops' distance is broken and code collision happens. With the code-based time-spread broadcasting schedule, broadcasting collision happens when the colliding nodes (nodes within two hops' distance and having the same code) broadcast in the same time slot. To further alleviate broadcasting collision caused by node mobility, we modify the above code-based time-spread broadcasting with a second random time-spread. In specific, given a random time-spread window size L, each broadcasting slot is further expanded into L slots. In this way, a node with code channel index i broadcasts in time slot starting from $T + (i-1)LT_x + (k-1)T_x$, where k is an integer generated at each node and uniformly distributed in $[1, L]$. The introduction of a random time-spread factor k makes it possible that broadcasting collision from colliding nodes can be effectively alleviated, if not eliminated. After code-based time-spread and random time-spread, the original one broadcasting slot is expanded into $M \times L$ slots, where M and L are two design parameters. Fig. 2 demonstrates the modified broadcasting schedule when network topology changes. Assume the random time-spread window size, L, is set to 2. When node N6 moves into the neighborhood of node N4, nodes N3 and N6 form a pair of hidden nodes with the same assigned code 3. At the beginning of connectivity update phase, each node generates a random integer k which is uniformly distributed in $[1, 2]$. Then with 50% probability, nodes N3 and N6 may broadcast in different time slot and the reception collision at node N4 is avoided. Because nodes N1 and N4 are still outside of two hops' distance from each other, they can always achieve successful broadcasting whenever they transmit in the same time slot or in different time slots, and the same for nodes N2 and N5.

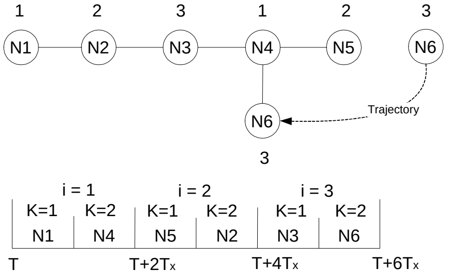

Fig. 2. Broadcasting order for mobile networks

As the time-spread broadcasting expands the time interval for connectivity update phase, one objective is to minimize the broadcasting interval with satisfaction of the broadcasting collision probability requirement. There are two time-spreads, one is based on the number of codes required, M, to achieve different code assignment in two hops' distance, and the other is a random time-spread with window size L to alleviate broadcasting collision caused by node movement. It is noted that M and L are a pair of design parameters that need to be optimized to achieve minimum length broadcasting schedule. A large value of M enlarges the first time-spread, but implies a small code collision probability in neighborhood so that a smaller random time-spread window size can be used; on the other hand, a small number of codes results in higher code collision probability, and will require a much larger random time-spread window size to achieve the same broadcasting collision probability. To minimize broadcasting length, the problem can be formulated as:

$$\begin{array}{ll} \text{Min.} & M \times L \\ \text{st.} & P_{bc}, \end{array}$$

where P_{bc} is the broadcasting collision probability requirement.

IV. DISTRIBUTED CODE ASSIGNMENT AND COLLISION RESOLUTION

A proper code channel assignment assists broadcasting scheduling. With successful reception of the periodical broadcasting message on common channel, a new incoming node can select an available code and the potential code collision can be detected and resolved.

A. Code Assignment

We propose a distributed code assignment scheme that assumes knowledge from one hop neighboring nodes only. It is noted that a node's one hop neighboring nodes are at most two hops' away from each other. In the following discussion, when we refer to a node's one hop neighboring nodes, they are inherently within two hops' distance. To do code assignment, each node maintains two lists. One is the neighboring node list which includes the neighboring nodes and the codes being assigned to them. The other is the available code channel list which lists the codes that are not being used by any of its neighboring nodes and can be assigned to a new comer in this node's neighborhood. Let

$$\begin{aligned} A &\equiv \{\text{total code channels available}\}\,, \\ B &\equiv \{\text{code channels used by neighboring nodes}\}\,, \\ C &\equiv \{\text{available code channels}\}\,, \end{aligned}$$

then, $A = B \cup C$. Here, a node is treated to be a neighbor of itself.

Each node broadcasts a NOTICE message to update connectivity information, to identify the neighboring nodes and to exchange the available code channel information. A NOTICE message is formatted as shown in Fig. 3:

Node ID	Code channel	Available code channel list

Fig. 3. NOTICE format

- Node ID: node identity
- Code channel: preassigned dedicated code to this node
- Available code channel list: a set of codes that are not used in the neighborhood of this node

When a new node comes within the transmission range of an existing node, it will follow these steps to get admission:

1. Receive NOTICE messages from neighboring nodes;
2. Based on the neighboring node list and the available code list, select a dedicated code channel, and send out a NOTICE message with the code selection after a random backoff;

Then the neighboring nodes update their neighboring node list and available code channel list with the received NOTICE message.

