IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 26, 2017, accepted February 24, 2017, date of publication March 1, 2017, date of current version March 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2676126

Cluster-Based Control Information Exchange in Multi-Channel Ad Hoc Networks With Spectrum Heterogeneity

XUESONG JONATHAN TAN[1] AND WEN ZHAN[2]
[1]National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Department of Electronic Engineering, City University of Hong Kong, Hong Kong 999077, China

Corresponding author: X. J. Tan (xtan@uestc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61471104 and Grant 61001085 and in part by the National High Technology Research and Development Program (863 Program) of China under Grant 2015AA01A705 and Grant 2014AA01A701.

ABSTRACT To overcome the constraint of spectrum heterogeneity, i.e., different spatial locations may have different available spectrum resources, nodes in a multi-channel ad hoc network (MCAHN) should exchange necessary control information. To facilitate this exchange, this paper first develops a novel distributed mechanism for MCAHNs to randomly aggregate the topology and spectrum information (TSI) of all network nodes into a unique one, then proposes two heuristic algorithms for the randomly selected node to perform clustering by solving a constrained set covering problem (SCP), and finally establishes a Hamiltonian cycle over the resulting clusters to afford an ordered flow of inter-cluster control information. Numerical simulation shows that, compared with the existing mechanisms under heterogeneous spectrum availability, the proposed distributed mechanism of information aggregation is more efficient in both time and energy consumption, while the proposed SCP-based clustering algorithms yield a better tradeoff between the efficiency and robustness for cluster-based control information exchange. Moreover, compared with the existing mechanism of control information exchange, the proposed mechanism based on a cluster-based Hamiltonian cycle incurs less packet collisions as well as shorter time delay and is more suitable to provide the quality of service guarantee for various types of traffic.

INDEX TERMS Ad hoc network, Hamiltonian cycle, set covering, spectrum heterogeneity.

I. INTRODUCTION

Being multi-hop, self-organizing, and lacking of infrastructure, ad hoc networks can flexibly utilize multiple disjoint spectrum channels for communications under distributed control and are especially suitable for various applications with changeable topology, such as device-to-device (D2D) communications, vehicle networking, emergency rescue, and military surveillance. In traditional ad hoc networks, neighboring nodes normally share a common availability for their accessible spectrum channels [1]. However, the increasingly wide deployment of wireless networks with fixed infrastructure, e.g., micro, pico, and femto cells, is making an increasingly complex electromagnetic environment for newly emerging ad hoc networks, e.g., machine-to-machine (M2M) networks [2], vehicle ad hoc networks (VANETs) [3], and cognitive radio ad hoc networks (CRAHNs) [4]. As the radio coverage of various cells becomes gradually small, neighboring ad hoc nodes under their coverage may have heterogeneous spectrum resources [5].

For example, Fig. 1 depicts a multi-hop CRAHN under the coverage of multiple primary base stations (PBSs). Because the interference to each PBS at its licensed channels is always prohibited, neighboring ad hoc nodes, e.g., N_C and N_D, under the coverage of two PBSs, i.e., PBS_2 and PBS_1, respectively, have to shun from the licensed channels, i.e., channels 1 and 2, being occupied by these two PBSs and hence have different sets of available channels, i.e., {2, 3} and {1, 3}. In particular, if N_C and N_D tune their radio transceivers to channels 2 and 1, respectively, then they can never communicate although they are within the radio coverage of each other.

To handle with this difficulty of spectrum management, a multi-channel ad hoc network (MCAHN) normally has to exchange a large amount of control information ([6], [7]),

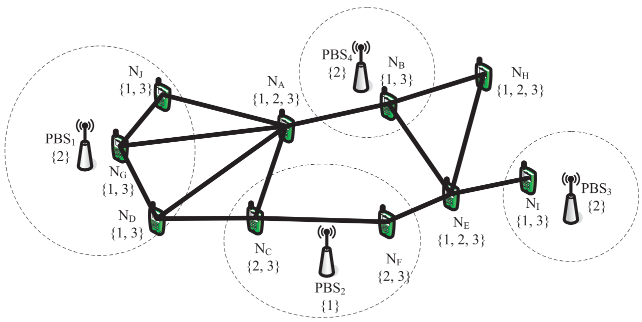


FIGURE 1. **A multi-channel ad hoc network (MCAHN) with heterogeneous spectrum resource, where each number below a primary base station (PBS) labels the licensed channel it is occupying and that below an ad hoc node labels one of its available channels.**

such as locally available spectrum channels, network topology, time clock, and channel reservation, among network nodes for negotiating appropriate communication opportunities. To facilitate this exchange, most existing media access control (MAC) protocols, e.g., [8]–[11], require an MCAHN to exploit a global control channel (GCC), which is available for all ad hoc nodes and free of interference. However, the probability for an MCAHN to establish such a GCC in the environment of spectrum heterogeneity is very small.

To overcome this drawback, one typical non-GCC approach is to divide all nodes in an MCAHN into multiple clusters, each consisting of neighboring nodes with similar spectrum resource, and establish at least one local control channel (LCC) for each cluster to exchange intra-cluster control information. While most traditional clustering algorithms [12] only apply to MCAHNs with homogeneous spectrum availability, various clustering methods ([13]–[18]) have been designed for MCAHNs with heterogeneous spectrum availability. For example, [13] and [14] propose two distributed methods for neighboring nodes to negotiate a LCC recursively and establish a cluster over those nodes with the LCC being available. However, recursive negotiation is time-consuming and incurs high communication overheads. To reduce the clustering overhead, [15] initializes the clusters of an MCAHN by distributed beacon broadcast such that each node, which has successfully broadcasted a beacon, will become a cluster head (CH) and each of the remaining nodes will join one neighboring CH as a member node. Since the initial clustering is not optimal, each CH will continue to optimize its host and neighboring clusters by searching a dominating set (DS) of nodes as new CHs. However, this approach is still sub-optimal because the existing CHs randomly selected for performing DS-based optimization are generally not the most appropriate ones. This increases the number of clusters and also the overhead for ensuing cluster-based control information exchange. Meanwhile, [16] proposes to first execute the DS-based clustering at each spectrum channel and then merge the clustering results at all channels into one, which requires an extra overhead for cluster merging and may again yield a relatively large number of clusters. To further reduce the number of clusters, [17] proposes to form multi-hop clusters, instead of one-hop clusters in [13]–[16], such that part of member nodes in a cluster may be multi-hop away from the CH. While the clustering result in [17] can reduce the overhead for inter-cluster exchange of control information, it incurs extra overheads for cluster generation and intra-cluster exchange of control information. Moreover, the clustering approaches in [13]–[17] share a common problem of robustness, i.e., each resulting cluster is equipped with one LCC only, at which the interference can easily interrupt the control information exchange within this cluster. On the other hand, the more LCCs in each cluster, the smaller the average number of nodes in each cluster, the larger the average number of clusters in an MCAHN, and the less efficient the inter-cluster exchange of control information. Thus there exists a tradeoff in the robustness and efficiency of cluster-based control information exchange.

To achieve this tradeoff, [18] formulates two heuristic algorithms, namely spectrum opportunity clustering (SOC) and constrained-SOC (C-SOC), based on bipartite graph theory. However, both SOC and C-SOC are inefficient in the sense that every node in an MCAHN has to perform clustering computation and fully exchange its clustering decision with all its neighbors for at least one time. Moreover, given a required minimal number of LCCs in each cluster, the average number of clusters generated by SOC or C-SOC is far beyond the minimum. In view of this, the first objective of this paper is to design efficient clustering algorithms for achieving a better tradeoff between the efficiency and robustness for cluster-based control information exchange.

Intuitively, the optimality of a clustering approach depends on the completeness of the topological and spectrum information (TSI) it aggregates from an MCAHN. As the existing approaches ([13]–[15], [18]) only utilize the TSI of one-hop neighboring nodes for clustering, their clustering results are far from the optimal. To increase the clustering optimality, it is necessary to aggregate the TSI of an MCAHN as complete as possible before clustering. While most existing algorithms for information aggregation [19] only apply to ad hoc networks with homogeneous spectrum availability, the distributed coordination protocol (DCP) [17] aggregates the network TSI under heterogeneous spectrum availability. In DCP, each CH first floods a common channel invitation (CCI) message to inform all nodes within a certain number of hops to report their TSI. After a node receives a CCI for a certain time length, it will begin to report its collected TSI back to the CH along the progression route of the CCI. One problem of this algorithm is that both CCI flooding and TSI reporting incur unnecessary packet transmissions and collisions. Moreover, the centralized control for TSI aggregation normally requires an extra overhead to establish and maintain the routes for TSI reporting and may not be robust enough for adapting to the dynamically changing network environment. In view of this, the second objective of the present paper is to develop an efficient and robust distributed mechanism to aggregate the complete TSI of an MCAHN with heterogeneous spectrum availability.

Once an MCAHN is clustered, it should adopt a specific mechanism for control information exchange among all clusters. However, the literatures [13], [14], [17], and [18] do not consider the design of this mechanism. In [15], a cluster-based MCAHN can maintain cluster structure, transmit data, sense channel status, and exchange intra- and inter-cluster control information in each time frame. Since the inter-cluster exchange of control information normally requires the hand-off of control channels between neighboring clusters with different LCCs and experiences a relatively long time delay for multi-hop transmission, it yields the major overhead for cluster-based control information exchange in an MCAHN. Thus the third objective of this paper is to devise an efficient mechanism for the inter-cluster exchange of control information.

In summary, the present paper is devoted to affording a complete solution for cluster-based control information exchange in an MCAHN with heterogeneous spectrum availability. In this solution, we first propose a distributed mechanism for aggregating the TSI of all nodes to a randomly selected one, then develop two greedy heuristic algorithms for the selected node to perform clustering by solving a constrained set covering problem (SCP) based on the aggregated TSI, and finally design an efficient mechanism for exchanging the inter-cluster control information along a directed approximated Hamiltonian cycle established over all clusters. The main contributions of this paper are summarized as follows:

- Compared with the DCP in [17], the proposed distributed mechanism for TSI aggregation is more robust in the sense that it does not have to establish and maintain any fixed route for TSI reporting. Numerical simulation also shows that the proposed mechanism incurs a shorter average time delay as well as a smaller average number of packet transmissions and collisions than the DCP and hence is more efficient in both time and energy consumption for TSI aggregation.
- Compared with the existing clustering algorithms ([15], [16], [18]) for MCAHNs with heterogeneous spectrum availability, the proposed two SCP-based ones yield a smaller average number of clusters subject to a required minimal number of LCCs within each cluster and hence achieve a better tradeoff between the efficiency and robustness for cluster-based control information exchange. Moreover, as our algorithms only require one randomly selected node to perform clustering computation for one time, they are more efficient than SOC and C-SOC in [18], which require every node to perform clustering computation and fully exchange its clustering decision with all its neighbors for at least one time.
- Compared with the mechanism in [15], the proposed Hamiltonian-cycle-based mechanism incurs less packet collisions and shorter average time delay for the inter-cluster exchange of control information and hence provides a better quality of service (QoS) guarantee for various types of traffic. Numerical simulation also shows that our mechanism offers a reasonable robustness and hence incurs limited maintenance overheads in dynamically changing spectrum environment.

The remaining of this paper is organized as follows. Section II prepares the system model. Section III first presents the distributed mechanism for aggregating the network TSI into a randomly selected node, based on which the selected node can perform the SCP-based clustering in Section IV. Then Section V proposes the mechanism for exchanging inter-cluster control information along a Hamiltonian cycle over the resulting clusters. Finally, Section VI compares the existing and proposed mechanisms via numerical simulation and Section VII concludes the main outcome of this paper.

