

Maximizing Lifetime of Connected-Dominating-Set in Cognitive Radio Networks

Zhiyong Lin^{1,2,*}, Hai Liu², Xiaowen Chu², Yiu-Wing Leung², Ivan Stojmenovic^{3,4}

¹Dept of Computer Science, GuangDong Polytechnic Normal University, China

²Dept of Computer Science, Hong Kong Baptist University, Hong Kong
{zylin, hliu, chxw, ywleung}@comp.hkbu.edu.hk

³SITE, University of Ottawa, Canada, ivan@site.uottawa.ca

⁴EET, Faculty of Technical Sciences, University of Novi Sad, Serbia

Abstract. Connected-dominating-set (CDS) is a representative technique for constructing a virtual backbone of wireless networks. Most of existing works on CDS aim at minimizing the size of the CDS, i.e., constructing the minimum CDS (MCDS), so as to reduce the communication overhead over the CDS. However, MCDS may not work well in cognitive radio networks (CRNs) where communication links are prone to failure due to the unpredictable activities of primary users. A MCDS without consideration of stochastic activities of primary users easily becomes invalid when the primary users reclaim the licensed spectrum. In this work, we assume that the activities of primary users follow the exponential distribution. Our problem is to maximize the lifetime of the CDS while minimizing the size of the CDS, where the lifetime of a CDS is defined as the expected duration that the CDS is maintained valid. We show that the problem is NP-hard and propose a three-phase algorithm. Our basic idea is to apply a pruning-based approach to maximize the lifetime of the CDS. Given a CRN, we prove that our algorithm can compute a CDS such that i) the lifetime of the CDS is maximized (optimal); and ii) the size of the CDS is upper-bounded. To the best of our knowledge, it is the first time in the literature that CDS in CRNs is studied and an effective algorithm is proposed.

Keywords: cognitive radio networks, connected-dominating-set, lifetime.

1 Introduction

Cognitive radio network (CRN) has been proposed as a new kind of wireless networking paradigm, aiming at alleviating the severe scarcity in unlicensed spectrum as well as improving the efficiency of licensed spectrum usage. A CRN is a group of unlicensed users (or cognitive users, CUs) equipped with cognitive radios which coexist with the licensed users (or primary users, PUs) in the same geographic area. With the cognitive radios, CUs are able to sense the licensed spectrum and opportunistically access the idle channels in the licensed spectrum of PUs without causing interference to the PUs. CUs must vacate all the related channels once these channels are reclaimed by PUs whenever the PUs become active. In a CRN, therefore, the sets of available channels for CUs dynamically change over time due to unpredictable activities of PUs. Such a unique characteristic distinguishes CRNs from traditional wireless networks where all nodes usually operate over the same and static channels.

The concept of connected-dominating-set (CDS) plays a crucial role in the management and maintenance of various networks, e.g., wireless sensor networks [1] [2] and Wavelength Division Multiplexing networks [3]. A dominating set (DS) of a given graph G is a set of nodes such that each node of G is either in the set or is adjacent to a node of the set. A CDS is defined as a connected DS. The nodes in CDS are referred to as *dominators* and the nodes other than dominators are referred to as

dominatees. CDS usually serves as a virtual backbone of wireless networks to facilitate tasks such as broadcasting, routing and connectivity management [4]. For instance, broadcasting in a wireless ad hoc network could be simplified by letting each node of a CDS transmit once.

Extensive works have been done in constructing CDS for ad hoc networks, sensor wireless networks and mesh wireless networks [5][6][7][8][9][10]. Most of these works concern the minimum CDS (MCDS), i.e., the CDS with the minimized size. Basically, the existing works can be classified into centralized (e.g. [6]) and decentralized approaches. The decentralized approaches can be further divided into two categories: pruning based algorithms (e.g. [8]) and maximum-independent-set (MIS) based algorithms (e.g. [10]). A good survey on MCDS construction algorithms can be found in [11]. Notice that the communication tasks are undertaken by the nodes in CDS. The CDS with the minimized size reduces the overall communication overhead and thus prolong the lifetime of the networks. In this sense, MCDS does work well in the wireless networks, where all the nodes operate over the same and static channels and the communication links are usually static.

However, MCDS may not be a suitable virtual backbone to CRNs. Notice that available channels of CUs dynamically change over time due to unpredictable activities of PUs. The communication link between two adjacent CUs is broken once there is no commonly-available channel to the CUs. Failure of communication links could cause invalidity of the previously-constructed MCDS. A CDS becomes invalid whenever i) it is no longer connected (i.e., the dominators are not connected); or ii) it is no longer a dominating set (i.e., some dominatee cannot be dominated by the dominators). Therefore, a MCDS without consideration of dynamic activities of PUs easily becomes invalid when the PUs become active and reclaim the spectrum. Notice that it takes considerable communication overhead to maintain or re-construct a CDS if the CDS becomes invalid. Rather than the MCDS with the minimum size, the CDS with the maximum operation duration (lifetime) is more desired in CRNs.

This work addresses the problem of constructing the CDS with the maximum lifetime in CRNs. We assume that a PU behaves at state of ON (active on the channel) or OFF (inactive on the channel) with respective probabilities, and the time lengths of the ON and OFF states follow a specific distribution (e.g., the exponential distribution [12]). The communication link of two adjacent CUs is broken if there is no commonly-available channel to the CUs. In this sense, the lifetime of a communication link is the maximum duration of channels in the link. Given a CRN, the problem of our concern is to compute a CDS such that the lifetime of the CDS is maximized and the size of the CDS is minimized, where the lifetime of a CDS is defined as the expected duration that the CDS is maintained valid. We prove the NP-hardness of the problem and propose a three-phase approach to the problem.

The contribution of this work is threefold. i) *New concept and new problem*: We introduce a new concept of lifetime for CDS which takes the stochastic activities of PUs into account. Based on this new concept, we address a new problem of maximizing the lifetime of the CDS for CRNs. ii) *Optimality of the lifetime of the CDS*: We propose a three-phase approach to the problem. Given a CRN, the proposed solution can compute a CDS with the maximum lifetime. iii) *Theoretical bound on the*

size of the CDS: In addition to the optimality of the lifetime, the proposed solution can achieve an approximation ratio with respect to the size of the CDS.

