Cooperative MAC Scheduling in CDMA-MANETs with Multiuser Detection*

Jun Li, Yifeng Zhou, Mathieu Déziel, and Louise Lamont

Communications Research Centre Canada
3701 Carling Ave. Ottawa, ON. K2H 8S2 Canada
{`jun.li,yifeng.zhou,mathieu.deziel,louise.lamont`}`@crc.gc.ca`

Abstract. Code division multiple access mobile *ad hoc* networks (CDMA-MANETs) will be a next-generation wireless networking architecture to connect various military platforms. The classic contention-based MAC protocols are inappropriate for tactical *ad hoc* networks, where more rigid requirements for the quality of service (QoS) (*e.g.*, guaranteed packet delivery) have to be satisfied. In this paper, we propose a contention-free medium access control (MAC) scheduling framework for CDMA-MANETs where each mobile unit is capable of multiuser detection (MUD) as well. In this MAC scheduling scheme, how and when a pending data packet is going to be transmitted are cooperatively determined by the respective transmitter-receiver pair. Furthermore, to fully utilize the functionality provided by multiuser detection, our proposed cooperative MAC scheduling scheme is able to schedule multiple transmitters to simultaneously transmit packets to a same receiver. Computer simulations are carried out to demonstrate the performance of the proposed cooperative MAC scheduling framework. It is confirmed from simulation results that the packet average delay increases with either the packet generation rate or the network size. More importantly, the proposed cooperative MAC scheduling framework is more suitable for MUD-enabled CDMA-MANETs with heavier network traffic and possibly a larger number of network nodes.

Keywords: Code division multiple access (CDMA), mobile *ad hoc* networks (MANETs), multiuser detection (MUD), medium access control (MAC), packet scheduling, cooperative communications.

1 Introduction

Multiuser detection (MUD) techniques can significantly improve the performance and capacity of code division multiple access (CDMA) networks, which have long been used in both the military and the civilian domain for one-hop communications [7]. Recently multi-hop mobile *ad hoc* networks (MANETs) are receiving increasing research attention due to their ubiquitous applications to military and civilian networks [15]. It is envisioned that multi-hop code division

---

* This work was supported by Defence Research and Development Canada (DRDC).

D. Simplot-Ryl et al. (Eds.): ADHOCNETS 2011, LNICST 89, pp. 131–146, 2012.

multiple access mobile *ad hoc* networks (CDMA-MANETs) will serve as a next-generation networking architecture used to network various military platforms, such as manned/unmanned aerial vehicles, manned/unmanned ground vehicles, and soldiers, and thus some more recent research has focused on this specific type of *ad hoc* network. (See, *e.g.*, [3,5,6,8,9,11,13,17,18] for studies on the data link layer and [4,10,12,14] for studies on the network layer.)

It is noted that a majority of studies [3,6,8,9,11,13,17,18] on the medium access control (MAC) layer have been based on the RTS/CTS handshaking mechanism, as used in the IEEE 802.11 standard [2], to scheduling transmission of packets, and are contention-based scheduling in nature. Some drawbacks of contention-based MAC schemes include that they experience a high packet collision probability under heavier network traffic and the quality of service (QoS) cannot be guaranteed in general. In a military CDMA-MANET, However, it is often explicitly specified that one or more rigid QoS requirements (*e.g.*, guaranteed delivery of command and control data with a delay limit) have to be satisfied, and thus it is impractical to use a contention-based MAC scheduling protocol in such a network. Recently, some non-contention based MAC designs have been reported in the literature. Among them is a hybrid token CDMA MAC protocol based on token-passing schemes [5]. By circulating the token around the network in a pre-defined order, a node with packets to transmit obtains a CDMA code dynamically and is ensured to transmit without contending to access channels. The multiuser detection techniques were not considered in [5].

In this paper, we propose a cooperative MAC scheduling framework in the context of military CDMA-MANET where each mobile unit is capable of multiuser detection. In the network, a unique control message, referred to the token frame, circulates continuously around the network in a non-predetermined order. Through circulating the token frame, each node obtains the information required for the MUD functionality, and based on the traffic information of the neighborhood, a potential transmitter-receiver pair cooperatively determines how and when a pending packet is going to be transmitted. To fully utilize the functionality provided by multiuser detection, our proposed cooperative MAC scheduling scheme schedules multiple transmitters to simultaneously transmit packets to a same receiver. Computer simulations are carried out to demonstrate the performance of the proposed cooperative MAC scheduling framework. It is confirmed from simulation results that the packet average delay increases with either the packet generation rate or the network size. More importantly, the proposed cooperative MAC scheduling framework is more suitable for MUD-enabled CDMA-MANETs with heavier network traffic and possibly a larger number of network nodes.