II. SYSTEM MODEL

The MCAHN in this paper is abstracted as a connected graph $(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the set of all nodes and $\mathbf{E}$ the set of communication links among the nodes in $\mathbf{V}$. All nodes in $\mathbf{V}$ can access a common set $\Phi = \{Ch_1, Ch_2, \ldots, Ch_K\}$ of K disjoint spectrum channels, each with a uniform bandwidth, and always keep a fixed transmission range by adjusting its transmission power over different channels. Denote by $\mathrm{AC}_i \in \Phi$ the set of locally available channels at each node $\mathrm{N}_i \in \mathbf{V}$. The spectrum heterogeneity of the MCAHN implies that any two sets AC_i and AC_k for $i \neq k$ may be different. If there exists a communication link in $\mathbf{E}$ between N_i and N_k, then these two nodes should be within the transmission range of each other and have at least one commonly available channel, i.e., $\mathrm{AC}_i \cap \mathrm{AC}_k \neq \emptyset$. For example, the MCAHN of Fig. 1 has $\mathbf{V} = \{\mathrm{N}_A, \mathrm{N}_B, \mathrm{N}_C, \mathrm{N}_D, \mathrm{N}_E, \mathrm{N}_F, \mathrm{N}_G, \mathrm{N}_H, \mathrm{N}_I, \mathrm{N}_J\}$, $\Phi = \{Ch_1, Ch_2, Ch_3\}$, and $K = 3$. In particular, as N_C and N_D are within the transmission range of each other and $\mathrm{AC}_C \bigcap \mathrm{AC}_D = \{Ch_3\}$, there exists a communication link in $\mathbf{E}$ between them.

In the initialization of an MCAHN, each node N_i first senses all channels in Φ to obtain the set AC_i of locally available channels, then discovers its one-hop neighbors in a distributed fashion, e.g., [18] for synchronized nodes and [20] for unsynchronized ones, and finally exchanges local sensing result with its neighbors. At the end of neighbor discovery, each N_i should have a set NB_i of its neighboring nodes and a set AC_k of available channels for each neighbor $\mathrm{N}_k \in \mathrm{NB}_i$. This local TSI at each node will be utilized for the MCAHN to first perform clustering and then establish a Hamiltonian cycle in a distributed and efficient fashion.

III. DISTRIBUTED MECHANISM FOR TOPOLOGY AND SPECTRUM INFORMATION AGGREGATION

A. TOPOLOGY AND SPECTRUM INFORMATION TABLE

After neighbor discovery, each N_i initializes a local TSI Table (TSIT), denoted by $\mathrm{TSIT}_i^{(0)}$. To aggregate the complete TSI of an MCAHN, each N_i should transmit its TSIT to at least one of its neighbors. On the other hand, once N_i successfully receives a TSIT that is destined to it, it will always merge the received TSIT with the local TSIT into a

Destination Node (DN): $\mathrm{N_C}$			
Source Node (SN)	Available Channels of SN	Neighbors of SN (NB)	Available Channels of NB
$\mathrm{N_D}$	1, 3	$\mathrm{N_A}$	1, 2, 3
		$\mathrm{N_C}$	2, 3
		$\mathrm{N_G}$	1, 3

(a)

Destination Node (DN): (Void)			
Source Node (SN)	Available Channels of SN	Neighbors of SN (NB)	Available Channels of NB
$\mathrm{N_C}$	2, 3	$\mathrm{N_A}$	1, 2, 3
		$\mathrm{N_D}$	1, 3
		$\mathrm{N_F}$	2, 3

(b)

Destination Node (DN): (Void)			
Source Node (SN)	Available Channels of SN	Neighbors of SN (NB)	Available Channels of NB
$\mathrm{N_C}$	2, 3	$\mathrm{N_A}$	1, 2, 3
		$\mathrm{N_D}$	
		$\mathrm{N_F}$	2, 3
$\mathrm{N_D}$	1, 3	$\mathrm{N_A}$	
		$\mathrm{N_C}$	
		$\mathrm{N_G}$	1, 3

(c)

FIGURE 2. In the MCAHN of Fig. 1, upon receiving (a) $\mathrm{TSIT}_D^{(0)}$ transmitted by N_D, N_C will merge it with the local (b) $\mathrm{TSIT}_C^{(0)}$ into a new table (c) $\mathrm{TSIT}_C^{(1)}$. For the compactness of information storage, $\mathrm{TSIT}_C^{(1)}$ only shows the set of available channels for N_A, N_C, N_D, N_F, or N_G once.

new TSIT. Denote by $\mathrm{TSIT}_i^{(t)}$, $t \geq 1$, the TSIT generated by N_i right after N_i successfully receives the t^{th} TSIT that is destined to it.

Each $\mathrm{TSIT}_i^{(t)}$, $t \geq 0$, should include at least five types of information: the destination node of $\mathrm{TSIT}_i^{(t)}$, the set $\mathrm{SN}_i^{(t)}$ of source nodes that contribute to the content of $\mathrm{TSIT}_i^{(t)}$, the set of available channels for each $\mathrm{N}_i \in \mathrm{SN}_i^{(t)}$, the set NB_i of neighbor nodes for each N_i, and the set of available channels for each $\mathrm{N}_k \in \mathrm{NB}_i$. Among these information, the destination node of $\mathrm{TSIT}_i^{(t)}$ should be a neighbor of N_i and is normally determined right before the transmission of $\mathrm{TSIT}_i^{(t)}$. Meanwhile, the remaining four types of information can be classified into $|\mathrm{SN}_i^{(t)}|$ rows, each corresponding to a different $\mathrm{N}_j \in \mathrm{SN}_i^{(t)}$. In particular, because $\mathrm{TSIT}_i^{(0)}$ is initialized by N_i, $\mathrm{SN}_i^{(0)} = \{\mathrm{N}_i\}$. For example, Fig. 2(a, b) depict $\mathrm{TSIT}_D^{(0)}$ and $\mathrm{TSIT}_C^{(0)}$, respectively, which are initialized by the nodes N_D and N_C in Fig. 1, i.e., $\mathrm{SN}_D^{(0)} = \{\mathrm{N}_D\}$, $\mathrm{NB}_D^{(0)} = \{\mathrm{N}_G, \mathrm{N}_C, \mathrm{N}_A\}$, $\mathrm{SN}_C^{(0)} = \{\mathrm{N}_C\}$, and $\mathrm{NB}_C^{(0)} = \{\mathrm{N}_A, \mathrm{N}_D, \mathrm{N}_F\}$. On the other hand, if N_i generates $\mathrm{TSIT}_i^{(t)}$, $t \geq 1$, by merging the local $\mathrm{TSIT}_i^{(t-1)}$ with a received $\mathrm{TSIT}_j^{(\tau)}$, $\tau \geq 0$, then the set $\mathrm{SN}_i^{(t)}$ should at least include N_i and N_j. Fig. 2(c) depicts the $\mathrm{TSIT}_C^{(1)}$, which is generated by merging $\mathrm{TSIT}_C^{(0)}$ with $\mathrm{TSIT}_D^{(0)}$ and has $\mathrm{SN}_C^{(1)} = \{\mathrm{N}_C, \mathrm{N}_D\}$.

B. A MULTI-CHANNEL TSIT TRANSMISSION SCHEME

To facilitate the TSIT exchange under heterogeneous spectrum availability, we formulate a multi-channel TSIT transmission scheme without the aid of any global common channel (GCC). In this scheme, each node N_i independently sets the length T_i of a local timeslot as $T_i = |\mathrm{AC}_i| \triangle t$, where $|\mathrm{AC}_i|$ is the number of available channels of N_i and $\triangle t$ is the length of a minislot. Normally, $\triangle t$ should be long enough for N_i to first hop from one channel to another, then transmit a TSIT, and finally receive an ACK replied by its receiver. For all nodes, the length $\triangle t$ is a common knowledge.

Once N_i decides to transmit its TSIT, it should first randomly select an initial delay of $d_i \in [0, W-1]$ timeslots, where W is the size of contention window, to reduce the collision probability of TSIT transmissions among neighboring nodes and then begin to hop among the $|\mathrm{AC}_i|$ local available channels according to a certain hopping sequence. In each minislot of this delay, N_i should first hop to a new available channel, say $Ch_c \in \mathrm{AC}_i$, and then keep listening at the channel Ch_c until the end of this minislot according to the following rule:

(1) Once N_i successfully receives a table $\mathrm{TSIT}_j^{(\tau)}$, of which the destination node is N_i, it will first reply an ACK to the sender of $\mathrm{TSIT}_j^{(\tau)}$ and then merge $\mathrm{TSIT}_j^{(\tau)}$ with the local TSIT into a new one.

This process of channel hopping and listening can prevent N_i from staying at a channel, which is not available for its neighbors, too long and help N_i to receive a TSIT without knowing the exact transmission channel.

At the end of the d_i-timeslot delay, N_i will choose one of its neighbors as the destination node of its TSIT according to the following rule:

(2) Among all its neighboring nodes, N_i should always prefer the ones that it has not yet exchange TSITs with. Moreover, if N_i has already exchanged TSIT with all its neighbors, then it will randomly choose one of its neighbors for transmitting TSIT.

This rule is for each N_i to exchange TSIT with its neighbors as many as possible and therefore accelerate the distributed aggregation process of network TSI.

Given that a neighbor node $\mathrm{N}_k \in \mathrm{NB}_i$ has been selected for TSIT transmission, N_i will randomly choose a channel Ch_c from the set $\mathrm{AC}_i \cap \mathrm{AC}_k$ for TSIT transmission. In the first minislot of TSIT transmission, N_i should first handoff to the channel Ch_c, then transmit its newest TSIT, and finally listen at this channel until the end of this minislot. If N_i receives the ACK from N_k, it will stop the TSIT transmission immediately; else, if the TSIT retransmission times has not yet reached a preset maximum value, N_i will retransmit its TSIT over the same channel in the next minislot; else, it will restart the TSIT transmission by reselecting an initial delay d_i.

In general, the selection of the maximal value for TSIT retransmission times depends on whether N_i knows the channel hopping sequence of N_k and whether the time clocks of N_i and N_k are synchronized. For instance, if N_i knows the channel hopping sequence of N_k from neighbor discovery and the time clocks of N_i and N_k are synchronized, then N_i can know the exact minislot for N_k to appear at Ch_j and hence transmit TSIT only in this minislot; else, N_i has to preset a larger limit of TSIT retransmissions so as to make sure that N_k can finally hop to the channel Ch_j for receiving TSIT.

