Asynchronous Multi-channel MAC Design with Difference Set based Hopping Sequences

Fen Hou, Lin X. Cai, Xuemin Shen, Jianwei Huang
Date Submitted: 7 August 2009
Date Published: 11 August 2009

Updated information and services can be found at:

These include:

Subject Classification Vehicular Technology > Wireless Networks

Keywords Difference set, multi-channel, asynchronous; Medium access control;

Submitting Author's Comments IEEE Infocom

Comments You can respond to this article at:
http://engine.lib.uwaterloo.ca/ojs-2.2/index.php/pptvt/comment/add/546/0

Copyright Copyright © Date Submitted: 7 August 2009 Fen Hou et al. This is an open access article distributed under the Creative Commons Attribution 2.5 Canada License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Asynchronous Multi-channel MAC Design with Difference Set based Hopping Sequences

Fen Hou∗, Lin X. Cai∗, Xuemin (Sherman) Shen∗, and Jianwei Huang∗
University of Waterloo, Waterloo, Canada∗; The Chinese University of Hong Kong∗
{fhou, lcai, xshen}@bbr.uwaterloo.ca; jwhuang@ie.cuhk.edu.hk

Abstract – Most existing multi-channel medium access control (MAC) protocols have at least one of the following 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’ different channel selection strategies. In this paper, we propose a difference set based multi-channel MAC protocol (DSMMAC) which overcomes the above performance bottlenecks. Our proposed protocol achieves high system throughput, low access delay, and good fairness performance, without the need for global synchronization or dedicated control channel/phase. We analyze the performance of our proposed protocol based on a Markove chain model, and show the proposed DSMMAC can achieve up to 100% improvement in system throughput and 150% reduction in channel access delay compared with an existing multi-channel MAC protocol.

Index Terms – Difference set, media access control, multi-channel, asynchronous

I. INTRODUCTION

Multi-channel communication system has been widely adopted today to efficiently support concurrent transmissions in the frequency domain and thus achieve high network throughput (e.g., [1], [2], [3]). For example, three and twelve non-overlapping channels are specified in the widely deployed IEEE 802.11b and 802.11a wireless local area networks (WLANs). In IEEE 802.11p, a seven channel structure is defined for intelligent transportation systems. The promising cognitive radio network is inherently a multi-channel system, where available frequency bands are accessed opportunistically by the secondary users. A key design challenge for these systems is to have efficient and distributed media access control (MAC) protocols that can exploit the multi-channel structure and achieve the system capacity.

Although various multi-channel MAC protocols have appeared in the literature, most of them share one or more of the following features (detailed discussion of related work can be found in Section VII):

• The need for a dedicated control channel for signaling exchange (e.g., [5]), which may significantly degrade the performance of a system where only a few channels are available and becomes the performance bottleneck when the control traffic load is heavy.
• Using a dedicated control phase (time period) to exchange signaling information (e.g., [8]), which reduces the channel utilization especially when the control phase is long.

• The need for global clock synchronization among all users (e.g., [6], [7]), which is difficult to achieve in large-scale distributed networks.
• Requiring the knowledge of channel selection strategies (e.g., hopping sequences) of other users in the network (e.g., [9], [10]), which is typically not trivial.

In this paper, we overcome the above performance bottlenecks by proposing a new multi-channel MAC protocol based on the concept of difference set (DSMMAC protocol). The key idea of DSMMAC is to assign all users the same hopping sequences generated from multiple difference sets. The unique rotation closure property of difference set (e.g., [12], [13]) enables any two users to communicate (i.e., meet or rendezvous) with a certain probability without global synchronization or a dedicated control channel. More detailed definitions and discussions on the properties of difference sets will be given in Section III. To the best of our knowledge, this is the first paper that uses the difference set concept in the multi-channel MAC design. The main advantages of the proposed DSMMAC include:

1) Asynchronization: Our protocol does not require synchronization among users, and thus is robust to implementation in distributed networks.
2) Multiple parallel rendezvous: Multiple source-destination pairs can rendezvous and communicate over different channels simultaneously without the centralized coordination. This improves the channel utilization and system performance.
3) Less signaling overhead: All users deploy the same hopping sequence, which eliminates the need for exchanging hopping sequence information and thus reduces the signaling overhead.
4) No dedicated control channel: Since DSMMAC requires no control channel for channel coordination, it can fully utilize all available channels for data transmissions.

The remainder of the paper is organized as follows. Section II describes the system 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, followed by the related work in Section VII. We finally conclude in Section VIII.

II. SYSTEM MODEL

We consider a multi-channel network consisting of $L$ channels and multiple nodes, as shown in Figure 1. Each node represents an end-user which can be a mobile phone, a mobile
PDA, a laptop, or a sensor. Each user is equipped with one tunable half-duplex radio transceiver which can switch between different channels. Two users that want to communicate with each other are called a source-destination (S-D) pair.

An S-D pair can communicate with each other only when they meet (or rendezvous) over a channel, i.e., both the source and destination hop over the same channel at the same time. Two S-D pairs can communicate over different channels without interference with each other. Meanwhile, based on the protocol interference model [11], multiple S-D pairs can communicate simultaneously over the same channels without interference with each other if one destination is not the one-hop neighbor of any other sources of ongoing transmissions over the same channel.