The rest of the paper is organized as follows: In Section 2, we describe a CDMA *ad hoc* network model and the problem to be solved in this study. A cooperative MAC scheduling framework is proposed in Section 3. Performance evaluation of the proposed cooperative MAC scheduling scheme is carried out in Section 4 using simulations. Finally, concluding remarks are given in Section 5.

2 Network Model and Problem Statement

In this section, we describe a CDMA-MANET model followed by a statement of the problem to be studied in this paper.

2.1 Network Setup and Assumptions

As shown in Fig. 1, a number N of nodes, which can be aircraft, or ground vehicles, or soldiers, are deployed in the battlefield and form a military mobile *ad hoc* network. All nodes are synchronized at packet level. The synchronization can be achieved by tracing a common timing source such as the global positioning system (GPS). The distribution of codes is shown in Fig. 2. A total of $M + 1$ CDMA codes are allocated for message transmissions, among which M codes (code 1 to M in Fig. 2) are assigned for data packets and one code, referred to as the Common Control Channel (CCC) in Fig. 2, is for disseminating control packets. Each node is equipped with two half-duplex transceivers of a same transmission range. One transceiver operates on code 1 to M and is capable of CDMA multiuser detection; the other is a simplified system that operates on CCC, and is not required to have the multiuser detection capability. All wireless links are assumed to be error-free so that only the node mobility can cause the link breakage, *e.g.*, two nodes are out of the transmission range of each other.

The network is assumed to be (fully or partially) connected although the network topology changes over time due to the mobility of the nodes in the network. It is also assumed that one of the mobile nodes is selected as the backbone node (BN), as shown in Fig. 1. BN typically has long-lasting power and serves as a gateway to another network, such as the secured Internet. We assume that BN is alive all the time. In the proposed MAC, the backbone node also plays the role of the token lead who initiates the token. In addition, the MUD capacity of a node is assumed to be greater than the maximum number of neighboring nodes of the node. The MUD capability of a node refers to the maximum number of transmission sources that the node can jointly detect. The assumption of greater MUD capability allows simultaneous transmissions of all neighbors of a node. When a node has data packets to transmit, it needs a CDMA code to be assigned by a code assignment mechanism. Because of the limited size of CDMA code set, reuse of a code is desired and necessary in CDMA *ad hoc* networks. To avoid collisions between two simultaneous transmissions, a code assignment mechanism has to guarantee that any two nodes that are either one- or two-hop away from each other are not assigned to a same code. In this paper, we assume that the code assignment mechanism (CAM) in [16] is implemented in each node through broadcasting control messages (*i.e.*, the token frame detailed in Section3.1).

2.2 Problem Statement

In CDMA-MANETs, the implementation of multiuser detection at the MAC layer has the following two requirements.

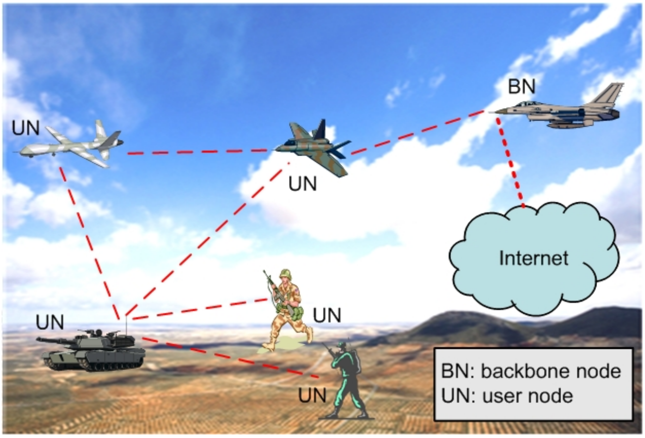

Fig. 1. Military CDMA Mobile Ad Hoc Network

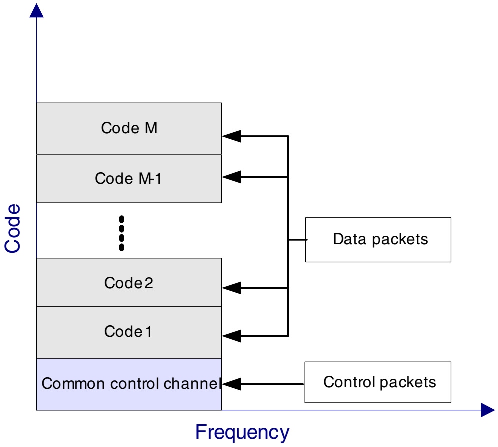

Fig. 2. Channel Structure

1. **Knowledge of code at the receiving side:** For a data transmission process to be conducted for a transmitter-receiver pair, the receiver has to know the corresponding code assigned to the transmitter for the transmission. Due to the dynamic feature of code assignment in CDMA-MANETs, the receiver does not have knowledge of code until the transmitter has been explicitly assigned one. Therefore, after assignment of code at the transmitter, a MAC scheduling mechanism is required to inform the receiver of the code that this transmission uses.
2. **Starting time of an expected communication:** Due to a limited capacity of multiuser detection (*i.e.*, the number of packets that the receiver can simultaneously receive is limited), the receiver needs to decide when a

data transmission process can start and inform the transmitter the starting time of the expected communication. This is done by a MAC scheduling mechanism as well.

