Asynchronous Multichannel MAC Design With Difference-Set-Based Hopping Sequences

Fen Hou, *Member, IEEE*, Lin X. Cai, *Member, IEEE*, Xuemin (Sherman) Shen, *Fellow, IEEE*, and Jianwei Huang, *Member, IEEE*

***Abstract*—Most existing multichannel medium access control (MAC) protocols have at least one of the following four performance bottlenecks: 1) global synchronization among users; 2) dedicated control channel for signaling exchange; 3) dedicated control phase for signaling exchange; and 4) complete knowledge of all users' channel selection strategies. In this paper, we first design a hopping sequence by combining multiple difference sets to ensure a high rendezvous probability of users over multiple channels. Applying the hopping sequence to all users, we then propose a difference-set-based multichannel MAC (DSMMAC) protocol to overcome the performance bottlenecks. Because all users use the same sequence for frequency hopping and channel access, significant signaling overheads can be reduced. The proposed protocol achieves high system throughput and low access delay without the need for global synchronization or a dedicated control channel/phase. Our analytical and simulation results show that the proposed DSMMAC protocol can achieve up to a 100% improvement in system throughput and a 150% reduction in channel access delay compared with an existing multichannel MAC protocol.**

***Index Terms*—Asynchronous, difference set, medium access control (MAC) , multichannel.**

I. Introduction

SUPPORTING successful concurrent transmissions of several close-by wireless links in the same channel is often difficult due to mutual interference. Exploiting the multichannel capability is an efficient approach to make concurrent transmissions possible. It is a general recognition that any pair of channels with at least 25-MHz frequency spacing are nonoverlapping channels and can simultaneously be used without mutual interference [1]. Multichannel communication systems have widely been adopted to efficiently support concurrent transmissions in the frequency domain and achieve high network throughput (e.g., [2]–[5]). For example, three and 12 nonoverlapping channels are specified in the widely deployed IEEE 802.11b and 802.11a wireless local area networks, respectively. In IEEE 802.11p, a seven-channel structure is defined for intelligent transportation systems. The promising cognitive radio network is inherently a multichannel system, where available "spectrum holes" are opportunistically accessed by the secondary users. A key design challenge for these systems is to have efficient medium access control (MAC) protocols that can coordinate the behavior of users and achieve good system throughput.

In a multichannel system, different users can use different channels to avoid excessive mutual interference. To perform the data transmission, the sender and receiver nodes of a source–destination (S–D) pair must successfully negotiate with each other beforehand and agree on a common channel to operate. However, it is quite challenging to ensure simple, robust, and efficient negotiations between S–D pairs, particularly for large-scale wireless networks. Most of the existing multichannel MAC protocols try to reach this goal based on one or more of the following assumptions:

- the existence of a dedicated common control channel for signaling exchange;
- the existence of a dedicated control phase (time period) to exchange signaling information;
- the availability of global clock synchronization among all users;
- every user's channel selection strategy (e.g., hopping sequence), which is known by all other users in the network.

For a system with only a few channels available, using a dedicated control channel will significantly limit the available resource and, thus, the network performance. In addition, heavy control traffic may make the control channel the performance bottleneck for the whole network. Furthermore, it is not always possible to assign a dedicated control channel if prior coordination among users is not possible (e.g., in cognitive radio networks). The global synchronization can help define the dedicated control phase, as shown in [16], but it is often difficult to achieve in large distributed networks and will also introduce additional implementation complexity.

To achieve the efficient S–D negotiation and avoid the global time synchronization and dedicated control channel, we introduce a difference-set-based frequency hopping in this paper. Difference set has widely been used in the design of power-saving protocols to reduce the wake-up duty cycle while

Manuscript received July 11, 2010; revised December 07, 2010; accepted January 21, 2011. Date of publication February 24, 2011; date of current version May 16, 2011. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under a discovery grant and by the General Research Funds through Projects 412308 and 412509, established under the University Grant Committee of Hong Kong. The review of this paper was coordinated by Dr. T. Taleb.

F. Hou and J. Huang are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: fhou@ie.cuhk.edu.hk; jwhuang@ie.cuhk.edu.hk).

L. X. Cai is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail:lincai@princeton.edu).

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: xshen@bbcr.uwaterloo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2011.2119384

maintaining a probability of communicating with neighboring nodes [6], [7]. Different from sleep-scheduling algorithms, which aim to minimize the energy consumption, our objective is to achieve high channel utilization and network throughput in a multiple-channel system. Recently, [8] has proposed to apply quorum-based channel hopping (QCH) in a synchronous multichannel system for control channel establishment. An asynchronous QCH scheme for a two-channel system is also introduced. To the best of our knowledge, asynchronous distributed MAC design in a system with more than two channels is still an open issue. Moreover, with more channels, how difference-set-based sequence design can be applied to ensure efficient, robust, and fair channel access among multiple users becomes more challenging.

In this paper, we propose a novel difference-set-based asynchronous multichannel medium access control (DSMMAC) protocol for distributed wireless networks without control channels. The unique rotation closure property of difference sets enables any two nodes to communicate (i.e., meet or rendezvous) with a nonzero probability without global synchronization or a dedicated control channel (the detailed discussion on the rendezvous probability is given in Section III). By combining multiple disjoint difference sets, we construct a hopping sequence that is suitable for multichannel networks. Aside from the performance metric of time to rendezvous, we are also interested in network throughput, channel utilization, channel access fairness, and power consumption. In particular, the main advantages of the proposed DSMMAC protocol are listed as follows.

1) *Asynchronization*. DSMMAC does not require global synchronization among users and, thus, is suitable for implementation in distributed networks.
2) *Multiple parallel rendezvous*. Multiple S–D pairs can simultaneously rendezvous and communicate over different channels without a centralized coordination. This condition improves the channel utilization and system performance.
3) *Less signaling overhead*. All users deploy the same hopping sequence in DSMMAC, which eliminates the need for exchanging hopping sequence information and reduces the signaling overhead.
4) *No dedicated control channel*. Because DSMMAC requires no dedicated common control channel, it can fully utilize all available channels for data transmission.

The remainder of this paper is organized as follows. Section II describes the network model. The difference-set-based hopping sequence is discussed in Section III, followed by the proposed DSMMAC protocol in Section IV . An analytical model is developed in Section V. Simulation results are presented in Section VI. Section VII presents the related work. We finally conclude in Section VIII.

II. Network Model

We consider a multichannel network that consists of sets $\mathcal{C} = \{c_1, \ldots, c_L\}$ of L channels and $\mathcal{N} = \{n_1, \ldots, n_{2M}\}$ of $2M$ nodes. One example is shown in Fig. 1, where $L = 4, M = 4$. Each node represents a user, which can be a mobile phone, a

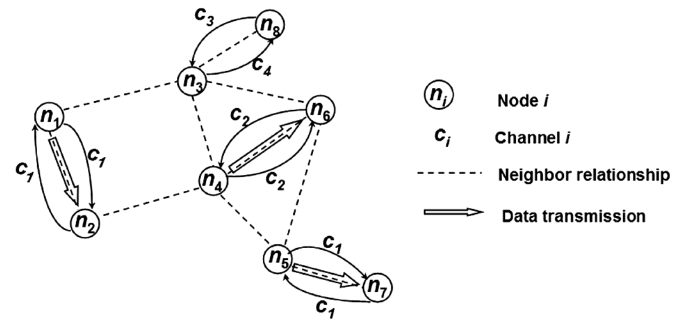


Fig. 1. Generic multichannel wireless network with eight nodes and four channels.

mobile personal digital assistant (PDA), a laptop, or a sensor. Two nodes that want to communicate with each other are called an S–D pair. Each node is equipped with one tunable half-duplex radio transceiver, which can switch between different channels. Whether an S–D pair can communicate depends on their selected working channels. As shown in Fig. 1, the nodes n_3 and n_8 turn their working channels to c_4 and c_3, respectively. The destination node n_8 cannot listen to the information transmitted by the source node n_3 over the channel c_4. Therefore, the S–D pair $n_3 - n_8$ cannot communicate with each other. An S–D pair can communicate only when both the source and the destination nodes simultaneously turn to the same channel (i.e., they meet or rendezvous over a channel). As shown in Fig. 1, the S–D pair $n_4 - n_6$ select the same channel c_2. They can conduct the data transmission after the successful negotiation over the channel c_2.

