Chapter 7
Control Channel Management in Dynamic Spectrum Access-Based Ad Hoc Networks

Tao Chen, Honggang Zhang, and Zhifeng Zhao

Abstract In this chapter we introduce the concept of the control channel cloud to solve the control channel problem in dynamic spectrum access (DSA)-based ad hoc networks. A DSA-based ad hoc network is an infrastructure-less wireless network based on DSA and featured by self-organization, self-configuration, and self-healing. One of the challenges in such a network is the common control channel problem, which is caused by the opportunistic spectrum sharing nature of secondary users (SU) in the network. Without a common control channel, it is a challenge to coordinate the behaviors of SU nodes in a DSA-based ad hoc network. The control channel cloud approach, which relies only on the local information exchange to function, aligns the control channel of SU nodes to the same channel in a distributed way if a common control channel exists. It provides a simple but scalable way to synchronize the control channel in a dynamically changed radio environment. The convergence of the proposed approach is proved. The performance of proposed algorithms is studied by simulation.

7.1 Introduction

Powered by *cognitive radio* (CR) [10], which is capable of sharing spectrum resource flexibly and efficiently, dynamic spectrum access (DSA) [17] is regarded as a promising remedy to solve the emerging spectrum scarcity problem. The paradigm shift of the radio spectrum from the conventional command and control allocation to DSA will have fundamental impact on the whole wireless ecosystem. To enable DSA, the adaptation has to be introduced into most layers of the network protocol stack in a system [1], leading to significant changes on the system and network design [3, 5, 14].

Designed for different purposes, wireless systems show great diversity. Obviously, when applying DSA, different types of wireless systems have their own requirements on the system design. Roughly dividing wireless systems into infrastructure based and ad hoc systems, we study the DSA-based ad hoc networks in this chapter. A DSA based ad hoc network is an infrastructure-less wireless network

T. Chen (✉)
VTT, Oulu, Finland
e-mail: tao.chen@vtt.fi

F.R. Yu (ed.), *Cognitive Radio Mobile Ad Hoc Networks*,
DOI 10.1007/978-1-4419-6172-3_7,

which uses DSA for spectrum access. Like a conventional wireless ad hoc network, such a network is featured by self-organization, self-configuration, and self-healing. However, it is more flexible on spectrum, energy, and network resource usage, therefore being superior to wireless ad hoc networks on performance and resource efficiency.

At present the study on DSA-based ad hoc networks is still on its early phase. Many open problems are remaining. We identify the control channel problem as one of the main challenges in DSA-based ad hoc networks. Indeed, in a DSA environment where primary users (PU) and secondary users (SU) of spectrum are assumed, because SUs opportunistically share spectrum with PUs, the network cannot rely on a global common control channel for operation, which distinguishes itself from conventional wireless networks.

In this chapter, we analyze the common control channel problem in DSA-based ad hoc networks and propose feasible solutions. We propose a control channel cloud concept for the control channel selection. A *control channel cloud* is formed by a group of connected SUs using a common control channel. The main reason to introduce the control channel cloud concept is to help the control channels chosen by individual SU nodes aggregate to the same control channel as possible in a dynamic changed radio environment. We introduce four basic operations to manage clouds in the network and provide details to merge clouds in different network dynamics. The algorithms based on four basic operations are presented and their correctness is proven. The performance of the proposed approach is studied by simulation.

The chapter is structured as follows: DSA and impacts on DSA-based ad hoc networks are introduced in Section 7.2; the control channel problem in DSA-based ad hoc networks is examined in Section 7.3; the studied problem and the corresponding system model are presented in Section 7.4; the requirements on the control channel in DSA-based ad hoc networks are summarized in Section 7.5; after in Section 7.6 we propose a new concept on the control channel management, i.e., the control channel cloud concept; the cloud approach under the dynamics of the network is described in Section 7.7; the algorithms and correction of the algorithms are provided in Sections 7.8 and 7.9, respectively; in Section 7.10 the advantages of the cloud approach are emphasized; the performance of the proposed approach is studied by simulation in Section 7.11; finally, the conclusion is drawn in Section 7.12.

7.2 Dynamic Spectrum Access and Impacts on Ad Hoc Networks

DSA is a general term to a new set of spectrum access principles and methods as compared to the conventional command and control spectrum access model [17]. It allows the previously exclusively allocated spectrum resource to be reused among heterogenous systems under predefined constraints on time, space, and interference conditions, and thus significantly improves the usage of precious spectrum resource. The enormous success of Wi-Fi access networks built upon the unlicensed

Industrial, Scientific and Medical (ISM) band and Unlicensed National Information Infrastructure (U-NII) band indicates a bright future of DSA.

According to the use right on the spectrum, DSA can be roughly categorized to three models: dynamic exclusive use model, open sharing model, and hierarchical access model. In the dynamic exclusive use model, spectrum bands are dynamically allocated to spectrum users for exclusive use. It can be seen as a flexible extension of the command and control access model, which enables spectrum trading. The open sharing model, as the name suggested, allows spectrum being equally shared among spectrum users. The spectrum in this context is common and no license is required to access this part of spectrum. The hierarchical access model is the most widely used model in the CR research. It allows an SU of a spectrum to opportunistically share the licensed spectrum with PU of that spectrum when the SU only generates tolerant interference to PUs. The definitions of PU and SU can be found in [1]. Underlay or overlay spectrum share is used in this model.

There is certainly no limitation on which DSA model a wireless ad hoc network should be built upon. However, it is more interesting to study wireless ad hoc networks under the hierarchical access model. As we can see, networks under the dynamic exclusive use model have no big difference from networks under the command and control access model. Moreover, the open sharing model can be regarded as a special case of hierarchical access model without PUs. Therefore in this chapter we assume the use of the hierarchical access model.

7.2.1 DSA-Based Ad Hoc Networks

A wireless ad hoc network by a general definition is a self-organized wireless network without the support of infrastructure. It is normally a multi-hop network where data are routed from the source to the destination through multiple intermediate nodes by distributed algorithms. Wireless ad hoc networks have been studied intensively in the last two decades and the focus is on the routing problem in a single channel radio environment [15]. As shown by Gupta and Kumar in their seminal paper [9], the capacity of a wireless ad hoc network is limited by the number of nodes in the network. This conclusion is drawn based on the assumption that all nodes in the network share limited amount of spectrum. Ad hoc networks based on DSA have potential to increase the network capacity as DSA allows more spectrum to be used by networks. But introducing DSA into wireless networks is more than the capacity improvement. DSA provides a more flexible way for wireless networks to use the radio resource and therefore enabling new applications and use scenarios. DSA-based wireless ad hoc networks are able to share spectrum with infrastructure-based wireless networks. Even better, they can act as bridges between heterogenous networks and help the penetration of conventional infrastructure-based services to a wider area.