In this paper we propose a MAC scheduling scheme, which explicitly considers the above requirements, for CDMA-MANETs with multiuser detection. By fast circulating the token frame, a type of scheme control message, the proposed scheduling scheme belongs to the group of contention-free scheduling mechanisms and enables cooperative scheduling of multiple transmissions destined to a same receiver which is enabled for multiuser detection.

3 Cooperative MAC Scheduling

In this section, we first define control messages, scheme buffers and scheme timers that are needed by our MAC scheduling scheme. We then elaborate the cooperative MAC scheduling scheme that fully utilizes the MUD capability in CDMA-MANETs.

3.1 Scheme Control Messages

Below we introduce two types of control messages, token frame and HELLO message, to be used in the proposed cooperative MAC scheduling scheme. Due to the use of a dedicated transceiver operating on CCC, there are no collisions between data and control packets, and parallel processing of data and control packets is allowed. The parallel processing of data and control packets is expected to improve the scheme performance such as a decreased packet delay and an increased system throughput. The token frame is created by BN at the beginning of the network operation. The token continuously circulates around the network. HELLO messages are transmitted in one hop, *i.e.*, between neighboring nodes.

Token Frame. As shown in Fig. 3, the token frame contains in total seven fields that start with the *preamble* field used for piloting/synchronization and end up with the *end_of_token* field indicating the end of the token. The *source* field is the address of the node that forwards the token, while the *destination* field contains the address of the node to which the token is forwarded. The *RTR* (ready to receive) field is used to indicate whether the *source* node is ready to receive data packets. The *NOC* (number of codes) field is an integer number representing the number of code indices listed in the *code_list* field. The *assigned_code* field contains the index of the code assigned to the *source* node, or a number, referred to as *NaC* (not a code), if no code is assigned to the *source* node. The *code_list* field lists all indices of the codes currently available for assignment.

We shall note that, among these fields in the token frame, *NOC* and *code_list* fields are exclusively used by the code assignment mechanism in [16], which is assumed in this study. As will be seen in Section 3.4, all other fields in the token frame will be used in our proposed cooperative MAC scheduling scheme to schedule transmission of data packets.

preamble	source	destination	RTR	NOC	assigned_code	code_list	end_of_token

Fig. 3. Token Frame

HELLO Message As shown in Fig. 4, a HELLO message contains three fields starting with *preamble* and ending up with *end_of_message*. The *source* field is the address of the node that broadcasts the HELLO message. The size of a HELLO message is very small. In addition, a node broadcasts a HELLO message only when it loses connection to the network (see Section 3.6). Hence, the chance that the token frame collides with a HELLO message is very small. In this paper, we assume that the token frame and HELLO messages are never colliding with each other.

preamble	source	end_of_message

Fig. 4. HELLO Message

3.2 Scheme Buffers

Two buffers, successor list (SL) and transmitter list (TL), are created in a node to facilitate operation of our proposed MAC scheduling scheme in the node.

Successor List (SL). SL in a node contains all neighbors of the node and provides information about the destination to which the node forwards the token. As will be seen from the scheme, the neighbor on the top of SL will always be chosen as the *destination* of the token. Under the assumption that the network is always connected, SL is never empty. In addition, when a node overhears the token circulating among the network, SL in the node is updated accordingly.

Transmitter List (TL). TL in a node maintains the information about the nodes that have data to transmit to the node as well as the codes to be used by these transmitters. In other words, each item in TL, containing the address of a node and a code index, corresponds to a transmission to be started.

3.3 Scheme Timers

As a result of node movements, a link between two nodes can be up and down. In addition, a node could enter an area where all its neighbors are new. To deal with these two possible changes in the network topology, we set the following two timers in our proposed MAC scheduling scheme

Timer for Link Breakage (*TimerOne*). For each time a node forwards the token to its successor, the node activates *TimerOne* and sets the value to 3τ, where τ represents the one-way transmission and processing time of the token from the source to its successor. The value of τ can be obtained from the last round of the token circulation. *TimerOne* is deactivated when the node overhears transmission of the token by its successor. If the timer is active and expires, the node assumes that the recent token forwarding has failed due to link breakage, in which case the node will retransmit the token to a next successor.