2 System Model and Problem Formulation

We consider a CRN consisting of N , $N \geq 2$, CUs who coexist with M , $M \geq 1$, PUs in the same geographical area. The PUs independently operate over non-overlapping licensed spectrum. For simplicity, the licensed spectrum of each PU i , $1 \leq i \leq M$, is denoted by channel c_i . That is, the whole set of potentially available channels is denoted by $\mathcal{C} = \{c_1, c_2, \dots, c_M\}$, where c_i represents the channel licensed to PU i ($i=1, 2, \dots, M$). We assume that each CU is equipped with one cognitive radio, by which the CU can access the idle channels in the licensed spectrum of the PUs.

There are two states to describe PUs' activities. A PU could behave at state of ON (active on the channel) or OFF (inactive on the channel) with respective probabilities. Channel c_i is said to be *available* to CUs if PU i is at state OFF, i.e., not active on channel c_i . Thus, the duration of c_i being continuously available is a random variable, denoted by x_i ($i=1, 2, \dots, M$).

A CRN is modeled by a graph $G(V, E)$, where $V = \{v_1, v_2, \dots, v_N\}$ denotes the set of CUs (v_i corresponding to CU i , $i=1, 2, \dots, N$) and E is the set of edges (communication links). There is an edge e_{ij} between v_i and v_j if and only if v_i and v_j are within the transmission range of each other and there is at least one commonly-available channel to v_i and v_j . We assume that edge e_{ij} is broken if all commonly-available channels become unavailable. We have the following definition.

Definition 1 (Lifetime of an Edge). Suppose that v_i and v_j are within the transmission range of each other and there are k channels, say, c_1, c_2, \dots, c_k , commonly-available to v_i and v_j . The lifetime of edge e_{ij} is defined as $\rho(e_{ij}) = \max\{E(x_1), E(x_2), \dots, E(x_k)\}$.

Remarks on Definition 1: i) The lifetime $\rho(e_{ij})$ is introduced to measure the duration that e_{ij} is preserved. That is, edge e_{ij} is assumed to be broken after $\rho(e_{ij})$ time units lapses. Notice that, the break of edge e_{ij} could be caused by the movement of v_i and v_j . Therefore, in general we should take these two factors into account when addressing the lifetime of an edge. In this study, however, we assume the users are static or move with relatively low speeds and the change of channel availability is the dominator factor to link failure, since we believe that this distinguished characteristic of CRNs is more important. ii) It should be emphasized that Definition 1 gives one reasonable and feasible approach to estimate the duration length of an edge. Other possible approaches could be adopted in the proposed solution in this work and do not affect its correctness.

Given a connected network, with each edge associated with a lifetime, we define its lifetime as the duration that this network is maintained connected.

Definition 2 (Lifetime of a Connected Network). Given a connected network G , its lifetime $\rho(G)$ is determined as follows: i) deleting all the edges with lifetime less than $\rho(G)$ cannot cause G disconnected; ii) but deleting all the edges with lifetime less than or equal to $\rho(G)$ will cause disconnection of G . That is, $\rho(G) = \max\{\rho \mid \text{deleting in } G \text{ all the edges with lifetime less than } \rho \text{ cannot cause } G \text{ disconnected}\}$.

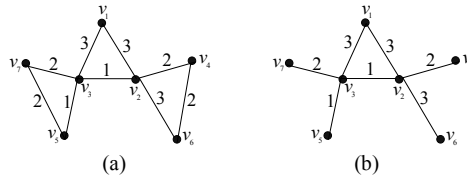


Figure 1. An illustration example.

Intuitively, the lifetime of a network indicates the longest time that the network is maintained connected. Take example in Fig. 1(a), the number associated with each edge denotes the lifetime of the edge (in time unit). When t time units lapse, all the edges with lifetime less than or equal to t are broken. It is easy to identify that the lifetime of the network is 2 (time units).

Based on Definition 1 and Definition 2, we introduce a new concept of lifetime for CDS. Notice that a CDS becomes invalid whenever i) it is no longer connected. That is, the *internal* connectivity of the dominators is not preserved; or ii) it is no longer a dominating set. That is, some *external* links (between the dominators and the dominatees) are broken and some dominatee cannot be dominated by the dominators. Therefore, the lifetime of a CDS not only depends on the duration of its internal connectivity, but also the duration of its external links. Accordingly, we have the following definitions.

Definition 3 (Internal Lifetime of a CDS). Suppose that \mathcal{S} is a CDS of $G(V, E)$, the internal lifetime of \mathcal{S} is defined as $\rho_{in}(\mathcal{S}) = \rho(G[\mathcal{S}])$, where $G[\mathcal{S}]$ denotes the subgraph of G induced by \mathcal{S} .

According to Definition 3, the internal lifetime of a CDS is exactly the lifetime of the connected subgraph induced by the CDS.

Definition 4 (External Lifetime of a CDS). Suppose that \mathcal{S} is a CDS of $G(V, E)$, the external lifetime of \mathcal{S} is defined as $\rho_{ex}(\mathcal{S}) = \min\{\tau(v) \mid v \in V - \mathcal{S}\}$, where $\tau(v) = \max\{\rho(e) \mid e \in E, \text{ one endpoint of } e \text{ is dominatee } v \text{ and the other endpoint is a dominator in } \mathcal{S}\}$ is used to denote the longest time that dominatee v can be dominated by the CDS \mathcal{S} .

Based on Definition 3 and Definition 4, we define the lifetime of a CDS as the minimum duration of its internal lifetime and its external lifetime to ensure the validity of the CDS.

Definition 5 (Lifetime of a CDS). Suppose that \mathcal{S} is a CDS of $G(V, E)$, the lifetime of \mathcal{S} is defined $\rho(\mathcal{S}) = \min\{\rho_{in}(\mathcal{S}), \rho_{ex}(\mathcal{S})\}$.

Definition 6 (Subgraph Spanned by a CDS). Suppose that \mathcal{S} is a CDS of $G(V, E)$, the subgraph spanned by \mathcal{S} , denoted by $G\langle\mathcal{S}\rangle$, is the graph which contains all nodes in V and the only edges associated with at least one endpoint in \mathcal{S} . That is, $G\langle\mathcal{S}\rangle = (V, E')$, where $E' = \{e \in E \mid \text{at least one endpoints of } e \text{ is a dominator in } \mathcal{S}\}$.

The following theorem states that the lifetime of a CDS is equal to the lifetime of the subgraph spanned by the CDS.

Theorem 1. Let \mathcal{S} be a CDS of $G(V, E)$, then $\rho(\mathcal{S}) = \rho(G\langle\mathcal{S}\rangle)$, where $G\langle\mathcal{S}\rangle$ is the subgraph spanned by \mathcal{S} .