For example, Fig. 3 depicts the possible scenarios of TSIT transmission in Fig. 1, where the node N_D with $\mathrm{AC}_D = \{Ch_1, Ch_3\}$ transmits its TSIT to the node N_C with

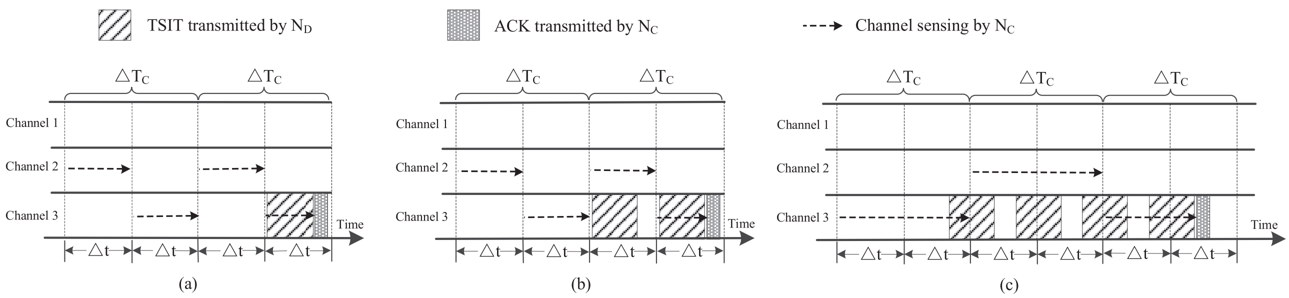


FIGURE 3. In the transmission of TSIT from N_D to N_C over Ch_3 in Fig. 1, the timetable of the multi-channel TSIT transmission scheme is illustrated if N_D and N_C have (a) a synchronized time clock and known channel hopping sequences; (b) a synchronized time clock and unknown channel hopping sequences; and (c) asynchronous time clocks.

$AC_C = \{Ch_2, Ch_3\}$ over the selected channel Ch_3. In the case of synchronized time clock, if N_D knows the exact hopping sequence of N_C, it only needs to transmit its TSIT to N_C in the minislot when N_C hops to Ch_3, as shown in Fig. 3(a); else, at least $|AC_C| = 2$ times of TSIT transmissions in Fig. 3(b) are necessary. On the other hand, Fig. 3(c) illustrates that, in the case that N_C and N_D have asynchronous time clocks, if N_C chooses a channel hopping sequence such that it has to listen at one channel within AC_C for two consecutive minislots before it can hop to a new channel, then $4(= 2|AC_C|)$ times of TSIT transmissions will still be enough for N_D. This example can be generalized into the following theorem:

Theorem 1: In the proposed TSIT transmission scheme, if a TSIT receiver N_R periodically hops among all its $|AC_R|$ available channels by always listening at each available channel for two consecutive minislots, then its neighboring node N_T can always send a TSIT successfully to it by repeatedly transmitting the TSIT at any channel $Ch_j \in AC_T \bigcap AC_R$ for a maximal number of $2|AC_R|$ retransmissions, no matter whether N_T and N_R are synchronized or not.

Proof: Without the loss of generality, assume that the first appearance of N_T at the channel Ch_j lag that of N_R at the same channel for δ minislots, where δ can be any nonnegative real number. If $\delta \in [0, 2 \triangle t - t_{TSIT}]$, where t_{TSIT} is the time length for N_T to transmit a TSIT, then N_R will successfully receive the TSIT transmitted by N_T during its first appearance at the channel Ch_j; else, if $\delta \in (2[(k-1)|AC_R| + 1] \triangle t - t_{TSIT}, 2(k|AC_R|+1)\triangle t - t_{TSIT}]$ for any positive integer k, then N_R will wait until its $(k+1)^{st}$ appearance at the channel Ch_j for receiving the TSIT successfully because it always takes a period of $2|AC_R|$ minislots for N_R to hop among all its $|AC_R|$ available channels. ■

C. DISTRIBUTED DECISION ON TSIT TRANSMISSION AND CLUSTERING QUALIFICATION

The multi-channel transmission scheme in Section III-B only specifies the operation of each node N_i during a TSIT transmission. To facilitate the distributed process of TSIT aggregation, each N_i should at least guarantee a minimum degree of TSIT exchange with its neighbor nodes. For this purpose, it should adopt the following rule:

(3) In the distributed process of TSI aggregation, each node N_i should successfully transmit its TSIT to one of its neighbor nodes for at least one time no matter whether it has received a TSIT.

This implies:

(*) During the distributed aggregation of network TSIT, there may exist multiple TSITs concurrently transmitted in a MCAHN.

On the other hand, in order to guarantee that only one randomly selected node can finally aggregate the complete TSI of the whole MCAHN and perform clustering computation, exactly one TSIT transmission should exist at the end of the TSIT aggregation process. This, together with (*), means that, when a node N_i receives a TSIT destined to it and updates its local TSIT according to the rule (1), it may or may not transmit the updated TSIT to one of its neighbors and, if not, then the total number of TSITs concurrently transmitted in the MCAHN will be reduced by 1. Under distributed control, this decision should be made based on a typical information contained by its received and local TSITs only. For this purpose, the following rule adopts the MAC addresses inside these two TSITs for each node N_i to make its decision on TSIT transmission as well as determine whether it has collected the complete TSI of the whole MCAHN:

(4) Assume that, upon receiving a $TSIT_j^{(\tau)}$ successfully, N_i generates a new table $TSIT_i^{(t)}$, $t \geq 1$, by (1). If the smallest MAC address of the nodes in the set $SN_j^{(\tau)}$ is larger than that of the nodes in the set $SN_i^{(t)}$, then N_i will continue its channel hopping without transmitting $TSIT_i^{(t)}$; else, if $SN_i^{(t)} = \bigcup_{N_q \in SN_i^{(t)}} NB_q^{(t)}$, then N_i will stop channel hopping and begin to execute clustering computation; else, it will begin to transmit $TSIT_i^{(t)}$.

Intuitively, the rule (3) guarantees that the node with the smallest MAC address, say N_s, in an MCAHN can always transmit its TSIT, of which one source node is N_s, to at least one of its neighbor nodes. By the rule (4), upon successfully receiving a TSIT with N_s being a source node, every node N_i will always generate a new TSIT with N_s being a source node by (1) and further transmit this TSIT to one of its neighbor nodes. This unicast process will continue until the first time when a TSIT with N_s being a source node reaches a node N_{i^*} that generates a new table $\mathrm{TSIT}_{i^*}^{(t)}$ with $\mathrm{SN}_i^{(t)} = \cup_{\mathrm{N}_q \in \mathrm{SN}_i^{(t)}} \mathrm{NB}_q^{(t)}$. At this time, N_{i^*} should have aggregated the complete TSI of the whole MCAHN and is qualified for executing clustering computation. Below we give a rigorous proof for the uniqueness of N_{i^*}.

Theorem 2: If every node in an MCAHN follows the rules (1)-(4) to exchange its TSIT with its neighbor nodes, then there will exist exactly one node that can aggregate the complete TSI of the MCAHN and execute clustering computation.

Proof: If a table $\mathrm{TSIT}_i^{(t)}$ generated by N_i satisfies

(5) $\mathrm{SN}_i^{(t)} = \cup_{\mathrm{N}_q \in \mathrm{SN}_i^{(t)}} \mathrm{NB}_q^{(t)}$,

then it means that each neighbor of a source node in $\mathrm{SN}_i^{(t)}$ is also a source node and hence $\mathrm{TSIT}_i^{(t)}$ has aggregated the TSI of all nodes. Thus we only need to show that (1)-(4) can always result in a unique node that generates a TSIT satisfying (5) and hence be qualified to execute clustering computation.

Let N_s be the node with the smallest MAC address in the MCAHN. From (1), before a node N_n, $\forall n \neq s$, receives a TSIT with N_s and N_n being the source and destination nodes, respectively, it can only generate a TSIT without N_s being a source node, which, however, cannot satisfy (5). On the other hand, after N_n receives such a TSIT, if N_n further receives a TSIT without N_s being a source node, it will neither check whether its TSIT can satisfy (5) nor transmit its TSIT by the rule (4). In this case, N_n still cannot perform clustering computation no matter whether its TSIT satisfies (5) or not. Thus we only need to consider those nodes N_n that are receiving a TSIT with N_s and N_n being the source and destination nodes, respectively.

Denote by TSIT(k), $k \geq 1$, the k^{th} successfully received TSIT in the MCAHN, which includes N_s as a source node, and by N(k) the destination node of TSIT(k). Thus TSIT(1) = $\mathrm{TSIT}_s^{(j)}$ for some $j \geq 0$. From (3), N_s should always transmit TSIT(1) to one of its neighbors successfully, i.e., N(1), no matter whether TSIT(1) satisfies (5). From (1) and (4), upon receiving TSIT(1), N(1) should first generate a new TSIT with N_s being a source node and then check if the new TSIT satisfies (5). If yes, N(1) will be the unique node satisfying (5) and executing clustering computation; else, it should further transmit TSIT(2) to N(2).

By induction on $k = 2, 3, \ldots$, upon receiving the table TSIT(k), where $k \geq 2$, N(k) should first generate a new TSIT with N_s being a source node and then check if the generated TSIT can satisfy (5). If yes, it will be the unique node that satisfies (5) and executes clustering computation; else, it should transmit TSIT($k+1$) to N($k+1$). Thus, at any time, there exists at most one node in the MCAHN, which can generate a TSIT satisfying (5). This fact, together with (2), implies that the sequential transmissions of TSIT(1), TSIT(2), ..., can finally yield a unique TSIT satisfying (5). ∎

From the proof of Theorem 1, along with the progression of a unique TSIT containing the smallest MAC address throughout an MCAHN, a unique node with complete network TSI will be automatically selected in a fully random fashion. Thus our mechanism for TSI aggregation is more robust than the distributed coordinated protocol in [17], which has to establish and maintain multiple fixed routes for TSI reporting.

IV. CLUSTERING BY CONSTRAINED SET COVERING

Once a unique node, say N_e, is selected and equipped with the complete TSI of an MCAHN, it will begin to execute clustering computation. As shown in [18], when an MCAHN has a smaller average number of clusters or, equivalently, a larger average cluster size, the overhead for inter-cluster exchange of control information will become lower but all nodes within each cluster will share less local common channels (LCCs). On the other hand, a larger number of LCCs in a cluster can better avoid the intra-cluster exchange of control information from being interrupted by the dynamic variation of spectrum availability and hence reduce the re-clustering overhead. Thus there exists a tradeoff between the efficiency of inter-cluster exchange of control information and the robustness of intra-cluster exchange. This section is devoted to the design of clustering algorithms for minimizing the number of resulting clusters while guaranteeing at least $m \in [1, M]$ LCCs within each cluster, where $M \in [1, K]$ denotes the smallest number of available channels for every node in an MCAHN and K the total number of channels accessible by every node in $\mathbf{V}$.

To overcome the constraint of spectrum heterogeneity, N_e can transform the original graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ into multiple subgraphs with homogeneous spectrum availability as follows. Denote by $\Phi_1^{(m)}, \Phi_2^{(m)}, \ldots, \Phi_{C_K^m}^{(m)}$ the C_K^m possible subsets of m channels in Φ and further call each of these subsets as a *spectrum layer*. For the spectrum layer $\Phi_j^{(m)}$, $j \in [1, C_K^m]$, generate a subgraph $\mathbf{G}_j^{(m)} = (\mathbf{V}_j^{(m)}, \mathbf{E}_j^{(m)})$ of $\mathbf{G}$, where the set $\mathbf{V}_j^{(m)} \subseteq \mathbf{V}$ consists of all nodes with every channel of $\Phi_j^{(m)}$ being available and the set $\mathbf{E}_j^{(m)} \subseteq \mathbf{E}$ all communication links among the nodes in $\mathbf{V}_j^{(m)}$. Through this way, the graph $\mathbf{G}$ is transformed into C_K^m subgraphs, each corresponding to a different spectrum layer. For example, Fig. 4 depicts the 3 subgraphs of the MCAHN in Fig. 1 over the 3 spectrum layers $\Phi_1^{(2)} = \{Ch_1, Ch_2\}$, $\Phi_2^{(2)} = \{Ch_2, Ch_3\}$ and $\Phi_3^{(2)} = \{Ch_1, Ch_3\}$ of $\Phi = \{Ch_1, Ch_2, Ch_3\}$.