We adopt the carrier-sense multiple access (CSMA) with a ready-to-send/clear-to-send (RTS/CTS) access control mechanism in our system model. All nodes have the same transmission and interference ranges. Multiple S–D pairs can simultaneously communicate over the same channel if the interferers do not cause harmful interference to the ongoing transmissions, i.e., they are outside the interference range of the transmitting S–D pairs. As shown in Fig. 1, two S–D pairs (i.e., $n_1 - n_2$ and $n_5 - n_7$) can simultaneously communicate over channel c_1, because these two pairs are far from, and do not interfere with each other. In a multichannel system, multiple S–D pairs can concurrently communicate over different channels due to the channel orthogonality.

To make an S–D pair rendezvous, the source and destination independently hop over channels based on a predefined hopping sequence. Detailed hopping sequence and multichannel MAC design will be presented in Sections III and IV.

III. Difference-Set-Based Hopping Sequence

The key idea behind our proposed multichannel MAC protocol is to design the hopping sequence based on the concept of difference sets. Different from the use of difference sets in the design of power-saving sleep scheduling algorithms [9], where the main objective is to minimize a node's wake-up duty cycle, our objective is to maximize the rendezvous probability of S–D pairs. Moreover, we need to consider the design in both frequency and time domains to guarantee a successful rendezvous.

In the following sections, we first review the properties of difference sets [10], [11]. Then, we present the design of difference-set-based hopping sequence for a multichannel system.

A. Definitions and Properties of Difference Sets

Definition 1—Difference Set: A difference set is defined by the following three elements: 1) the set cycle v; 2) the set size k; and 3) the time number λ. A (v, k, λ) difference set includes k integers, denoted by $\mathcal{D} = \{d_1, \ldots, d_k\}$. For each value r that is smaller than v (i.e., $r \in \{1, \ldots, v-1\}$), we can find exact λ ordered pairs of (d_i, d_j) in the set such that the difference is r (i.e., $r = (d_i - d_j) \mod v$, where $d_i, d_j \in \mathcal{D}$).

For example, the set $\{1, 2, 4\}$ is a (7, 3, 1) difference set. There are three elements in this set, with a cycle of 7. For any integer $r \in \{1, 2, \ldots, 6\}$, we can find exactly one ordered pair of elements in the set, which gives the difference of r. For example, the ordered pair (1, 2) gives a difference of 6 (i.e., $6 = (1-2) \bmod 7$), and the ordered pair (2,1) gives a difference of 1.

Definition 2—Complementary Set of a Difference Set: Let $\mathcal{D} = \{d_1, \ldots, d_k\}$ be a difference set with a cycle of v. The complementary set of $\mathcal{D}$ is defined with respect to the set of $\mathcal{Z} = \{1, 2, \ldots, v\}$, i.e.,

$$\overline{\mathcal{D}} = \{l : l \in \mathcal{Z} \setminus \mathcal{D}\}.$$

Take the set $\{1, 2, 4\}$ as an example. It is a difference set with a cycle of 7. Therefore, its complementary set is $\{3, 5, 6, 7\}$.

Definition 3—Shift Set of a Difference Set: Let $\mathcal{D} = \{d_1, \ldots, d_k\}$ be a difference set with a cycle of v. Its μth shift set is defined as $\mathcal{D}^{\mu} = \{d_1 + \mu, \ldots, (d_k + \mu)(\bmod v)\}$, $\mu \in \{1, \ldots, v\}$.

The shift set of a difference set is generated by conducting a shift operation with the mod of v. For instance, the first shift set of the difference set $\{1, 2, 4\}$ is $\{2, 3, 5\}$.

Property 1: The complementary set of a difference set is a difference set. If $\mathcal{D}$ is a difference set with a cycle of v, then its complementary set $\overline{\mathcal{D}}$ is also a difference set with a cycle of v.

We can use this property to generate multiple disjoint difference sets with the same cycle.

Property 2—Rotation Closure Property: A (v, k, λ) difference set $\mathcal{D}$ and any of its shift sets have λ overlapping elements in a cycle of v.

When applying the rotation closure property to the design of hopping sequence, two nodes that use the same difference set in either a synchronous or an asynchronous fashion will have a rendezvous probability λ/v without prior coordination. Examples are shown in Fig. 2 and are further elaborated as follows.

Fig. 2 illustrates the rotation closure property in a single-channel case (e.g., channel c_1). Nodes A and B follow the same hopping sequence generated from a $(11, 5, 2)$ difference set $\mathcal{D} = \{1, 3, 4, 5, 9\}$. In a cycle of 11 time slots, nodes hop over channel c_1 in their first, third, fourth, fifth, and ninth time slots, which are depicted as solid rectangles. An empty rectangle represents an inactive time slot, in which users switch off their transceivers.

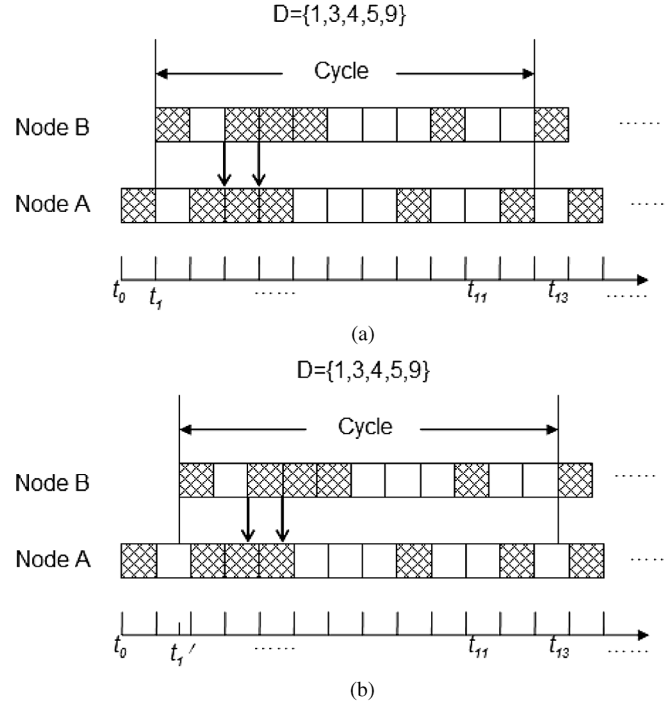


Fig. 2. Rotation closure property for the difference set $D = \{1, 3, 4, 5, 9\}$. (a) Aligned boundaries of time slots. (b) Nonaligned boundaries of time slots.

- *Synchronization case*. As shown in Fig. 2(a), nodes A and B have aligned time slot boundaries. They start to hop to channel c_1 at times t_0 and t_1, respectively, based on the same hopping sequence but with different time shifts. Let us consider a cycle of 11 time slots. We can see that two nodes rendezvous on the channel c_1 twice (the beginnings of node B's time slots 3 and 4). The long-term rendezvous probability is 2/11.
- Asynchronization case. As shown in Fig. 2(b), the time slot boundaries are not aligned. Nodes A and B join the network and hop to channel c_1 at times t_0 and t_1', respectively. Within a cycle of 11 time slots, two nodes rendezvous on the channel c_1 twice (at the beginning of node B's time slots 3 and 4). The long-term rendezvous probability is still 2/11.

The aforementioned example illustrates that the rendezvous probability λ/v is independent of synchronization.

B. Design of the Difference-Set-Based Hopping Sequence

A hopping sequence determines which channel a user should access during each time slot of a cycle. In the proposed DSMMAC protocol, all users use the same frequency-hopping sequence so that users do not need to exchange hopping sequences with each other. The hopping sequence is designed with multiple difference sets to support communications over multiple channels. In addition, it is desirable to achieve a high rendezvous probability of each S–D pair so that they have a higher chance of successfully communicating with each other. Moreover, we will try to support several concurrent transmissions to exploit the capability of multichannel networks. The main design steps are given as follows.

Step 1—Select Multiple Difference Sets: To guarantee successful node rendezvous in both time and frequency domains, we need to combine multiple difference sets with the *same* cycle to generate a hopping sequence that covers all channels. That is, for a network with L available channels, we will choose L difference sets with the same cycle v. In other words, we combine L difference sets for channel access in the frequency domain, and each difference set has a cycle v that ensures the rendezvous probability λ/v over each channel. We propose two criteria for the sequence design as follows.

Criterion 1—High Rendezvous Probability: We select difference sets with high rendezvous probabilities (i.e., large values of λ/v) to ensure that users are very likely to meet with each other over each channel.

Criterion 2—Empty Intersection: Any two chosen difference sets (denoted as $\mathcal{D}_i$ and $\mathcal{D}_j$ $(i \neq j)$) should satisfy

$$\mathcal{D}_i \bigcap \mathcal{D}_j = \emptyset \qquad \forall i, j \in \{1, 2, \ldots L\}.$$

In a multichannel network, two users can communicate when they hop on the same channel at the same time (i.e., meet in both frequency and time domains). Criterion 1 ensures that nodes meet with high probabilities. Criterion 2 ensures that there is no ambiguity when assigning channels to time slots in a hopping sequence. One numerical example will clearly illustrate this point.