Proof. It can be directly obtained from Definition 2–6. ■

Take example in Fig. 1(a) again, $\mathcal{S} = \{v_1, v_2, v_3\}$ is a CDS of the graph. After 1 unit time lapses, edges (v_3, v_2) and (v_3, v_5) are broken and thus dominatee E can no longer be dominated (\mathcal{S} become invalid). We can see the lifetime of \mathcal{S} is 1. The subgraph spanned by \mathcal{S} is shown in Fig. 1(b). According to Definition 2, its lifetime is 1 which equals the lifetime of \mathcal{S} . In this example, MCDS is $\{v_2, v_3\}$ with lifetime 1 while CDS $\{v_1, v_2, v_3, v_7\}$ is with the maximum lifetime 2.

In this work, we study the problem of maximizing lifetime of CDS in CRNs. Notice that two CDSs with the same lifetime may have different internal lifetimes and external lifetimes. When the external lifetime expires, individual dominatee cannot be

dominated and this has only local effect. In contrast, the whole CDS is disconnected and thus does not work if the internal lifetime expires. Therefore, we believe that the internal lifetime is more important than the external lifetime. Our problem is formally formulated as follows.

Maximizing Lifetime of CDS (MLCDS). Given a CRN $\mathcal{G}(V, E)$ and the stochastic activities of PUs, the MLCDS problem is to compute a CDS \mathcal{S} of \mathcal{G} such that: i) the lifetime of \mathcal{S} is maximized, i.e., $\max \rho(\mathcal{S})$; ii) the internal lifetime of \mathcal{S} is maximized, i.e., $\max \rho_{\text{in}}(\mathcal{S})$; iii) the size of \mathcal{S} is minimized, i.e., $\min |\mathcal{S}|$.

The MLCDS problem is a multi-objective optimization problem. The three objectives are difficult to be optimized simultaneously. In practice, a multi-objective problem is usually tackled by optimizing the objectives sequentially. In CRNs, we believe that the lifetime of a CDS is more important than the size. It is because that a CDS with short lifetime is prone to failure, which may cause considerable communication overhead to maintain and re-construct the CDS. To tackle the MLCDS problem, therefore, we first compute a CDS with the maximum lifetime and then try to minimize the size of the CDS with the maximum lifetime preserved. The following theorem shows the MLCDS problem is NP-hard.

Theorem 2. The MLCDS problem is NP-hard.

Proof. Due to limited space, please refer to the supplementary in [17]. ■

3 A three-phase algorithm

We develop a three-phase algorithm to the MLCDS problem. Given a CRN $\mathcal{G}(V, E)$ and the stochastic activities of PUs, we first determine the lifetime of each edge in \mathcal{G} . According to [12], activity of a PU can be modeled as a continuous time semi-Markov process. That is, the time lengths of the ON and OFF states follow the exponential distribution. Thus, the duration of channel c_i being continuously available to CUs, denoted by variable x_i , also follows the exponential distribution. The probability density function (pdf) of x_i is represented as $\lambda_i e^{-\lambda_i x_i}$ ($x_i \geq 0$) ($i=1, 2, \dots, M$). We assume that each CU is aware of the pdf of x_i either by estimation based on the historical data or by the notification of the spectrum server [13]. Two neighboring CUs can exchange information to identify their commonly-available channels by rendezvous algorithms (e.g., [14] and [15]). Once commonly-available channels are identified and duration of each channel ($E(x_i)=\lambda_i$) is determined, the CUs can compute the lifetime of the edge by Definition 1.

With each edge associated with a lifetime, our task is to sequentially optimize the three objectives of the MLCDS problem, i.e., maximizing the lifetime, maximizing the internal lifetime, and minimizing the size. Accordingly, our algorithm consists of three phases which aim at the three objectives, respectively. Specifically, given a CRN \mathcal{G} , in the first phase, we compute a connected subgraph \mathcal{G}' of \mathcal{G} such that CDS \mathcal{S} of \mathcal{G} has the maximum lifetime if and only if \mathcal{S} is a CDS of \mathcal{G}' . That is, the CDS of \mathcal{G} with the maximum lifetime is guaranteed to be included in subgraph \mathcal{G}' (we reduce the searching scope). Based on \mathcal{G}' , in the second phase, we compute a subgraph \mathcal{G}'' of \mathcal{G}' such that any CDS of \mathcal{G}' (also the CDS of \mathcal{G}) constructed within \mathcal{G}'' can further maximize the internal lifetime with the maximum lifetime preserved. Finally, in the third phase, we compute a CDS of \mathcal{G}' within \mathcal{G}'' and minimize its size, with the maximum lifetime and the maximum internal lifetime both preserved.

For convenience, given two sets $S_1 \subseteq V$ and $S_2 \subseteq V$ in graph $G(V, E)$, hereafter we say S_2 is dominated by S_1 if each node in S_2 either belongs to S_1 or is adjacent to a node in S_1 .

3.1 Phase one: maximize the lifetime

According to Theorem 1, the lifetime of CDS S of G , i.e. $\rho(S)$, is equal to $\rho(G\langle S \rangle)$, where $G\langle S \rangle$ denotes the subgraph of G spanned by S . That is, maximizing the lifetime of S is equivalent to maximizing the lifetime of $\rho(G\langle S \rangle)$. On the other hand, according to Definition 2, lifetime of a connected network (graph) is determined by the lifetimes of edges of the graph. Therefore, to construct the CDS of G with the maximum lifetime, we should avoid considering those edges with small lifetimes in G . Inspired by this intuitive observation, we design a greedy edge pruning algorithm, called *MaxLifetime*, which can generate a subgraph G' of G such that constructing CDS of G' is equivalent to constructing CDS of G with the maximum lifetime. The algorithm works as follows. Given a connected graph $G(V, E)$, we first sort all the edges of G in ascending order according to their lifetimes. Note that the edges may have the same lifetime. Suppose there are l different lifetime levels, which are denoted by $\rho_1 < \rho_2 < \dots < \rho_l$. Let $G' = (V, E')$ be a subgraph of G with E' initialized as E . Then, we continually delete the edges in E' according to lifetime level from ρ_1 to ρ_l as long as such edge deletion will not cause disconnection of G' . When the algorithm ends, we get a connected subgraph G' of G . The pseudo code of *MaxLifetime* is presented in Fig .2.

MaxLifetime Algorithm

INPUT: a connected graph $G(V, E)$ in which each edge is associated with a lifetime.

OUTPUT: $G' = (V, E')$, a connected subgraph of G .