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

III. DIFFERENCE-SET BASED HOPPING SEQUENCE

The key idea behind our proposed multi-channel MAC protocol is to use the nice properties of difference set. Some previous works applied difference set in the design of power-saving sleep scheduling algorithms in wireless ad hoc networks, where the main objective is to minimize a node’s wake-up duty cycle while maintaining a certain probability that two users can communicate with each other when they wake up [14], [15]. Different from sleep scheduling algorithm design, our objective of multi-channel MAC design with difference set is to maximize the rendezvous probability of S-D pairs and achieve high channel utilization over various channels. Moreover, we need to consider the design in both frequency and time domains to guarantee a successful rendezvous, while a sleep scheduling algorithm only considers a wake-up rendezvous in the time domain.

In this section, we briefly review the properties of difference set [12], [13] and present the design of difference set based hopping sequence for a multi-channel system.

A. Difference set properties

Definition 1: Difference Set - Let \( v, k \) and \( \lambda \) represent integers no less than 1. A \((v,k,\lambda)\) difference set is defined as a \( k \)-element set \( D = \{d_1, \cdots, d_k\} \) such that every \( d \in \{1, \cdots, v-1\} \) can be expressed in the form of \( d = d_i - d_j \equiv d \text{ (mod } v) \) with exactly \( \lambda \) ordered pairs of \((i,j)\), where \( i,j \in \{1, \cdots, k\} \) denote the indices in \( D \). \( v \) is the cycle of the difference set.

\[
D = \{d_1, d_2, \cdots, d_k\} \quad \text{be a} \quad (v,k,\lambda) \quad \text{difference set. The incidence vector of a} \quad D \quad \text{is denoted as} \quad \sum_{i=0}^{v-1} a_i = 1 \quad \text{if} \quad i \quad \text{is an element of} \quad D \\
\quad 0 \quad \text{otherwise.} 
\]

Definition 2: Incident vector of a difference set - The incident vector of a \((v,k,\lambda)\) difference set \( D \) is a binary sequence \( \{a_i\}_{i=0}^{v-1} \)

\[
a_i = \begin{cases} 
1 & \text{if } i \text{ is an element of } D \\
0 & \text{otherwise} 
\end{cases}
\]

Definition 3: Complementary set of a difference set - Let \( D = \{d_1, d_2, \cdots, d_k\} \) be a \((v,k,\lambda)\) difference set. The complementary set of \( D \) in \( Z_v = \{1, 2, \cdots, v\} \) is

\[
\overline{D} = \{d \in Z_v | d \notin D\}
\]

Definition 4: Shift set of a difference set - Let \( D = \{d_1, d_2, \cdots, d_k\} \) be a \((v,k,\lambda)\) difference set. Its \( \mu^{th} \) shift set is defined as \( D^\mu_v = \{d_1 + \mu, \cdots, (d_k + \mu \text{ (mod } v))\} \), \( \mu \in \{1,2,\cdots,v-1\} \).

An example of \((11,5,2)\) difference set is \( D = \{1,3,4,5,9\} \). Its complementary set and \( 1^{st} \) shift set are \( \overline{D} = \{2,6,7,8,10,11\} \) and \( D^1_v = \{2,4,5,6,10\} \), respectively, and its incident vector is \( \{1,0,1,1,0,0,0,1,0,0\} \).

Property 1: Complementary set of a difference set is a difference set - If \( D \) is a \((v,k,\lambda)\) difference set, its complementary set \( \overline{D} \) is a \((v, v-k, v-2k+\lambda)\) difference set, called the complementary difference set.

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

Property 2: Rotation closure property - If \( a_i \) is the incidence vector of a \((v,k,\lambda)\)-difference set, its autocorrelation function satisfies

\[
R_a(j) = \begin{cases} 
k & \text{if } (j \text{ mod } v) \equiv 0 \\
\lambda & \text{Otherwise} 
\end{cases}
\]

where \( R_a(j) = \sum_{i=0}^{v-1} a_i a_{(i+j) \text{ mod } v} \).

Rotation closure property ensures that a \((v,k,\lambda)\) difference set and any of its shift sets have \( \lambda \) overlapping elements in a cycle of \( v \). Thus, \( \lambda/v \) is denoted as the rendezvous probability when a difference set is applied in hopping sequence design. This unique feature of difference set ensures two users using the same hopping sequence generated from one or more difference sets can meet with each other with a certain probability without prior coordination.

Figure 2 illustrates the rotation closure property in a single channel case (e.g., Channel \( C_i \)). Users A and B follow the same hopping sequence generated from a \((11,5,2)\) difference set \( D = \{1,3,4,5,9\} \). In a cycle of 11 time slots, users hop over channel \( C_1 \) in their 1, 3, 4, 5, and 9 time slots depicted as solid rectangles. A blank rectangle represents an inactive time slot in which users switch off their transceivers.

- Synchronization Case: As shown in Figure 2(a), users A and B have aligned time slot boundaries. They start to hop to channel \( C_3 \) at the time \( t_0 \) and \( t_1 \), respectively, based on the same pre-defined hopping sequence with different time shifts. Let us consider a cycle of 11 time slots. We can see that two users rendezvous on the same channel \( C_1 \) twice (the beginnings of user B’s time slot 3 and 4). This means that the long-term rendezvous probability is 2/11.
Asynchronization Case: As shown in Figure 2(b), users A and B join the network and hop to channel $C_1$ at the time $t_0$ and $t_1'$, respectively. We can see that in a cycle of 11 time slots, two users rendezvous on the same channel $C_1$ twice (at the beginning of user B’s 3 and 4 time slots). This means that the long-term rendezvous probability is still $2/11$.