It is seen that for a node to select a code channel, only knowledge from its first hop neighboring nodes is required. Assume around this node, there are N first hop neighboring nodes with their available code channel list as:

$$\begin{aligned} C_i &= \text{available code channel list of neighboring node } i, \\ i &= 1, 2, ..., \text{N} \end{aligned}$$

A code channel from $C_1 \cap ... C_i \cap ... \cap C_N$ is selected and assigned. A node fails to get a code channel cannot be identified by its neighboring nodes though it physically exists, which causes node blocking. When no NOTICE message is detected during connectivity update phase, a node is assumed to move out of the neighborhood, and the code channel will be available for potential new comers. It is noted that only neighboring nodes within two hops' distance are taken into account in code channel assignment, and the same code channel can be reused outside two hops' distance without affecting each other. Therefore, the total number of code channels needed to identify all the neighboring nodes is irrespective of the network size, but constrained by the node degree, D, namely, network density.

B. Code collision Resolution

Node mobility has the risk of potential code collision, which can cause reception collision in both broadcasting and data transmission phases due to hidden or exposed nodes problem. The reception collision in broadcasting phase is alleviated with time-spreading as discussed in the previous section. To resolve the reception collision in data transmission phase, a code collision resolution scheme is proposed in this section.

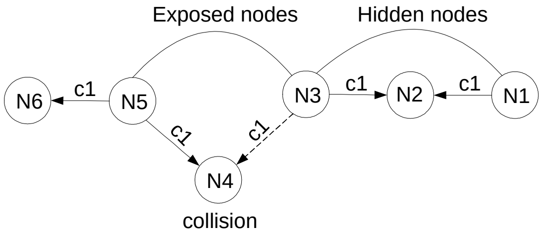

Fig. 4. Hidden/Exposed nodes

Fig. 4 illustrates the hidden and exposed nodes problem in Ad Hoc networks where only two nodes are considered to be either hidden or exposed. In practice, hidden and exposed nodes problem can happen among more than two nodes. As shown in Fig. 4, in periodic connectivity update process, a hidden nodes problem is detected when N2 notices that in its neighboring node list, two neighboring nodes, N1 and N3, are using the same code c1. An exposed nodes problem can be detected when N3(N5) finds that N5(N3) is using the same code c1 as itself. To determine which neighboring node, either N1 or N3 in hidden nodes problem, or N3 or N5 in exposed nodes problem, should be reallocated a code channel, an easy way is to randomly select a node to do code channel reallocation. However, there is possibility that a code reassignment fails due to lack of available codes, which causes the node to be disconnected from the network temporarily. Therefore, a more reasonable method to achieve low node blocking probability is proposed. As a node selects a code which is in the available code channel list of all its neighboring nodes, colliding nodes with smaller node degree are expected to have high chance for successful reassignment and should be required to do code reselection. We refer to this selection criterion as node degree-based code reassignment selection. Therefore, the node with the maximum node degree keeps the code and the remaining colliding nodes reselect an available code. If there is reselection collision, the node with the second maximum node degree keeps the code and the others reselect again till each colliding node has a unique code. It is noted that a node with small node degree has large available code channel list which is carried in NOTICE message. So, there is no more information exchange needed. The process to reselect a code channel is the same as when a node joins the system for the first time. Node blocking happens when the reselection turns out to be a failure.

Before we discuss in detail the proposed code collision resolution protocol, we first define a control node and three control messages. A control node, defined as a node who directs code reassignment, is chosen in the following way. If there are colliding nodes in a node's neighborhood, including the case that this node itself is a colliding node, this node serves as a control node; if there are only colliding nodes in one hop's distance, the node with higher node degree serves as the control node. Fig. 5 shows the control node selection in these two cases. In case (a), when N1 detects that N2, N4 and itself are using the same code c1, it determines to be a control node. While in case (b), N1 and N2 are one hop away, and N2 has higher node degree. So N2 will act as a control node to direct collision resolution.

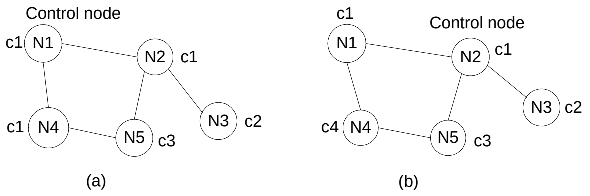

Fig. 5. Control node selection

Three control messages, CR, RR and ACK, used in collision resolution protocol are defined as follows.