Timer for Entering the Network (*TimerTwo*). For each time a node forwards the token to its successor, the node sets the value of *TimerTwo* to $3N\tau$. When *TimerTwo* expires, the node assumes that it has lost connection to the network. That is, the links previously connecting the node to its neighboring nodes are down and all current neighbors of this node are new. In this case, the node initiates the *entering* procedure to rejoin the network, which is detailed at the end of Section 3.4.

3.4 MAC Scheduling Scheme

In this section a cooperative MAC scheduling scheme is proposed for CDMA-MANETs with the capability of multiuser detection.

The key idea in our proposed cooperative MAC scheduling scheme is the "receive-forward" module. This is a major difference between our proposed MAC scheduling scheme and a classical token ring MAC protocol (*e.g.*, IEEE 802.5 [1]) that is based on the "receive-hold-forward" module. In the "receive-hold-forward" module, only a node capturing the token can transmit data, and the node, after capturing the token, holds the token until the transmission process is terminated. In comparison, with the "receive-forward" module, a node, when receiving the token, will not hold the token during a data transmission process, but simply forward the token to a successor (*i.e.*, *destination* of the node), disregarding the status of the node (*i.e.*, transmitting, receiving or idle state). The token successor corresponds to the address at the top of SL in the node. Due to dynamic updates of the successor list, the token circulates in a non-predetermined order. The core part of the cooperative MAC scheduling scheme is described as follows.

If the current node A destined by the token is idle and has data packets to transmit to a node C, which is one neighbor of A, A obtains a code via CAM and moves C to the top of A's successor list. If A is transmitting, it removes the top node from SL to the *destination* field of the token and sets the RTR field to 0 (meaning not ready for data reception). Otherwise (i.e., A is either receiving data packets or idle), it removes the top node from SL to the *destination* field of the token frame and sets the RTR field to 1 (meaning ready for data reception). After the modification of the token frame, A sets itself as *source* of the token frame and forwards the token to its successor. After receiving the token, the successor repeats modification of the token frame according to the above description and forwards it to a next successor. While transmitting the token, all neighboring nodes of source node A will overhear the token, from which whether data transmission to A can be conducted will be determined from the RTR field of the token. If a neighboring node finds that the RTR field is one, and if it has data packets to transmit to A, is currently idle, and has been assigned to a code for this data transmission, it starts to transmit data packets to A immediately. Otherwise, the overheard token frame is ignored. We assume that this “receive-forward” module is processed almost instantaneously in each node such that a source node can always overhear the token transmitted by its successor.

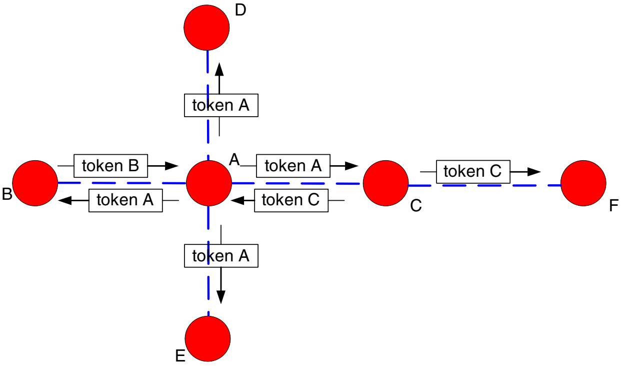

Fig. 5. A Network with Cooperative MAC Scheduling

Fig. 5 illustrates the MAC scheduling scheme described above. We assume that, when receiving the token from B (token B in Fig. 5), A is idle and has data packets to transmit to C. The nodes in A's SL is (B, D, C, E) with B on the top of buffer and E in the bottom. TL in A is empty. Then, A moves B to the bottom of SL so that the list becomes (D, C, E, B). Moreover, since A has data packets for C, C is moved to the top of SL in A, which results in (C, D, E, B) in A's SL. A then acquires a code via CAM and assigns its corresponding code index x to *assigned_code* of the token. Since A is not transmitting at the moment, RTR of the token is set to 1. The the resulting token to be sent by A (token A in Fig. 5) is shown in Fig. 6, where the NOC and *code_list* fields are addressed in CAM. Node A also moves C to the bottom of its SL after the token is sent out. Each of the neighbors of A (B, C, D, E in Fig. 5) overhears the token and moves node A to the bottom of the SL in the node. Moreover, when receiving the token sent by A, C updates its TL by running the *TL_update* procedure in Algorithm 2, after which an item $(A; x)$ is added in C's TL. Given that node C is not transmitting at the moment, C signals its neighbors that it is ready for receiving data packets by setting RTR to 1. (If C is transmitting, RTR is set to zero.) The contents of the token sent out by C (token_C in Fig. 5) is shown in Fig. 7. When A overhears the token sent by C, A will start to transmit data packets to C using the channel code of index x.