In general, if two users use the same hopping sequences generated from difference sets, they can always meet with a certain probability regardless of the synchronization.

**B. The design of difference set based hopping sequence**

In the proposed DSMMAC protocol, all users in the network use the same frequency hopping sequence designed based on difference sets. To improve the resource utilization of a multi-channel network, it is desirable to achieve a high rendezvous probability of any S-D pair so that they have a higher chance to successfully communicate with each other. Moreover, to fully exploit the capability of concurrent transmissions in multi-channel networks, we use a hopping sequence with high rendezvous probability. The main design steps include:

**Step 1:** Select multiple difference sets - To guarantee successful user rendezvous in both time and frequency domains, we need to combine multiple difference sets with the same cycle to generate a hopping sequence covering 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 $v$ which assures the rendezvous probability $\lambda/v$ 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 $\lambda/v$) to assure that users are very likely to meet with each other over each channel.

**Criterion 2:** Empty intersection - Any two chosen difference sets (denoted as $D_i^v$ and $D_j^v$ (i $\neq$ j)) should satisfy

$$D_i^v \cap D_j^v = \emptyset, \text{ for all } i, j \in 1, 2, \cdots L.$$
time slots of $D_7^7$ (i.e., time slots 3, 5, 6, and 7). The resulting hopping sequence is $H=[C_1 \ C_1 \ C_2 \ C_1 \ C_2 \ C_1 \ C_2 \ C_1 \ C_2 \ C_2 \ C_2]$. Shown in Figure 3.

As a more complicated example that will be used in Section VI, we discuss the hopping sequence design for 8 channels (denoted as $C_1$ through $C_8$). We first choose 8 difference sets as follows

$$D_S = \{D_1^1, D_2^2, D_3^3, D_4^4, D_5^5, D_6^6, D_7^7, D_8^8\}$$

$$= \{\{2 \ 3 \ 5 \ 9 \ 17 \ 33 \ 38 \ 56 \ 65\},$$

$$\{4 \ 7 \ 13 \ 20 \ 24 \ 25 \ 39 \ 47 \ 49\},$$

$$\{6 \ 8 \ 11 \ 15 \ 21 \ 29 \ 40 \ 41 \ 57\},$$

$$\{10 \ 19 \ 37 \ 42 \ 58 \ 66 \ 70 \ 72 \ 73\},$$

$$\{12 \ 16 \ 22 \ 31 \ 43 \ 45 \ 48 \ 61\},$$

$$\{14 \ 27 \ 32 \ 44 \ 52 \ 53 \ 59 \ 63\},$$

$$\{18 \ 34 \ 35 \ 46 \ 50 \ 60 \ 64 \ 67 \ 69\},$$

$$\{26 \ 28 \ 36 \ 50 \ 51 \ 55 \ 62 \ 68 \ 71\}\}.$$ These 8 difference sets have the same cycle (i.e., 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 corresponding to $D_1^1$ (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 8 difference sets, we find that time slot 1 remains unassigned (i.e., $D_S^1 = \{1\}$). Therefore, we randomly assign one frequency channel to time slot 1. Without loss of generality, we allocate channel $C_1$ to the time slot 1. The final hopping sequence is $H=[C_1 \ C_1 \ C_2 \ C_1 \ C_2 \ C_1 \ C_2 \ C_1 \ C_2 \ C_1 \ C_2 \ C_2 \ C_2]$. Shown in Figure 3.

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

Algorithm 1 Source node
1: Randomly select a hopping channel;
2: Channel sensing;
3: if Channel is idle for a DIFS then
4: Send a RTS;
5: if A CTS is received after a SIFS then
6: while Channel occupation time $\leq T_{max}$ and data buffer is not empty do
7: Transmit data over the channel;
8: end while
9: Release the channel;
10: end if
11: end if
12: Hop to the next channel based on the designed hopping sequence;
13: Go to Line 2;

IV. DIFFERENCE-SET BASED MULTI-CHANNEL MAC DESIGN

To design an efficient and robust MAC, it is critical to reduce the signaling overhead as well as release the requirement of synchronization. To achieve these, we propose a difference set based asynchronous multi-channel MAC (DSMMAC). In the proposed DSMMAC, all users use the same hopping sequence to access various channels in a distributed manner. Thus, users do not need to exchange the hopping sequence information of each other, which can efficiently reduce signaling overhead. The pseudo codes of the sources and destinations are presented in Algorithm 1 and Algorithm 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, a source node will switch to the next hopping channel based on the hopping sequence and start to sense the channel again. If the channel is sensed idle for a DIFS (DCF InterFrame Space) interval, implying currently there is no ongoing transmissions over the channel, the source node initiates a RTS (Ready To Send) transmission. If no CTS (Clear To Send) is received after a SIFS (Short InterFrame Space) interval, it continues channel sensing. Notice that it is also possible that the transmitted RTS/CTS are lost in the error-prone wireless channel or due to hidden terminal problems in a multi-hop network. If the channel condition between an S-D pair is poor, it is impossible to start data transmission in this channel at this time. In addition, with multiple channels operating in parallel, the collision probability on any one of the channel is greatly reduced. Therefore, in a multi-channel scenario, when a source node sends a RTS but receive no CTS after a SIFS, it is more likely that the tagged destination does not access the same channel at this moment. Thus, it switches to other channels. If a CTS is received after a SIFS, which means the tagged destination also accesses the channel at this time, 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 current channel, users resume channel hopping based on the hopping sequence. Users keep hopping over various channels according to the pre-defined common hopping sequence until a successful handshake between an S-D pair occurs. Because of the rotation closure property of difference sets, any two users can meet with each other or rendezvous in one channel with a guaranteed probability in each cycle, regardless of the initial hopping channels they select and whether their system clocks are synchronized or not.