1. CR message: Code Reassignment request. CR message is multicast from the control node to all the colliding nodes that need to do code reassignment.

Node ID1	Node ID2	······	Available code channel set

Fig. 6. CR format

2. RR message: Reassignment Result. RR message is sent back from colliding node to control node to report the code reassignment result.

Node ID	Code channel

Fig. 7. RR format

3. ACK message: ACKnowledgement. ACK message is multicast from control node to resolve reception collision.

Node ID1	Node ID2	······

Fig. 8. ACK format

- Node ID: colliding node identity

Fig. 9 shows an example of code collision resolution where there are three colliding nodes, $N1, N2$ and $N3$, with node degree of $1, 2$ and 3, respectively. Here, we assume that N3 acts as a control node. Based on the proposed code assignment criterion, N3 has the maximum node degree among the colliding nodes and keeps the code. N3 sends out a CR message for N1 and N2 to reselect available codes. After reassignment, N1 and N2 send out a RR message informing the newly assigned code after a random backoff to alleviate potential reception collision, and then start a counter to wait for ACK message. On reception of RR message, each neighboring node updates its neighboring node list and the available code channel list accordingly. If the collision is not completely resolved because two or more nodes select the same code again as shown in case (a), another CR message will send out to the node with smaller node degree, here N1. The attached available code channel list informs N1 of the updated code assignment. The counter is reset at the reception of CR message and N1 executes code reassignment for the second time. The introduction of a counter is to guarantee that RR message is successfully received. As shown in case (b) in Fig. 9, when RR messages from N1 and N2 arrive at N3 simultaneously, both RR messages are destroyed. Before the counter expires, neither an ACK message for successful reassignment, nor CR message for code reassignment is received at N1 and N2, and a reception collision is detected. In this case, N1 and N2 set another random backoff and send out RR message again.

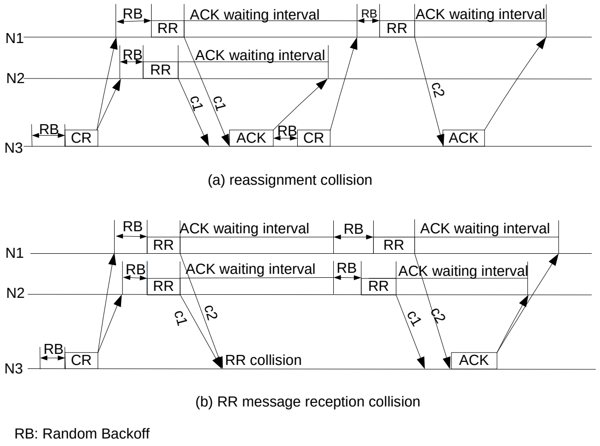

Fig. 9. Code collision resolution protocol

To cope with the reception collision at nodes which are asked to do code reassignment by more than one control nodes, random backoff can be applied before each CR message. If we ignore the signal processing delay, the time needed for collision resolution is:

$$\begin{aligned} T_{cr} &= (T_{CR} + +2T_{RB} + T_{RR} + T_{ACK}) \times n \\ &+ (T_{RB} + T_{RR} + T_{count.}) \times m, \end{aligned}$$

where T_{CR}, T_{RR} and T_{ACK} are the message transmission and propagation delay, and T_{RB} is the random backoff window size. n is the number of code reselections. In worst case, n is equal to the number of colliding nodes. $T_{count.}$ is the ACK waiting interval, and m is the number of reception collisions. In practical Ad Hoc networks, a small collision resolution length is preferred as periodical connectivity update cycle interrupts regular data transmission. The number of collision resolution rounds, n in Fig. 9 case (a) and m in Fig. 9 case (b) is chosen to satisfy the node blocking probability requirement, P_b, which is formulated as:

$$\begin{aligned} &\text{Min.} && \text{collision resoluction length} \\ &\text{st.} && P_b. \end{aligned}$$

V. SIMULATION RESULTS

Consider an Ad Hoc network where nodes are randomly distributed. A mobility model mimicing the human and vehicle movement [9] is applied. The average node degree is 16. We assume a reliable wireless communication and a free space propagation mode so that the signal attenuation is caused exclusively by transmission distance. Each node has the maximum transmission power of 7w. The minimum required received power to identify a neighbor is set to 10^{-6}w. To promptly update connectivity information for neighboring nodes identification, the connectivity update, code assignment and potential code collision resolution process is executed in every 10ms.