Using an algorithmic format the cooperative MAC scheduling scheme is summarized in Algorithm 1. We assume that the algorithm is run by some node A, which can be either the BN or a UN. Besides the core part described above, the scheme also includes the following procedures.

Algorithm 1. Cooperative MAC Scheduling Scheme

```
while (algorithm running in A) do
    if (TimerOne is active and expires) then
        remove the node, referred to as X, in the bottom of A's SL from the list;
        if (A's TL has an item containing X) then remove the item from the list;
        set the node on the top of A's SL as destination of the token and move that node to the bottom of the list;
        send the token to A's successor;
        reset the value of TimerOne to $3\tau$;
    end
    if (A is a UN and its TimerTwo expires) then
        empty SL and TL in A;
        call entering procedure in Section 3.6;
    end
    if (A overhears token sent by B) then
        if (TimerOne in A is active) then deactivate TimerOne;
        if (B is in A's SL) then
            move B to the bottom of A's SL;
        else
            add B to A's SL in the bottom of the list;
        end
        if (RTR of the token is 1 and A is idle and A has data waiting for transmission to B and A has been assigned to a code) then A starts data transmission to B;
        if (A is the destination of the token) then
            if (A is a UN) then deactivate TimerTwo;
            if (assigned_code is not NaC) then perform TL_update procedure (Algorithm 2);
            if (A is idle and A has data to transmit to C) then
                move C to the top of A's SL;
                obtain a code via CAM and set the code index to assigned_code of the token;
            else
                set assigned_code to NaC;
            end
            if (A is transmitting) then
                set the RTR field of the token to 0;
            else
                set the RTR field of the token to 1;
            end
            assign A as source of the token;
            set the node at the top of SL as destination and move that node to the bottom of A's SL;
            send the token to its successor;
            activate TimerOne and set the value to $3\tau$;
            if (A is a UN) then set TimerTwo to $3N\tau$;
        end
    end
end
```

preamble	A	C	1	NOC	x	code_list	end_of_token

Fig. 6. Token A

preamble	C	F	1	NOC	NaC	code_list	end_of_token

Fig. 7. Token C

3.5 Update of Scheme Buffers

The scheme buffers need to be updated accordingly in the MAC scheduling scheme. First of all, both SL and TL in a node are updated if either $TimerOne$ or $TimerTwo$ of the node expires. For instance, when $TimerTwo$ of node A expires, A's SL and TL will be emptied. Because SL in node A is a priority list of A's neighbors, A will update its SL each time the token frame sent by a node, denoted by B, is overheard. That is, B will be put into the bottom position of A's SL. In addition, the node associated with the *destination* field of the token sent by A has to be moved to the bottom of A's SL. When A is the destination of the token sent node B, A's TL is updated according to Algorithm 2. That is, if the *assigned_code* field in the token contains a valid code index, A looks up its TL. If TL does not contain an item corresponding to node B, a new item containing B and code index equal to *assigned_code* is created and added to A's TL. If TL contains an item corresponding to node B, then B's code index in that item is reassigned to *assigned_code*.

Algorithm 2. *TL_update* Procedure

```
begin
    if (A's TL has an item containing B) then
        replace B's code index with assigned_code of the token;
    else
        create a new item containing B and assigned_code;
        add the item to A's TL;
    end
end
```

3.6 Management of Node Mobility:

Some node X that wants to enter the network has to initiate the *entering* procedure. In the procedure, node X starts to periodically broadcast a $HELLO$ message, which contains the address of node X, over the common control channel. If a neighboring node Y that is already in the network receives the $HELLO$ message, node Y updates the successor list by simply adding X to the top of Y's SL. In the next time node Y receives the token, node X will be Y's successor who receives the token. By this way, node X joins the network.

4 Performance Evaluation

In this section, we describe a performance study on the proposed cooperative MAC scheduling scheme using simulations implemented in MATLAB.