Algorithm 2 Destination node
1: Channel sensing;
2: if Channel is idle for a DIFS then
3: Randomly select a hopping channel;
4: Channel occupation time $\leq T_{max}$ and data buffer is not empty do
5: Transmit data over the channel;
6: end while
7: Release the channel;
8: end if
9: Hop to the next channel based on the designed hopping sequence; 
10: Go to Line 2;

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

V. PERFORMANCE ANALYSIS

We develop an analytical model to investigate the performance of the proposed DSMMAC in terms of network throughput and channel utilization. Main notations used in the paper are listed in TABLE I for easy reference.
Algorithm 2 Destination node
1: Randomly select a hopping channel;
2: Channel sensing;
3: if Channel is idle then
4: Wait a time slot for a RTS;
5: if Receive a RTS targeted to it then
6: Resend a CTS;
7: if Receive data after a SIFS then
8: repeat
9: Receive data
10: until Data transmissions complete
11: end if
12: end if
13: end if
14: Hop to the next channel based on the designed hopping sequence;
15: Go to Line 2;

We divide time into small “virtual” slots. Let \( \tau_0 = 1, 2, \ldots \) be the time instant at the beginning of the time slot \( b \). Notice that time slot introduced here is for analysis purpose. In practical implementation, each user keeps a local time-slotted system where the boundaries of the slots are not necessarily synchronized across users. The only requirement is the length of a time slot is the same for all users. We observe the system at \( \tau_0 \) and denote the system state as \( \zeta \), which consists of the set of all channels being used for data transmission at \( \tau_0 \). Assume that system state transition occurs at the beginning of each time slot. The sequence of observed system states forms a discrete Markov chain.

Let the \( S_i = \{\phi_1^i, \phi_2^i, \ldots, \phi_{C_i}^i\} \) denote the system state \( i \), where \( \phi_1^i, \ldots, \phi_{C_i}^i \) represent channels involving in the data transmission at this state. For instance, \( S_{10} = \{C_1, C_2, C_8\} \) means that channels \( C_1, C_2 \) and \( C_8 \) are being 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 \( \emptyset, \{C_1\}, \{C_2\}, \{C_1, C_2\} \). These four system states can be labeled as the states 1, 2, 3, and 4, respectively.

Thus, one-step transition probability from the state \( i \) to \( j \) is defined as
\[
p_{ij} = Pr[\zeta = S_j, t = \tau_{k+1} | \zeta = S_i, t = \tau_b]
\]
where \( S_i = \{c_1^i, \ldots, c_{C_i}^i\} \) and \( S_j = \{c_1^j, \ldots, c_{C_j}^j\} \). \( m \) and \( n \) represent the total number of data transmission channels in the state \( i \) and \( j \), respectively. \( \tau_b \) is the beginning instant of an arbitrary time slot \( b \).

Additional notations are needed to derive the values of transition probability \( p_{ij} \). Let \( \phi_{ij}^s \) and \( \phi_{ij}^f \) denote the sets of data transmission channels started and released at this state transition, respectively, and the number of channels in the sets of \( \phi_{ij}^s \) and \( \phi_{ij}^f \) are denoted as \( k = |\phi_{ij}^s| \) and \( l = |\phi_{ij}^f| \), respectively. In other words, during this state transition, two events occur: (1) data transmissions over \( l \) channels in \( \phi_{ij}^f \) are released; and (2) \( k \) data transmissions start over the channels in \( \phi_{ij}^s \). For instance, given that \( S_i = \{C_1, C_3, C_4\} \) and \( S_j = \{C_1, C_5, C_8\} \), we have \( \phi_{ij}^s = \{C_5, C_8\} \), \( \phi_{ij}^f = \{C_3, C_4\} \), \( m = 3, n = 3, k = 2, \) and \( l = 2 \). Let \( Pr[R_{ij}] \) and \( Pr[S_{ij}] \) be the probabilities of releasing the \( l \) data transmission channels and starting the \( k \) new data transmissions, respectively. The one step transition probability from the state \( i \) to \( j \) is obtained as
\[
p_{ij} = Pr[R_{ij}]Pr[S_{ij}].
\]

1) Derivation of \( Pr[R_{ij}] \): An S-D pair starts data transmission after successful exchange of RTS and CTS messages over the same channel. Let random variable \( X \) represent the duration of each data transmission. We assume that it follows the 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 one. Define the discrete random variable \( Y = \lfloor X/T_{slot} \rfloor \), where \( T_{slot} \) is the duration of a time slot, and \( \lfloor \cdot \rfloor \) represents the largest integer less than or equal to the argument. Since \( X \) is an exponential random variable with mean \( T \), \( Y = \lfloor X/T_{slot} \rfloor \) also follows an exponential distribution with mean \( \alpha = T/T_{slot} \). In other words, we observe the system at the beginning of each time slot, each data transmission may release or continue with probability \( q \), and \( 1 - q \) is the probability of this event is given by
\[
Pr[R_{ij}] = q^k (1-q)^{m-k},
\]
where \( k = |\phi_{ij}^f| \) is the number of data channels released at the next time slot, and \( m = |\phi_{ij}^s| \) is the total number of channels used for data transmissions in state \( i \).