Based on the C_K^m subgraphs, the original clustering problem for $\mathbf{G}$ can be formulated as a constrained set covering problem (SCP) [21], which searches for a group $\mathbf{S}^{(m)}$ of multiple subsets of the node set $\mathbf{V}$, i.e., $\mathbf{S}^{(m)} = \{\mathbf{S}_1^{(m)}, \mathbf{S}_2^{(m)}, \ldots, \mathbf{S}_q^{(m)}\}$ with $\mathbf{S}_k^{(m)} \subseteq \mathbf{V}$, $\forall k \in [1, q]$, to cover the node set $\mathbf{V}$, i.e., $\bigcup_{k=1}^{q} \mathbf{S}_k^{(m)} = \mathbf{V}$, subject to:

(6) For each $\mathbf{S}_k^{(m)} \subseteq \mathbf{V}$, where $k \in [1, q]$, there exists at least one subgraph $\mathbf{G}_j^{(m)}$, $j \in [1, C_K^m]$, of $\mathbf{G}$ such that $\mathbf{S}_k^{(m)} \subseteq \mathbf{V}_j^{(m)}$.

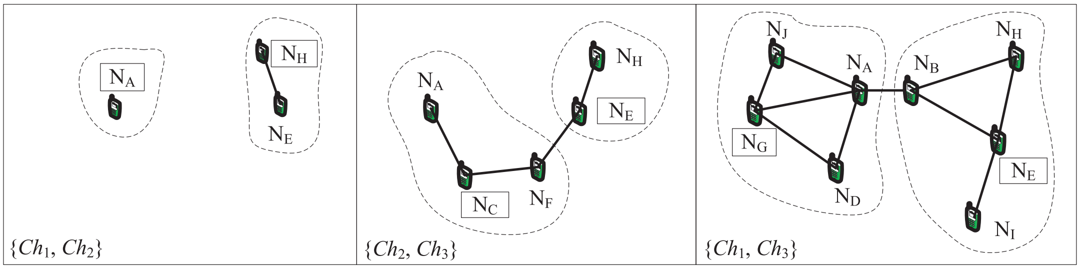


FIGURE 4. The MCAHN in Fig. 1 can be transformed into the depicted three subgraphs at the three spectrum layers $\{Ch_1, Ch_2\}$, $\{Ch_2, Ch_3\}$ and $\{Ch_1, Ch_3\}$, each of which is further clustered based on its minimum dominating set (MDS). In this figure, each cluster is surrounded by a dotted circle and each node in a square denotes a cluster head.

(7) For each $\mathbf{S}_k^{(m)}$, there exists at least one node N_h that is neighboring to all other nodes in this subset.

Here the constraints (6) and (7) together guarantee that N_h can generate from the subset $\mathbf{S}_k^{(m)}$ a cluster, of which N_h serves as a CH and the m LCCs are those channels in $\Phi_j^{(m)}$. Thus the optimization target of this section is to minimize the number of subsets in $\mathbf{S}^{(m)}$ under the constraints (6) and (7).

Obviously, N_e can exhaustively search all possible $\mathbf{S}^{(m)}$, which satisfy (6) and (7), from the C_K^m subgraphs to find the optimal one with the minimum value of $q = |\mathbf{S}^{(m)}|$. This, however, incurs high complexity. In view of this, Sections IV-A and IV-B propose two greedy heuristic clustering algorithms, both of which are more efficient than the clustering via exhaustive search at the expense of yielding a little bit more clusters than the latter. Section IV-C then compares the clustering efficiency of the proposed algorithms with the existing ones.

A. CLUSTERING BY SEARCHING MINIMAL DOMINATING SET AND BRANCH-AND-BOUND SET COVERING

The first greedy heuristic clustering is to solve the constrained SCP via the following two steps. First, perform clustering for each subgraph $\mathbf{G}_j^{(m)}$, $j \in [1, C_K^m]$, such that the node set of each resulting cluster can satisfy (6) and (7). Second, select appropriate clusters from the clustering results at all C_K^m spectrum layers to cover the graph $\mathbf{G}$. Through this way, the number of the resulting clusters can be minimized by first minimizing the number of clusters in each subgraph and then selecting a minimal number of clusters to cover the node set $\mathbf{V}$.

For the first step, a classic approach for minimizing the number of clusters in the subgraph $\mathbf{G}_j^{(m)}$ is to search a minimum dominating set (MDS) $\mathbf{D}_j^{(m)} \subseteq \mathbf{V}_j^{(m)}$ such that each node $N_i \in \mathbf{V}_j^{(m)}$ either belongs to $\mathbf{D}_j^{(m)}$ or is adjacent to a node in $\mathbf{D}_j^{(m)}$. In literature, there exist various exponential-time algorithms ([22], [23]) for finding an MDS in $\mathbf{G}_j^{(m)}$. Once N_e finds an MDS $\mathbf{D}_j^{(m)}$ from $\mathbf{V}_j^{(m)}$, it can first set each node in $\mathbf{D}_j^{(m)}$ as a cluster head (CH) and then randomly designate each $N_i \in \mathbf{V}_j^{(m)} \setminus \mathbf{D}_j^{(m)}$ as a member of one neighboring CH. Through this way, the subgraph $\mathbf{G}_j^{(m)}$ can be divided into $|\mathbf{D}_j^{(m)}|$ clusters and the total number of clusters so generated for all C_K^m subgraphs is $Q = \sum_{j=1}^{C_K^m} |\mathbf{D}_j^{(m)}|$.

Denote by $\text{Cluster}_j^{(m)}$, $j \in [1, Q]$, the j^{th} cluster resulted in the first step. Thus the problem for selecting a minimal number of clusters to cover $\mathbf{V}$ in the second step can be formulated as follows:

$$
\begin{aligned}
&\min \ \sum_{j=1}^{Q} x_j, &&(**)\\
&s.t., \ \sum_{j=1}^{Q} a_{i,j} x_j \geq 1, \quad \forall N_i \in \mathbf{V},\\
&\quad x_j \in \{0, 1\}, \quad \forall j \in [1, Q],\\
&\quad a_{i,j} \in \{0, 1\}, \quad \forall N_i \in \mathbf{V} \text{ and } j \in [1, Q],
\end{aligned}
$$

where $x_j = 1, j \in [1, Q]$, means that $\text{Cluster}_j^{(m)}$ is selected and $x_j = 0$ the opposite, while $a_{i,j} = 1$ means that the node N_i belongs to the cluster $\text{Cluster}_j^{(m)}$ and $a_{i,j} = 0$ the opposite. In particular, the constraint $\sum_{j=1}^{Q} a_{i,j} x_j \geq 1$ in (**) is to guarantee that each N_i is covered by at least one selected cluster. The optimal solution of this problem (**) can be directly obtained by applying the classic branch-and-bound algorithm [24].

Assume that the optimal solution for the problem (**) is $x_{f(1)} = x_{f(2)} = \ldots = x_{f(r)} = 1$ and every $x_j = 0$, where $j \in \{1, 2, \ldots, Q\} \setminus \{f(1), f(2), \ldots, f(r)\}$, $r \in [1, Q]$ and the function f is a mapping from $\{1, 2, \ldots, r\}$ to $\{1, 2, \ldots, Q\}$. This implies that the r clusters $\text{Cluster}_{f(1)}^{(m)}$, $\text{Cluster}_{f(2)}^{(m)}$, ..., $\text{Cluster}_{f(r)}^{(m)}$ are selected to cover $\mathbf{V}$. To solve the possible overlapping among the r selected clusters, N_e can further apply the following two rules to yield the r final clusters, to be denoted by $FC_1^{(m)}, FC_2^{(m)}, \ldots, FC_r^{(m)}$.

(8) The CH of each selected cluster $\text{Cluster}_{f(j)}^{(m)}$, $j \in [1, r]$, will become that of the final cluster $FC_j^{(m)}$ and the member nodes of this cluster, which do not belong to any other selected cluster, will join $FC_j^{(m)}$.

(9) If a node N_h is both the member node of one selected cluster $\text{Cluster}_{f(j)}^{(m)}$ and the CH of another, then it will not join $FC_j^{(m)}$; else, if N_h is concurrently a member node of multiple selected clusters, then it will randomly choose one cluster $\text{Cluster}_{f(j^*)}^{(m)}$ from them and join the final cluster $FC_{j^*}^{(m)}$.

From (8), if a node N_h is concurrently the CH of $k \in [1, r]$ clusters selected in the second step, then it will remain as the CHs of k final clusters, of which the subsets of m LCCs are

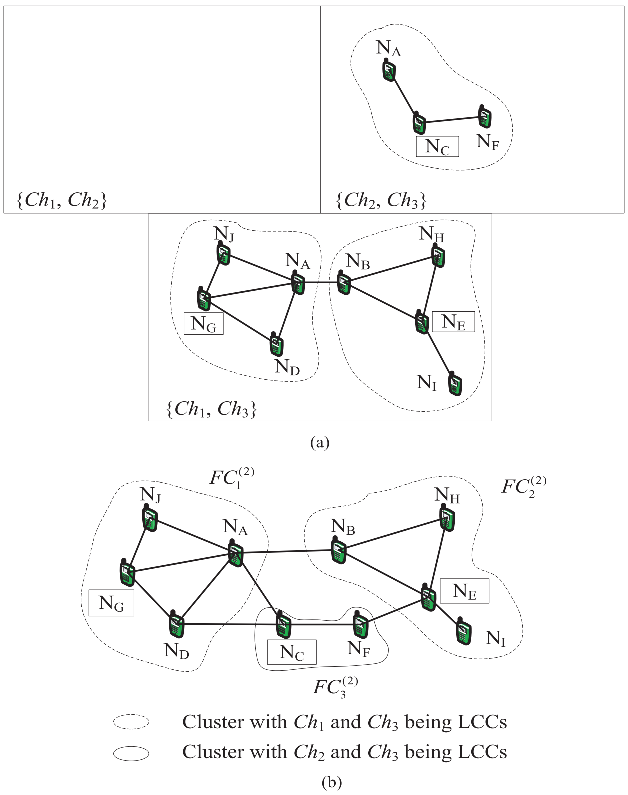


FIGURE 5. (a) The MCAHN in Fig. 1 is covered by selecting a minimum number of three clusters from the clustering result in Fig. 4; (b) After applying (8) and (9) to the three selected clusters in (a), this MCAHN is finally divided into three non-overlapping final clusters, each equipped with two local common channels (LCCs). In this figure, each node in a square denotes a cluster head.

different. This can facilitate the exchange of control information among all final clusters with N$_h$ being their CHs. Moreover, (9) guarantees that, if two final clusters share a common node, then this node wil neither be the member node of both clusters nor be the member node of one cluster and, meanwhile, the CH of the other cluster. This leads to:

Lemma 1: For any two final clusters generated by (8) and (9), if they share a common node, then this node can only be the CH of both clusters.

For example, Fig. 4 depicts an MDS-based clustering result at each spectrum layer of the MCAHN in Fig. 1. Based on this result, Fig. 5(a) then shows that a minimum number of 3 clusters selected by Step 2 of Algorithm 1 is enough to cover the whole MCAHN. This selection excludes the two clusters at the spectrum layer $\{Ch_1, Ch_2\}$ shown in the leftist part of Fig. 4 and hence yields an empty box in the upper-left part of Fig. 5(a). The application of (8) and (9) to the 3 clusters in Fig. 5(a) finally yields the ultimate clustering result in 5(b).