To show the performance improvement of the proposed time-spread broadcasting schedule, Fig. 10 compares the broadcasting collision probability with the proposed time-spread scheme and a totally random broadcasting scheme. Here, the mobile speed limit is set to 120km/h. The total number of codes is set to 50, which guarantees that the node blocking probability is small enough and each physically existing node can be identified with a unique code. The random time-spread window size, L varies from 1 to 10. Therefore, the broadcasting length varies from 50 slots to 500 slots. For the totally random scheme, each node broadcasts in a time slot uniformly distributed in 1 and the broadcasting length. It is shown that with the same broadcasting length, the proposed time-spread scheme can significantly decrease broadcasting collision probability compared to the totally random scheme. This is because in time-spread broadcasting schedule, the unique code assignment within two hops distance helps avoid broadcasting collisions, and the only reason for broadcasting collision comes from code collisions due to node mobility. While for a totally random broadcasting scheme, both the first and the second class collisions cause severe broadcasting collisions. Fig. 10 verifies that the time-spread broadcasting schedule is efficient in collision alleviation.

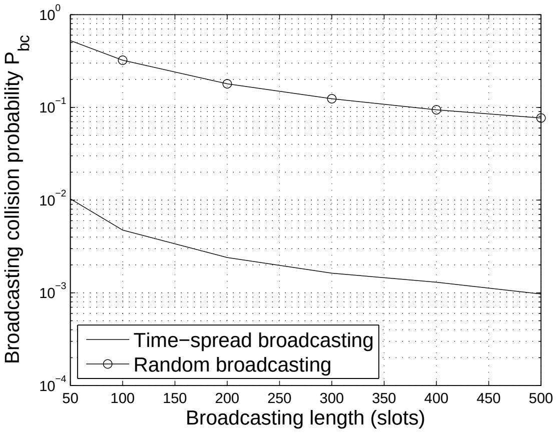

Fig. 10. Broadcasting collision probability comparisons

Fig. 11 demonstrates the effectiveness of the random time-spread in broadcasting collision alleviation. The speed limit is set to 60km/h, 120km/h and 200km/h, respectively. It is seen that as the mobile speed increases, nodes with the same code are more likely to come into neighborhood and cause higher broadcasting collision probability. As the random time-spread window size, L, increases, the broadcasting collision probability decreases significantly. Compared to the totally deterministic code-based time-spread broadcasting where $L = 1$, broadcasting collision probability can be decreased by approximately 50% when L is set to 2. As shown in Fig. 11, the broadcasting collision probability curves can be closely matched by exponentially decreasing functions. This verifies that the introduction of random time-spread is an efficient way to alleviate collisions.

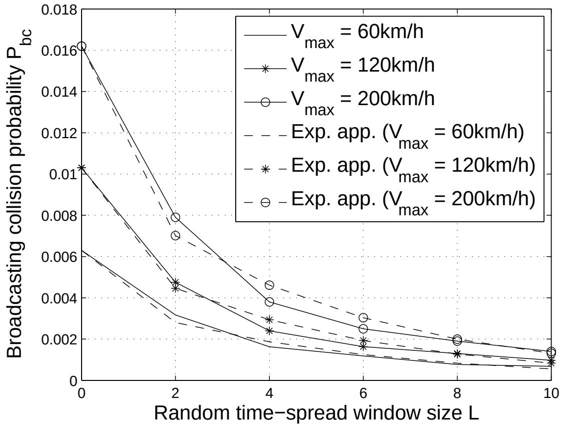

Fig. 11. Broadcasting collision probability vs. random time-spread window size

Fig. 12 shows the broadcasting collision probability with different number of codes, M, and different random time-spread window size, L. The mobile speed limit is fixed at 120km/h. Besides the fact that the broadcasting collision probability decreases exponentially with L, it also decreases linearly with the number of codes. The preferred selection of M and L should minimize the broadcasting length, $M \times L$, and at the same time guarantee the broadcasting collision probability P_{bc}. Based on Fig. 12, a minimum broadcasting length of 280 slots with combination of 70 codes and the random time-spread window size of 4 is required to guarantee broadcasting collision probability of 0.002.