---

**TABLE I**

**Table of Notations.**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L )</td>
<td>the total number of channels in the system</td>
</tr>
<tr>
<td>( F )</td>
<td>the total number of S-D pairs (i.e., flows)</td>
</tr>
<tr>
<td>( S_i )</td>
<td>system state ( i )</td>
</tr>
<tr>
<td>( \phi_{ij}^s )</td>
<td>the set of data channels started when the system transits from state ( i ) to ( j )</td>
</tr>
<tr>
<td>( \phi_{ij}^f )</td>
<td>the set of released data channels when the system transits from state ( i ) to ( j )</td>
</tr>
<tr>
<td>( k, l )</td>
<td>the number of channels in ( \phi_{ij}^s ) and ( \phi_{ij}^f ), respectively</td>
</tr>
<tr>
<td>( Pr[R_{ij}] )</td>
<td>Probability of releasing all data channels in ( \phi_{ij}^f )</td>
</tr>
<tr>
<td>( Pr[S_{ij}] )</td>
<td>Probability of starting data transmissions over all channels in ( \phi_{ij}^s )</td>
</tr>
<tr>
<td>( Pr[W_{ij}] )</td>
<td>Probability that the ( z^{th} ) S-D pair fails to negotiate given system state ( i )</td>
</tr>
<tr>
<td>( Pr[S_i^{r,h}] )</td>
<td>Probability that the ( z^{th} ) S-D pair succeeds to negotiate over channel ( h ) given state ( i )</td>
</tr>
<tr>
<td>( H )</td>
<td>The set of hopping sequence</td>
</tr>
<tr>
<td>( h_i )</td>
<td>The ( i^{th} ) hopping channel in ( H )</td>
</tr>
<tr>
<td>( C )</td>
<td>The set of all available channels in the system</td>
</tr>
<tr>
<td>( C_i )</td>
<td>The ( i^{th} ) channel in the system</td>
</tr>
<tr>
<td>( q )</td>
<td>Probability of releasing a data channel</td>
</tr>
<tr>
<td>( r )</td>
<td>Data transmission rate</td>
</tr>
<tr>
<td>( Th )</td>
<td>System throughput</td>
</tr>
<tr>
<td>( Th_f )</td>
<td>Achieved throughput of the ( f^{th} ) flow</td>
</tr>
<tr>
<td>( T )</td>
<td>Mean of each data transmission (in unit of time slot)</td>
</tr>
<tr>
<td>( I )</td>
<td>Jain’s Fairness Index</td>
</tr>
</tbody>
</table>
2) Derivation of \( \Pr[S_{ij}] \): A data transmission starts when an S-D pair successfully rendezvous over one channel. Due to asynchronous transmission of multiple users in a multi-channel scenario, the probability that two or multiple sources attempt to transmit with the same channel simultaneously is negligible. In other words, the probability of a successful rendezvous mainly depends on the difference set based hopping sequence and the total number of S-D pairs attempting to access the channel. Consider a difference set based hopping sequence \( H = [h_1, \cdots, h_v] \) with \( h_g \in \{C_1, C_2, \cdots, C_L\} \), where \( v \) is the cycle of difference sets used in the design of \( H \), \( h_g \) is the \( g \)th hopping channel in the hopping sequence \( H \). Let \( S \) be the total number of channels in the system. Let \( \Pr[S_i^{z,h}] \) be the probability that the \( z \)th S-D pair starts data transmission over channel \( h \) at the next time slot given that the system is in state \( i \) at the current time slot. It equals to the probability of successful rendezvous over channel \( h \) for an S-D pair. Since the initial hopping channel is randomly selected, an S-D pair has different \( \Pr[S_i^{z,h}] \) when they select different initial hopping channels. Let \( \eta_s \) and \( \eta_d \) be the indices of initial hopping channels in \( H \) selected by the source and destination, respectively. For instance, \( \eta_s = 2 \) means that the source selects \( h_2 \) as its initial hopping channel. Thus, the probability of successful rendezvous over channel \( h \) is given as

\[
\Pr[S_i^{z,h}] = \sum_{\gamma_1=1}^{v} \sum_{\gamma_2=1}^{v} \Pr[\eta_s = \gamma_1] \Pr[\eta_d = \gamma_2] \Pr[S_i^{z,h} | \eta_s, \eta_d]
\]

(4)

Given \( \eta_s \) and \( \eta_d \), the conditional probability \( \Pr[S_i^{z,h} | \eta_s, \eta_d] \) depends on the used hopping sequence \( H \). For instance, given \( \eta_s = 5 \), \( \eta_d = 1 \), \( H = [C_1, C_1, C_2, C_1, C_2, C_2] \), and the system state is \( S_i = \{C_1\} \), a source meets its rendezvous using its destination channel \( C_2 \) at its 3rd time slot. Therefore, the conditional probability \( \Pr[S_i^{z,h} | \eta_s = 5, \eta_d = 1] = 1/3 \). Thus, the probability that the \( z \)th S-D pair fails to negotiate for data transmission given the system is at state \( i \), \( \Pr[W_i^{z}] \), is given by

\[
\Pr[W_i^{z}] = 1 - \sum_{h \in \overline{S}_i} \Pr[S_i^{z,h}]
\]