1. Sort the lifetimes $\{\rho(e) \mid e \in E\}$ in ascending order and get l lifetime levels: $\rho_1 < \rho_2 < \dots < \rho_l$;
 2. Initialize $E' = E$;
 3. **FOR** $i = 1: l$ //edge deletion
 4. $E_i = \{e \mid \rho(e) = \rho_i, e \in E'\}$; //set of the edges with lifetime ρ_i
 5. **IF** graph $(V, E' - E_i)$ still keeps connected
 6. $E' = E' - E_i$; //delete all the edges with lifetime ρ_i
 7. **ELSE** break; //out of the for-loop
 8. **RETURN** $G' = (V, E')$;
-

Figure 2. Pseudo code of the *MaxLifetime* algorithm

Given graph $G(V, E)$, the *MaxLifetime* algorithm can generate a connected subgraph $G'(V, E')$ of G . Let $\eta = \min\{\rho(e) \mid e \in E'\}$ be the smallest lifetime of edges in G' and $\rho^* = \max\{\rho(S) \mid S \text{ is a CDS of } G\}$ be the maximum lifetime of CDSs of G . The following lemma indicates that both G and G' have the same lifetime, which is equal to η , and this value also equals ρ^* .

Lemma 1. Given $G(V, E)$, let $G'(V, E')$ be the output subgraph by the *MaxLifetime* algorithm for G . We have: $\rho(G) = \rho(G') = \eta = \rho^*$, where $\eta = \min\{\rho(e) \mid e \in E'\}$ and $\rho^* = \max\{\rho(S) \mid S \text{ is a CDS of } G\}$.

Proof. Due to limited space, please refer to the supplementary in [17]. ■

Based on Lemma 1, we prove that the subgraph G' , which is output by the *MaxLifetime* algorithm for G , has the following property: constructing CDS of G' is equivalent to constructing CDS of G with the maximum lifetime. This result is formally described by the following theorem.

Theorem 3. Given $G(V, E)$, let $G'(V, E')$ be the output subgraph by the *MaxLifetime* algorithm for G . We have: $S \subseteq V$ is a CDS of G with the maximum lifetime (i.e., $\rho(S)=\rho^*$) if and only if S is a CDS of G' .

Proof. “ \Rightarrow ” Suppose $S \subseteq V$ is a CDS of G and $\rho(S)=\rho^*$. According to Definition 5, $\rho(S)=\min\{\rho_{in}(S), \rho_{ex}(S)\}$, which implies $\rho_{in}(S)\geq\rho^*$ and $\rho_{ex}(S)\geq\rho^*$. According to Definition 3, $\rho_{in}(S)=\rho(G[S])$, where $G[S]$ is the subgraph of G induced by S . Since $\rho(G[S])=\rho_{in}(S)\geq\rho^*$, S can be connected by a set of edges with lifetime not less than ρ^* . According to Lemma 1, ρ^* is exactly equal to the minimum lifetime of edges in $G'(V, E')$, that is, $E'=\{e \mid e \in E, \rho(e)\geq\rho^*\}$. Therefore, S can be connected in G' , namely, $G'[S]$ is connected. On the other hand, S should dominate $(V-S)$ in G' . Otherwise, there exists a dominatee v in $(V-S)$ such that v cannot be dominated by S in G' . This implies that $\tau(v)<\rho^*$, where $\tau(v)=\max\{\rho(e) \mid e \in E, \text{one endpoint of } e \text{ is dominatee } v \text{ and the other endpoint is a dominator in } S\}$. According to Definition 4, this also implies $\rho_{ex}(S)<\rho^*$, which leads to a contradiction. Therefore, S can dominate $(V-S)$ in G' . Since we have shown $G'[S]$ is connected, S is thus a CDS of G' .

“ \Leftarrow ” Suppose $S \subseteq V$ is a CDS of $G'(V, E')$. Since $E' \subseteq E$, obviously, S is also a CDS of G , and $\rho(G'<S>)\leq\rho(G<S>)$. According to Lemma 1, the minimum lifetime of edges in G' equals ρ^* . Hence, $\rho(G'<S>)\geq\rho^*$, which implies $\rho(G<S>)\geq\rho(G'<S>)\geq\rho^*$. According to Theorem 1, $\rho(S)=\rho(G<S>)$. Thus, $\rho(S)\geq\rho^*$. Moreover, we should have $\rho(S)\leq\rho^*$ as ρ^* is the maximum lifetime of CDSs of G . So, $\rho(S)=\rho^*$. ■

3.2 Phase two: maximize the internal lifetime

After the optimization in phase one, we may obtain a connected subgraph $G'(V, E')$ of $G(V, E)$. Any CDS of G' is also CDS of G . Furthermore, according to Theorem 3, to construct the CDS of G with the maximum lifetime, we just need to construct the CDS of G' instead. However, since the lifetime of a CDS is the minimum of its internal lifetime and external lifetime, the CDS with the maximum lifetime does not necessarily achieve the maximum internal lifetime. Hence, to maximize the internal lifetime, we need further optimization.

MaxInternalLifetime Algorithm

INPUT: a connected graph $G'(V, E')$ in which each edge is associated with a lifetime.

OUTPUT: $G''=(V, E'')$, a subgraph of G' .

1. Sort the lifetimes $\{\rho(e) \mid e \in E'\}$ in ascending order and get l lifetime levels: $\rho_1 < \rho_2 < \dots < \rho_l$;
 2. Initialize $E''=E'$;
 3. **FOR** $i=1:l$ //edge deletion
 4. $E_i=\{e \mid \rho(e)=\rho_i, e \in E''\}$; //set of the edges with lifetime ρ_i
 5. **IF** graph $(V, E''-E_i)$ still has a component containing a CDS of G'
 6. $E''=E''-E_i$; //delete all the edges with lifetime ρ_i
 7. **ELSE** break; //out of the for-loop
 8. **RETURN** $G''=(V, E'')$;
-

Figure 3. Pseudo code of the *MaxInternalLifetime* algorithm

Given a CDS S of G' , according to Definition 3, its internal lifetime $\rho_{in}(S)$ is determined by $\rho(G'[S])$, where $G'[S]$ is the subgraph of G' induced by S . Intuitively, to maximize $G'[S]$, we should ensure that S can be connected by edges with lifetime as large as possible. That is, similar to the optimization in phase one, in this phase we also need to appropriately prune edges with small lifetime in G' . With this basic idea, we devise a greedy algorithm called *MaxInternalLifetime*. The proposed algorithm can

generate a subgraph G' of the given graph G such that any CDS of G constructed within G' can achieve the maximum internal lifetime. The algorithm works in a similar way to the *MaxLifetime* algorithm, namely, both are edge pruning based. Specifically, given $G'(V, E')$, we also first sort the edges of G' in ascending order according to their lifetime. Suppose there are l different lifetime levels, which are denoted by $\rho_1 < \rho_2 < \dots < \rho_l$. Let $G''=(V, E'')$ be a subgraph of G' with E'' initialized as E' . Then, we try to delete the edges in E'' according to lifetime level from ρ_1 to ρ_l . The edge deletion terminates when any further edge deletion would result in that no component of G'' can still include a CDS of G . When the algorithm ends, we get a subgraph G'' of G' in which there exists at least one component that contains a CDS of G . The pseudo code of *MaxInternalLifetime* is presented in Fig. 3.