Step 2—Match the Frequency Channels With the Selected Difference Sets: Let $\mathcal{D}_1, \mathcal{D}_2, \ldots$ and $\mathcal{D}_L$ be L difference sets that are selected for a network with L available channels (denoted as $c_1, c_2, \ldots, c_L$). We will associate one channel with each difference set and allow access of this channel only during the time slots of the corresponding difference set. For instance, $D_1 = \{1, 3, 6\}$ means that the node will only access channel c_1 during time slots 1, 3, and 6 of a cycle.

Step 3—Randomly Allocate the Frequency Channels to the Remaining Unassigned Time Slots: It is possible that some time slots remain unassigned after step 2 if the total number of elements of all L different sets is smaller than v. In other words, let $\Psi = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_L\}$ be the set of selected difference sets and $\overline{\Psi}$ be the complementary set of Ψ in $\{1, 2, \ldots, v\}$. If $\overline{\Psi} \neq \emptyset$, we randomly assign channels to the time slots in $\overline{\Psi}$ for easy implementation.

As one simple example, we first elaborate the hopping sequence design for a network with $L = 2$ channels (denoted as c_1 and c_2). In this case, two difference sets with the same cycle are needed. By applying Property 1 of the difference set, we choose difference set $\{1, 2, 4\}$ and its complementary difference set $\{3, 5, 6, 7\}$ (i.e., $\mathcal{D}_1 = \{1, 2, 4\}$ and $\mathcal{D}_2 = \{3, 5, 6, 7\}$). Both difference sets have the same cycle of seven time slots. The rendezvous probabilities of the two difference sets are 1/7 and 2/7, respectively. Moreover, these two difference sets do not intersect with each other (i.e., criterion 2 in design step 1). We allocate the frequency-hopping channels c_1 and c_2 based on these two difference sets as follows. Channel c_1 is allocated to the time slots of $\mathcal{D}_1$ (i.e., time slots 1, 2, and 4), and channel c_2 is allocated to the time slots of $\mathcal{D}_2$ (i.e., time slots 3, 5, 6, and 7).

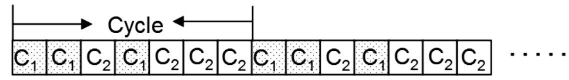


Fig. 3. Hopping sequence for two channels based on two difference sets.

The resulting hopping sequence is $\mathrm{H} = [c_1\ c_1\ c_2\ c_1\ c_2\ c_2\ c_2]$, as shown in Fig. 3.

As a more complicated example, we consider the hopping sequence design for eight channels (which is denoted as c_1–c_8). We first choose eight difference sets as follows:

$$\begin{aligned}\Psi &= \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5, \mathcal{D}_6, \mathcal{D}_7, \mathcal{D}_8\}\\ &= \{\{2\ 3\ 5\ 9\ 17\ 33\ 38\ 56\ 65\}\\ &\quad \{4\ 7\ 13\ 20\ 24\ 25\ 39\ 47\ 49\}\\ &\quad \{6\ 8\ 11\ 15\ 21\ 29\ 40\ 41\ 57\}\\ &\quad \{10\ 19\ 37\ 42\ 58\ 66\ 70\ 72\ 73\}\\ &\quad \{12\ 16\ 22\ 23\ 31\ 43\ 45\ 48\ 61\}\\ &\quad \{14\ 27\ 30\ 32\ 44\ 52\ 53\ 59\ 63\}\\ &\quad \{18\ 34\ 35\ 46\ 54\ 60\ 64\ 67\ 69\}\\ &\quad \{26\ 28\ 36\ 50\ 51\ 55\ 62\ 68\ 71\}\}.\end{aligned}$$

These eight difference sets have the same cycle (i.e., $v = 73$ time slots) and empty intersections (i.e., criterion 2). The frequency-hopping channels are allocated based on the difference sets. For instance, at the time slots that correspond to $\mathcal{D}_1$ (i.e., the time slots 2, 3, 5, 9, 17, 33, 38, 56, and 65), the frequency-hopping channel is c_1. After allocating channels based on all eight difference sets, we find that time slot 1 remains unassigned (i.e., $\overline{\Psi} = \{1\}$). Therefore, we randomly assign one frequency channel to time slot 1. Without loss of generality, we allocate channel c_1 to time slot 1. The final hopping sequence is $\mathrm{H} = [c_1\ c_1\ c_1\ c_2\ c_1\ c_3\ c_2\ c_3\ c_1\ c_4\ c_3\ c_5\ c_2\ c_6\ c_3\ c_5\ c_1\ c_7\ c_4\ c_2\ c_3\ c_5\ c_5\ c_2\ c_2\ c_8\ c_6\ c_8\ c_3\ c_6\ c_5\ c_6\ c_1\ c_7\ c_7\ c_8\ c_4\ c_1\ c_2\ c_3\ c_3\ c_4\ c_5\ c_6\ c_5\ c_7\ c_2\ c_5\ c_2\ c_8\ c_8\ c_6\ c_6\ c_7\ c_8\ c_1\ c_3\ c_4\ c_6\ c_7\ c_5\ c_8\ c_6\ c_7\ c_1\ c_4\ c_7\ c_8\ c_7\ c_4\ c_8\ c_4\ c_4]$.

IV. Difference-Set-Based Multichannel Medium Access Control Design

The key idea of our proposed DSMMAC protocol is to design a smart difference-set-based hopping sequence. Then, all nodes use this sequence for channel hopping and channel access as follows. When a source node joins the network, it randomly selects an initial channel and attempts to access the channel by starting a handshake process. If the source node successfully exchanges handshake information with its intended destination node, the S–D pair stops hopping and starts data transmissions over this channel. Otherwise, the source node selects the next hopping channel in the following time slot according to the predesigned hopping sequence. The node keeps channel hopping until it successfully shakes hands with its destination.

The proposed DSMMAC has several advantages. First, all nodes use the same hopping sequence that is predesigned (e.g., by the manufacturer or the service provider); thus, there is no need for further information exchange, and no central controller or centralized allocation is needed. Second, the unique rotation closure property of difference set ensures that a source node can successfully meet its destination with a certain probability, even without synchronization. Third, although all nodes use a common hopping sequence, nodes join the network at different time instants and initiate channel accesses over different channels, which helps achieve multiple parallel rendezvous and improves the channel utilization. The detailed algorithms of the source and destination nodes are presented in Algorithms 1 and 2, respectively.

When a node joins the network, it randomly selects a hopping channel and starts channel sensing. If the channel is sensed busy, which implies that the current channel is occupied by other S–D pairs, the source node will switch to the next hopping channel based on the hopping sequence and resume the channel sensing at the next time slot. If the channel is sensed idle for a distributed coordination function interframe space (DIFS) interval, implying that currently there are currently no ongoing transmissions over the channel, the source node initiates an RTS transmission. If no CTS is received after a short interframe space (SIFS) interval, it continues channel hopping. Notice that it is also possible that the transmitted RTS/CTS is lost in the error-prone wireless channel or due to hidden terminal problems in a multihop network. If the channel condition between an S–D pair is poor, it is not desirable to start data transmission in the first place. In addition, with multiple channels operating in parallel, the collision probability on any of these channels is greatly reduced. Therefore, when a source node sends an RTS but receives no CTS after an SIFS, most likely, the corresponding destination does not access the same channel at this moment. Thus, the source node should switch to a different channel. If a CTS is received after an SIFS, which means that the corresponding destination also accesses the channel at this time, after the successful handshake, the source node starts data transmissions to its destination.

To allow multiple S–D pairs to fairly share the wireless medium, users release the channel when a predefined time duration $T_{\max}$ is reached or its data buffer becomes empty. After releasing the current channel, a node resumes channel hopping based on the hopping sequence until there is a successful handshake. Because of the rotation closure property of difference sets, any two users can meet each other on one channel with a certain probability, regardless of the initial choice of hopping channels and channel synchronization.