(5)

where \( \overline{S}_i \) is the set of channels not used for data transmission in state \( i \), which is the complimentary set of \( S_i \) in \( \{C_1, C_2, \cdots, C_L\} \). Therefore, the probability that \( k \) data transmissions start over channels in \( \phi_{ij} \), \( \Pr[S_{ij}] \), is given by

\[
\Pr[S_{ij}] = \sum_{a=1}^{(F-m)} \prod_{z \in \Omega_a, z' \in \Omega_a \setminus \Omega, h \in \phi_{ij}} \Pr[S_i^{z,h}] \Pr[W_i^{z'}]
\]

(6)

where \( F \) is the total number of S-D pairs, and \( m \) is the number of channels used for the data transmission at state \( i \). Therefore, \( (F - m) \) is the number of S-D pairs attempting to access the channels. Let \( \Omega \) represent the set of all these \( (F - m) \) S-D pairs. \( k \) is the number of channels in \( \phi_{ij} \). That is, there are \( k \) S-D pairs successfully negotiate for their data transmissions at the next time slot. There are \( (F - m)^k \) possible combinations of selecting \( k \) out of \( (F - m) \) S-D pairs. Let \( \Omega_a \) represent the set of S-D pairs corresponding to the \( a \)th combination, and \( \overline{\Omega_a} \) be the complementary set of \( \Omega_a \) in \( \Omega \). Therefore, \( \prod_{z \in \Omega_a, z' \in \overline{\Omega_a}, h \in \phi_{ij}} \Pr[S_i^{z,h}] \Pr[W_i^{z'}] \) is the probability that S-D pairs in \( \Omega_a \) successfully negotiate over the channels in \( \phi_{ij} \) and S-D pairs in \( \overline{\Omega_a} \) fail their negotiations. We sum over all possible combinations from \( a = 1 \) through \( (F-m)^m \), and obtain the probability \( \Pr[S_{ij}] \).

As an illustration example, we assume the total number of channels in the network is \( L = 2 \), denoted as \( C_1 \) and \( C_2 \). The number of S-D pairs is \( F = 3 \). When we observe the system at the beginning of two continuous time slots, the system states are \( S_i = \{C_1\} \) and \( S_j = \{C_1, C_2\} \), respectively. Therefore, we have \( S_{ij} = \emptyset \), \( \phi_{ij} = \{C_2\} \), \( \Omega = \{1, 2\} \), \( F = 3 \), \( L = 2 \), \( m = 1 \), \( n = 2 \), \( k = 1 \), and \( l = 0 \). Thus, \( \Pr[S_{ij}] \) is obtained as

\[
\Pr[S_{ij}] = \prod_{a=1}^{(F-m)} \sum_{z \in \Omega_a, z' \in \overline{\Omega_a}, h \in \phi_{ij}} \Pr[S_i^{z,h}] \Pr[W_i^{z'}] = \prod_{z \in \{1\}, z' \in \{3\}, h \in \{C_2\}} \Pr[S_i^{z,h}] \Pr[W_i^{z'}] + \prod_{z \in \{3\}, z' \in \{1\}, h \in \{C_2\}} \Pr[S_i^{z,h}] \Pr[W_i^{z'}] = \Pr[S_i^{1,2}] \Pr[W_i^{3}] + \Pr[S_i^{3,2}] \Pr[W_i^{1}]
\]

(7)

3) System throughput and channel utilization: Based on the derived probabilities of releasing and starting data transmissions, the one-step transition probability matrix is

\[
P = \begin{bmatrix} p_{ij} \end{bmatrix}_{N \times N}; i, j \in \Theta
\]

(8)

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

Let \( \pi_i \) denote the steady-state probability for the system staying at the state \( i \). Based on the developed one-step transmission probability matrix and system balance equations, the system steady-state probability, \( \Pi = [\pi_1, \pi_2, \ldots, \pi_N] \), is derived as

\[
\Pi = Q([I - P]\Lambda + U)^{-1};
\]

(9)

where \( Q \) is a 1-by-\( N \) zero matrix except that the last element is one, \( N \) is the total number of states of the Markov chain, \( I \) is a \( N \)-by-\( N \) identity matrix, \( P \) is the one-step transition matrix, \( \Lambda \) is a matrix with the first \( (N - 1) \) elements of the diagonal set equal to one, and other elements are zero, \( U \) is \( N \)-by-\( N \) zero matrix except that all elements in the last column are ones. \([\cdot]^{-1}\) represents the inverse of a matrix. Thus, the average number of channels used for the data transmissions is

\[
\mathcal{T} = \sum_{i \in \Theta} n_i \pi_i
\]

(10)

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

The average throughput of the system is

\[
Th = r \cdot \sum_{i \in \Theta} n_i \pi_i
\]

(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 involving in the data transmissions to the total number of channels in the system, which is

$$\mu = \frac{\sum_{i \in \Theta} n_i \pi_i}{L}$$  \hspace{1cm} (12)

where $L$ is the total number of channels in the system.