B. CLUSTERING BY MAXIMAL DEGREE OF CONNECTIVITY

In Algorithm 1, either finding an MDS within each subgraph of $\mathbf{G}$ or searching the fewest node sets to cover $\mathbf{V}$ is known as an NP-hard problem. To improve the clustering efficiency, we further formulate a greedy heuristic algorithm, i.e., Algorithm 2, which simply searches the CHs and their associated sets of LCCs based on the number of neighbors, i.e., the degree of connectivity at each of the C_K^m spectrum layers.

Algorithm 2 can satisfy the constraints (6) and (7) of the constrained SCP by constructing each cluster from neighboring nodes at one spectrum layer in Step 3. As Algorithm 2 only ends when every node in $\mathbf{V}$ has been covered by a constructed cluster, its resulting clusters can always cover the graph $\mathbf{G}$. For example, Fig. 6 depicts the 3 final clusters resulted by recursively applying Algorithm 2 to the MCAHN in Fig. 1. By Step 4 in Algorithm 2, all final clusters resulted by Algorithm 2 will not overlap with each other. This is different from Algorithm 1, which may yield multiple clusters with a common CH.

Note that [15] has proposed to optimize the established clusters in an MCAHN in a partially similar way as Algorithm 2. That is, to reduce the total number of its host and neighboring clusters, an existing CH, say N$_h$, first selects one of its available channels, say Ch_j, at which N$_h$ has the maximal number of neighbors, and then forms a new cluster by N$_h$ and its one-hop neighbors with Ch_j being available. After eliminating the new cluster from the MCAHN, N$_h$ can

Algorithm 1 Greedy Heuristic Clustering by Searching MDS and Branch-and-Bound Set Covering

1: For each subgraph $\mathbf{G}_j^{(m)}$ of $\mathbf{G}, j \in [1, C_K^m]$, search an MDS $\mathbf{D}_j^{(m)}$, set each node in the MDS as a CH, and randomly designate each node N$_v$ $\in \mathbf{V}_j^{(m)} \setminus \mathbf{D}_j^{(m)}$ as the member node of exactly one neighboring CH.

2: Apply the branch-and-bound algorithm [24] for solving the problem (**) so as to select a minimal number of clusters resulted by Step 1 to cover $\mathbf{V}$.

3: Apply (8) and (9) to the selected clusters in Step 2 to resolve the possible overlapping among them.

Algorithm 2 Greedy Heuristic Clustering by Degree of Connectivity

1: For each node N$_i$ $\in \mathbf{V}$, identify the number $d_i(j)$ of its neighbor nodes in each subgraph $\mathbf{G}_j^{(m)}, j \in [1, C_K^m]$. Initialize the parameter $k = 1$.

2: Search the C_K^m spectrum layers for a node N$_{i^*}$ in a subgraph $\mathbf{G}_{j^*}^{(m)}$ such that $d_{i^*}(j^*) = \max_{\mathrm{N}_i \in \mathbf{V}, j \in [1, C_K^m]} d_i(j)$. Select N$_{i^*}$ as a CH.

3: Construct a cluster $FC_k^{(m)} = (\mathbf{FV}_k^{(m)}, \mathbf{FE}_k^{(m)})$, where the set $\mathbf{FV}_k^{(m)} \subseteq \mathbf{V}$ consists of N$_{i^*}$ and its neighbor nodes in the subgraph $\mathbf{G}_{j^*}^{(m)}$ and $\mathbf{FE}_k^{(m)} \subseteq \mathbf{E}$ all communication links among the nodes in $\mathbf{FV}_k^{(m)}$.

4: Let $\mathbf{V} = \mathbf{V} \setminus \mathbf{FV}_k^{(m)}, \mathbf{E} = \mathbf{E} \setminus \mathbf{FE}_k^{(m)}, \mathbf{V}_j^{(m)} = \mathbf{V}_j^{(m)} \setminus \mathbf{FV}_k^{(m)}$, and $\mathbf{E}_j^{(m)} = \mathbf{E}_j^{(m)} \setminus \mathbf{FE}_k^{(m)}, \forall j \in [1, C_K^m]$.

5: If $\mathbf{V} \neq \emptyset$, then let $k = k + 1$ and go back to Step 2; else, end the clustering process.

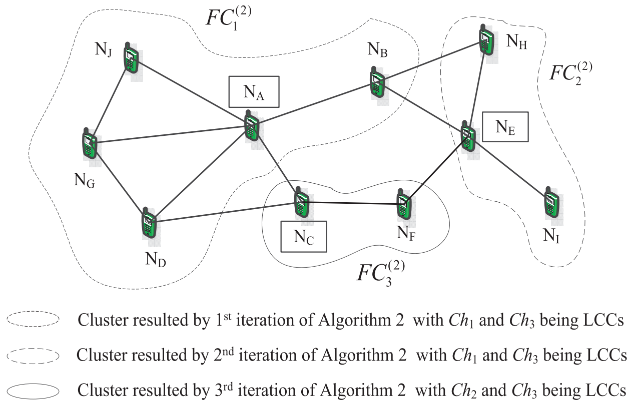


FIGURE 6. Iterative application of Algorithm 2, which incurs lower computational complexity than Algorithm 1, to the MCAHN in Fig. 1 yields three clusters sequentially, where each node in a square denotes a cluster head.

iteratively generate more clusters by first selecting the node with the maximal degree of connectivity as a new CH and then forming a new cluster over the selected CH and its neighbor nodes. This re-clustering process will continue until each node in the old host and neighboring clusters of N_h joins a new cluster or the number of new clusters exceeds that of old clusters. Since N_h itself may not have the maximal degree of connectivity and the re-clustering process only utilizes partial TSI of an MCAHN, [15] in most time cannot reduce the number of clusters as effectively as Algorithm 2.

C. ANALYSIS OF CLUSTERING EFFICIENCY

Having formulated two greedy heuristic clustering algorithms, we can further compare their execution efficiency with the existing clustering algorithms under heterogeneous spectrum availability.

For Algorithm 1, the existing algorithms for finding an MDS in each subgraph $\mathbf{G}_j^{(m)}, j \in [1, C_K^m]$, are known to incur an exponential computational complexity, e.g., $\mathcal{O}(2^{0.955N})$ in [22] and $\mathcal{O}(2^{0.598N})$ in [23], where $N = |\mathbf{V}_j^{(m)}|$. Meanwhile, the branch-and-bound algorithm [24] incurs a computational complexity of at most $\mathcal{O}(2^Q)$ for solving the problem (**), where Q is the total number of clusters resulted by Step 1. Finally, Step 3 only requires a computational time of $\mathcal{O}(|\mathbf{V}|)$. Thus the computational complexity for Algorithm 1 is at the order of $\mathcal{O}(C_K^m 2^{|\mathbf{V}|} + 2^Q)$. In contrast, Algorithm 2 only incurs a computational complexity of $\mathcal{O}\left((C_K^m)^2|\mathbf{V}|^2\right)$. As Algorithm 2 only selects CHs to form a dominating set (DS), which is not necessarily a minimal one, it normally yields more clusters than Algorithm 1. Thus there exists a tradeoff between the optimality and efficiency of clustering computation under heterogeneous spectrum.

On the other hand, [18] has proposed two heuristic clustering algorithms, i.e., spectrum-opportunity clustering (SOC) and constrained-SOC (C-SOC), under heterogeneous spectrum availability. Compared with Algorithms 1 and 2 that require only one selected node, i.e., N_e, to execute clustering computation only once, both SOC and C-SOC incur more clustering overhead in the sense that every node in the MCAHN has to execute clustering computation and fully exchange its clustering decision with its neighbors for at least one time. A similar conclusion also applies to the DS-based clustering algorithm in [15], which requires multiple CHs to execute clustering optimization and exchange their decisions for at least one time.

Moreover, the computational complexities of SOC and C-SOC at each node are $\mathcal{O}(K^2|\mathbf{V}|^2)$ and $\mathcal{O}(mK|\mathbf{V}|^2)$, respectively, while that of the DS-based clustering algorithm at each CH is $\mathcal{O}(K^2|\mathbf{V}_S|^2)$, where $\mathbf{V}_S$ is the set of nodes on which the CH executes the clustering optimization. Thus, as for computational complexity, the SOC, C-SOC, or DS-based clustering at each node is similar as Algorithm 2 and lower than Algorithm 1.

V. EXCHANGE OF CONTROL INFORMATION OVER CLUSTER-BASED HAMILTONIAN CYCLE

After clustering computation, the execution node N_e can further design a specific mechanism for inter-cluster exchange of control information, e.g., spectrum sensing, time clock, channel reservation, and network topology. The design of this mechanism, however, is omitted in most existing literatures, e.g., [13], [14], [17], and [18], on cluster-based control information exchange. To effectively control the overhead of this exchange and improve its reliability, this section proposes to construct a directed Hamiltonian cycle over the resulting clusters for guiding the flowing of inter-cluster control information.

In graph theory, a Hamiltonian cycle is a closed path that visits each vertex of a graph exactly once. By regarding each cluster in the MCAHN as a vertex, N_e can construct a cluster-based Hamiltonian cycle to provide an ordered path for avoiding the extra overhead caused by the unordered inter-cluster exchange of control information.

A. CONSTRUCTION OF CLUSTER-BASED HAMILTONIAN CYCLE

Based on the clustering result, N_e can generate a cluster-based graph $\mathbf{G}^{(C)} = (\mathbf{V}^{(C)}, \mathbf{E}^{(C)})$, where the set $\mathbf{V}^{(C)}$ consists of all final clusters resulted by Algorithm 1 or 2 and the set $\mathbf{E}^{(C)}$ the possible communication links among clusters. Recall from Lemma 1 that Algorithm 1 may result in multiple clusters with a common CH. Thus the inter-cluster communication links can be classified as the following two types:

Type I. When two clusters share a common CH, the CH itself qualifies as an inter-cluster link;
Type II. For two clusters $FC_i^{(m)}$ and $FC_j^{(m)}$, if there exist two neighboring nodes $N_a \in FC_i^{(m)}$ and $N_b \in FC_j^{(m)}$ that share at least one common available channel, then the link between N_a and N_b qualifies as one link between $FC_i^{(m)}$ and $FC_j^{(m)}$.

In general, Algorithm 1 may generate both type-I and type-II links while Algorithm 2 generates type-II links only.

As a branch of Traveling Salesman Problem (TSP) [25], searching a Hamiltonian cycle in the graph $\mathbf{G}^{(C)}$ is an

NP-hard problem. To reduce the searching cost, we adopt an existing algorithm [26] to generate an approximated Hamiltonian cycle with a computational complexity of $\mathcal{O}(|\mathbf{V}^{(C)}|^2\sqrt{|\mathbf{V}^{(C)}|})$. The generated cycle may visit part of clusters in $\mathbf{V}^{(C)}$ for more than one time and consist of a number of clusters no more than 1.5 times that in the smallest Hamiltonian cycle.

For any two adjacent clusters in the Hamiltonian cycle, N_e should further select a default communication link and at least one alternative link, if any, by preferring a type-I link to a type-II one and a type-II link with more common available channels to that with less channels. This selection principle can reduce the overhead for inter-cluster communications and improve its robustness. Either end node of the default link will become a gateway node.