4.1 Simulation Setup

In simulations the network size N varies between 4 and 24 nodes. The wireless channel connecting a pair of nodes is assumed to be bidirectional and has a constant data rate 1 $Mbps$. Due to the mobility of the network nodes, the wireless channel between the pair of nodes is up and down over time. The period of channel status changes is set to 40 milliseconds, which equals the duration of one slot. We assume that the network topology changes over time according to a Markov model. That is, the channel status process $\{X_{ij}(t) : t = 1, 2, \cdots\}$ for a pair of nodes i and j is a two-state (*i.e.*, up and down) Markov chain with the transition probability matrix $\mathbf{P} = \begin{bmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{bmatrix}$. The parameter t in the process denotes the slot numbers in discrete time. In addition, two channel status processes for two different pairs are assumed to be independent of each other. All data packets are assumed to have the same size of 2500 bytes, which corresponds to a transmission delay of 20 ms for each data packet. The size of a control packet is 62.5 bytes, which corresponds to a transmission delay of 0.5 ms for each control packet. During a simulation run each node generates data packets with the number of data packets generated per second being a binomial random variable with parameters $(10, p)$. The arrival probability p varies between 0.1 and 1 in simulations. This implies that the packet arrival rate in each node is between 1 and 10 packets per second. Each simulation runs 300 seconds, and we consider the following two performance metrics.

1. **Packet Average Delay:** The packet average delay is the average duration between the time a data packet is generated and the time of reception by its destination. The packet delay of a data packet is the sum of waiting time of the packet in its source node and the transmission time, which is assumed to be a constant of 20 ms in this study.
2. **Packet Delivery Ratio:** The packet delivery ratio is defined as the ratio of the total number of packets received by all network nodes to the total number of packets generated by them.

4.2 Simulation Results

In Fig. 8 we plot the packet average delay results when packet arrival rate varies from 1 to 10 and the network size N is fixed as 16. The solid line segments correspond to the simulation results of our proposed cooperative MAC scheduling scheme when multiuser detection is enabled in network nodes, while the dashed line segments correspond to the results when network nodes are not capable of multiuser detection. As we expect, the packet average delay increases with the packet arrival rate. This is because, on average, a data packet will wait longer for service in its source node for a larger packet arrival rate. Meanwhile, it is observed from Fig. 8 that the packet average delay results for MUD enabled networks are always smaller than these for networks without MUD capability. This is because multiple transmitters can simultaneously transmit packets to a same receiver in a MUD enabled network, which in turn results in a reduced packet waiting time in the source node. Furthermore, the larger the value of the packet arrival rate, the larger the gap between the MUD enabled packet delay results and the delay results without MUD capability. When the packet arrival rate becomes larger, there are more chances that more than one node have data packets destined to a same receiver at the same time, and thus the MUD capability can be more frequently utilized in the network. This shows that our proposed cooperative MAC scheduling can be more beneficial for MUD CDMA-MANETs with heavier traffic.

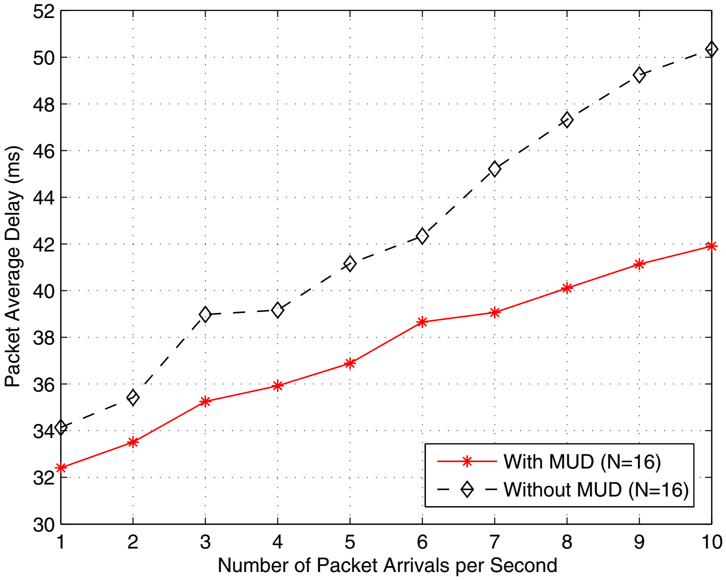

Fig. 8. Packet Average Delay versus Arrival Probability p

Fig. 9 plots the packet average delay results for the network size varying from 4 to 24 and p set to 0.4. The bottom curve in the figure corresponds to the network with MUD while the top curve corresponds to the network without MUD. It is observed that the packet average delay increases with the network size. For a network with a fixed size N, the packet average delay with MUD is smaller than that without MUD. The gap between these two results is more or less the same for different network sizes. This shows that our proposed cooperative MAC scheduling is suitable for MUD CDMA-MANETs of a large network size.