VI. Numerical Results

We evaluate the performance of the proposed DSMMAC in terms of system throughput, channel utilization, and fairness for both single hop and multi-hop network scenarios using an event-driven C code simulator. In the single-hop scenario, all users are within the one-hop range of any other one. We use the hopping sequence $H=[C_1 C_1 C_2 C_1 C_2 C_2 C_2]$ designed in Section III-B. $C_1$ and $C_2$ represent two available channels. In the multi-hop scenario, some users are more than one-hop away from other ones and thus it is possible for multiple S-D pairs to communicate over the same channel simultaneously without generating interferences to each other. We consider a system with 8 channels and apply a hopping sequence with a cycle of 73, derived in Section III-B. In both cases, the system consists of 200 users, which are randomly distributed in a $2 \times 2$ km area. All S-D pairs (flows) are randomly selected within one-hop neighbors in the system. The number of flows range from 1 to 20. We repeat each experiment for 20 runs with different random seeds and calculate the average value. The confidence intervals with 95% confidence level are given to indicate the reliability of the simulation results. The main system parameters are listed in Table II.

Figures 4 and 5 show the performance of the proposed DSMMAC in the single-hop scenario. From Figure 4, we observe that the system throughput increases with the duration of each data transmission. This is because if an S-D pair successfully negotiates, they can occupy the channel for a longer time and thus improve the transmission efficiency since less overhead involved in each data transmission.

Figure 5 shows the channel utilization with different number of flows. Due to various nice properties of DSMMAS (i.e., parallel rendezvous, asynchronization, etc), multiple S-D pairs attempting to access the media are efficiently separated in the frequency and time domains. Therefore, the increase of flow number does not negatively impact the channel utilization. In fact, the probability of starting a data transmission over a channel (and thus the channel utilization) increases with the number of flows. We also notice that channel utilization increases with the data transmission duration.

Figures 4 and 5 also show that the simulation results match well with our analytical model, especially when the average transmission time $T$ is large. This is because the inaccuracy introduced by converting the continuous exponential distribution to discrete geometric distribution becomes negligible when $T$ is large.

Figures 6 to 10 show the performance in a multi-hop scenario. To further verify the efficiency of the proposed DSMMAC, we compare it with the multi-channel MAC based on the hopping sequence proposed in [16] (Denoted as CompScheme). In CompScheme, a pre-defined hopping sequence is adopted by all users to reduce the overhead of exchanging hopping information among users and release the synchronization requirement. The comparison between our algorithm and CompScheme should be fair due to similar features and objectives of both algorithms.

Figures 6 and 7 show the impacts of the flow number on the system throughput and channel access delay. Here, channel access delay is defined as the average duration from the time instant that a source attempts to access the channel to the time it successfully communicates with its destination. From Figure 6, it is observed that the total system throughput increases with the number of flows in both DSMMAC and

TABLE II

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel number $L$</td>
<td>2, 8</td>
<td>SIFS</td>
<td>16 $\mu$s</td>
</tr>
<tr>
<td>Flow number $F$</td>
<td>1 – 20</td>
<td>DIFS</td>
<td>34 $\mu$s</td>
</tr>
<tr>
<td>PHY preamble</td>
<td>192 bits</td>
<td>RTS</td>
<td>20 bytes</td>
</tr>
<tr>
<td>Switch time $T_{\text{max}}$</td>
<td>20 $\mu$s</td>
<td>CTS</td>
<td>14 bytes</td>
</tr>
<tr>
<td>Slot duration</td>
<td>418 $\mu$s</td>
<td>RTS/CTS rate</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Mean of transmission duration $T$</td>
<td>5 – 20(slots)</td>
<td>Data rate $r$</td>
<td>54 Mbps</td>
</tr>
</tbody>
</table>

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

Fig. 5. Channel utilization with different number of flows.
CompScheme, due to the nice features of multiple rendezvous
and asynchronous transmissions. Meanwhile, the proposed
DSMMAC always outperforms CompScheme with different
numbers of flows. The system throughput of DSMMAC is
twice of that of CompScheme when the number of flow is 10.
In addition, the channel access delay increase lightly with the
increasing of flow number. With a larger number of flows, the
probability that an S-D pair fails to negotiate due to channel
busy increases, which leads to a longer access delay. We
can see that DSMMAC achieves about 150% reduction in
channel access delay compared with CompScheme.

Fairness is an important performance metric to evaluate
a MAC protocol. We adopt the Jain Fairness Index [17] to
evaluate the fairness performance of the proposed DSMMAC,
which is defined as
\[ I = \frac{\left( \sum_{f=1}^{F} (Th_f)^2 \right) / (F \sum_{f=1}^{F} (Th_f))^2}{(F \sum_{f=1}^{F} (Th_f))^2} \],
where F is the total number of flows, and Th_f is the achieved
throughput of the fth flow. The fairness performance of
DSMMAC is shown in Figure 8, where the number of flows
in the system is set from 1 to 10. It is observed that the
values of fairness index of DSMMAC approach 1. All flows
have roughly equal chances to access the channel and the
achieved flow throughputs are close to each other. In addition,
the fairness performance does not degrade with the increase
of flow numbers, which shows the good scalability of the
proposed DSMMAC.

The achieved throughput of each flow is shown in Figure 9.
It is observed that each flow achieves similar throughput using
the same data transmission duration. It can also be seen that the
achieved throughput increases with a longer data transmission
duration.