As aforementioned, in this chapter the hierarchical access model is applied. Under this model, a DSA-based ad hoc network is more like a multi-channel network where the availability of channels depends on the radio environment. The network coordination in multi-channel systems has been well investigated [4, 12, 13]. Proposals for conventional multi-channel wireless systems usually assume channels are available all the time. This assumption greatly simplifies efforts for coordination. However, it may not always hold in DSA-based ad hoc networks. DSA-based ad hoc networks use the radio resource in an opportunistic nature. They have the following common features and associated challenges:

- Distributed spectrum sensing is required in those networks in order to support DSA. Due to the ad hoc nature, each node should be able to sense the radio environment and determine its available spectrum resource independently. The basic requirement is that the spectrum sensing should be accurate enough to avoid unnecessary interference to PUs while minimizing false alarms on the spectrum hole detection.
- In a DSA-based ad hoc network, a common control channel may not be always available in the network. Network coordination relies on the local share channels of neighboring nodes. This imposes a prominent challenge on the network management.
- Connectivity in a DSA-based ad hoc network is not only influenced by mobility and power control of SU nodes but also by the activities of PUs. Spectrum management becomes a necessary function in a DSA-based ad hoc network to enable adaptive medium access and routing.
- New hidden terminal problems occur in DSA based ad hoc networks. On the one hand, PUs are a new type of hidden terminals to SUs, which cannot be avoided by using explicit signalings like Request to Send (RTS)/Clear to Send (CTS). On the other hand, SUs should not be the hidden terminals of PUs. This is normally solved by spectrum sensing.

7.3 Control Channel Problems in DSA-Based Ad Hoc Networks

Due to dynamics introduced by self-coordination activities, the control problem is critical in distributed networks. Until now most of proposed spectrum control protocols designed for the DSA scenario assume the availability of a common control channel [1]. For instance, Jing et al. [11] used common spectrum coordination channel (CSCC) etiquette protocol for coexistence of IEEE 802.11b and 802.16a networks. Moreover, the cognitive pilot channel (CPC) concept was proposed in [7] for the exchange of the spectrum information among nodes.

As aforementioned, a common control channel may not always exist in DSA-based ad hoc networks. Correspondingly, the topology management of DSA-based ad hoc networks is affected by two main factors: the absence of a common control channel in the network and the frequent topology changes according to the presence of PUs and SUs. In a DSA-based ad hoc network, SUs normally rely on local share channels for the network coordination.

However, until now only few proposals are made under the non-common control channel assumption. Zhao et al. observed that though very limited number of global common channels exist in a network, local neighbors may share numerous channels with others [16]. They proposed a distributed grouping scheme to solve the common control channel problem [16]. Bian et al. [2] used the concept of the segment, which is a group of nodes sharing common channels along a routing path, to organize control channels. In [6], this problem was tackled by a cluster-based approach, where the local users sharing common channels form a dynamic one-hop cluster and the spectrum is managed by cluster heads. DaSilva et al. (chapter 19 in [8]) used the sequence-based rendezvous mode in cognitive radio networks to synchronize SUs. The idea is to assign a well-designed channel access sequence to each SU so that two SUs can meet on certain channels following their channel hopping sequences.

In DSA-based ad hoc networks, the control channel problem is related to physical, medium access control (MAC), and network layer. The air interface structure, signal processing, and spectrum sensing ability in the physical layer determine the reach range and coexisting method of control signals. The MAC layer relies on those control signals to fulfill its channel access functions. The network layer, which builds its routing functions on top of the date delivery service of the link layer, uses network layer control messages to establish and maintain routing. It is easy to see the reliability of control channel and efficiency of control mechanism determine the performance of the network.

We focus in this chapter on the non-common control channel problem in DSA-based ad hoc networks. The key challenges in this problem include the synchronization of control channels, the reliability and scalability of control channel management mechanisms, and the trade-off between performance and control overhead on control channel management. In the following, we will use the control channel cloud concept and cluster-based network formation to deal with the common control channel problem.

7.4 Control Channel Management in DSA-Based Ad Hoc Networks

We give the system model before proceeding to the details of the control channel management method. As aforementioned, we study a DSA-based ad hoc network under a hierarchical spectrum access model. The nodes in studied network are SUs of spectrum, which opportunistically share the spectrum with PUs of the spectrum. The spectrum sharing of SUs can be spectrum overlay and underlay [1]. The principle is that SUs should not violate the tolerant interference level of PUs. We do not study the spectrum sensing here but assume the perfect spectrum sensing in each SU. That assumes synchronization is achieved among SUs to perform the spectrum sensing and avoid false alarm.

SUs coexist with PUs in a given range of spectrum. For SUs, the spectrum is divided into a series of non-overlapped channels. Without losing generality, we assume each channel has equal size. Every channel can be used to transmit control and data messages. A unique channel ID is assigned to each channel, ordered by its frequency range. After spectrum sensing, an SU gets a list of available channels, identified by the channel IDs. A channel is available to a pair of SUs only when the transmitter in the pair does not violate the interference limit of any neighboring PU receiver. When considering the interference at a PU receiver, the sources from multiple SU transmitters must be taken into account. We assume the spectrum sensing function of the SU will handle this problem.

No global common channel is assumed for the control purpose. An SU in the network should be ready to vacate any channel once a PU is detected on the channel. Consequently, SUs must rely on the local common channels for control and coordination.

For the studied DSA-based ad hoc network to operate, the link layer and network layer functions, for instance, neighbor discovery, MAC, and routing, are needed. In this chapter, we only focus on the control channel-specific problems, e.g., control channel selection, mobility, and maintenance problems, at the link layer.

To support the control message exchange, we extend the reference link layer model from [6]. Following the idea in [6], a DSA-based ad hoc network is organized by clusters. As shown in Fig. 7.1, a cluster is a group of nodes managed by a centric node called the cluster head. A cluster has the following properties: the cluster head is selected from the members of the cluster which is one-hop away from all other nodes in the cluster; all nodes in a cluster share at least one common channel; the cluster uses one common channel as the control channel; all control messages of the cluster are transmitted over this channel. We call the control channel of the cluster as the master channel of the cluster. The reasons to organize the DSA-based ad hoc network by clusters have been explained in [6].

As shown in Fig. 7.2, each cluster organizes the channel access time by superframes. In each superframe, there is a period for neighborhood broadcast, in which each node in the cluster is assigned a slot to broadcast the status information about the cluster and the node itself. The status information includes the master channel of the cluster, the available channels of the cluster, the available channels of the node, and the available channels and their associated clusters of neighboring nodes.

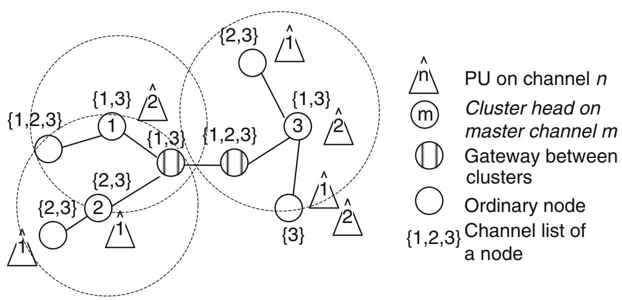

Fig. 7.1 DSA-based ad hoc network formed by clusters

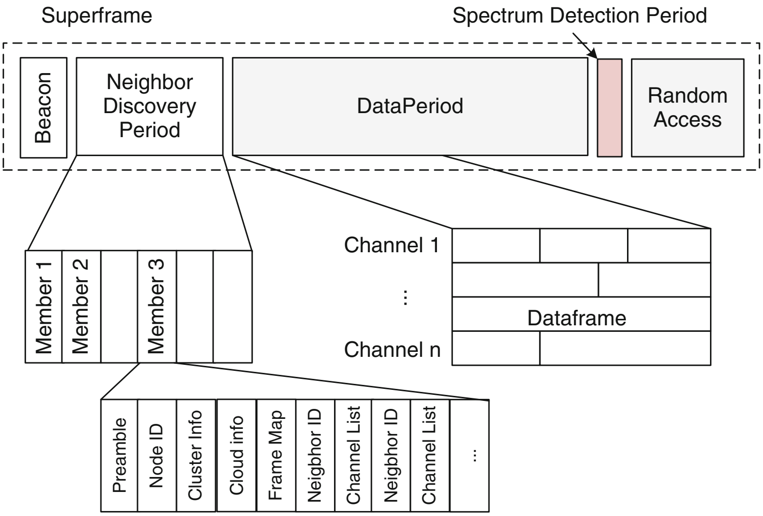

Fig. 7.2 Superframe structure of studied network

Using the status information, nodes are able to establish and maintain clusters. The detail information regarding the clustering in a DSA-based ad hoc network is referred to [6].