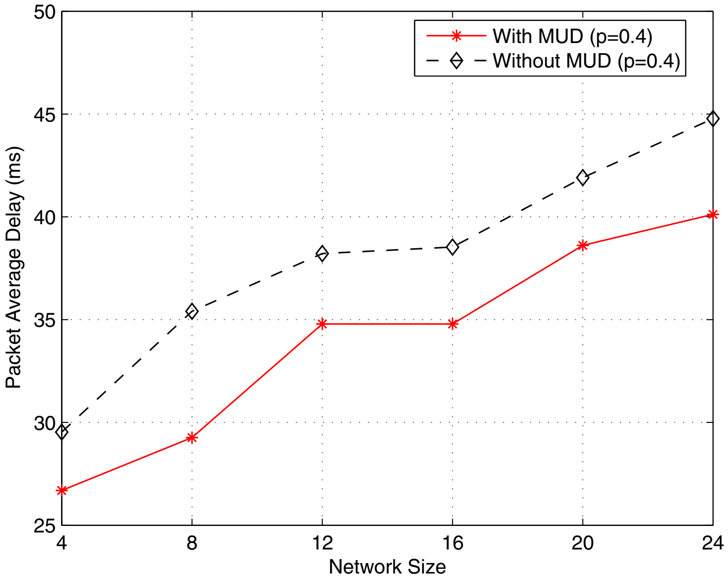

Fig. 9. Packet Average Delay versus Network Size

We plot the simulation results of the packet delivery ratio in Fig. 10 for the packet arrival rate varying from 1 to 10 and N equal to 16. There are two reasons which cause a data packet to not be received by its destination. One reason is due to changes in the network topology. If the channel between a packet's source node and its destination is down when the packet is to be transmitted by the source node, the packet will not be received by the destination node. The other reason is due to longer waiting times of packets in their source nodes. At the end of a simulation run, if a data packet, which has been generated before that time, has not been transmitted due to too many packets waiting for transmission, the packet will not be counted in the total number of received packets. In this simulation work, the ratio of the undelivered packets caused by the first reason is the same for both MUD-enabled and -disabled networks. Then a major difference of the packet delivery ratio between MUD-enabled and -disabled networks results from the second cause. It is observed from Fig. 10 that the packet delivery ratio in a network with MUD is always larger than that in a network without MUD as a consequence of a smaller packet average delay in the MUD enabled network. The simulation results of the packet delivery ratio are plotted in Fig. 11 when the network size varies from 4 to 24 and p is set to 0.4. We observe that the packet delivery ratio with MUD is never larger than that without MUD for each network size value.