Given graph $G'(V, E')$, the *MaxInternalLifetime* algorithm generates a subgraph $G''(V, E'')$ of G' . Let $\gamma = \min\{\rho(e) \mid e \in E''\}$ be the smallest lifetime of edges in G'' and $\rho_{in}^* = \max\{\rho_{in}(S) \mid S \text{ is a CDS of } G\}$ be the maximum internal lifetime of CDSs of G , respectively. The following lemma shows that γ is equal to ρ_{in}^* .

Lemma 2. Given $G'(V, E')$, let $G''(V, E'')$ be the output subgraph by the *MaxInternalLifetime* algorithm for G' . We have: $\gamma = \rho_{in}^*$, where $\gamma = \min\{\rho(e) \mid e \in E''\}$ and $\rho_{in}^* = \max\{\rho_{in}(S) \mid S \text{ is a CDS of } G\}$.

Proof. Due to limited space, please refer to the supplementary in [17]. ■

Based on Lemma 2, we obtain the following theorem, which shows that the *MaxInternalLifetime* algorithm is guaranteed to generate a subgraph G'' such that i) any CDS of G which also keeps connected in G'' will achieve the maximum internal lifetime and ii) any CDS of G with the maximum internal lifetime must be contained in a component of G'' .

Theorem 4. Given $G'(V, E')$, let $\rho_{in}^* = \max\{\rho_{in}(S) \mid S \text{ is a CDS of } G\}$ and $G''(V, E'')$ be the output subgraph by the *MaxInternalLifetime* algorithm for G' . We have: i) if $S \subseteq V$ is a CDS of G and $G''[S]$ is connected, then $\rho_{in}(S) = \rho_{in}^*$; ii) if S is a CDS of G and $\rho_{in}(S) = \rho_{in}^*$, then $G''[S]$ is connected.

Proof. i) Since $G''[S]$ is connected, we have $\rho(G''[S]) \geq \gamma$ (γ is the minimum lifetime of edges in G''). According to Lemma 2, $\gamma = \rho_{in}^*$. Thus, $\rho(G''[S]) \geq \rho_{in}^*$. Notice that $E'' \subseteq E'$, which implies $\rho(G'[S]) \geq \rho(G''[S])$. So, $\rho_{in}(S) = \rho(G'[S]) \geq \rho(G''[S]) \geq \rho_{in}^*$. Obviously, we have $\rho_{in}(S) \leq \rho_{in}^*$. Thus, $\rho_{in}(S) = \rho_{in}^*$.

ii) According to Lemma 2, $\rho_{in}^* = \gamma$. Since $\rho_{in}(S) = \rho(G'[S]) = \rho_{in}^*$, we have $\rho(G'[S]) = \gamma$. This implies that $G'[S]$ is still connected even if we delete in it all the edges with lifetime less than γ , namely, S can be connected by a set of edges with lifetime not less than γ . Thus, $G''[S]$ must be connected. ■

Given graph $G(V, E)$, we perform the *MaxLifetime* algorithm in phase one and obtain a connected subgraph $G'(V, E')$ of G . Based on $G'(V, E')$, we perform the *MaxInternalLifetime* algorithm in phase two and obtain a subgraph $G''(V, E'')$ of G' . Theorem 3 shows that constructing CDS of G with the maximum lifetime can be transferred to constructing CDS of G' . Theorem 4 shows that constructing CDS of G' within G'' can maximize the internal lifetime of the CDS. Furthermore, according to the *MaxLifetime* algorithm, it is clear that a CDS of G' with the maximum internal lifetime is also the CDS of G with the maximum internal lifetime (both maximums are the same). Based on these observations, we obtain the following theorem.

Theorem 5. Given $G(V, E)$, let $G^d(V, E^d)$ be the output subgraph by the *MaxLifetime* algorithm for G and $G^d(V, E^d)$ be the output subgraph by the *MaxInternalLifetime* algorithm for G^d . We have: $S \subseteq V$ is a CDS of G with the maximum lifetime and the maximum internal lifetime if and only if S is a DS of G^d and $G^d[S]$ is connected.

Proof. It is immediately followed by Theorem 3 and Theorem 4. ■

3.3 Phase three: minimize the size with the maximum lifetime preserved

ConstructMCDS_R Algorithm

INPUT: $G(V, E)$ and its subgraph G^d consisting of k components $g_i(V_i^d, E_i^d)$ ($i=1, 2, \dots, K$).

OUTPUT: S which is a subset of V and a CDS of G^d .

```

1:  $S=V$ ;
2: FOR  $i=1, 2, \dots, K$ 
3:   IF  $V_i$  can dominate  $(V-V_i^d)$  in  $G^d$ 
4:      $X_i=ConstructMCDS(g_i)$ ; //comment 1
5:     IF  $X_i$  can dominate  $(V-V_i^d)$  in  $G^d$ 
6:        $S_i=X_i$ ;
7:     ELSE
8:        $U=\{v|v \in (V-V_i^d) \text{ and } v \text{ is not dominated by } X_i\}$ ;
9:        $Y_i=FindMCS(V_i^d-X_i, U)$ ; //comment 2
10:       $S_i=X_i+Y_i$ ;
11:     IF  $|S_i| < |S|$ 
12:        $S=S_i$ ;
13: RETURN  $S$ ;
```

Figure 4. Pseudo code of the *ConstructMCDS_R* algorithm

Remark: comment 1—*ConstructMCDS(g)* is the subroutine to find MCDS of graph g and it return a CDS of g (e.g., we can use the approach proposed by Guha *et al.* in [6]); comment 2—*FindMCS(V^d, U)* is the subroutine to find a minimum number of nodes in V^d to cover/dominate all nodes in U and it return a subset of V^d (e.g., we can use the well-known greedy algorithm for minimum-set-cover problem [16]).