Inside a cluster, the collision can be avoided by the coordination of the cluster head. Among clusters, collisions may happen and harm the transmission of control messages. Although it is still an open question, we assume the mechanism like Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) is applied to reduce the collision among clusters. Therefore a node can listen to different channels and decode neighborhood broadcast messages correctly. Each node will gradually know its surrounding. According to the update of the neighbor information, nodes interact with each other to organize the network in a better way, for instance, using better cluster structure and selecting better control channels.

Depending on the size, purpose, and architecture of the network, there are various criteria to select control channels in multi-channel ad hoc networks [4, 12, 13]. In a DSA-based ad hoc network, it is reasonable to maintain the control channels of a group of nodes at the same channel as possible. The reasons are following: first, if a group of nodes can contact each other, using the same control channel provides a reliable and timely way to exchange control information; second, the common control channel approach simplifies the neighbor discovery process; third, the common control channel approach eases the way of routing and reduce overhead; fourth, the common control channel approach can help to manage the channel variation. If a stable common control channel is used by most of nodes, it is easy for them to deal with channel variation occurring in other channels.

The idea in this chapter is to make the nodes of the whole network select common control channels as possible so that efforts for communication cross different control channels are reduced. We assume the activities of PUs are semi-stationary. Therefore the common channels among a group of nodes are relatively stable.

7.5 Requirements of Control Channel Management in DSA-Based Ad Hoc Networks

We have an assumption that the DSA-based ad hoc network studied here is organized by clusters. As shown in Fig. 7.1, a cluster is constructed by a group of nodes voluntarily. Each cluster has a master channel acting as the control channel of all its members. Between two neighboring clusters different master channels may be used even they share common channels. The basic idea for the control channel management in studied network is to develop a distributive mechanism that has neighboring clusters selected the same master channel as possible so that the nodes in the network aggregate to few control channels as possible. It is a distributed consensus process in which nodes by clusters agree to use the same control channel only based on the local information exchange. In an extreme case if there is a global common channel in a fully connected network, using the mechanism all nodes will tune to the same control channel in a long run.

Several requirements need to be considered in the development of such a mechanism:

- The mechanism should work in a distributive way. It should only rely on the exchange of local information to function.
- The mechanism should be reliable. If a stable common channel is available in an area including multiple connected clusters, those clusters will finally converge to a common control channel.
- The mechanism must adapt to the channel changes. After channel changes due to the presence or absence of PUs, the mechanism is able to re-organize the common control channels so that less common control channels are used.
- The mechanism should be robust to the fluctuation of channels. The fluctuation of channels in short time intervals should not trigger the frequent change of control channels in the whole network.
- The mechanism should avoid the oscillation of control channels. A cluster should not bounce among two or more control channels frequently if those channels are always available.

We use a the concept of *cloud* to manage the control channels of the clusters in the network. We define the cloud as following. In a cluster, in addition to the master channel, all members may share other channels. Let us define the share channels of the neighboring cluster are the share channels of all members in a cluster, and the share channels of two neighboring clusters are the intersected share channels of two clusters. If two clusters share the same master channel and other channels, we put those two clusters into a cloud, identified by their master channel and share channels. We include the cloud identity in the neighborhood broadcast messages.

Since a neighborhood broadcast message defined in [6] has included the master channel, we only need to add a new field to include the share channel list of the cloud. Two clusters are in the same cloud if they are connected, not necessary in neighborhood, and have the same master channel and share channel list in neighborhood broadcast messages. Two connected clusters sharing the same master channel but different share channel lists are not in the same cloud. We describe the details of the cloud-based control channel management in the next section.

7.6 Cloud-Based Control Channel Management

We introduce the cloud concept into the DSA-based ad hoc network for the control channel management. The purpose is to have nodes in the network synchronized on the same control channel as possible in a distributive and self-organizing way. A cloud starts from one cluster and grows as more neighboring clusters joining. If two clouds sharing common channels meet, they can be merged to a single one. By the growing and merging processes, the clouds in the network continue evolving. If the channels in network are stable enough, they finally reach the stable state. It is an optimization process cross the network with the objective to put nodes on few common control channel as possible. In this process, the cloud becomes a method to share common channel information beyond one-hop neighbors and makes it possible at the network wide level to synchronize the control channel. All those are done by simply exchanging the cloud information among neighbors.

The cloud idea can be illustrated by a simple example. When a new cluster is formed, it first broadcasts its master channel and share channels in neighborhood broadcast messages. As it learns neighboring clusters from neighborhood broadcast messages of other clusters, it will know the cloud information of other clusters. There may be several clouds around it. It makes a decision to join another cloud or insist on it own, according to the factors like the choices of neighboring clusters, the connections with neighboring clusters, the need to shift the master channel, and others. If it decides to join another cloud working on a different master channel, the cluster head will configure the cluster to shift the master channel starting from the next superframe. The updated cloud information is broadcasted in the next round of neighborhood broadcast messages. As a result the cluster move from it own cloud to other cloud. It continues monitoring the cloud state by collecting the cloud information from neighboring clusters. Once detecting neighboring clusters change their clouds, the cluster will take action to keep staying in a desired cloud. Moreover, the cluster will update its cloud information if the available channels of the cluster are affected by the activities of PUs. In this case, the cluster creates a new cloud and interact with other clusters to deal with the changes.

As we can see, the key idea in the cloud based management is to merge clouds when possible. There are several basic situations where a mergence may occurs. We describe them in details before developing the whole idea of the cloud-based management.

7.6.1 Basic Cloud Operations

We say two clouds meet when a cluster of one cloud detects that there is a neighboring cluster belonging to another cloud. Two clusters are in different clouds if either the master channel or share channels of the cloud are different. The merging process happens if two clouds meet. There are four basic situations: two clouds are built upon the same master channel but different share channels; two clouds relies on different master channels and at least one master channel in their share channels; two clouds use different master channels and none of the master channels are in their share channels; cloud self-refresh. The fourth situation is a special case of the first situation. We introduce it separately since it is an additional approach for self-healing of clouds. Before describing the basic operations, several rules are defined to make the merging process move in one direction.