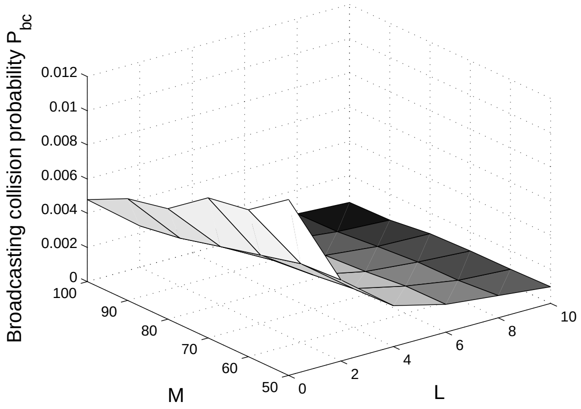

Fig. 12. Broadcasting collision probability vs. broadcasting length

To demonstrate the proposed code assignment and potential code collision resolution scheme, we assume a broadcasting collision free environment. As a well designed random backoff can exponentially decrease reception collisions of CR and RR messages, Fig. 13 focuses on the node blocking probability caused either by unsolved code collision or lack of codes. Comparisons are made between code reassignment with random node selection and with the proposed node degree-based selection. It shows that the node blocking probability decreases with the increasing number of code channels. This is because with the increasing number of code channels, the number of identified nodes increases which increase the network density. And a dense network can improve the code channel reuse efficiency. As expected, Fig. 13

shows that node degree-based code reassignment can outperform random-based code reassignment with a smaller node blocking probability. This figure also shows that one round of code reassignment is enough to resolve the potential code collision and the node blocking is mainly caused by lack of code channels. Therefore, for the simulated network, the time consumption for code collision resolution is kept minimal.

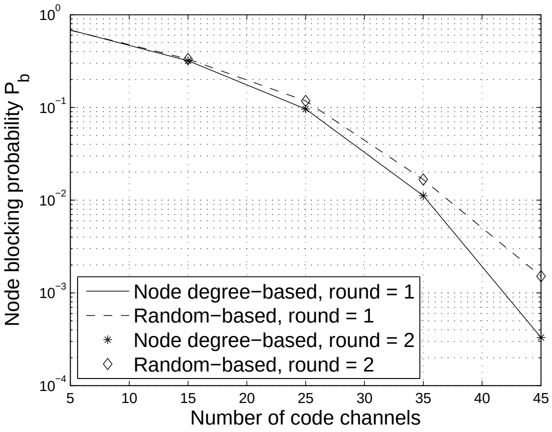

Fig. 13. Node blocking probability

VI. CONCLUSIONS

We proposed an enhanced time-spread broadcasting schedule for code assignment and the potential code collision resolution scheme in mobile Ad Hoc networks. By combining deterministic time slot allocation and random time-spread in broadcasting schedule, the broadcasting collision probability is shown to be significantly reduced, which also guarantees high probability of code collision detection. The potential code collision can be resolved with the proposed node degree-based code reassignment. The algorithms also minimize the time consumption and are simple to implement in practical systems.

REFERENCES

[1] C. Y. L. Y. C. Tseng, S. L. Wu and J. P. Sheu, “A multi-channel MAC protocol with power control for multi-hop mobile Ad Hoc networks,” in *Int’l. Conf. Distributed Comp. Sys. Wksp.*, pp. 419–424, 2001.

[2] K. T. Jin and D. H. Cho, “Multi-code MAC for multi-hop wireless Ad Hoc networks,” in *Vehic. Tech. Conf. IEEE*, vol. 2, pp. 1100–1104, 2002.

[3] A. A. Bertossi and M. A. Bonuccelli, “Code assignment for hidden terminal interference avoidance in multihop packet radio networks,” *IEEE/ACM Transactions on Networking*, vol. 3, pp. 441–449, August 1995.

[4] A. Ephremides and T. Truong, “Scheduling broadcasts in multihop radio networks,” *IEEE Transactions on Communications*, vol. 38, p. 460, April 1990.

[5] J.-H. Ju and V. O. K. Li, “An optimal topology-transparent scheduling method in multihop packet radio networks,” *IEEE/ACM Transactions on Networking*, vol. 6, pp. 298–306, June 1998.

[6] J. Zhang, Z. Dziong, F. Gagnon, and M. Kadoch, “Multiuser detection based MAC design for Ad Hoc networks,” *Accepted for publication in IEEE Transactions on Wireless Communications*, May 2008.

[7] J. Zhang, Z. Dziong, F. Gagnon, and M. Kadoch, “Receiver initiated MAC design for Ad Hoc networks based on multiuser detection,” in *Proceedings of QShine’08, Hongkong, China,*, July 28-31 2008.

[8] J. Zhang, Z. Dziong, F. Gagnon, and M. Kadoch, “Performance evaluation of a multiuser detection based MAC design for Ad Hoc networks,” in *Proceedings of the IEEE 66th Vehicular Technology Conference VTC2007-Fall, Baltimore, USA*, September 30 - October 3 2007.

[9] J. Zhang, J. M. Mark, and X. Shen, “An adaptive resource reservation strategy for handoff in wireless cellular CDMA networks,” *Canadian J. Electrical and Computer Engineering*, vol. 29, pp. 77–83, Jan./April 2004.