Once the cluster-based Hamiltonian cycle is established, N_e can broadcast the clustering results, the detail of this cycle and the beginning time for inter-cluster exchange of control information to all nodes. To reduce the broadcasting overhead by avoiding unnecessary transmissions, N_e should establish a tree, of which N_e itself is the root and each of all other nodes should be an internal node or a leaf, for guiding the broadcast of clustering decision and time clock information throughout the whole network. Moreover, during the broadcasting procedure, each internal node of the tree should follow the multi-channel transmission scheme proposed in Section III-B for unicasting clustering decision and time clock information to its sons at the tree sequentially. This can effectively avoid the ACK implosion problem, i.e., multiple sons of an internal node may reply their ACKs concurrently and hence cause ACK collisions, and also adapt for the possible spectrum heterogeneity among these sons. Because all nodes can be synchronized by this broadcast, they can begin to implement the Hamiltonian cycle simultaneously.

B. INTER-CLUSTER EXCHANGE OF CONTROL INFORMATION

At the beginning of inter-cluster exchange of control information, the CH of N_e will initialize a Hamiltonian control packet (HCP) and transmit it to the next cluster in the Hamiltonian cycle. In general, a HCP includes the latest information on available channels, time clock, channel reservation, and topology change of each cluster it has trespassed. After receiving the HCP from the previous cluster in the Hamiltonian cycle, a cluster will first renew the HCP based on the local information and then transmit the updated HCP to the next cluster on the cycle.

To avoid possible conflict between HCP and data transmissions, each node should be equipped with two pairs of transceivers, namely the *HCP* and *data transceivers*. In a cluster, all HCP transceivers in a cluster always hop among the LCCs according to a common channel hopping sequence for exchanging the HCP and intra-cluster control information. Meanwhile, each data transceiver can access any local non-control channel to exchange data or inter-cluster information of channel reservation.

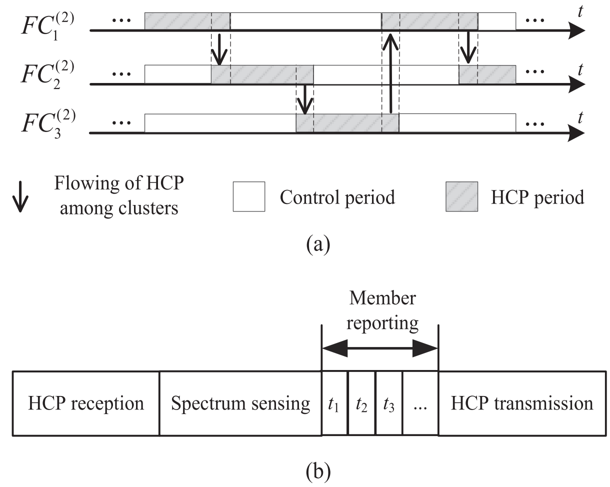


FIGURE 7. When a Hamiltonian control packet (HCP) flows along a Hamiltonian cycle over the three clusters in Fig. 6 (a) time of each cluster is periodically divided into the HCP and control periods alternatively and (b) each HCP period consists of 4 phases, where t_i, $i \geq 1$, denotes the reporting minislot of a specific member node.

As illustrated by Fig. 7(a), time of every cluster is divided into multiple frames with uniform length, each consisting of a *control period* and a *HCP period*. The former is for every HCP transceiver in a cluster to exchange the intra-cluster control information other than HCP and the latter for this cluster to receive, renew and transmit the HCP. All nodes neighboring to a cluster should always shun from the LCCs of this cluster during its HCP period and can access its non-control channels at any other time or its LCCs during its control period based on a competitive fashion, i.e., CSMA. This can avoid possible interference to the HCP transmission and, meanwhile, guarantee fair channel utilization among neighboring clusters.

During its control period, each node N_i can make channel reservation with its neighbor node N_j for data transmission, no matter whether they belong to one common cluster or not. More specifically, if N_i and N_j belong to a same cluster, they can adopt their HCP transceivers to directly exchange RTS/CTS/RES, of which the detail is specified in [27], for reserving a local non-control channel; else, if at least one LCC of N_j is also available for N_i, then N_i will wait for the control period of N_j, handoff its data transceiver to this LCC, and exchange RTS/CTS/RES with N_j to reserve a non-control channel for both of them; else, N_i has to report the communication request to its CH in its HCP period, which will then incorporate this request into the HCP such that N_j can finally receive this request and reply N_i by renewing the received HCP.

On the other hand, as illustrated by Fig. 7(b), a HCP period of each cluster $FC_k^{(m)}$ consists of 4 phases, i.e., HCP reception, spectrum sensing, member reporting, and HCP transmission. In the HCP reception phase, the gateway node of $FC_k^{(m)}$ receives the HCP from the previous cluster in the Hamiltonian cycle and relays it to the CH of $FC_k^{(m)}$ for broadcasting to other member nodes. In the spectrum sensing phase, each node in $FC_k^{(m)}$ will temporarily stop data transmission and adopt both HCP and data transceivers to sense all channels in Φ. In the member reporting phase, all members of $FC_k^{(m)}$ will report their latest information of spectrum sensing, channel reservation, and topological changes based on Time Division Multiple Access (TDMA) to the CH, which will renew the HCP based on the information from all member nodes and broadcast the renewed HCP to them again. Upon receiving this HCP, if a member node realizes that its present data transmission becomes impossible because of topology change or spectrum agility, then it will immediately stop data transmission to avoid more collisions. In the HCP transmission phase, which partially overlaps with the HCP reception phase of the next cluster in the Hamiltonian cycle, the gateway node of $FC_k^{(m)}$ will further transmit the renewed HCP to the next cluster.

C. MAINTENANCE FOR HAMILTONIAN CYCLE

After the cluster-based Hamiltonian cycle is established by the MCAHN, the Hamiltonian cycle structure may be continuously affected by the change of channel availability or the movement of ad hoc nodes. Thus appropriate mechanisms should be designed for maintaining the operation of the Hamiltonian cycle.

1) INTERRUPTION ON INTER-CLUSTER LINK

Once a node detects that a previously available LCC becomes unavailable because of the change of the channel availability, it will report this information to its CH via any of the remaining available LCCs. The CH then will temporarily eliminate the interrupted LCC from the local list of LCCs and broadcast this change to all member nodes. Moreover, if a gateway node reports that its link to the neighbor cluster in the Hamiltonian cycle has been interrupted, then the CH will choose an alternative gateway to resume the communication along the Hamiltonian cycle.

2) ENTRANCE OF NEW NODE

When a new node N_i wants to join the MCAHN, it should periodically listen to all channels in Φ for at least one frame, i.e., a control period plus a HCP period, for discovering its neighbors. If N_i cannot detect any CH during neighbor discovery, then it should immediately generate a new cluster with the CH being itself, choose one or two of its neighbor clusters in the existing Hamiltonian cycle to establish inter-cluster communication link, and transmit this information to the chosen clusters. After each chosen cluster confirms the new inter-cluster links, the whole Hamiltonian cycle is updated in a distributed manner.

On the other hand, if N_i detects at least one CH during neighbor discovery, it should further choose one cluster with the minimal number of member nodes, determine a HCP period of this cluster, and competitively transmit a request packet over the detected channel to the CH in the ensuing control period. Upon receiving this request, the CH will designate to N_i a new minislot in the member reporting phase of the next HCP period and reply a confirmation packet to N_i.

3) LEAVE OF NODES

When a member node N_j wants to leave from a cluster, it does not need to inform its CH. Once the CH cannot receive the reporting from N_j during a HCP period, it will regard N_j as a leaving node, adjust the minislots for member reporting, and finally update this leaving information into the HCP. On the other hand, when a gateway or CH will leave, the CH should also choose an alternative gateway or CH, inform its member nodes, and update the HCP accordingly.

4) RECONSTRUCTION OF HAMILTONIAN CYCLE

Once all LCCs in a cluster are interrupted in Part (1) or the CH cannot find an alternative gateway/CH to replace the already ineffective one in Part (1) or (3), this cluster should be reconstructed for adapting to the new topology and spectrum resources. At the beginning of this reconstruction, the old CH should designate a member node, say N_h, to take charge of the whole reconstruction process and inform this decision to its member nodes and neighboring clusters. Upon receiving this decision, N_h can adopt the DS-based clustering algorithm in [15] to establish new clusters over all remaining members of the old cluster, form appropriate links among the new clusters and their neighboring clusters, and select a shortest one-way route to connect the new clusters so as to establish a new cluster-based Hamiltonian cycle.

The benefit of this re-clustering process is that it only affects the member nodes of the old host cluster of N_h and hence incurs relatively small overhead. However, since the number of clusters yielded by the DS-based algorithm is far beyond the minimum, it would be desirable to periodically execute the distributed TSI aggregation in Section III, the SCP-based clustering in Section IV, and the establishment of cluster-based Hamiltonian cycle in Section V-A once a cluster-based Hamiltonian cycle operates enough long.

VI. NUMERICAL SIMULATION

This section simulates the mechanisms proposed in Sections III, IV, and V based on MATLAB. In this simulation, we consider a synchronized MCAHN, in which all nodes randomly locate within a square area of $100m \times 100m$ and can access a set $\Phi = \{Ch_1, Ch_2, Ch_3, Ch_4, Ch_5\}$ of 5 disjoint spectrum channels with uniform bandwidth. During the distributed aggregation of network TSI, each node has two randomly selected channels in Φ always being locally available and each of the remaining 3 ones in Φ being locally available with the probability 0.5. Once a node N_i decides to transmit a TSIT, it should randomly select an initial delay of $d_i \in [0, W-1]$ timeslots before the TSIT transmission, where W is the size of contention window, wait for the backoff counter to reduce to 0. Following the rules (1)$\sim$(4) in Section III, if two nodes transmit their TSITs in a same minislot, each node within the overlapping of their transmission ranges can never decode these TSITs successfully. To avoid pos-

TABLE 1. Simulation parameters.

Contention window size W	10
Transmission range of each node	$40m$
Required minimum no. of LCCs in a cluster	1 or 2
Length of TSIT	600 bytes
Length of RTS	20 bytes
Length of CTS	14 bytes
Length of RES	14 bytes
Length of ACK	14 bytes
Transmission rate of each channel	1 Mbps
Length of a minislot	$0.01s$

sible fluctuation, each simulation in the sequel is repeated for at least 500 times to generate average results. Other simulation parameters are summarized in Table 1, where TSIT and ACK are for distributed aggregation of network TSI while RTS, CTS and RES are for the control packet exchange after the MCAHN has been clustered.

A. DISTRIBUTED MECHANISM OF TSI AGGREGATION

To evaluate the performance of the TSIT-based aggregation mechanism proposed in Section III, we compare it with the distributed coordination protocol (DCP) in [17]. In DCP, each CH first floods a common channel invitation (CCI) message to inform all nodes, which are within a certain number of hops away from the CH, to report their TSI. After a node receives a CCI for a certain time length, it will begin to report its collected TSI back to the CH along the opposite direction of the progression route of the CCI. When the required number of hops is large enough, then the CH will collect the complete TSI of the MCAHN. In each simulation trial, each node generates its channel hopping sequence independently and randomly. We record the time length from the beginning of the aggregation process to the end where the complete TSI of the MCAHN is collected, and we also keep track of the number of transmitted packets and collisions.

Fig. 8(a) compares the average aggregation time for the TSIT-based mechanism and DCP to aggregate the complete TSI of an MCAHN and Fig. 8(b) their average number of packet transmissions and collisions, where the initial back-off counter for each node is randomly selected from [0, 10] timelsots and all nodes are synchronized. Compared with DCP, the TSIT-based mechanism always incurs less average aggregation time delay and a smaller average number of packet transmissions/collisions, no matter whether each node knows the exact channel hopping sequences of its neighbors, and hence is more efficient in both time and energy consumption. Moreover, if each node knows the channel hopping sequences of its neighbors, the average aggregation time delay and the average number of packet transmissions/collisions in the TSIT-based mechanism can be further reduced.