- The rule of the master channel. It includes several cases: when there are multiple options on the master channel for a cloud, the channel with the lowest frequency wins the game; if two clouds merge and the master channel of one cloud is in their intersected channels but the other is not, the master channel in the share channels is inherited by the new cloud.
- The rule of keeping previous cloud information. For the basic operations of cloud, it is required that each cluster remembers the previous cloud information of its own and neighboring clusters, which is used in the merging process. This rule sets that the previous cloud information of a cluster will be only kept for a given time period and then replaced by the state of the current cloud. This time period is a system parameter. We call a cloud of which the previous cloud information is equal to the current cloud information as a *persistent cloud*.
- The rule for back-off timer to change cloud. A rule is set that difference back-off timer are used to make changes to clouds. The use of back-off timers can be referred to Section 7.8. The back-off timer to change the master channel is longer than that to only update the share channel list.
- The rule of subset cloud overriding. We said one cloud is a *subset* of the other cloud if two clouds share the same master channel, and the share channels of one cloud are the subset of the share channels of the other cloud. We also define the cloud broadcasting as a cluster receives a neighborhood broadcast message from another cluster. The rule is that if a cloud receives a cloud broadcasting from a cloud which is its subset but is not its previous cloud, the cloud will convert to the new cloud.
- The rule of holding in a cloud. If a cluster joins a new cloud but receiving cloud broadcasting messages from its previous cloud, it will keep staying in the new cloud.
- The rule of overriding persistent cloud. It is that a persistent cloud will convert to a cloud with more share channels than it only when the cloud's previous cloud is a persistent cloud.

7.6.1.1 Operation 1: Same Master Channel But Different Share Channels

This situation can be divided into two cases: two clouds meet as they grow, or a new cloud is created inside a cloud due to the activities of PUs. Since a PU can be active or inactive, the new cloud can have less or more channels than the original cloud.

A cluster can distinguish if a new cloud is created and growing inside another cloud or two clouds meet as they grow based on the records of previous cloud information. If a new cloud is created from an old one, a cluster belonging to the new cloud will keep staying in the new one even it receives cloud broadcasting messages from the old cloud since it knows its previous cloud is the old one. For a cluster in the old cloud, it will join the new cloud as it treats the new cloud as a chance to adapt to channel changes. At the end, the old cloud will evolve to the new one. In a case a cluster finds there are more than one new cloud to join, it will join the cloud with most neighboring clusters on it. Otherwise it will join the first one who sends the cloud broadcasting message. We have to emphasize that the new cloud here is a cloud uses the same master channel but different channel lists from the original cloud, and therefore having no impact on the delivery of control messages.

It also worth noting that a new cloud may have more share channels than the old cloud. This provides an opportunity for a cloud to increase its share channels. As we can get from the common sense, if two clouds are merged, the share channels of the new cloud are normally reduced. At the end, if there is no mechanism to refresh the share channel lists of clouds at all according to the changes of the radio environment, clouds will become stale and finally not function correctly. By allowing a new cloud increases the share channels of an old cloud, we can introduce a cloud refreshing mechanism in which the clusters of a cloud can split from the cloud, create a new cloud with their local share channels, and refresh the old cloud by mergence.

If two clouds grow and finally meet, two neighboring clusters belonging to each cloud will detect this happening. Those two clusters will negotiate and create a new cloud, which uses the intersected share channels of two clouds as its share channels. If one cloud is a subset of another cloud, no new cloud will be created but the one cloud with less share channels will gradually convert the other one. We can use the previous cloud information to guarantee the mergence of two clouds occur correctly. Without lose generality, let us say a new cloud is created among two clouds. The clusters of the new cloud remember that its previous cloud is one of those old clouds. If a cluster in the new cloud receives a cloud broadcasting from its previous cloud, it will ignore this message. On the other hand, if it receives a cloud broadcasting from the other old cloud, it finds that the new cloud is the subset of this old cloud and then does nothing. Therefore it will take no action when receiving a cloud broadcasting from old clouds. The clusters in old clouds, while receiving a cloud broadcasting from the new cloud, will join the new cloud as the new cloud is not their previous clouds. As a result, two clouds will finally be merged to one cloud.

7.6.1.2 Operation 2: Different Master Channels But at Least One Master Channel in Share Channels

In this situation if two clouds meet, a new cloud will be created using the master channel of one cloud. There are two cases in this situation: two clouds have the same share channels or two clouds share some of their share channels. According to the rule of the master channel, in the former the cloud with the master channel of a lower frequency will convert the other cloud. For the latter, a new cloud using the intersected share channels of two clouds is created and grows among two clouds. We need to decide the master channel in the new cloud. It depends on the share channels and master channels of two clouds. If the share channels of one cloud are included in the other cloud, the master channel of the cloud will be inherited by the new cloud. Otherwise, the rule of the master channel is applied.

We can show that in this situation two clouds will finally be merged. If two clouds have equal share channels, the one with the master channel of a lower frequency remains untouched but the clusters in the other one will gradually move the first one. If two clouds have no equal share channels, according to the rule of subset cloud overriding, the clusters in two clouds will join the new cloud. For the clusters in the new cloud, assuming they previously belong to the cloud one, if they receive cloud broadcasting from the cluster one, it will take no action since their previous cloud is the cloud one; if they receive cloud broadcasting from the cluster two, since from their previous cloud information they will find the share channels of the current cloud is a subset of those in the cloud two, they conclude that their current cloud is derived from the cluster two and thus take no action. Following the same deduction, the clusters will take no action if they previously belong to the cluster two. Therefore two clouds will be merged.

7.6.1.3 Operation 3: Different Master Channels and No Master Channel in Share Channels

If no master channel of two clouds is in their share channels, a new master channel has to be selected for the new created cloud. According to the rule of the master channel, the share channel with the lowest frequency becomes the new master channel of the new cloud. Finally the new cloud will merge two old clouds.

It is easy to show that the merging process happens in one direction. For a cluster in any of two clouds, if it finds the new cloud, it will move to it since the new cloud is its subset. For a cluster in the new cloud, if it receives cloud broadcasting from any of two old clouds, it will take no action as its previous cloud information shows it just moves from another cloud.

7.6.1.4 Operation 4: Cloud Self-Refreshing

The aforementioned operations only occur when two external clouds meet. That means a cloud can change itself only when meeting with another cloud. If the clouds of a network are evolved to a stable situation, the evolution of clouds will be stopped even there is a chance to move to a more optimized cloud configuration. That chance happens when PUs release their spectrum so corresponding channels returns to the network. A self-refreshing operation continuously makes the network find new opportunities for better cloud configuration.

This operation works as follows. According to the rule of keeping previous cloud information, after a given time of staying in a cloud, a cluster will replace its previous cloud information with the current cloud information. After a given time when a cluster's previous cloud is its current cloud, the cluster will include all its available channels into the share channel list of its cloud. A new cloud is created if the available channels are more than the share channels of the cloud. According to the rule of overriding persistent cloud, the old cloud will convert to the new cloud. For the new cloud, if it receives cloud broadcasting from the old cloud, it knows it is from its previous cloud and takes no action. If a new channel is available to a cloud, according to this operation, the new cloud will include that new channel. When the new cloud meets other external clouds, the other three operations are applied.

7.7 Dynamics of Network

We show in this section how the operations of cloud are applied in a network under different network conditions. Those conditions include the initiation of the network, the loss and return of channels due to the PU activities, and the topology change due to nodes joining or leaving the network. We also introduce how clouds refresh themselves in this section.

7.7.1 Initiation

A DSA-based ad hoc network is organized in an autonomous way. Every node in the network is an equal entity from the communication viewpoint. A network is formed when messages can be delivered from one node to any other nodes in the network. As the channel availability, topology and nodes of the network undergo dynamic changes during the life time the network, the network is expected to experience frequent network partition and re-union. Therefore it is hard to define the initiation phase of the network since it may happen all the time during the life time of the network. To illustrate how a cloud is formed at the beginning, we show here a special initiation phase of the network where all nodes in the network start joining the network at the same time.