Given graph $G(V, E)$, after optimization in phase one and phase two, we obtain the subgraphs $G^d(V, E^d)$ and $G^d(V, E^d)$, respectively. Notice, any CDS of G^d constructed within G^d can achieve the maximum lifetime as well as internal lifetime. Therefore, in phase three, the remaining problem is to compute a MCDS of G^d in G^d with the maximum lifetime preserved. It is worth emphasizing that, the way constructing CDS in this phase is different to that in the traditional MCDS problem, since the former is restricted in G^d , i.e., a subgraph of G^d , to find the desired CDS of G^d , while the latter has no such restriction. In viewing of this, to distinguish from the traditional MCDS problem, we refer to the problem in phase three as *MCDS with Restriction (MCDS_R) Problem*, which is formally described as follows.

MCDS_R Problem. Given graph $G^d(V, E^d)$ and its subgraph G^d , the MCDS_R problem is to find a subset of V , say S , which satisfies the following requirements: i) S is a DS of G^d ; ii) the subgraph induced by S in G^d , i.e. $G^d[S]$, is connected; iii) the size of S , i.e. $|S|$, is minimized.

According to Theorem 5, requirements i) and ii) ensure that S is a CDS of G with the maximum lifetime as well as internal lifetime preserved. Requirement iii) ensures that the size of S is minimized. For convenience, hereafter we use $MCDS_R(G^d, G^d)$ to denote the MCDS of G^d with restriction in G^d . The following theorem shows the NP-hardness of the MCDS_R problem.

Theorem 6. The MCDS_R problem is NP-hard.

Proof. Due to limited space, please refer to the supplementary in [17]. ■

We next present an approximation algorithm, called *ConstructMCDS_R*, to the MCDS_R problem. The basic idea of the algorithm is as follows. Given graph G' , we check each component g_i of G' . We skip g_i if the nodes in g_i cannot dominate V , i.e., g_i does not contain a CDS of G' . Otherwise, we attempt to compute a MCDS of g_i , i.e., $\text{MCDS}(g_i)$. If $\text{MCDS}(g_i)$ can dominate V , we let $S_i = \text{MCDS}(g_i)$ which is also a CDS of G' . Otherwise, additional nodes in g_i need to be added into S_i for dominating V . To minimize the number of nodes in the CDS, we add as few as possible nodes to dominate V . This can be done by employing any existing algorithm to the *minimum-set-cover* problem [16]. Nodes in $\text{MCDS}(g_i)$ along with these additional nodes are included into S_i which is a CDS of G' . Finally, we output S which is the S_i with the minimum size. The pseudo code of the *ConstructMCDS_R* algorithm is described in Fig. 4.

Theorem 7. Let α and β denote the approximation ratios of any algorithms to the MCDS problem and the minimum-set-cover problem, respectively. The *ConstructMCDS_R* algorithm achieves the approximation ratio not greater than $(\alpha + \beta)$.

Proof. Given graph $G'(V, E')$ and its subgraph G'' , without loss of generality, we assume each component $g_i(V_i'', E_i'')$ of G'' contains a CDS of G' , i.e., V_i'' can dominate V , since otherwise g_i is actually skipped by the algorithm (see Line 3 in Fig. 4). In the subroutine *ConstructMCDS* invoked by *ConstructMCDS_R*, we can employ the algorithm with approximation ratio α for the MCDS problem. Then, for the fixed component g_i , according to Line 4 in *ConstructMCDS_R*, we have

$$|X_i| \leq \alpha |\text{MCDS}(g_i)| \quad (1)$$

Note that $\text{MCDS}_R(G', g_i)$ is also a CDS of g_i , that is, $|\text{MCDS}(g_i)| \leq |\text{MCDS}_R(G', g_i)|$. Based on Inequality (1), we further have

$$|X_i| \leq \alpha |\text{MCDS}_R(G', g_i)| \quad (2)$$

In case X_i cannot dominate $(V - V_i'')$, those nodes that have not yet been dominated by X_i form the set U (see Line 8 in Fig. 4). Some additional nodes in $(V_i'' - X_i)$, which finally form the set Y_i , should be added to cover/dominate U . Y_i is obtained by invoking the subroutine *FindMSC* to resolve the corresponding minimum-set-cover problem (see Line 9 in Fig. 4). Since we can employ in *FindMSC* the algorithm with approximation ratio β for the minimum-set-cover problem, we have

$$|Y_i| \leq \beta \times \text{opt}_i \quad (3)$$

where opt_i denotes the minimum number of nodes to cover/dominate U .

Note that $\text{MCDS}_R(G', g_i) - X_i \subseteq V_i'' - X_i$ since $X_i \subseteq V_i''$ and $\text{MCDS}_R(G', g_i) \subseteq V_i''$. Moreover, since $\text{MCDS}_R(G', g_i)$ dominates V , $\text{MCDS}_R(G', g_i) - X_i$ should also cover/dominate U . Thus, we have

$$\text{opt}_i \leq |\text{MCDS}_R(G', g_i) - X_i| \leq |\text{MCDS}_R(G', g_i)| \quad (4)$$

Combining Inequality (3) with Inequality (4), we obtain,

$$|Y_i| \leq \beta |\text{MCDS}_R(G', g_i)| \quad (5)$$

Note that we either have $S_i = X_i$ or have $S_i = X_i + Y_i$ (see Lines 6 and 10 in Fig. 4). Based on Inequalities (2) and (5), we get

$$|\mathcal{S}_i| \leq |X_i| + |Y_i| \leq (\alpha + \beta) |\text{MCDS}_R(\mathcal{G}', \mathbf{g}_i)| \quad (6)$$

According to Lines 11~12 in the *ConstructMCDS_R* algorithm, the finally returned \mathcal{S} satisfies that $|\mathcal{S}| \leq |\mathcal{S}_i|$ for any $i=1, 2, \dots, K$. Furthermore, note that $\text{MCDS}_R(\mathcal{G}', \mathcal{G}'')$ must be identical to some $\text{MCDS}_R(\mathcal{G}', \mathbf{g}_i)$. Combining these observations with Inequality (6), we infer that $|\mathcal{S}| \leq (\alpha + \beta) |\text{MCDS}_R(\mathcal{G}', \mathcal{G}'')|$, i.e., *ConstructMCDS_R* achieves the approximation ratio not greater than $(\alpha + \beta)$. ■

3.4 Overall Algorithm and Numerical Example

Based on the *MaxLifetime*, *MaxInternalLifetime* and *ConstructMCDS_R* algorithms, we present the overall algorithm in Fig. 5.