Algorithm 1: Source node

```
Randomly select a hopping channel;
while There are data to send do
    Sense the channel;
    if Channel is idle for a DIFS then
        Send an RTS;
        if A CTS is received within an SIFS then
            while Channel occupation time ≤ T_max and data buffer is not empty do
                Transmit data over the channel;
            end while
            Release the channel;
        end if
    end if
    Hop to the next channel based on the designed hopping sequence;
end while
```

Algorithm 2: Destination node

```
Randomly select a hopping channel;
Sense the channel;
if Channel is idle then
    Wait a time slot for an RTS;
    if Receive an RTS targeted to it then
        Respond with a CTS;
        if Receive data within a SIFS then
            repeat
                Receive data
            until Data transmissions complete
        end if
    end if
end if
Hop to the next channel based on the designed hopping sequence;
```

As shown in Algorithm 2, the operation of the destination node is similar. The main difference is that a destination node will wait for receiving a potential RTS for a time slot and respond with a CTS.

In the proposed MAC protocol, frequency hopping is only used to find the rendezvous channel. After a successful rendezvous has occurred between an S–D pair, they stop the frequency hopping and conduct their data transmission over the common channel. In other words, data transmission is not based on the frequency-hopping technique but over a single channel. Therefore, the proposed protocol has much less signaling overhead than the traditional frequency-hopping-technique-based MAC protocol (i.e., frequency-hopping spread spectrum in IEEE 802.11). In the latter case, data are sent over a sequence of the hopping channels that corresponds to an agreed hopping *pattern* between an S–D pair.

TABLE I
TABLE OF NOTATIONS

L	the total number of channels in the system
c_i	The i^{th} channel in the system
M	the total number of S-D pairs
H	Hopping sequence
h_i	The i^{th} hopping channel in H
$\boldsymbol{s}_i$	system state i
ϕ_{ij}^{f}	the set of channels that finish data transmission at the system transition from the state i to j
ϕ_{ij}^{b}	the set of channels that begin data transmission at the system transition from the state i to j
T	Mean of each data transmission (in unit of time slot)
q	Probability of finishing data transmissions at a time slot
$Pr[F_{ij}]$	Probability of finishing the transmissions over all channels in ϕ_{ij}^{f}
$Pr[B_{ij}]$	Probability of beginning the transmissions over all channels in ϕ_{ij}^{b}
$Pr[W_i^m]$	Probability that the m^{th} S-D pair fails to negotiate given system state i
$Pr[B_i^{m,l}]$	Probability that the m^{th} S-D pair begins data transmission over channel l given system state i
μ	Channel utilization
r	Data transmission rate
Th	System throughput
Th_m	Achieved throughput of the m^{th} S-D pair

V. PERFORMANCE ANALYSIS

We analyze the performance of the proposed DSMMAC in terms of network throughput and channel utilization. The main notations that are used in the paper are listed in Table I.

We divide time into small "virtual" slots, denoted as time slot t $(t = 1, 2, \ldots)$. Notice that the time slots are introduced here for analytical convenience. In practical implementations, each node keeps a local time-slotted system, and the boundaries of time slots are not necessarily synchronized across users. The only requirement is that the length of a time slot is the same for all users. We observe the system at each time slot and assume that system-state transition occurs only at the beginning of each time slot. Denote a system state, e.g., system state i, as $\boldsymbol{s}_i \in \boldsymbol{s}$, which consists of a set of channels used for data transmissions at this time slot.

The sequence of observed system states then forms a discrete Markov chain. Without loss of generality, in system state i, $\boldsymbol{s}_i = \{c_i^1, c_i^2, \ldots, c_i^I\}$, where $c_i^1, \ldots, c_i^I$ represent the channels that are involved in the data transmissions at this state. For instance, $\boldsymbol{s}_{10} = \{c_1, c_2, c_8\}$ means that channels c_1, c_2, and c_8 are used for active data transmission, and this system state is labeled as state 10 in the Markov chain. The labeling of system state is simply a mapping from the system state to an integer. For instance, for a network with two available channels, the possible system states are $\emptyset$, $\{c_1\}$, $\{c_2\}$, and $\{c_1, c_2\}$. These four system states can be labeled as states 1, 2, 3, and 4, respectively.

The one-step transition probability from states i to j is defined as

$$p_{ij} = \Pr\left[s(t+1) = \boldsymbol{s}_j | s(t) = \boldsymbol{s}_i\right] \tag{1}$$

where $\boldsymbol{s}_i = \{c_i^1, \ldots, c_i^I\}$, and $\boldsymbol{s}_j = \{c_j^1, \ldots, c_j^J\}$. Here, I and J represent the total number of active data transmission channels at states i and j, respectively. The value of p_{ij} is independent of the time index t.

Additional notations are needed to derive the values of transition probability p_{ij}. Let ϕ_{ij}^f and ϕ_{ij}^b denote the sets of channels that finish and begin data transmissions at the state transition from i to j, respectively. Therefore, the following two events occur during this state transition: 1) active S–D pairs over channels in ϕ_{ij}^f finish their transmissions and 2) S–D pairs begin new transmissions over channels in set ϕ_{ij}^b. For instance, given that $\boldsymbol{s}_i = \{c_1, c_3, c_4\}$ and $\boldsymbol{s}_j = \{c_1, c_5, c_8\}$, we have $\phi_{ij}^f = \{c_3, c_4\}$ and $\phi_{ij}^b = \{c_5, c_8\}$. Let $\Pr[F_{ij}]$ be the probability that data transmissions over channels in $|\phi_{ij}^f|$ finish and $\Pr[B_{ij}]$ be the probability that new data transmissions over channels in $|\phi_{ij}^b|$ begin. The one-step transition probability from states i to j is

$$p_{ij} = \Pr[F_{ij}]\Pr[B_{ij}]. \tag{2}$$

A. *Derivation of* $\Pr[F_{ij}]$

Let random variable X represent the duration of a data transmission. To simplify the analysis, in this paper, we assume that durations of all data transmissions follow the same exponential distribution with mean T. To formulate the state transition into a discrete Markov model, we convert the continuous random variable X to a discrete random variable $Y = \lceil X/T_{slot} \rceil$, where T_{slot} is the duration of a time slot, and $\lceil \cdot \rceil$ represents the largest integer that is less than or equal to the argument. Because X is an exponential random variable with mean T, X/T_{slot} also follows an exponential distribution with mean $\alpha = T/T_{slot}$, and the discrete random variable follows geometric distribution with parameter $q = (1 - e^{-1/\alpha})$. In other words, we observe the system at the beginning of each time slot, and each data transmission may finish or continue with probabilities $q = (1 - e^{-1/\alpha})$ and $1 - q = e^{-1/\alpha}$, respectively. The accuracy of this approximation will be demonstrated in Section VI.

During one-step transition from states i to j, the data transmissions over channels in ϕ_{ij}^f finish, whereas the data transmissions over other channels in $\boldsymbol{s}_i$ are still active. The probability of this event is

$$\Pr[F_{ij}] = q^{|\phi_{ij}^f|}(1-q)^{|\boldsymbol{s}_i| - |\phi_{ij}^f|} \tag{3}$$

where $|\cdot|$ represents the cardinality of a set.

B. Derivation of $\Pr[B_{ij}]$

Data transmission starts when an S–D pair successfully conducts rendezvous over one channel. In a synchronous network, the transmission boundaries of RTS signaling are aligned, which leads to a nonzero probability of concurrent transmissions over the same channel (i.e., collision). In this paper, we consider that S–D pairings are fixed and focus on the asynchronous case, where the probability that two or more sources attempt to simultaneously transmit over the same channel is negligible. In this case, the probability of a successful rendezvous only depends on the hopping sequence and the total number of S–D pairs that attempt to access the channel. Let $\Pr[B_i^{m,l}]$ be the probability that the mth S–D pair begins data transmission over channel c_l at the next time slot, given that the system is in state i at the current time slot. It equals the probability of successful rendezvous over channel c_l for an S–D pair.

The value of $\Pr[B_i^{m,l}]$ depends on how the source and the destination nodes of the mth S–D pair select their initial hopping channels. Let η_s and η_d be the indices of initial hopping channels selected by the source and the destination, respectively. For instance, consider a hopping sequence $H = [h_1, \ldots, h_g, \ldots, h_v]$, where v is the cycle of the hopping sequence, and $h_g \in \{c_1, c_2, \ldots, c_L\}$ represents the gth hopping channel in the hopping sequence H. $\eta_s = 2$ means that the source node selects h_2 as its initial hopping channel. Thus, the probability of successful rendezvous over channel c_l is

$$\Pr[B_i^{m,l}]=\sum_{\gamma_1=1}^{v}\sum_{\gamma_2=1}^{v}\Pr[\eta_s=\gamma_1]\Pr[\eta_d=\gamma_2]Pr\left[B_i^{m,l}|\eta_s,\eta_d\right]. \tag{4}$$

Given η_s and η_d, the conditional probability $\Pr[B_i^{m,l}|\eta_s, \eta_d]$ depends on the hopping sequence H. As an example, assume that $\eta_s = 5$, $\eta_d = 1$, and $H = [c_1, c_1, c_2, c_1, c_2, c_2, c_2]$. The source and the destination nodes choose channels $h_5 = c_2$ and $h_1 = c_1$ as their initial hopping channels, respectively. Therefore, they will meet at the third time slot, in which both nodes hop to channel c_2. The conditional probability is $\Pr[B_i^{m,2}|\eta_s = 5, \eta_d = 1] = 1/3$.