As described in [6], a node will first detect available channels and listen to those channels for available clusters. If it cannot detect any working clusters on its available channels, it will create the cluster of its own by choosing a proper master channel, which can be the channel of the lowest frequency, the channel of the best quality, or a randomly chosen channel. Otherwise, it will join the first cluster that it detects. A new cluster forms its own cloud at the beginning by using its share channels as the share channels of the cloud. The cloud information is broadcasted in the neighborhood discovery process. As the new cluster continuously monitors the radio environment, it will gradually know neighboring clusters on different channels. The cluster optimization algorithm, for instance, minimum dominating set (MDS) algorithm in [6], is run to optimize the cluster structure of the network. Meanwhile, the cloud operations 1–3 are performed among clusters to align nodes on the same control channel if possible. The initiation phase of the network can be described as the initiation and optimization of clusters and clouds to establish optimized cluster structure on few control channel as possible.

7.7.2 Loss and Return of Channels

We assume the loss and return of channels in the network are mainly affected by the activities of PUs. Nodes in the area have to vacate a channel when it is occupied by PUs in that area. On the other hand, the channel can be re-used by SU nodes if nodes sense the channel and make a conclusion that no surrounding PU is in its active state.

If nodes have to vacate a channel, the available channels of their clusters may be reduced. If the master channel of a cluster has to be vacated, the cluster will be dismissed. Nodes of the cluster join the network by forming new clusters. Clouds will be created in new clusters and interact with old clouds using the operation 1–3 to find a new optimized cloud distribution. If the vacated channel is not the master channel but a channel in the share channel list of a cloud, the affected clusters will update the share channel list of their clouds, and the operation 1 is applied to update other clusters in their clouds.

The return of a channel to a node will not immediately affect the cloud distribution of the network. However, if the new channel increases the share channel of a cluster, in self-refreshing process, the operation 4 will bring the new channel to the cloud. The network then re-optimizes the cloud distribution using the operation 1–3. The best result is that more nodes synchronize on the same master channel. However the worst case is that the return of a channel will have no impact on the cloud distribution.

7.7.3 Node Joins or Leaves Network

A node follows the procedure described in Section 7.7.1 to join an established network. That is it will join a working cluster or establish the cluster of its own. The impact on the cloud dynamics when a node join the network depends on the contribution of the node to the connectivity of the network and its available channels. If the node does not create new neighboring clusters for its cluster, it will have no impact on the master channel of the cloud, but it may reduce the share channels of the cloud if it changes the share channels of the cluster. The operation 1 will be applied in this case. If the new node connects two previous separated networks, it may have a profound impact on the cloud distribution of the whole network. If two previous separated networks use different master channels but now it is possible to move to a common master channel, the operation 1–3 will be applied to drive the cloud evolution.

In our network, a node can be a cluster head or a normal node in a cluster. The impact on the cloud when a node leaving network relies on the role of the node in the cluster. If a cluster head leaves the network, the cluster will be dismissed and the remaining nodes of the cluster will follow the initiation process to join the network. In this process, the operation 1–3 will be used to re-organize clouds in the local area. If the leaving of the node makes it possible to merge clouds of the network, the operation 4 will initialize this process and then the operation 1–3 will be invoked to complete the process. If a normal node in a cluster leave the network, there is no impact on the cloud at the beginning. But later on the operation 4 will be applied to refresh the cloud of the cluster. If after the self-refreshing, it is possible to merge more clouds in the network, the operation 1–3 will be applied. Note that the leaving of a node may create network partition. The use of the operation 4 will drive the evolution of two separated networks independently.

7.7.4 Refresh Cloud

As a cloud merges with others, it will normally reduce its share channel list. As a result, a cluster may have more share channels than its cloud. The refreshing of a cloud refers to a cluster replacing the share channels of its cloud with its own share channels. It is the way a cloud can increase its share channels when more channels are available to the cloud. For instance, the share channels of a cloud may increase if PUs release channels to SUs or nodes leave the network. The refreshing of cloud provides a way to make cloud adapt to channel changes.

The refreshing process can be regarded as two steps: first, a new cloud is created by a cluster using the self-refreshing operation; second, the new cloud start merging surrounding clouds by applying the rule of overriding persistent cloud. The introduction of the persistent cloud concept and mechanism makes the refreshing happen only when clouds are stable for enough long. It gives the self-refreshing cloud merging a lower priority than other merging. The reason is straightforward: if other merging happens, it will override the self-refreshing merging.

7.8 Algorithms for Control Channel Management

The whole algorithms for control channel management include the algorithms on the ordinary nodes of a cluster and those on the cluster head. All algorithms are run in a distributed way. The algorithms in an ordinary node of a cluster performs neighboring cluster discovery and cloud broadcasting. The cloud operations are done in the cluster head. We only describe the algorithms on the cluster head in this section. The basic functions of the algorithms on the cluster head consist of three parts: initiation of the cloud, cloud merging, and cloud self-refreshing. The initiation of the cloud in a cluster is rather simple. It starts by building the own cloud of the cluster and then merging with other surrounding clouds. The own cloud of a cluster is built by using the share channels of the cluster as the share channels of the cloud.

In this section, we provide pseudo-codes for cloud merging and cloud self-refreshing, listed in Algorithms 1, 2. The output of those two algorithms is the updated cloud information, which will be broadcasted in the beacon period of the next superframe. The functions for broadcasting and receiving of cloud information

Algorithm 1: MergeCloud run on CH_i

Data: Cloud information on CH_i and neighboring CH_j
Result: Updated Cloud information of CH_i
begin
  if $mCh_i = mCh_j$ **and** $SC_i \neq SC_j$ **then**
    if $T_s = 0$ **then**
      if $SC_j \in SC_i$ **then**
        if $SC_i' \neq SC_j$ **then**
          $SC_i' = SC_i$;
          $SC_i = SC_j$;
      else if $SC_i \in SC_j$ **then**
        if $SC_i' = SC_i$ **and** $mCh_i' = mCh_i$ **then**
          $SC_i' = SC_i$;
          $SC_i = SC_j$;
      else
        $SC_i' = SC_i$;
        $SC_i = SC_i \cap SC_j$;
      Reset T_s;
    else
      $T_s = T_s - 1$
  else if $mCh_i \neq mCh_j$ **then**
    if $T_d = 0$ **then**
      if $mCh_i \in SC_j$ **and** $mCh_j \in SC_i$ **then**
        $mCh_i' = mCh_i$;
        $mCh_i = \min(mCh_i, mCh_j)$;
      else if $mCh_j \in SC_i$ **then**
        $mCh_i' = mCh_i$;
        $mCh_i = mCh_j$;
      else if $mCh_i \notin SC_j$ **then**
        $mCh_i' = mCh_i$;
        $mCh_i = \min(SC_i \cap SC_j)$;
      $SC_i' = SC_i$;
      $SC_i = SC_i \cap SC_j$;
      Reset T_d;
    else
      $T_d = T_d - 1$

Algorithm 2: SelfRefresh run on CH_i

```
Data: Cloud information on CH_i
Result: Updated Cloud information of CH_i
begin
    if SC'_i ≠ SC_i or mCh'_i ≠ mCh_i then
        if T_p = 0 then
            SC'_i = SC_i;
            mCh'_i = mCh_i
            Reset T_p;
        else
            T_p = T_p − 1
    else
        if T_r = 0 then
            SC_i = CC_i;
            Reset T_r;
        else
            T_r = T_r − 1
```

Table 7.1 Notation

CH_i	Cluster Head *i*
mCh_i	Master channel of CH_i
mCh'_i	Previous master channel of CH_i
CC_i	Share channel list of cluster CH_i
SC_i	Share channel list of cloud stored at CH_i
SC'_i	Previous share channel list of cloud stored at CH_i
T_s	Back-off timer for cloud merging on same master channel
T_d	Back-off timer for cloud merging on different master channels
T_p	Back-off timer to change to persistent cloud
T_r	Back-off timer for cloud self-refreshing

are not included in the algorithms. The notations used in the algorithms are given in Table 7.1.