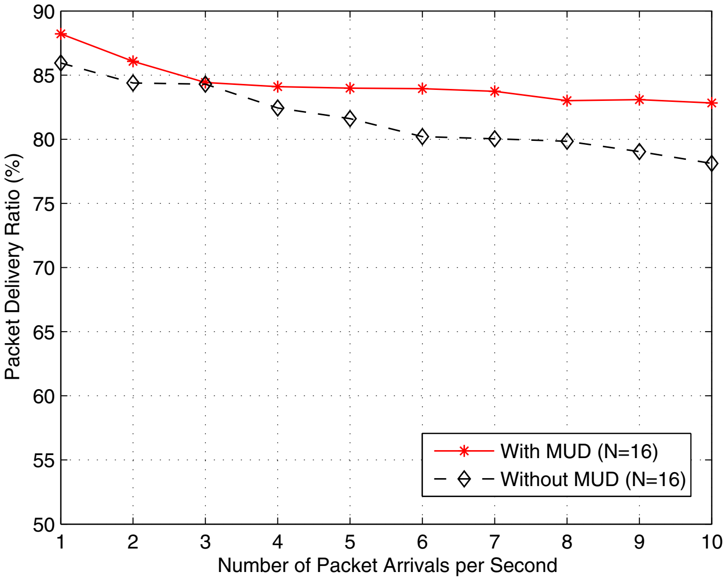

Fig. 10. Packet Delivery Ratio versus Arrival Probability p

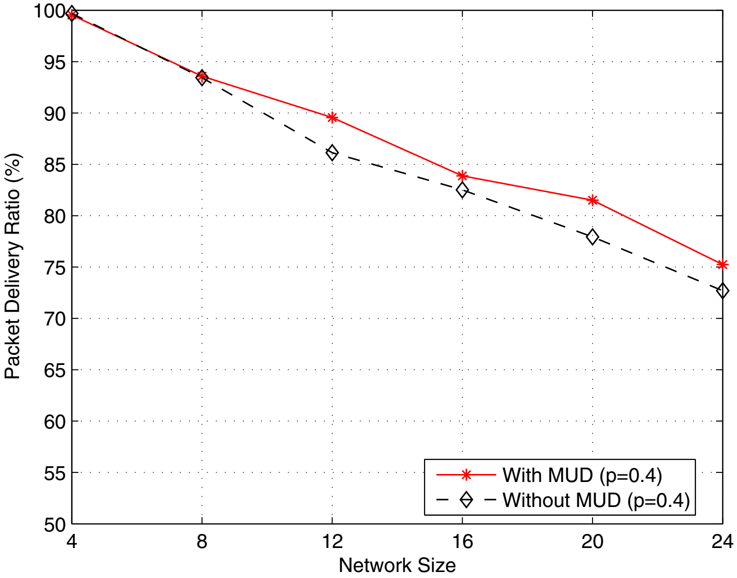

Fig. 11. Packet Delivery Ratio versus Network Size

5 Conclusion

In this paper, we proposed a MAC scheduling scheme that cooperatively schedules transmission of data packets in CDMA-MANETs with multiuser detection. Different from commonly used MAC schemes in wireless networks, the proposed cooperative MAC scheduling belongs to the group of non-contention based ones. By continuously circulating the token around the network, multiuser detection capability of the network can be fully utilized by each network node, for which multiple transmitters are allowed to transmit data to a same receiver at the same time. Simulation results are presented to demonstrate the performance of our proposed cooperative MAC scheduling scheme. It is observed from simulation results that the packet average delay increases with either network size or the intensity of network traffic. More importantly, it is shown that our proposed cooperative MAC scheduling is appropriate for MUD CDMA-MANETs with intense network traffic and possibly a large network size.

References

1. IEEE Standards Board. IEEE 802.5 - Token ring access method and physical layer specifications (March 1985)
2. IEEE Standards Board. IEEE 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications (August 1999)
3. Al-Meshhadany, T., Ajib, W.: New CDMA-based MAC protocol for ad hoc networks. In: Proc. of the IEEE 66th Vehicular Technology Conference (VTC 2007-Fall), pp. 91–95 (September 2007)
4. Comaniciu, C., Poor, H.V.: On the capacity of mobile ad hoc networks with delay constraints. IEEE Transactions on Wireless Communications 5(8) (August 2006)
5. Liu, I.S., Takawira, F., Xu, H.J.: A hybrid token-CDMA MAC protocol for wireless ad hoc networks. IEEE Transactions on Mobile Computing 7(5), 557–569 (2008)
6. Moon, Y., Syrotiuk, V.R.: A cooperative CDMA-based multi-channel MAC protocol for mobile ad hoc networks. Computer Communications 32(17), 1810–1819 (2009)
7. Moshavi, S.: Multi-user detection for DS-CDMA communications. IEEE Communications Magazine 34(10), 124–136 (1996)
8. Muqattash, A., Krunz, M.: CDMA-based MAC protocol for wireless ad hoc networks. In: Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2003), pp. 153–164 (June 2003)
9. Naqvi, S.H., Patnaik, L.M.: A medium access protocol exploiting multiuser-detection in CDMA ad-hoc networks. Wireless Networks 16(6), 1723–1737 (2010)
10. Nie, N., Comaniciu, C.: Energy efficient AODV routing in CDMA ad hoc networks using beamforming. In: Proc. of the IEEE 61th Vehicular Technology Conference (VTC 2005-Spring), vol. 4, pp. 2449–2453 (May 2005)
11. Qian, X.C., Zheng, B.Y., Yan, Z.Y., Yu, G.J.: Algorithm and application of multiuser detection for CDMA-based MANET. Journal of Shanghai University (English Edition) 11(2), 148–152 (2007)
12. Sankaran, C., Ephremides, A.: The use of multiuser detectors for multicasting in wireless ad hoc CDMA networks. IEEE Transactions on Information Theory 48(11), 2873–2887 (2002)

13. Su, Y.S., Su, S.L., Li, J.S.: Receiver-initiated multiple access protocols for spread spectrum mobile ad hoc networks. Computer Communications 28(10), 1251–1265 (2005)
14. Xiao, L.: CDMA bus lane: A cross-layer protocol for QoS routing in CDMA based mobile ad hoc networks. Ph.D. thesis, Department of Electronic Engineering, Queen Mary, University of London (September 2007)
15. Yi, J.: A survey on the applications of MANET. Technical report, `http://www.jiaziyi.com/documents/20080229_A_Survey_on_the_Applications_of_MANET.pdf`
16. Zhang, J., Dziong, Z., Kadoch, M., Gagnon, F.: Enhanced broadcasting and code assignment in multihop mobile ad hoc networks. In: Proc. of the 11th World Multi-Conference on Systemics, Sybernetics and Informatics (WMSCI 2007) (July 2007)
17. Zhang, J., Dziong, Z., Gagnon, F., Kadoch, M.: Receiver initiated MAC design for ad hoc networks based on multiuser detection. In: Proc. of the 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine 2008) (July 2008)
18. Zhang, J., Dziong, Z., Gagnon, F., Kadoch, M.: Multiuser detection based MAC design for ad hoc networks. IEEE Transactions on Wireless Communications 8(4), 1836–1846 (2009)