Let $\mathcal{S}_i$ and $\overline{\mathcal{S}}_i$ be the set of active channels at the system state $\boldsymbol{s}_i$ and its complementary set in $\{c_1, c_2, \ldots, c_L\}$, respectively. Thus, the probability that the mth S–D pair fails to negotiate for a data transmission, given that the system is at state $\boldsymbol{s}_i$, is

$$Pr\left[W_i^m\right] = 1 - \sum_{l\in\overline{\mathcal{S}}_i} Pr\left[B_i^{m,l}\right]. \tag{5}$$

Let Ω represent the set of S–D pairs that attempt to access the channels during the transition from states i to j. We have $\Omega = M - |\boldsymbol{s}_i|$, where M is the total number of S–D pairs, and $|\boldsymbol{s}_i|$ is the number of active S–D pairs at state i. When the system transits to state j, there are $|\phi_{ij}^b|$ S–D pairs that successfully negotiate for their data transmissions over channels in ϕ_{ij}^b. There exist $\binom{M-|\boldsymbol{s}_i|}{|\phi_{ij}^b|}$ possible combinations of selecting $|\phi_{ij}^b|$ out of $M - |\boldsymbol{s}_i|$ S–D pairs. Let Ω_φ represent the set of S–D pairs that correspond to the φth combination and $\overline{\Omega}_\varphi$ be the complementary set of Ω_φ in Ω . Therefore, $\prod_{m\in\Omega_\varphi, m'\in\overline{\Omega}_\varphi, l\in\phi_{ij}^b}\Pr[B_i^{m,l}]\Pr[W_i^{m'}]$ is the probability that S–D pairs in Ω_φ successfully negotiate over the channels in ϕ_{ij}^b and that S–D pairs in $\overline{\Omega}_\varphi$ fail their negotiations. By summing over all possible combinations from $\varphi = 1$ to $\binom{M-|\boldsymbol{s}_i|}{|\phi_{ij}^b|}$, we obtain the probability that channels in ϕ_{ij}^b begin data transmissions as

$$\Pr[B_{ij}] = \sum_{\varphi=1}^{\binom{M-|\boldsymbol{s}_i|}{|\phi_{ij}^b|}} \prod_{m\in\Omega_\varphi, m'\in\overline{\Omega}_\varphi, l\in\phi_{ij}^b} Pr\left[B_i^{m,l}\right] Pr\left[W_i^{m'}\right]. \tag{6}$$

As an illustrative example, consider a total number of $L = 2$ channels denoted as c_1 and c_2. The number of S–D pairs is $M = 3$. The system states in two consecutive time slots are $\boldsymbol{s}(t) = \boldsymbol{s}_i = \{c_1\}$ and $\boldsymbol{s}(t+1) = \boldsymbol{s}_j = \{c_1, c_2\}$, respectively. Therefore, we have $\phi_{ij}^f = \emptyset$, $\phi_{ij}^b = \{c_2\}$, and $\Omega = \{1, 2\}$. Thus, $\Pr[B_{ij}]$ is obtained as

$$\begin{aligned}
\Pr[B_{ij}] &= \sum_{\varphi=1}^{\binom{2}{1}} \prod_{m\in\Omega_\varphi, m'\in\overline{\Omega}_\varphi, l\in\phi_{ij}^b} Pr\left[B_i^{m,l}\right] Pr\left[W_i^{m'}\right] \\
&= \prod_{m\in\{1\}, m'\in\{3\}, l\in\{c_2\}} Pr\left[B_i^{m,l}\right] Pr\left[W_i^{m'}\right] \\
&\quad + \prod_{m\in\{3\}, m'\in\{1\}, l\in\{c_2\}} Pr\left[B_i^{m,l}\right] Pr\left[W_i^{m'}\right] \\
&= Pr\left[B_i^{1,2}\right] Pr\left[W_i^3\right] + Pr\left[B_i^{3,2}\right] Pr\left[W_i^1\right].
\end{aligned} \tag{7}$$

C. System Throughput and Channel Utilization

Based on the derived probabilities of finishing and beginning data transmissions, the one-step transition probability matrix is

$$\begin{aligned}
\mathbf{P} &= [p_{ij}]_{N\times N} \quad i, j \in \Theta \\
p_{ij} &= \Pr[F_{ij}]\Pr[B_{ij}]
\end{aligned} \tag{8}$$

where Θ represents the state space of the Markov chain, and $N = |\Theta|$ is the total number of states in Θ.

Let π_i denote the steady-state probability that the system stays at state i. The system steady-state probability $\mathbf{\Pi} = [\pi_1, \pi_2, \ldots, \pi_N]$ can be calculated as

$$\mathbf{\Pi} = \boldsymbol{Q}\left[(\boldsymbol{I} - \boldsymbol{P})\mathbf{\Lambda} + \boldsymbol{U}\right]^{-1} \tag{9}$$

where $\boldsymbol{Q}$ is a $1 \times N$ zero vector, except that the last element is one, $\boldsymbol{I}$ is an $N \times N$ identity matrix, $\boldsymbol{P}$ is the one-step transition matrix, $\mathbf{\Lambda}$ is an $N\times N$ matrix, with the first $(N-1)$ elements of the diagonal set equal to 1 and other elements are zero, and $\boldsymbol{U}$ is an $N\times N$ zero matrix, except that all elements in the last column are ones. The notation $[\cdot]^{-1}$ represents the matrix inverse. Thus,

TABLE II
TABLE OF THE MAIN SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Channel number L	2, 8	SIFS	16 μs
S-D pair number M	$1 - 20$	DIFS	34 μs
PHY preamble	192 bits	RTS	20 bytes
Channel switch time	40 μs	CTS	14 bytes
Slot duration	418 μs	ACK	14 bytes
Transmission/interference range	100 m	RTS/CTS rate	2 Mbps
Mean of transmission duration T	$5 - 20$(slots)	Data rate r	54 Mbps

the average number of channels used for the data transmissions is

$$\overline{L} = \sum_{i \in \Theta} n_i \pi_i \tag{10}$$

where n_i is the number of channels used for the data transmission when the system is in state i.

The average system throughput is

$$Th = r. \sum_{i \in \Theta} n_i \pi_i \tag{11}$$

where r is the average data transmission rate of an S–D pair and is assumed to be the same for all S–D pairs.

Define the channel utilization as the ratio of the channels involved in the data transmissions to the total number of channels in the system, i.e.,

$$\mu = \left(\sum_{i \in \Theta} n_i \pi_i \right) \Big/ L. \tag{12}$$

VI. NUMERICAL RESULTS

We evaluate the performance of the proposed DSMMAC in terms of system throughput, channel utilization, fairness, and power consumption overhead using an event-driven C simulator. The number of S–D pairs ranges from 1 to 20. We repeat each experiment for 20 runs with different random seeds and calculate the average value. The confidence intervals with a 95% confidence level are given to indicate the reliability of the simulation results. The main system parameters are listed in Table II. We consider both *single-hop* and *multihop* network scenarios.

1) *Single-hop scenario*. We simulate a system with 30 users and two channels. All users are within the communication range of each other. We apply the hopping sequence $H = [c_1\ c_1\ c_2\ c_1\ c_2\ c_2\ c_2]$, where c_1 and c_2 represent two available channels. S–D pairs are randomly selected from the users in the system.
2) *Multihop scenario*. We simulate a system with 8 channels and 200 users, which are randomly distributed in a 2 km $\times$ 2 km area. The transmission range of each user is 100 m. Therefore, some users are outside the communication range of other users, and thus, it is possible for multiple S–D pairs to simultaneously communicate over the same channel without interfering with each other. S–D pairs are randomly selected within one-hop neighbors in the system. We use the eight-channel hopping sequence derived in Section III.