Each cluster is identified by its cluster ID i. In the algorithms the host cluster running the algorithms is identified as CH_i, while neighboring clusters are identified as CH_j. Four back-off timers are introduced to postpone the change of a cloud. T_s is used to update the share channels of a cloud but keeping the master channel the same; T_d is used to move a cloud to a different master channel; T_p is used to turn a cloud to a persistent cloud after it becomes a non-persistent cloud; T_r is used to self-refresh a persistent cloud. The time unit for those timers is one superframe period. Note that those timers are set that $T_\text{s} < T_\text{d} << T_\text{p} < T_\text{r}$.

7.9 Correctness of Cloud Formation Algorithm

Definition 1 A connected component of a network is a group of network nodes in which any two nodes can be reached by a direct link or links from other nodes in the group and no other nodes of the network can be added in the group.

Proposition 1 *In a radio environment where the availability of channels is fixed, the operation 1–3 finally create a stable cloud distribution.*

Proof First we prove that after the operation 1–3, a cluster will never return to the previous clouds it has stayed. As we can see, those three operations will never increase the share channels of a cloud. If the share channels of the new cloud and the previous cloud is the same, which happens in the operation 2, the new cloud will use a lower frequency master channel. Otherwise the new cloud will reduce its share channels. As a cluster continuously moves between different clouds, the new cloud to which it jumps will either use lower frequency channel or reduce share channels. Since the number of channels is limited, a cluster will finally be fixed at one cloud. Therefore for the whole network, the clouds of all nodes will finally be fixed. It implies the operation 1–3 evolve clouds in one direction.

Lemma 1 *In a radio environment where the availability of channels is fixed, if a connected component has common channels, by the operation 1–3 and without self-refreshing it will finally converge to a single control channel.*

Proof The operation 1–3 guarantee the common channels remaining in the share channel lists of clouds. The operation 2 makes two clouds with the same share channel list using the same control channel. The Proposition 1 makes sure the final cloud distribution stable. As a result, the connected component will converge to a single control channel.

Lemma 2 *If a connected component has common channels, the self-refreshing operation does not affect the connected component to select its common control channel.*

Proof If a connected component has already use a common control channel, since the self-refreshing operation does not change the master channel, the common control channel of the connected component is not affected.

If a connected component has common channels but has not converge to a common control channel, the operation 1–3 will merge clouds. Let us say a new cloud created by the operation 1–3 merges a old cloud. The self-refreshing can be considered in two cases: the old cloud and new cloud start self-refreshing, respectively. For the former, since self-refreshing only increases the share channels of the cloud, the new cloud is still the subset of the refreshed cloud. Consequently, it will not change the result of merging. For the latter, we can show that it does not affect the merging process either. As described in the operation 4, the self-refreshing operation includes a cluster into a new cloud only when the cluster is in a persistent cloud. New clouds created by the operation 1–3 are not persistent clouds. They will immune from self-refreshing as long as they are not persistent clouds. Since the time to become a persistent cloud is much longer than that to convert the cloud of a cluster, the edge of a new cloud will not be affected by its self-refreshing and will continue converting the old cloud.

Since the self-refreshing operation does not affect the operation 1–3, combining with the Lemma 1, the conclusion holds.

Lemma 3 *If common channels exist in a connected component after the PU activities, and at least one of them keeps available enough long, the connected component will finally converge to one common control channel.*

Proof Note that the PU activities refer to a PU appears or disappears in/from the network. We only discuss the case a PU occupies the control channel since other cases have no direct impact on the control channel of connected component.

If a PU shows up and occupies the control channel in a region, new clouds will be formed at that region. Those clouds will interact with the old cloud in the unaffected area. According to the Lemma 1 and 2, a new cloud based on a new common control channel will finally be formed in the connected component. This also holds if there are many PUs in different areas of the connected component occupying common channels.

If PUs release their occupied channels, according to the operation 4, the clusters in the network will self-refresh their clouds. The Proposition 1 states if the availability of channels is stable enough, the cloud distribution will be stable. According to the rule of keeping previous cloud information, those clouds will become persistent clouds. The self-refreshing operation will finally return released channels to the cloud. If there are common channels again in the connected component, according to the Lemma 1 and 2, a new cloud based on a common control channel will eventually be formed.

Lemma 4 *If two connected components become connected and there exists at least one common channel available enough long, the new connected component will finally use a single common control channel.*

Proof If two connected components become connected, they are merged into a single connected components. According to the Lemma 3, the new connected component will finally use a single common control channel.

Theorem 1 *If common channels exist and they are available in the network for time enough long, a common control channel will finally be selected by the network.*

Proof A network is composed by one or several connected components. According to the Lemma 4, when it becomes a single connected component, and meanwhile common channels are available for time enough long, the network will converge to a common control channel.

7.10 Advantages of Cloud Approach

The cloud-based control channel management approach is specifically proposed for DSA-based ad hoc networks. It fulfills the needs of DSA and has the follow advantages to deal with the control channel problem in similar networks.

- There is no need to assign a unique id to each cloud. Instead, a cloud is identified by the master channel and its share channel list. This significantly eases the cloud management. Clouds on the same master channel but with different share channel lists are different clouds. Merging happens if two of such clouds meet.

- Including a share channel list in a cloud makes it possible to align the control channel multiple hops away only relying on local information exchange. The proposed algorithms are therefore scalable.
- The cloud is managed in a loose way based on the cluster. A cluster has its own freedom to create and merge a cloud. It enables robust and fast control channel management and makes it possible for clouds quickly adapting to environment changes.
- Managing the cloud at the cluster level simplifies the control channel management. The cluster is responsible for the channel management in a local group and thus reduces the overhead for cloud interaction at the node level.
- The combination of cloud merging and self-refreshing mechanisms balances the adaptive ability for rapid changes of channels and long-term selection of control channel.

7.11 Simulation Studies

The behaviors and performance of the proposed algorithms are studied by simulation. The simulation environment is settled in a 600 m $\times$ 600 m 2-dimensional playground, in which PUs and SUs are randomly distributed according to Poisson processes. The maximin reach range of a PU is set to 200 m and that of an SU is 150 m. A number of channels are shared by PUs and SUs. A PU randomly chooses a channel to operate. During its active time, the surrounding SUs which may cause intolerable interference to the PU have to vacate that channel. We assume the transmission of cloud broadcasting message is collision free. The neighboring SUs listening on that channel can decode that message successfully. Cloud broadcasting messages are deliver to cluster heads by member nodes in the following beacon period. The algorithms are run on each cluster head to decide its cloud configuration.

The simulation is run sufficient times and the average values of the statistic data are analyzed. In each simulation run, PUs fix their operating channels. SUs initialize clusters and evolve their cluster structure according to the MDS algorithm proposed in [6]. The cloud algorithms are run based on the established clusters. We compare the control channel and cluster distributions without and with the cloud algorithm. In the network without the use of the cloud concept, the cluster head decides its own master channel without referring to the choices of neighboring clusters. The simulation results are presented and analyzed as follows.