B. CLUSTERING BY SET COVERING

To evaluate the performance of Algorithms 1 and 2 in Section IV, we compare them with 4 existing clustering algorithms, i.e., SOC [18], C-SOC [18], layered clustering (LC) [16], and cogmesh clustering (CC) [15], where the CC consists of the generation of initial clusters via beacon broadcasting and the ensuing DS-based clustering optimization. In these algorithms, Algorithm 1, Algorithm 2 and C-SOC can guarantee a minimal number of $m \in [1, M]$ LCCs in each cluster, where $M \in [1, |\Phi|]$ is the smallest number of available channels for each node in an MCAHN, while SOC, LC and CC can only guarantee at least one LCC in each cluster. In each simulation trial, we randomly generate a network scenario according to the simulation setting specified at the beginning of this section and provide a node with the complete TSI of the MCAHN. This node then executes different clustering algorithms and obtain the results of interested in Fig. 9.

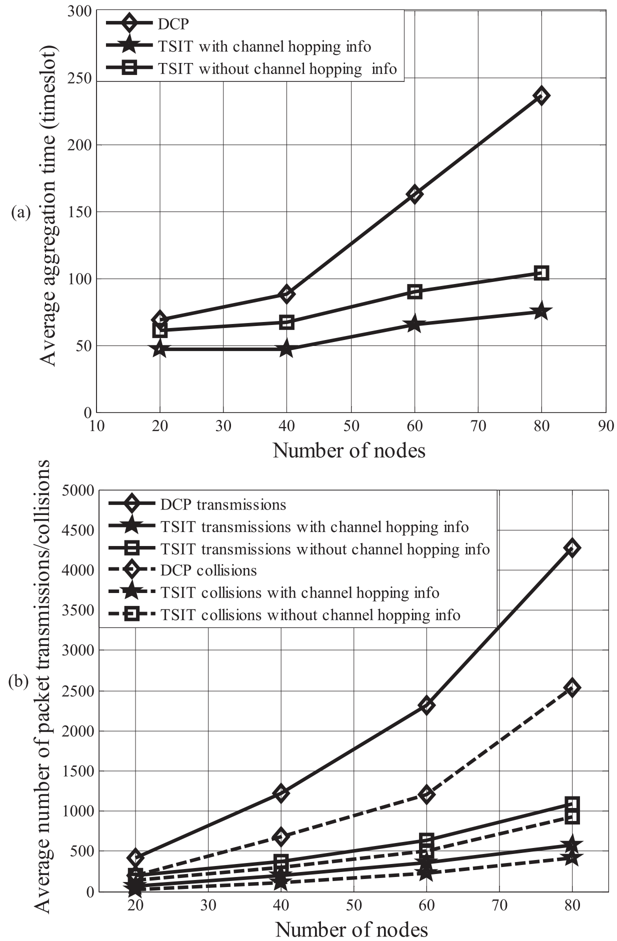


FIGURE 8. Simulation of the distributed coordinated protocol (DCP) in [17] and the proposed TSIT-based aggregation mechanism. (a) The average time delay and (b) the average number of packet transmissions and collisions.

Fig. 9(a) compares the average number of clusters resulted by these algorithms, while Fig. 9(b) the average number of local common channels (LCCs) in each cluster. They show that Algorithms 1 and 2 always outperform C-SOC by yielding a smaller average number of clusters or a larger average number of LCCs in each cluster, no matter whether the minimal number of LCCs in each cluster is set as 1 or 2. Meanwhile, when guaranteeing at least one LCC in each clus-

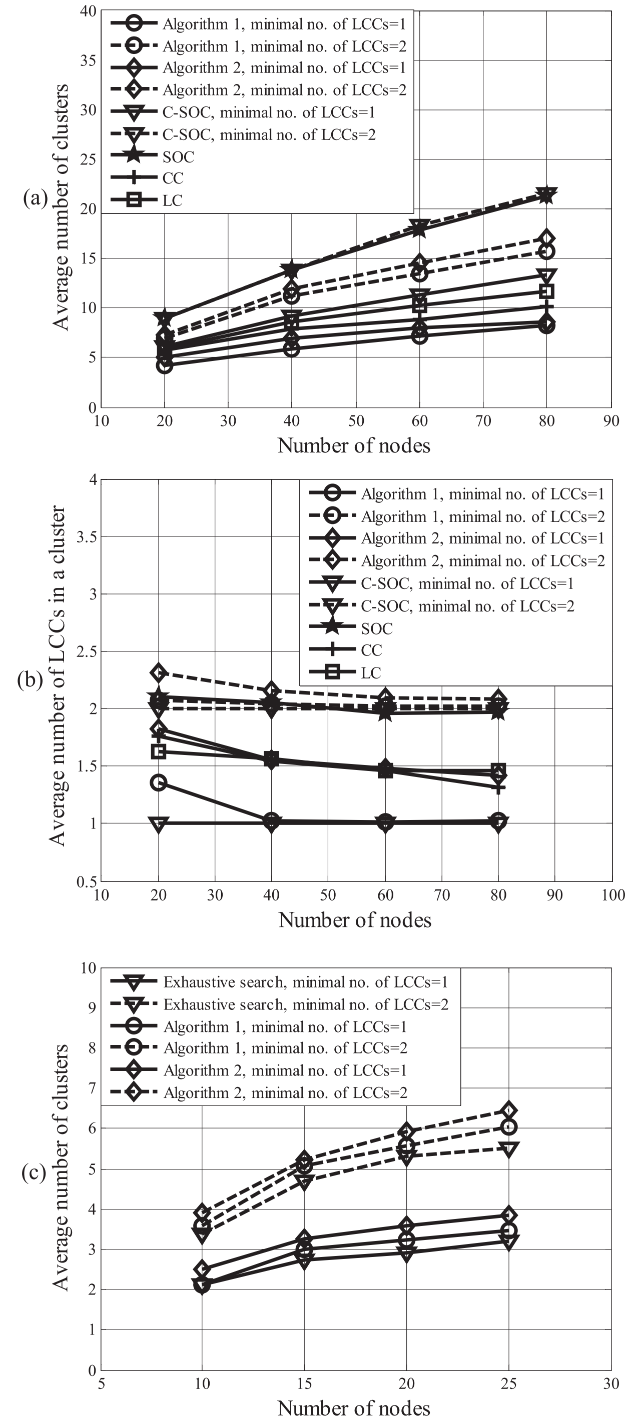


FIGURE 9. In the clustering by Algorithm 1, Algorithm 2, SOC [18], C-SOC [18], LC [16], CC [15], and exhaustive search, (a) and (c) compare the average number of clusters, while (b) shows the average number of actual local common channels (LCCs) in a cluster.

ter, Algorithms 1 and 2 also yield a smaller average number of clusters than SOC, LC, and CC. Thus, given a common requirement of clustering robustness, Algorithms 1 and 2 can incur less overhead than SOC, C-SOC, LC, and CC for inter-cluster exchange of control information. The reason is that the former two utilize the complete TSI of an MCAHN, while SOC, C-SOC, and CC distribute clustering computation to nodes with only partial network TSI. Moreover, the merging of overlapping clusters in LC is inefficient for reducing the number of clusters.

Fig. 9(b) also shows that, when the minimal number of LCCs in each cluster is set as 1, Algorithm 2, LC, and CC can offer a much larger average number of actual LCCs in each cluster than Algorithm 1 and C-SOC. The reason is that the former three iteratively select a node with the maximal degree of connectivity as a CH and hence are more possible to generate relatively small-scale clusters with relatively large number of actual LCCs than the latter two. Moreover, Fig. 9(b) shows that, while guaranteeing at least one LCC in each cluster, Algorithm 1 or 2 offers a larger average number of actual LCCs than C-SOC or CC, respectively. The reason also comes from the fact that the former utilizes more complete network TSI than the latter.

Finally, Fig. 9(c) compares Algorithms 1 and 2 with exhaustive search, which always yields a minimum number of clusters. It shows that the average number of clusters yielded by Algorithm 2 is larger than that of Algorithm 1, both of which are larger than that of exhaustive search. As described in Section IV-C, the polynomial-time complexity of Algorithm 2 is lower than the exponential-time complexity of Algorithm 1, which in turn is more efficient than that of exhaustive search. Thus there exists a tradeoff between the complexity and optimality of clustering computation under heterogeneous spectrum availability.

C. INTER-CLUSTER EXCHANGE OF CONTROL INFORMATION

To evaluate the performance of the Hamiltonian-cycle-based control information exchange proposed in Section V, we compare it with the exchange mechanism in [15]. In this mechanism, a cluster-based MCAHN can maintain cluster structure, transmit data, sense channel status, and exchange intra- and inter-cluster control information in each superframe. For the fairness of comparison, we assume that, in the exchange mechanism in [15], each node is equipped with two pairs of transceivers, one hopping on the LCCs for exchanging intra-cluster control information and the other on the remaining channels for exchanging inter-cluster control information or transmitting data. When a node N_i wants to communicate with N_j, if they belong to a common cluster, N_i will directly use their LCCs to exchange control information with N_j; else, N_i will first hop to the LCCs of N_j and then exchange control information with it. The exchange of inter- or intra-cluster control information is based on the handshake of RTS, CTS and RES [27].

Before this comparison, we adopt Algorithm 1 to generate 12 clusters over an MCAHN of 40 nodes, where each cluster is equipped with a minimal number of 2 LCCs. In the proposed Hamiltonian-cycle-based mechanism, the length of a HCP period is chosen to be 5 minislots and hence the length of a *Hamiltonian cycle period*, i.e., the time for an HCP to flow around the cycle once, is 60 minislots. The comparison simulates a heavy traffic scenario in which each

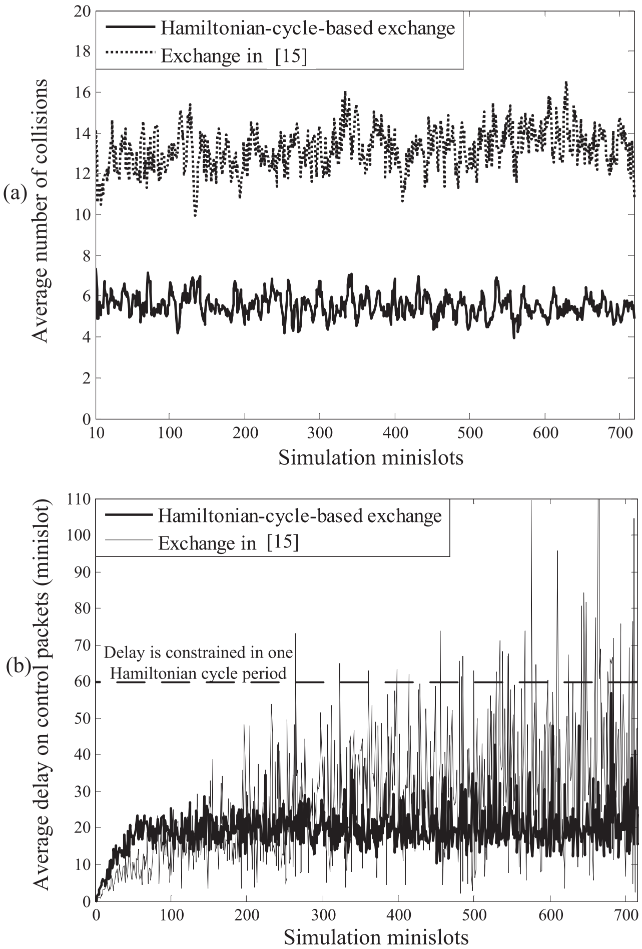


FIGURE 10. Comparison between the Hamiltonian-cycle-based mechanism for control information exchange and the exchange mechanism in [15] by (a) the average number of packet collisions and (b) the average delay of control packets.

node generates the control packet according to a Bernoulli distribution with parameter 0.7. The transmission delay of a successfully transmitted control packet is defined as the time length from its generation to its successful reception.