ConstructMLCDS Algorithm
INPUT: $\mathcal{G}(V, E)$ in which each edge is associated with a lifetime.
OUTPUT: \mathcal{S} which is a CDS of \mathcal{G} .
1: $\mathcal{G}'(V, E') = \text{MaxLifetime}(\mathcal{G})$;
2: $\mathcal{G}''(V, E'') = \text{MaxInternalLifetime}(\mathcal{G}')$;
2: $\mathcal{S} = \text{ConstructMCDS}_R(\mathcal{G}', \mathcal{G}'')$;
3: RETURN \mathcal{S} ;

Figure 5. Pseudo code of the *ConstructMLCDS* algorithm

Before analyzing the performance of the *ConstructMLCDS* algorithm, we give a numerical example to demonstrate the overall algorithm. The given graph $\mathcal{G}(V, E)$ is shown in Fig. 6(a) where there are 20 nodes in V and the lifetimes of edges in E vary from 1 to 4 (time units). We apply our three-phase algorithm in the graph.

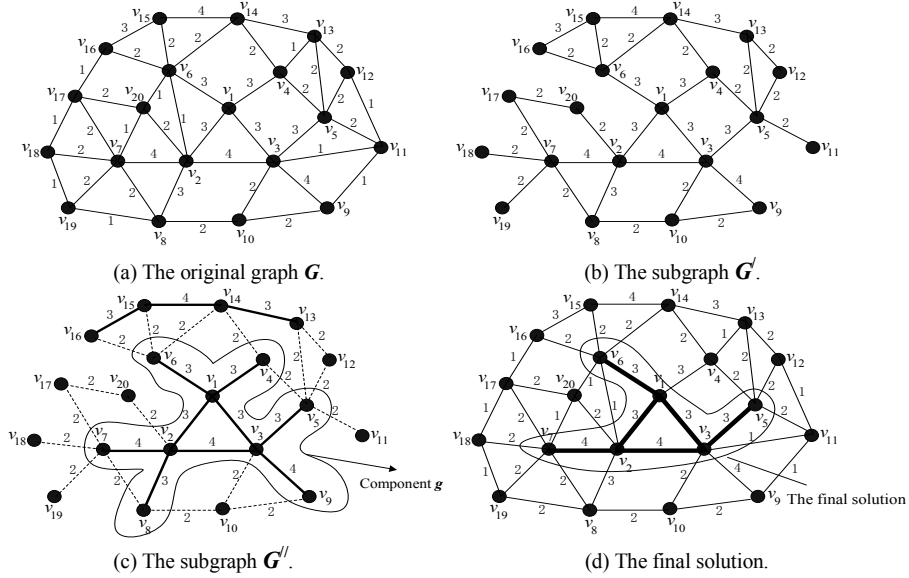


Figure 6. A numerical example for the *ConstructMLCDS* algorithm.

Phase one: $\mathcal{G}' = \text{MaxLifetime}(\mathcal{G})$.

In the first phase, we call the *MaxLifetime* algorithm to generate \mathcal{G}' which is a connected subgraph of \mathcal{G} . As shown in Fig. 6(b), \mathcal{G}' is obtained from \mathcal{G} by pruning all the edges with lifetime equal to 1, since further pruning of the edges with lifetime equal to 2 will cause disconnection of \mathcal{G}' .

Phase two: $\mathbf{G}'' = \text{MaxInternalLifetime}(\mathbf{G}')$.

In the second phase, we call the *MaxInternalLifetime* algorithm to generate \mathbf{G}'' which is a subgraph of \mathbf{G}' . As shown in Fig. 6(c), \mathbf{G}'' is obtained from \mathbf{G}' by pruning all the edges with lifetime equal to 2 (see the dashed edges). In \mathbf{G}'' , there is only one component, denoted by \mathbf{g} in Fig. 6(c), which contains the CDS of \mathbf{G}' . Obviously, further pruning of the edges with lifetime equal to 3 will result in that no component of \mathbf{G}'' can contain the CDS of \mathbf{G}' .

Phase three: $\mathbf{S} = \text{ConstructMCDS}_R(\mathbf{G}', \mathbf{G}'')$.

Based on \mathbf{G}' and \mathbf{G}'' , the third phase is to call the *ConstructMCDS_R* algorithm to compute $\text{MCDS}_R(\mathbf{G}', \mathbf{G}'')$. As we can find in Fig. 6(c), only component \mathbf{g} of \mathbf{G}'' can induce a CDS of \mathbf{G} . According to *ConstructMCDS_R*, we first construct the MCDS of \mathbf{g} by using a MCDS-constructing method, such as the one proposed by Guha *et al.* in [6]. Employing this algorithm on \mathbf{g} , we can easily obtain the corresponding CDS of \mathbf{g} , which is $\mathbf{S} = \{v_1, v_2, v_3\}$.

Obviously, \mathbf{S} is not a CDS of \mathbf{G}' . So, we need select additional nodes from $\{v_4, v_5, v_6, v_7, v_8, v_9\}$ and add them into \mathbf{S} to form a CDS of \mathbf{G}' . According to *ConstructMCDS_R*, this is actually reduced to solving the minimum-set-cover problem, where the universal set $\mathbf{U} = \{v_{11}, v_{12}, \dots, v_{19}\}$ and a family of subsets is: $\mathbf{U}_1 = \{v_{14}\}$, $\mathbf{U}_2 = \{v_{11}, v_{12}, v_{13}\}$, $\mathbf{U}_3 = \{v_{14}, v_{15}, v_{16}\}$, $\mathbf{U}_4 = \{v_{17}, v_{18}, v_{19}\}$, $\mathbf{U}_5 = \{v_{19}\}$ and $\mathbf{U}_6 = \{\}$. Note that \mathbf{U} is the set of nodes in \mathbf{G} that are not dominated by \mathbf{S} yet and subset \mathbf{U}_i is the set of nodes which can be dominated by node v_{i+3} ($i=1, 2, \dots, 6$). By employing the greedy strategy, i.e., choosing the set which contains the largest number of uncovered elements in \mathbf{U} , we can obtain $\{\mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_4\}$. We add the set of corresponding nodes $\{v_5, v_6, v_7\}$ into \mathbf{S} and obtain $\mathbf{S} = \{v_1, v_2, v_3\} \cup \{v_5, v_6, v_7\} = \{v_1, v_2, v_3, v_5, v_6, v_7\}$. The final solution \mathbf{S} is shown in Fig. 6(d). Clearly, \mathbf{S} is a CDS of \mathbf{G} and its lifetime $\rho(\mathbf{S})$ and internal lifetime $\rho_{\text{in}}(\mathbf{S})$ are equal to 2 and 3.