The number of connected components on all master channels is a good performance indicator of the algorithm. A connected component on a master channel refers to a group of connected SUs in which all of them share the same master channel and no other SUs on the same master channel are connected to this group. Figure 7.3 shows the connected components on all master channels as a function of SU number. It is the result based on the cloud algorithms under different available

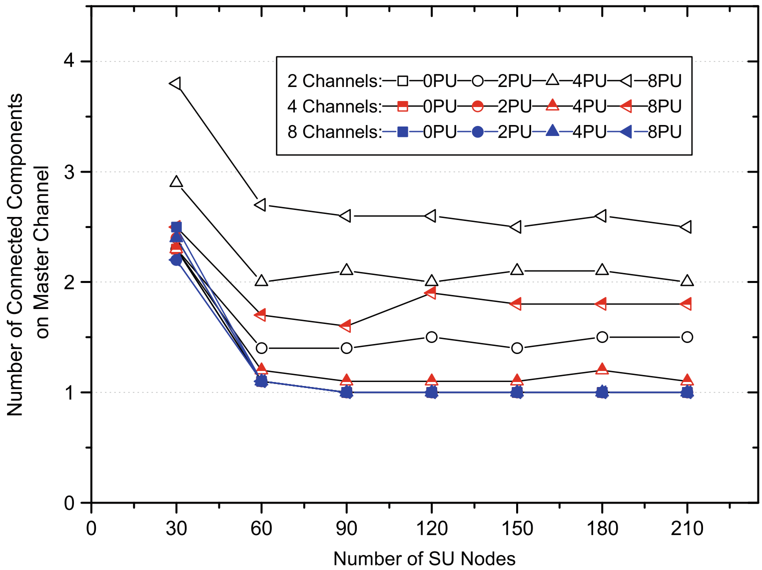

Fig. 7.3 Number of connected components on master channels as a function of number of SUs

channels and PUs setup. From the figure, it is easy to observe that more available channels lead to less connected components. In the case where the channel number is 8, there is only one connected component for the SU number greater than 90. That means only one control channel is used. We also notice the impact of PUs on the control channels. That is a larger PU number resulting in more connected components, which implies that more control channels have to be used. The figure shows the SU number of 30 has significantly more connected components than other larger SU numbers. It is because all nodes in the network cannot be fully connected when the SU number is 30 or less.

Figure 7.4 gives the comparison of the cloud and no cloud approach on the connected components on all master channels. We have already analyzed the performance of the cloud approach on connected components in Fig. 7.3. From Fig. 7.4, we can see when the PU number increases, the no cloud approach creates more connected components, therefore using more control channels. In the case of 8 channels, the cloud approach converges to a single control channel when the SU number is greater than 90, while for the no cloud approach, more than one control channels are used when the PU number is 4 and 8. That shows the effort of the cloud approach to drive SUs to the same control channel.

The other performance indicator of the cloud algorithms is the number of SUs on the dominant master channel. The dominant master channel refers to a channel selected by most of SUs as the control channel. Figure 7.5 provides the percentage of SUs on the dominant master channel as a function of SU number. The percentage is defined as the number of SUs on the master channel to the total number of SUs

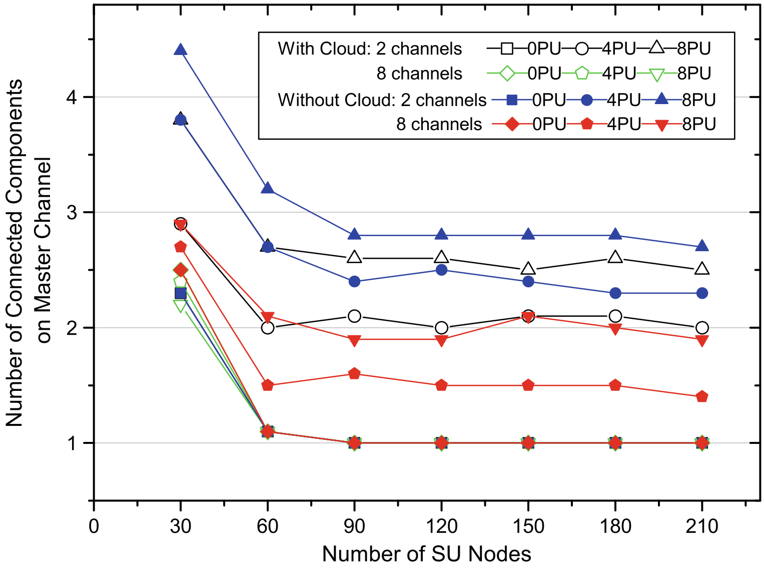

Fig. 7.4 Number of connected components on master channels as a function of number of SUs. Cloud approach vs. no cloud approach

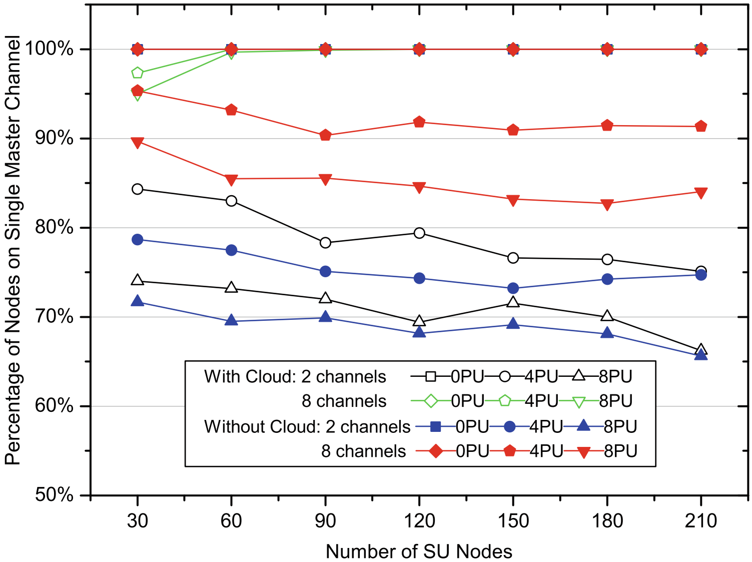

Fig. 7.5 Percentage of SUs on master channel selected by maximum number of SUs as a function of number of SUs. Cloud approach vs. no cloud approach

in the network. In line with Figs. 7.4, 7.5 shows that more PUs lead to less SUs on the dominant master channel. Moreover, in the no cloud approach, less SUs are on the dominant master channel than the cloud approach when the PU number is large. Since when the PU number is 0, the no cloud approach almost has the same results on control channels as the cloud approach, while in other cases their results are significantly different, it illustrates the cloud approach has advantage on the control channel selection.

Figure 7.6 reveals the impact of the cloud approach on the cluster formation. It shows that as the SU number increases, the number of clusters increases in both approaches. The increasing rates in both approaches are not linear and slow down when the SU number increases. Compared to the no cloud approach, the cloud approach slightly increases the number of clusters in the networks. That difference is prominent when there is only two channels and it becomes dimmish as available channels increase.