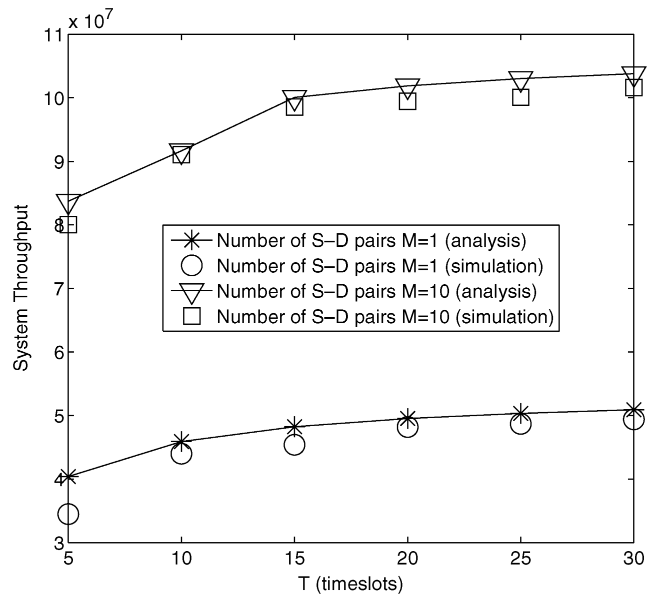


Fig. 4. System throughput versus the duration of each data transmission.

A. Performance Evaluation in the Single-Hop Scenario: Throughput and Channel Utilization

Fig. 4 shows the system throughput of the proposed DSMMAC in the single-hop scenario. It is shown that the system throughput increases with the duration of each data transmission (i.e., T). With a larger T , an S–D pair can occupy the channel for a longer time when they successfully negotiate, which improves the transmission efficiency due to less overhead per transmission in terms of time percentage and, thus, obtain a larger throughput.

Fig. 5 shows the channel utilization with different numbers of S–D pairs. Due to the excellent properties of DSMMAS (e.g., parallel rendezvous and asynchronization), multiple S–D pairs that attempt to access the media are efficiently distributed in the frequency domain. Meanwhile, the time instances that multiple source nodes send RTS message are separated in the time domain, which benefits the improvement of channel utilization with the increase of S–D pairs. The channel utilization (i.e., the

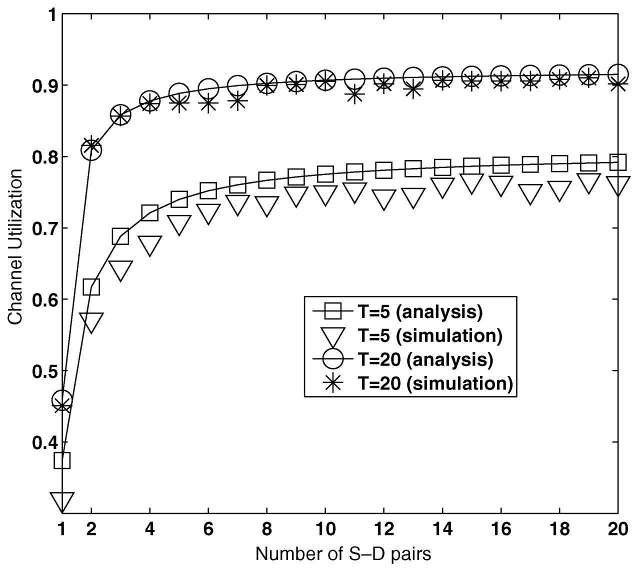


Fig. 5. Channel utilization with different numbers of S–D pairs.

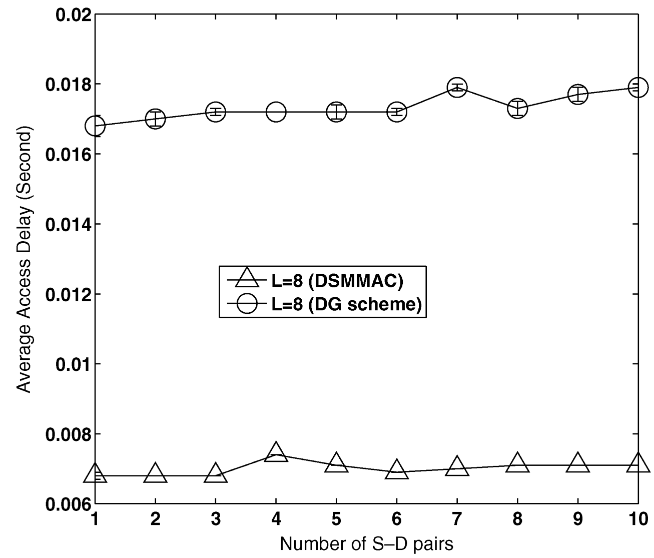


Fig. 7. Medium access delay with different numbers of S–D pairs.

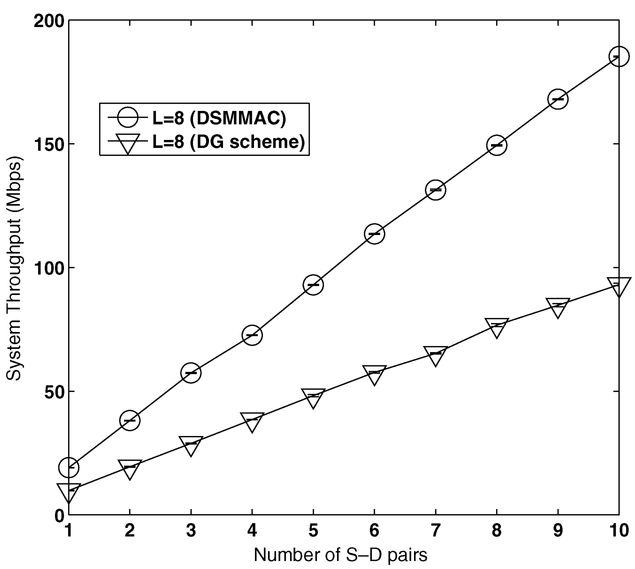


Fig. 6. System throughput with different numbers of S–D pairs.

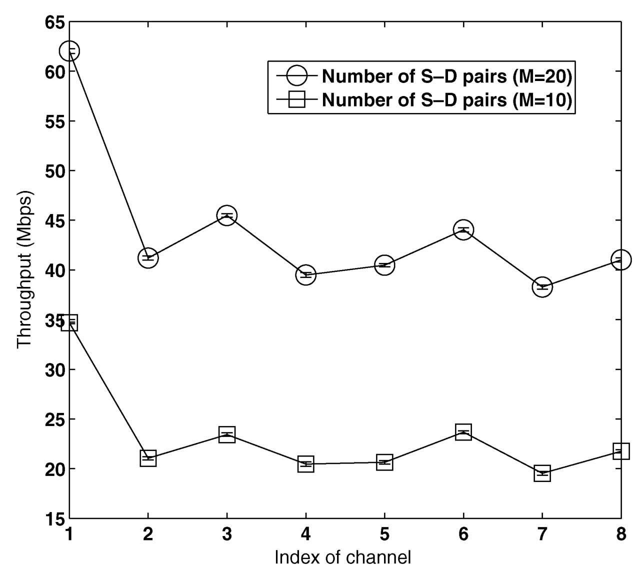


Fig. 8. Achieved throughput over each channel.

probability of starting a new data transmission over a channel) also increases with the number of S–D pairs. The analytical results match well with the simulation results, particularly when the average transmission time T is large. This case occurs because the inaccuracy introduced by converting the continuous exponential distribution to the discrete geometric distribution becomes negligible when T is large.

B. Performance Evaluation in the Multihop Scenario: Throughput, Access Delay, and Overhead

Figs. 6 –8 show the performance of the proposed DSMMAC in the multihop scenario. To further verify the efficiency of the proposed DSMMAC, we compare it with a multichannel MAC protocol based on the hopping sequence proposed by Dasilva and Guerreiro in [20] (which is denoted as the DG scheme). In the DG scheme, a predefined hopping sequence is adopted by all users to reduce the overhead of exchanging hopping information among users and eliminate the synchronization requirement. The comparison between our DSMMAC and the DG scheme is fair due to the similar features and objectives of both schemes.

Fig. 6 shows the system throughput with different numbers of S–D pairs. We observe that the total system throughput increases with the number of S–D pairs in both DSMMAC and the DG scheme due to the excellent features of multiple rendezvous and asynchronous transmissions. In addition, the proposed DSMMAC outperforms the DG scheme. The system throughput of DSMMAC is twice the system throughput of the DG scheme when the number of S–D pairs is 10.