The following theorem presents the time complexity of our *ConstructMLCDS* algorithm.

Theorem 8. The time complexity of *ConstructMLCDS* is not greater than $O(|V|^4)$.

Proof. Due to limited space, please refer to the supplementary in [17]. ■

Let α^* and β^* denote the theoretically best approximation ratio for the MCDS problem and that for the minimum-set-cover problem, respectively. The following theorem shows that any algorithms to the MLCDS problem cannot achieve approximation ratio lower than $\min\{\alpha^*, \beta^*\}$ with respect to the size of the CDS.

Theorem 9. Regarding the size of the CDS involved in the MLCDS problem, approximation ratio less than $\min\{\alpha^*, \beta^*\}$ is not achievable.

Proof. Due to limited space, please refer to the supplementary in [17]. ■

Theorem 10. For the MLCDS problem, *ConstructMLCDS* can compute the CDS such that i) the optimality on both the lifetime and the internal lifetime is achieved and ii) the approximation ratio on the size is not greater than $(\alpha^* + \beta^*)$.

Proof. Due to limited space, please refer to the supplementary in [17]. ■

4 Conclusions

In this paper, we study the MLCDS problem which aims at constructing the CDS with a long lifetime and a small size. We prove that the MLCDS problem is NP-hard. We

propose a three-phase algorithm which constructs a CDS for a given CRN, such that the lifetime of the CDS is maximized and the size of the CDS is upper-bounded by $(\alpha^* + \beta^*)$ of the optimal solution. To the best of our knowledge, it is the first time in the literature that CDS in CRNs is studied and an effective algorithm is proposed.

Acknowledgment

This work is partially supported by FRG2/10-11/111 (Minimum latency of broadcasting and beaconing in multihop wireless networks with multiple radios and multiple channels) and FRG1/10-11/036 (QoS-concerned rendezvous in cognitive radio networks) of Hong Kong Baptist University. Zhiyong Lin is supported by the Natural Science Foundation of Guangdong Province in China, Grant No. S2011040002890. Ivan Stojmenovic is supported by NSERC CRDPJ 386874-09 (Reliable and secure QoS routing and transport protocols for mobile ad hoc networks) and TR32016, Serbian Ministry of Science and Education (Innovative electronic components and systems based on inorganic and organic technologies embedded in consumer goods and products).

References

1. Liu, H., Wan, P., Jia, X.: Maximal Lifetime Scheduling for Sensor Surveillance Systems with K Sensors to 1 Target. *IEEE Trans on Parallel & Distributed Systems*. 17, 1526-1536 (2006)
2. Mini, S., Udgata, S. K., Sabat, S. L.: A Heuristic to Maximize Network Lifetime for Target Coverage Problem in Wireless Sensor Networks. *Adhoc & Sensor Wireless Networks*. 13, 251-269 (2011)
3. Hu, X., Shuai, T., Jia, X., Zhang, M.: Multicast Routing and Wavelength Assignment in WDM Networks with Limited Drop-offs. In: *Proceedings of IEEE INFOCOM 2004*. (2004)
4. Li, J., Zhou, Y., Lamont, L., Deziel, M.: A Novel Framework of Adaptive Routing Design in Cognitive Radio Military Networks. *Adhoc & Sensor Wireless Networks*. 12, 79-101(2011)
5. Shaikh, J. A., Solano-Gonzalez, J., Stojmenovic, I., Wu, J.: New Metrics for Dominating Set Based Energy Efficient Activity Scheduling in Ad Hoc Networks. In: *Proceedings of IEEE Conf. on Local Computer Networks/WLN 2003*, pp. 726-735. (2003)
6. Guha, S., Khuller, S.: Approximation Algorithms for Connected Dominating Sets. *Algorithmica*, 20, 374-387 (1998)
7. Kamrul, I., Selim, G. A.: A Local Algorithm to Compute Multiple Connected Dominating Sets in Wireless Sensor Networks. *International Journal of Parallel, Emergent and Distributed Systems*. 26, 369-380 (2011).
8. Wu, J., Wu, B., Stojmenovic, I.: Power-aware Broadcasting and Activity Scheduling in Ad Hoc Wireless Networks Using Connected Dominating Sets. *Wireless Communications and Mobile Computing*. 3, 425-438 (2003)
9. Simplot-Ryl, D., Stojmenovic, I., Wu, J.: Energy Efficient Backbone Construction, Broadcasting, and Area Coverage in Sensor Networks. In: Stojmenovic I. (ed.) *Handbook of Sensor Networks: Algorithms and Architectures*. Wiley (2005)
10. Alzoubi, K. M., Wan, P.-J., Frieder, O.: Distributed Heuristics for Connected Dominating Sets in Wireless Ad Hoc Networks. *Journal of Communications and Networks*. 4, 1-8 (2002)
11. Wu, Y., Li, Y.: Connected Dominating Sets. In: Liu, H., Leung, Y. W., Chu, X. (eds.) *Handbook of Ad Hoc and Sensor Wireless Networks: Architectures, Algorithms and Protocols*. Bentham Science (2009)
12. Geirhofer, S., Tong, L., Sadler, B.M.: Dynamic Spectrum Access in the Time Domain: Modeling and Exploiting White Space. *IEEE Communications Magazine*. 45, 66-72 (2007)
13. Stevenson, C.R., Chouinard, G., Lei, Z., Hu, W., Shellhammer, S.J., Caldwell, W.: IEEE 802.22: The First Cognitive Radio Wireless Regional Area Network Standard. *IEEE Communications Magazine*. 47, 130-138 (2009)
14. Lin, Z., Liu, H., Chu, X., Leung, Y.-W.: Jump-Stay Based Channel-Hopping Algorithm with Guaranteed Rendezvous for Cognitive Radio Networks. In: *Proceedings of IEEE INFOCOM 2011*, pp. 2444-2452. (2011)
15. Lin, Z., Liu, H., Chu, X., Leung, Y.W.: Ring-Walk Rendezvous Algorithms for Cognitive Radio Networks. *Ad Hoc & Sensor Wireless Networks*. (2012)
16. Slavík, P.: A Tight Analysis of the Greedy Algorithm for Set Cover. *Journal of Algorithms*. 25, 237-254 (1997)
17. http://www.comp.hkbu.edu.hk/~hliu/publications/supplemental/Networking12_Proof.pdf