The final figure shows how the number of PUs impacts the control channel selection of SUs. As seen from Fig. 7.7, the increasing of the PU number leads to the decreasing of percentage of SUs on the dominant master channel. When there are more channels, more SUs are aggregated on the dominant master channel. When the channel number is 8, the cloud approach results in almost all SUs on the dominant master channel. It also shows that the no cloud approach has less SUs on the dominant master channel in all cases when PUs are in active.

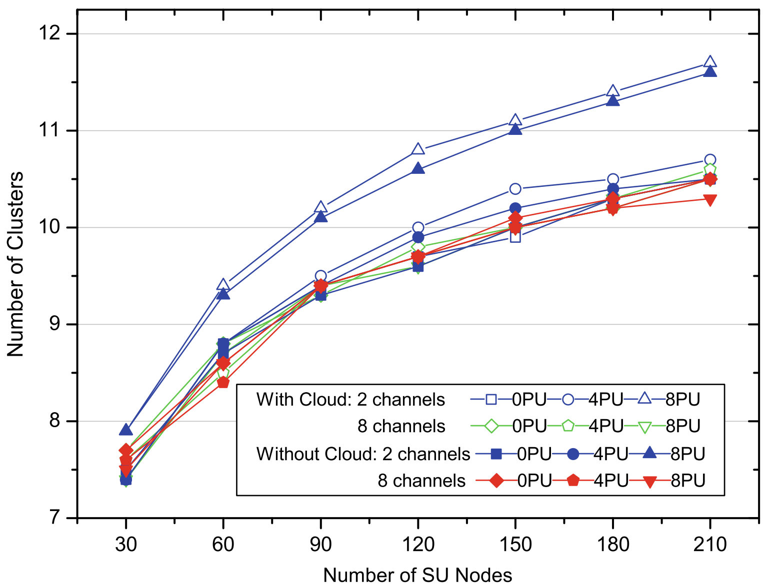

Fig. 7.6 Number of clusters as a function of number of SUs. Cloud approach vs. no cloud approach

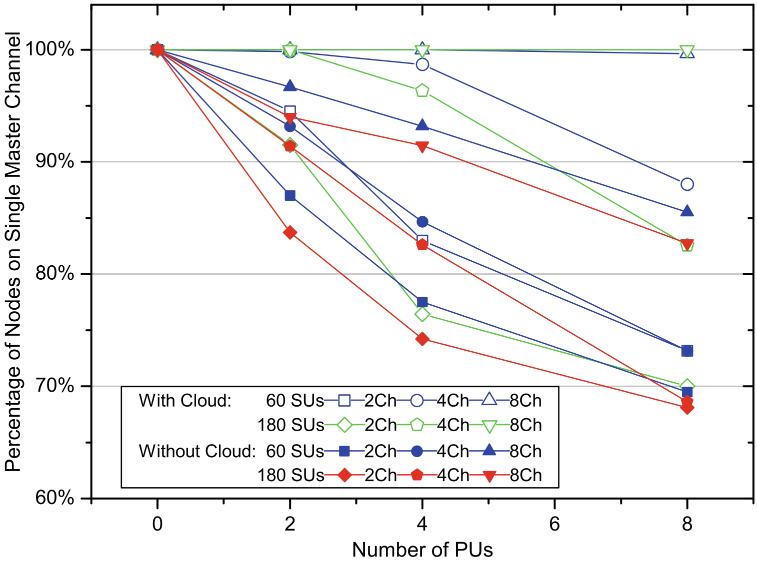

Fig. 7.7 Percentage of SUs on master channel with most SUs as a function of PUs. Cloud approach vs. no cloud approach

7.12 Conclusion

In this chapter, we study the control channel problem in DSA-based ad hoc networks. The control channel cloud approach is proposed to synchronize the control channel of SU nodes in the network to the same channel if a common channel exists. The motivation behind this approach is that if nodes in an ad hoc network using the same control channel, the signaling will be significantly simplified. We provide four basic operations to manipulate control channel clouds according to the cloud state of nodes in the network. Algorithms are developed based on the basic operations and implemented on a cluster based network. The convergence of the algorithms is proved. The simulation study shows the proposed approach converge the control channel under different channels, SUs, and PUs setups.

References

1. I.F. Akyildiz, W.Y. Lee, M.C. Vuran, and S. Mohanty. Next Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. *Computer Networks*, 50(13):2127–2159, 2006.
2. K. Bian and J. Park. Segment-Based Channel Assignment in Cognitive Radio Ad Hoc Networks. *CROWNCOM 2007, Orlando, August*, 2007.
3. D. Bourse, K. El-Khazen, A. Delautre, T. Wiebke, M. Dillinger, J. Brakensiek, K. Moessner, G. Vivier, and N. Alonistioti. European Research in End-to-End Reconfigurability. *Proc. of IST Mobile & Wireless Communications Summit, Lyon, France*, pages 27–30, June 2004.

4. S.P. Chaudhuri, R. Kumar, and A.K. Saha. A MAC protocol for Multi frequency Physical layer. Technical report, Technical Report, Rice University, Houston, TX, January 2003.
5. T. Chen, H. Zhang, M. Matinmikko, and M.D. Katz. CogMesh: Cognitive Wireless Mesh Networks. In *WMSN Workshop in conjunction with IEEE GLOBECOM 2008, New Orleans*, pages 1–6, 2008.
6. T. Chen, H. Zhang, X. Zhou, G.M. Maggio, and I. Chlamtac. CogMesh: A Cluster Based Cognitive Radio Mesh Network. F. Fitzek et al ed., Cognitive Wireless Networks Concepts, Methodologies and Visions: Inspiring the Age of Enlightenment of Wireless Communications, pages 657–678, Springer, Dordrecht, Netherlands, 2007, ISBN: 978-1-4020-5978-0.
7. P. Cordier, D. Bourse, K. Moessner, et al. E2R Cognitive Pilot Channel concept. *Proc. of IST Mobile & Wireless Communications Summit 2006, Myconos, Greece*, June 2006.
8. B.A. Fette. *Cognitive Radio Technology*. Academic, Burlington, MA, 2009.
9. P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.
10. S. Haykin. Cognitive Radio: Brain-Empowered Wireless Communications. *Selected Areas in Communications, IEEE Journal on*, 23(2):201–220, 2005.
11. X. Jing and D. Raychaudhuri. Spectrum Co-existence of IEEE 802.11 b and 802.16 a Networks Using the CSCC Etiquette Protocol. *DySPAN 2005, Baltimore*, MD, November 2005.
12. J. So and N.H. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver. *MobiHoc 2004, Tokyo, Japan*, pages 222–233, May 2004.
13. C.Z. Tang and JJ Garcia-Luna-Aceves. Hop-Reservation Multiple Access (HRMA) for Ad hoc Networks. *IEEE INFOCOM 1999, New York*, NY, March 1999.
14. RW Thomas, DH Friend, LA Dasilva, and AB Mackenzie. Cognitive Networks: Adaptation and Learning to Achieve End-to-End Performance Objectives. *Communications Magazine, IEEE*, 44(12):51–57, 2006.
15. C.K.K. Toh. *Ad Hoc Wireless Networks: Protocols and Systems*. Prentice Hall PTR Upper Saddle River, NJ, 2001.
16. J. Zhao, H. Zheng, and G. Yang. Distributed Coordination In Dynamic Spectrum Allocation Networks. *IEEE DySPAN 2005, Baltimore*, MD, November, 2005.
17. Q. Zhao and B.M. Sadler. A Survey of Dynamic Spectrum Access. *Signal Processing Magazine, IEEE*, 24(3):79–89, 2007.