Fig. 7 shows the channel access delay, which is defined as the average duration from the time instant that a source node attempts to access the channel to the time that it successfully communicates with its destination node. We observe that the

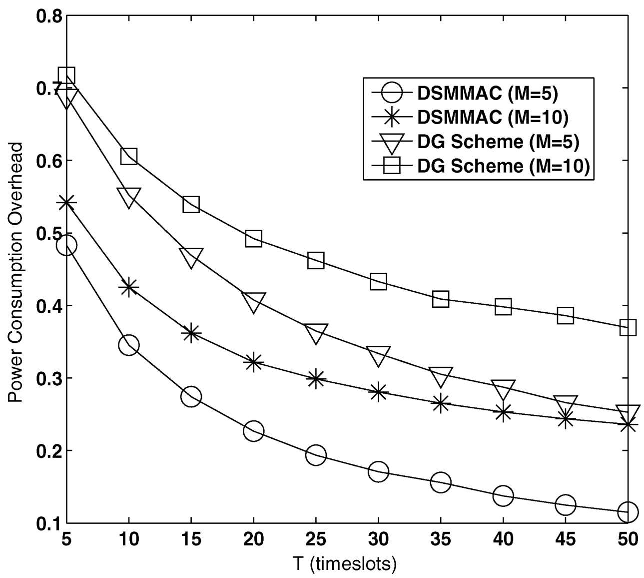


Fig. 9. Power consumption overhead versus the duration of each data transmission.

channel access delay slightly increases with the increase of S–D pairs. With a larger number of S–D pairs, the probability that an S–D pair fails to negotiate increases due to a busy channel increases, which leads to a longer access delay. In addition, the proposed DSMMAC protocol outperforms the DG scheme in terms of access delay due to a higher rendezvous probability of our designed difference-set-based hopping sequence. It is shown that DSMMAC achieves around a 150% reduction in channel access delay compared with the DG scheme.

Fig. 8 shows the throughput on each channel (not each S–D pair). Recall that we have used the eight-channel hopping sequence defined at the end of Section III. It is shown that channel c_1 achieves a higher throughput than other channels, because we allocate channel c_1 to the remaining time slot in the hopping sequence design, which means that channel c_1 will be used more often than the other channels. The throughputs achieved on channels c_2– c_8 are similar, which demonstrates fair use across channel resources.

For the proposed DSMMAC protocol, each source node sends an RTS message over its hopping channel at the beginning of a time slot to probe whether its destination node hops on the same channel. This probing may occur multiple times before an S–D pair successfully negotiates, which may lead to a significant overhead in terms of power consumption. Fig. 9 shows this overhead, which is defined as the ratio of the power consumed by the probing process to the power consumed by the probing process and data transmission. It is shown that, with a longer average data transmission time T , the power consumption overhead decreases. On the other hand, a larger number of S–D pairs (i.e., M) leads to a longer probing time for each S–D pair and, thus, increases the overhead. Compared with the DG scheme, the proposed DSMMAC protocol significantly decreases the power consumption overhead under the same setting of system parameters.

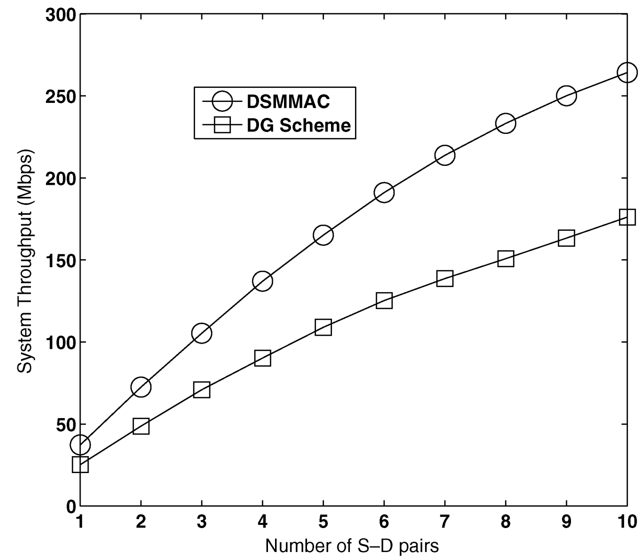


Fig. 10. System throughput with different numbers of S–D pairs.

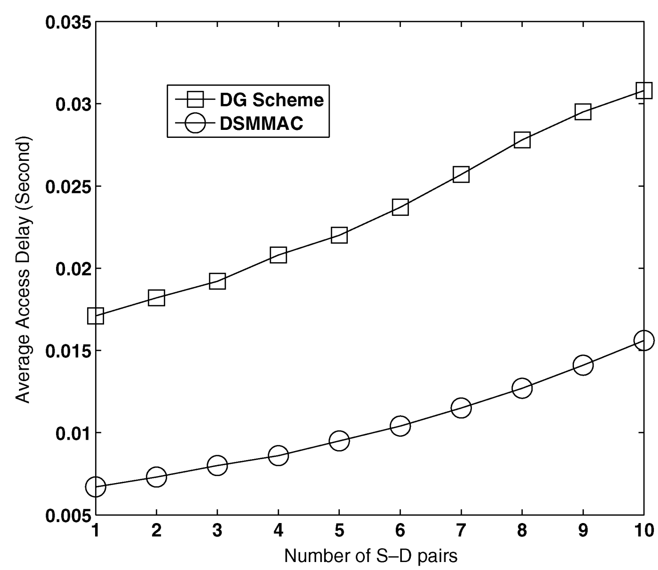


Fig. 11. Medium access delay with different numbers of S–D pairs.

C. Performance Evaluation With Deterministic Data Transmission Time: Throughput and Access Delay

The performance analysis and the aforementioned simulations assume exponential distribution of the data transmission time. Next, we show the performance of the proposed scheme based on a deterministic data transmission time (20 time slots) in Figs. 10 and 11 . A similar performance improvement of the proposed DSMMAC in terms of the system throughput and access delay is shown.

VII. Related Work

As the promising orthogonal frequency-division multiplexing (OFDM) technology opens doors for multichannel systems, MAC protocol design for multichannel systems has recently become a critical research issue and attracted great attention. Some previous works use a dedicated control channel for control message exchange to coordinate data transmissions over multiple data channels in the multichannel MAC design [12]–[16]. Li *et al.* [13] proposed a dynamic channel selection scheme using one dedicated frequency channel for control message exchanges. Moon and Syrotiuk [14] proposed a cooperative multichannel MAC using a control channel group to facilitate the channel selection. Alshamrani *et al.* [15] proposed an adaptive admission control and channel selection scheme using a secondary network controller to collect the information on the available channels and broadcast the allocation results through a local common control channel.

Another form of dedicated control channel was in the time domain, i.e., control and data messages were transmitted at different time slots. The multichannel MAC in [16] divided the time into an alternating sequence of control and data phases. Because no data transmission was allowed in the idle data channels during the control phases, such a protocol limited the network throughput achievable when the number of channels or the length of control phase was large.

Most multichannel MAC protocols required either tight or loose synchronization among all nodes. In the channel-hopping multiple access scheme in [17], all nodes used a predefined common hopping pattern, which eliminated heavy signaling overhead, but it still required tight synchronization among users. A MAC protocol with priority-based channel access is proposed in [18] to reduce the node contention and then improve the channel utilization, in which the loose synchronization among neighboring nodes is needed in the implementation.

There are MAC protocols that require the sources to know the hopping information of the destinations, which may involve significant signaling overheads. In the multiple rendezvous MAC protocol in [19], each node followed multiple hopping sequences in a time-multiplexed manner. When a node attempted to initiate a transmission to another node, it waited in a channel until their rendezvous arose over this channel. However, to make rendezvous happen, the sources needed to know their destinations' current hopping sequences through a seed broadcast mechanism, which increased the signaling overhead and degraded the system performance. Moreover, the reliability of broadcast transmissions could not be guaranteed over an error-prone wireless channel due to the lack of acknowledgment mechanisms.

Some recent work designed efficient hopping sequences without frequent hopping information exchanges among nodes. Dasilva and Guerreiro [20] designed hopping sequences that can ensure two nodes to meet without knowing each other's hopping sequence. They showed that a proper sequence design reduced the time to rendezvous compared with a simple random rendezvous. Bian *et al.* [8] proposed a QCH framework for the control channel establishment. They studied two optimal QCH systems in synchronous systems: One system minimized the time to rendezvous, and another system guaranteed the even distribution of the rendezvous points in both time and channel domains. Different from [8] and [20], this paper focuses on designing a distributed asynchronous MAC protocol to provide efficient robust channel access performances among different users over multiple channels. In addition, we analyze the protocol performance in terms of network throughput, access delay, channel utilization, and power consumption overhead.

VIII. Conclusion

In this paper, we have proposed a difference-set-based asynchronous MAC protocol for multichannel wireless networks. By allowing all users to use the same hopping sequence derived from difference sets, multiple S–D pairs can simultaneously rendezvous over different channels in a distributed manner so that the utilization of a multichannel system can be greatly improved. It has been demonstrated that the proposed MAC protocol can achieve high system throughput, low access delay, and good fairness among users under various network conditions. In addition, compared with previous proposed protocols, our protocol requires neither a dedicated control channel nor global synchronization among users, and thus, it is promising for practical deployment.