We also investigate the achieved throughput over each
channel, which is shown in Figure 10. It is observed that the
achieved throughput over channels C_2 through C_8 are similar,
which demonstrates that DSMMAC protocol can achieve a
fair channel utilization and throughput over these channels by
nicely selecting difference sets and making channels be equally
allocated in the hopping sequence. It is also seen that channel
C_1 achieves a little bit higher throughput than other channels.
This is because we allocate channel C_1 to the remaining time
slots in the hopping sequence design, which makes C_1 be used
as a hopping channel more frequently than other ones, such
that more rendezvous occur over C_1 and result in a higher
throughput.

VII. RELATED WORK

Protocol design for multi-channel system has been widely
studied in the literature [4] - [10]. Dedicated control channel
is commonly used in many proposed media access control
(MAC) protocols. A dynamic channel assignment scheme is
proposed in [5], where one channel is dedicated for control
message exchange. For a small number of available channels,
the network performance is highly limited by the exclusive
use of the control channel. Moreover, the control channel may
become bottleneck when the control load is heavy.
the achievable network capacity. Meanwhile, clock synchro-
transmission is allowed in other data channels, which limit s
all nodes. In addition, during the control phase, no data
it still requires time synchronization of the two phases amo ng
transceiver, which reduces the cost and complexity. However,
in the previous control phase. MMAC requires only one
control messages. During the data phase, all channels can be
phase, all nodes switch to the control channel to exchange
alternating sequence of control and data phases. In the cont rol
phase, phase, this work, and severe contention may
arise due to the common hopping sequence used by all nodes.

Multi-channel MAC (MMAC) in [8] divides the time into an
alternating sequence of control and data phases. In the control
phase, all nodes switch to the control channel to exchange
control messages. During the data phase, all channels can be
used for data transmissions based on the agreements made in
in the previous control phase. MMAC requires only one
transceiver, which reduces the cost and complexity. However, it
still requires time synchronization of the two phases among
all nodes. In addition, during the control phase, no data
transmission is allowed in other data channels, which limits
the achievable network capacity. Meanwhile, clock synchro-
nization among all users is difficult to achieve, especially in
large scale distributed networks.

Some other MAC protocols do not require synchronization,
but need the hopping information of the destinations, which
may not be easily obtained. A multiple rendezvous MAC
protocol is proposed in [9], where each node follows multiple
hopping sequences in a time-multiplexed manner. When a
node attempts to initiate transmissions to another node, it waits
in a channel until their rendezvous arises over this channel.
However, to make rendezvous happen, the sources needs to
know their destination’s current hopping sequences via a seed
broadcast mechanism, which increases the signaling overhead
and reduce the protocol performance. Moreover, the reliability
of broadcast transmissions cannot be guaranteed over an error-
prone wireless channel due to the lack of acknowledgment
mechanisms.

VIII. CONCLUSIONS AND DISCUSSIONS

In this paper, we have proposed a difference set based
asynchronous MAC protocol for multi-channel wireless net-
works. By allowing all users to use the same hopping sequence
derived from difference sets, multiple source-destination pairs
can rendezvous over different channels simultaneously in a
distributed manner, which achieves high system throughput,
low access delay, and good fairness among users. Compared
with previous proposed protocols, our protocol does not re-
require a dedicated control channel and there is no need for
global synchronization among users. It is thus quite promising
for practical deployment.

There are several interesting directions to extend this work.
For instance, cognitive radio network is essentially a multi-
channel system where the available frequency bands vary with
the behavior of primary users and a common control channel is
usually not available. The proposed DSMMAC can be readily
extended to the promising cognitive radio networks.

Another direction is to apply service differentiation mech-
anism in DSMMAC to provide quality of service (QoS) in
a heterogenous network. In this case, diverse rendezvous
probabilities of different users over multiple channels should
be considered, based on users’ QoS requirements.

REFERENCES

802.11-based multi-channel wireless mesh network,” in Proc. IEEE
INFOCOM, USA, March 2005.


ring MAC protocol for inter-vehicle communications”, in Proc. IEEE
GLOBECOM, USA, Nov. 2008


MAC protocol with on-demand channel assignment for mobile ad hoc
networks,” in Proc. Int. Symp. Parallel Architectures, Algorithms and
Networks (ISPAN), USA, Dec. 2000.

[6] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, “Channel-hopping mul-
tiple access”, in Proc. IEEE ICC, USA, June 2000.

protocol for IEEE 802.11 ad hoc wireless LANs”, in Proc. IEEE
Personal, Indoor and Mobile Radio Communications (PIMRC), China,

multi-channel hidden terminal using a single transceiver”, in Proc. ACM

hopping for capacity improvement in IEEE 802.11 ad hoc wireless

multi-channel MAC protocol”, in Proc. IEEE Wireless Comm. and
Networking Conf. (WCN), Hong Kong, March 2007.

protocol interference model for multi-hop wireless networks,” in Proc. 
 Mobihoc, USA, May 2009.

[12] L.D. Baumont, Lecture Notes in Mathematics, Cyclic Difference Sets, 

[13] Ian Aderson, Combinatorial Designs and Tournaments, Oxford Univer-

chronous wake-up schedulers and its applications in multi-hop wireless
networks”, IEEE Trans. on Mobile Computing, pp. 1228-1241, vol. 5, 
nos. 9, Sept. 2006.

and rendezvous for mobile sensing applications”, in Proc. ACM Conf.

spectrum access”, in Proc. IEEE Dynamic Spectrum Access Networks
(DsSPAN), USA, Oct. 2008.

[17] R. Jain, D. Chiu, and W. Hawe, “A quantitative measure of fairness and
discrimination for resource allocation in shared computer system,” DEC