Fig. 10(a) depicts the average number of collisions during control information exchange in terms of simulation minislots. It shows that, benefited from the ordered flow of inter-cluster control information along the Hamiltonian cycle, the proposed mechanism incurs a smaller average number of packet collisions than [15]. Fig. 10(b) depicts the average delay for control packet exchange in terms of simulation minislots. It shows that, with the passing of simulation time, the average delay in the proposed mechanism is always less than the length of a Hamiltonian cycle period, while that in [15] keeps increasing and finally surpasses that of a Hamiltonian cycle period after about 300 simulation minislots. The reason is that the proposed mechanism supports each node to reserve channels without the help of HCP or, if necessary, via the HCP updated by its CH. This can guarantee a successful channel reservation within one Hamiltonian cycle period. Moreover, Fig. 10(b) shows that, with the passing of simulation time, the variation of average time delay in [15] increases, while that in the proposed mechanism remains relatively stable. Thus the latter is more suitable for providing QoS guarantee for various types of traffic.

D. ROBUSTNESS OF CLUSTER-BASED HAMILTONIAN CYCLE

To evaluate the robustness of the cluster-based Hamiltonian cycle generated in Section VI-C, we keep those channels, which are unavailable for a node during the generation of the Hamiltonian cycle, as being still unavailable and deploy four additional jammers at the 4 locations $(25m, 25m)$, $(75m, 25m)$, $(25m, 75m)$, and $(75m, 75m)$, respectively, in the square area of $100m \times 100m$. By setting an interference radius $60m$ for each newly deployed jammer and randomly selecting one of the 5 channels in Φ for the jammer to occupy at any time, the flowing of HCP packet along the cycle faces with a dynamically changing spectrum environment.

From Section V-C, the interference of jammer activities on an existing Hamiltonian cycle can be classified as:

- Case A: The jammer activity has not yet interfered the existing Hamiltonian cycle.
- Case B: The jammer activity has interfered the existing Hamiltonian cycle and appropriate change on the spectrum channels for the inter-cluster links within the cycle suffices to recover it.
- Case C: The jammer activity has interfered the existing Hamiltonian cycle and appropriate change on the gateway nodes or CHs within the cycle suffices to recover it.
- Case D: The jammer activity has interfered the existing Hamiltonian cycle and it is necessary for a selected CR node to execute the DS-based clustering in [15] for reconstructing a new Hamiltonian cycle.

Obviously, while Case A requires no action of the MCAHN to maintain the existing Hamiltonian cycle, the change of spectrum channels in Case B incurs less overhead for the MCAHN to recover the cycle than the change of gateway nodes or CHs in Case C, which in turn incurs less overhead than the re-clustering by the DS-based algorithm [15] in Case D.

Fig. 11 depicts the occurrence ratio of Cases A-D for the cluster-based Hamiltonian cycle established in Section VI-C when the probability of jammer occupancy at any minislot is 0.1, 0.01, or 0.001 and the time length of each jammer occupancy follows a uniform distribution over [0, 4] minislots. It shows that, when the probability of jammer occupancy at any minislot increases, the ratio for Cases B, C or D also increases, while that for Case A decreases. The reason is that the more frequent the jammers occupy the channels in Φ, the more serious the existing Hamiltonian cycle is interfered by jammer occupancy, and the more difficult for the MCAHN to recover this cycle. Moreover, Fig. 11 also shows that, even when the probability of jammer occupancy is relative high, i.e., equal to 0.1, the ratio for Cases C and D, where the

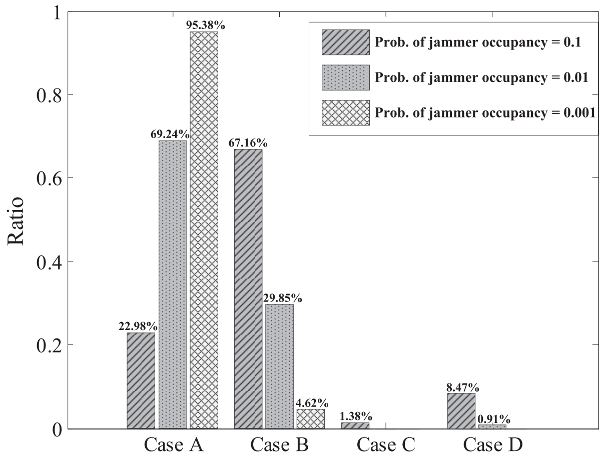


FIGURE 11. Ratios for four interference cases by jammer activity on an existing cluster-based Hamiltonian cycle.

flowing of HCP packet would be mainly interrupted, is still relative low, i.e., 9.85 percent in total. Thus the Hamiltonian-cycle-based exchange of inter-cluster control information has a reasonable robustness against the dynamically changing spectrum environment.

VII. CONCLUSION

To solve the difficulty of spectrum management for multi-channel ad hoc networks (MCAHNs) under the environment of heterogeneous spectrum availability, this paper proposes to establish an ordered flow of control information along a cluster-based Hamiltonian cycle, which can incur less packet collisions as well as shorter average delay and offer better QoS guarantee for various types of traffic than the existing mechanism [15] for control information exchange. Numerical simulation also shows that the cluster-based Hamiltonian cycle established for control information exchange also has a reasonable robustness under the dynamically changing spectrum environment. Moreover, to achieve a better trade-off between the efficiency and robustness for cluster-based control information exchange, we also propose two greedy heuristic clustering algorithms to yield a smaller average number of clusters than those in [15], [16], and [18] while guaranteeing each cluster with a minimal number of LCCs. Finally, to provide the complete topology and spectrum information (TSI) of an MCAHN for clustering, we design a novel distributed mechanism to randomly aggregate the TSI of all nodes into a unique node. The benefits of this mechanism include that it does not have to establish and maintain any fixed route for TSI reporting and is more efficient in time and energy consumption than the existing distributed aggregation mechanism [17] under heterogeneous spectrum availability. Therefore, the presented systematic solution for control information exchange in MCAHNs sheds important light on the practical system design for the network with spectrum heterogeneity, and is expected to be widely used in, for instance, multi-hop CRAHNs.

REFERENCES

[1] Y. Song and J. Xie, "BRACER: A distributed broadcast protocol in multi-hop cognitive radio ad hoc networks with collision avoidance," *IEEE Trans. Mobile Comput.*, vol. 14, no. 3, pp. 509–524, Mar. 2015.

[2] A. Rajandekar and B. Sikdar, "A survey of MAC layer issues and protocols for machine-to-machine communications," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 175–186, Apr. 2015.

[3] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: A survey and taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 4, pp. 19–41, 4th Quart., 2009.

[4] I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Netw.*, vol. 7, pp. 810–836, Jul. 2009.

[5] N. Francisco and F. Ramon, "Channel assignment algorithms for OSA-Enabled WLANs exploiting prioritization and spectrum heterogeneity," *IEICE Trans. Commun.*, vol. E95-B, no. 4, pp. 1125–1134, Apr. 2012.

[6] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Phys. Commun.*, vol. 4, no. 1, pp. 26–39, Mar. 2011.

[7] A. De Domenico, E. C. Strinati, and M.-G. Di Benedetto, "A survey on MAC strategies for cognitive radio networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 21–44, 1st Quart., 2012.

[8] J. Jia, Q. Zhang, and X. S. Shen, "HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 106–117, Jan. 2008.

[9] H. Su and X. Zhang, "Cross-layer based opportunistic MAC protocols for QoS provisionings over cognitive radio wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 118–129, Jan. 2008.

[10] B. Hamdaoui and K. G. Shin, "OS-MAC: An efficient MAC protocol for spectrum-agile wireless networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 8, pp. 915–930, Aug. 2008.

[11] M. Timmers, S. Pollin, A. Dejonghe, L. Van der Perre, and F. Catthoor, "A distributed multichannel MAC protocol for multihop cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 446–459, Jan. 2010.

[12] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, no. 14–15, pp. 2826–2841, Oct. 2007.

[13] J. Zhao, H. Zheng, and G.-H. Yang, "Distributed coordination in dynamic spectrum allocation networks," in *Proc. 1st IEEE Int. Symp. New Frontiers Dyn. Spectr. Access Netw. (DySPAN)*, Nov. 2005, pp. 259–268.

[14] T. Chen, H. Zhang, M. D. Katz, and Z. Zhou, "Swarm intelligence based dynamic control channel assignment in CogMesh," in *Proc. IEEE ICC Workshop*, May 2008, pp. 123–128.

[15] T. Chen, H. Zhang, G. M. Maggio, and I. Chlamtac, "CogMesh: A cluster-based cognitive radio network," in *Proc. IEEE DySPAN*, Apr. 2007, pp. 168–178.

[16] X. J. Tan and W. Zhan, "Control information exchange in cognitive radio ad hoc networks with heterogeneous spectrum," in *Proc. IEEE Globecom*, Dec. 2014, pp. 986–992.

[17] M.-R. Kim and S.-J. Yoo, "Distributed coordination protocol for ad hoc cognitive radio networks," *J. Commun. Netw.*, vol. 14, no. 1, pp. 51–62, Feb. 2012.

[18] S. Liu, L. Lazos, and M. Krunz, "Cluster-Based control channel allocation in opportunistic cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 10, pp. 1436–1449, Oct. 2012.

[19] P. Jesus, C. Baquero, and P. S. Almeida, "A survey of distributed data aggregation algorithms," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 381–404, 1st Quart., 2015.

[20] K. Bian and J. M. Park, "Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1294–1307, Jul. 2012.

[21] T. H. Cormen, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2001.

[22] F. V. Fomin, D. Kratsch, and G. J. Woeginger, "Exact (exponential) algorithms for the dominating set problem," in *Proc. Workshop Graph Theor. Concepts Comput. Sci.*, Jun. 2004, pp. 245–256.

[23] F. Grandoni, "A note on the complexity of minimum dominating set," *J. Discrete Algorithms*, vol. 4, no. 2, pp. 209–214, Jun. 2006.

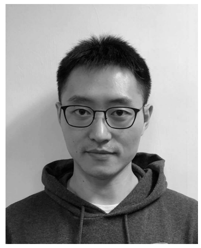
[24] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, Jul. 1960.

[25] R. Bellman, "Dynamic programming treatment of the travelling salesman problem," *J. ACM*, vol. 9, no. 1, pp. 61–63, 1962.

[26] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," Graduate School Ind. Admin., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. 388, Apr. 1976.

[27] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proc. ACM Mobihoc*, May 2004, pp. 222–233.

XUESONG JONATHAN TAN received the B.S. degree in electrical engineering from Sichuan University, Chengdu, China, in 1988, the M.S. degree in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, in 2001, and the Ph.D. degree in electrical engineering from The Chinese University of Hong Kong, Hong Kong, in 2004. He is currently an Associate Professor with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu. His research interests include communication antijamming technology of communication, communication network, and wireless and mobile communication.



WEN ZHAN received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. His research interests are in the area of wireless communications and networking.