There are several interesting issues based on this paper. For instance, in a cognitive radio network, the channel availability and channel conditions of secondary users may differ from each other, because they are distributed over various geographical locations. How we can design hopping sequences to adapt to the dynamics and heterogeneity of the available spectrum bands to the secondary users is very important.

References

[1] D. Valerio, F. Ricciato, and P. Fuxjaeger, "On the feasibility of IEEE 802.11 multichannel multihop mesh networks," *Comput. Commun.*, vol. 31, no. 8, pp. 1484–1496, 2008.

[2] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE-802.11-based multichannel wireless mesh network ," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 2223–2234.

[3] R. Maheshwari, H. Gupta, and S. R. Das, "Multichannel MAC protocols for wireless networks," in *Proc. IEEE SECON*, Sep. 2006, pp. 393–401.

[4] J. Mo, H. W. So, and J. Walrand, "Comparison of multichannel MAC protocols," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 50–65, Jan. 2008.

[5] Y. Bi, K. H. Liu, L. X. Cai, X. Shen, and H. Zhao, "A multichannel token ring protocol for QoS provisioning in intervehicle communications ," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5621–5631, Nov. 2009.

[6] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks," *Mobile Netw. Appl.*, vol. 10, no. 1/2, pp. 169–181, Feb. 2005.

[7] R. Zheng, J. C. Hou, and L. Sha, "Optimal block design for asynchronous wake-up schedulers and its applications in multihop wireless networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 9, pp. 1228–1241, Sep. 2006.

[8] K. Bian, J.-M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks ," in *Proc. MobiCom*, Beijing, China, 2009, pp. 25–36.

[9] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications ," in *Proc. ACM Conf. Embedded Netw. Sens. Syst.*, Nov. 2008, pp. 71–84.

[10] L. D. Baument, *Lecture Notes in Mathematics: Cyclic Difference Sets*. New York: Springer-Verlag, 1971.

[11] I. Aderson, *Combinatorial Designs and Tournaments*. London, U.K.: Oxford Univ. Press, 1997.

[12] T. Luo, M. Motani, and V. Srinivasan, "Cooperative asynchronous multichannel MAC: Design, analysis, and implementation," *IEEE Trans. Mobile Comput.*, vol. 8, no. 3, pp. 338–352, Mar. 2009.

[13] J. Li, D. Zhang, S. Ji, and L. Guo, "RCS: A random channel selection with probabilistic backoff for multichannel MAC protocols in WSNs ," in *Proc. IEEE GLOBECOM*, Dec. 2010, pp. 1–5.

[14] Y. Moon and V. Syrotiuk, "A cooperative dual-access multichannel MAC protocol for ad hoc networks," in *Proc. IEEE GLOBECOM*, Dec. 2010, pp. 1–5.

[15] A. Alshamrani, L. Xie, and X. Shen, "Adaptive admission-control and channel-allocation policy in cooperative ad hoc opportunistic spectrum networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 1618–1629, May 2010.

[16] J. So and N. Vaidya, "Multichannel MAC for ad hoc networks: Handling multichannel hidden terminal using a single transceiver," in *Proc. ACM MobiHoc*, Tokyo, Japan, May 2004, pp. 222–233.

[17] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "Channel-hopping multiple access," in *Proc. IEEE ICC*, Jun. 2000, pp. 415–419.

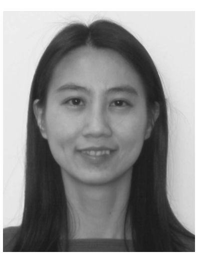
[18] J. Zhu, Z. Wang, and D. Xu, "A unified MAC and routing framework for multichannel multi-interface ad hoc networks ," *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4589–4601, Nov. 2010.

[19] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad hoc wireless networks," in *Proc. ACM MobiCom*, Sep. 2004, pp. 216–230.

[20] L. A. Dasilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *Proc. IEEE DySPAN*, Oct. 2008, pp. 1–7.

Fen Hou (M'10) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2008.

She is currently a Postdoctoral Fellow with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. Her research interests include resource allocation, connection admission control, scheduling schemes, and quality-of-service provisioning for multimedia communications.



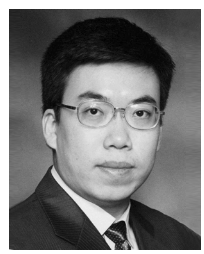
Lin X. Cai (M'10) received the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2005 and 2009, respectively.

In 2010, she received a Postdoctoral Fellowship from the Natural Sciences and Engineering Research Council of Canada and worked as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON. Since 2011, she has been a Postdoctoral Research Fellow with the Department of Electrical Engineering, Princeton University, Princeton, NJ. Her research interests include resource management for broadband multimedia networks, green communication and networking, cross-layer optimization, and quality-of-service provisioning.

Xuemin (Sherman) Shen (F'09) received the B.Sc. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees in electrical engineering from Rutgers University, New Brunswick, NJ, in 1987 and 1990, respectively.

He is currently a Professor and the University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is a coauthor of three books and has published more than 400 papers and book chapters in wireless communications and networks, control, and filtering. He also served as the Editor-in-Chief of *Peer-to-Peer Networking and Applications*, an Associate Editor for *Computer Networks* and *ACM/Wireless Networks*, and a Guest Editor for *ACM Mobile Networks and Applications*. His research interest is focused on resource management in interconnected wireless/wired networks, ultrawideband wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks.

Dr. Shen is a Registered Professional Engineer in Ontario, Canada, a Fellow of the Engineering Institute of Canada, and a Distinguished Lecturer of the IEEE Communications Society. He served as the Technical Program Committee Chair of the 2010 IEEE Vehicular Technology Conference, the Symposia Chair of the 2010 IEEE International Conference on Communications (ICC), the Tutorial Chair of the 2008 IEEE ICC, the Technical Program Committee Chair of the 2007 IEEE Global Communications Conference, a General Cochair of the 2007 International Conference in Communications and Networking in China, and the Third Annual Conference on Quality of Service in Heterogeneous Wireless/Wireline Networks, and the Founding Chair of the IEEE Communications Society Technical Committee on Peer-to-Peer Communications and Networking. He also served as a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and as a Guest Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and the IEEE COMMUNICATIONS MAGAZINE. He received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award in 2003 from the Province of Ontario, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo.



Jianwei Huang (M'05) received the B.S. degree in electrical engineering from the Southeast University, Nanjing, Jiangsu, China, in 2000 and the M.S. and Ph.D. degrees in electrical and computer engineering from the Northwestern University, Evanston, IL, in 2003 and 2005, respectively.

From 2005 to 2007, he was a Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University, Princeton, NJ. In the summer of 2004 and 2005, he was an Intern with Motorola, Arlington Heights, IL. Since 2007, he has been an Assistant Professor with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, Hong Kong, where he leads the Network Communications and Economics Laboratory. His research interests include nonlinear optimization and game-theoretical analysis of communication networks, with a current focus on network economics, cognitive radio networks, broadband communication networks, and multimedia over wireless.

Dr. Huang has served as an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2010–present), the Lead Guest Editor of the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS Special Issue on Game Theory in Communication Systems, and as a Guest Editor for several other journals, including Wiley *Wireless Communications and Mobile Computing*, the *Journal of Communications*, and the *Journal of Advances in Multimedia*. He has served as the Vice Chair of the IEEE Communications Society (ComSoc) Multimedia Communications Technical Committee (2010–2012), the Director of the IEEE Communications Society Multimedia Communications Technical Committee E-Letter (2010), the Student Activities Cochair of the 2011 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, a Technical Program Committee (TPC) Co-chair of the 2009 International Conference on Game Theory for Networks, a TPC Co-chair of the IEEE Global communications Conference (GLOBECOM) Wireless Communications Symposium 2010, a TPC Co-chair of the 2010 International Wireless Communications and Mobile Computing Conference Computing Symposium, and a TPC Member of several conferences such as International Conference on Computer Communications (INFOCOM), MobiHoc, International Conference on Communications (ICC), GLOBECOM, Dynamic Spectrum Access Networks (DySPAN), International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), and Wireless Communications and Networking Conference (WCNC). He is the recipient of the 2010 IEEE GLOBECOM Best Paper Award, the 2009 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award, the 2009 Asia-Pacific Conference on Communications Best Paper Award, and the Walter P. Murphy Fellowship from the Northwestern University in 2001.