Robust Provisioning of Multicast Sessions in Cognitive Radio Networks

Abdullah M. Almasoud [1, 2]   Ahmed E. Kamal [1]

[1] Dept. of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA
[2] Dept. of Electrical Engineering, Salman Bin Abdulaziz University, Al-Kharj, Saudi Arabia.
{almasoud, kamal}@iastate.edu

***Abstract*— Today's wireless networks use fixed spectrum over long term and fixed geographical regions. However, spectrum utilization varies by time and location, which leads to temporal and special spectrum underutilization. Cognitive radio is an emerging technology that enables dynamic sharing of the spectrum in order to overcome the spectrum underutilization problem. In this paper, we consider the problem of supporting the multicast service mode in cognitive radio networks. Moreover, we are concerned with supporting this mode of service such that it is robust in the face of failures. We develop two algorithms which provide robust multicasting in such networks. Our proposed algorithms are: 1) multicast sessions protection with link-sharing trees and 2) multicast sessions protection using rings. These algorithms provision multiple multicast sessions, and protect them against a single primary user interruption at a time. They also take into account that the activities of a primary user may disrupt communication in several groups, of secondary users, which are referred to as Shared Primary User Risk Group (SPURG). The objective of the proposed algorithms is to increase the number of sessions that can be accommodated in the network and minimize the cost of provisioning the sessions while protecting them against failures.**

Keywords*— *multicast; cognitive radio; resilience; protection.

I. INTRODUCTION

With the rapid increase in the demand for wireless networks and their applications, spectrum scarcity has emerged as a major challenge for this kind of networks. Even with spectrum scarcity, measurements have shown that spectrum utilization under fixed spectrum assignment policy varies in time and geographical location, and that variation ranges from 15% and 85% [1]. Hence, introducing efficient ways that utilize the underutilized portions of the spectrum by allowing spectrum sharing are needed for the next generation of wireless networks. The enabling technology for dynamic spectrum access that utilizes spectrum usage is cognitive radio.

The users in cognitive radio networks are classified into two types: primary users (PUs) and secondary users (SUs). A primary user is a licensed user who has a license to access a certain band without competition with other users. On the other hand, a SU is not licensed to use the licensed bands of the spectrum; however, a SU can use the spectrum opportunistically whenever the PU is idle.

Multicasting is an important service mode which is used by many existing and emerging applications. The implementation of multicasting in cognitive radio networks is challenging as several characteristics of cognitive radio networks should be taken into consideration [2]. In traditional wireless networks, all users can transmit on the same set of frequency bands. However, that may not be true in cognitive radio networks since each user has a different set of available channels. Hence, at least one common channel between any two users, which are within communication range, must be available so they can communicate with each other. One challenge in multicast over cognitive mesh networks is the heterogeneity of channel availability among SUs of one multicast group. As a result, multicast time may take longer time because of transmission over multiple channels. Hence, channel diversity between a source node and its one-hop neighbors in cognitive radio networks necessitates finding an effective way to handle one-hop multicast.

In [3], the authors propose an assistance strategy to mitigate channel heterogeneity in cognitive radio wireless mesh networks. An assisted-multicast scheduling in a single cell in wireless cognitive mesh networks is proposed. The authors proposed a solution for this problem in order to minimize the required multicast time over cognitive mesh networks. Resilient multicast routing in cognitive radio networks is proposed in [4] using a multilayer hyper-graph. Due to PUs' activities, it is important to protect the multicast session from the failure when one of the used channels becomes unavailable. The authors proposed a solution to support multicast in cognitive radio networks while protecting the multicast session from failures during transmission. Survivability is provided using reactive protection approach where the traffic is rerouted to preplanned backup path once a failure occurs and affects the multicast session. In [5], a cross-layer optimization is proposed to support video multicasting in infrastructure-based cognitive radio networks. The objectives of this work are to optimize the quality of received video, achieve proportional fairness between multicast users and protect PU from interference by keeping the interference below a certain threshold. A joint channel allocation and multicast routing scheme for a multi-hop cognitive radio network is proposed in [6]. The objective of this work is to maximize the multicast throughput

This research was supported in part by a grant number 11-INF1920-02 from the National Program for Science and Technology, Saudi Arabia.

978-1-4799-0959-9/14/$31.00 ©2014 IEEE

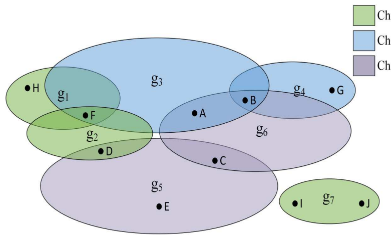

Fig. 1: Secondary users grouping based on locations and common channels.

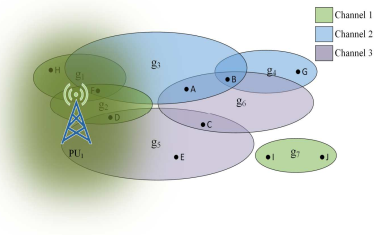

Fig. 2: Secondary users belonging to g_1 and g_2 share the same risk caused by PU_1.

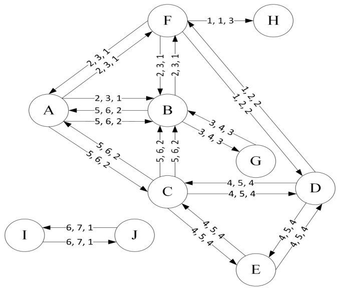

Fig. 3: Converting the network in Fig. 1 to a directed graph.

while taking into the consideration the dynamic change in channels availabilities.

We propose two algorithms that provision a robust multicasting for multiple sessions in cognitive radio networks by protecting all multicast sessions. Our proposed algorithms are: multicast sessions protection with link-sharing and multicast sessions protection using rings. The goals of our work are to provision multiple multicast sessions, protect them against single PU interruption at a time, minimize the cost of multicast sessions and increase the number of sessions that can be accommodated in the network.

The rest of this paper is organized as follows. System model and protection model are described in Sections II and III respectively. In Section IV, we introduce our proposed algorithms for provisioning robust multicast sessions in cognitive radio networks. Then, simulation results of our proposed algorithms are shown and explained in Section V. Finally, we conclude this paper in Section VI.

II. SYSTEM MODEL

We consider a cognitive radio network with a set of SU nodes and a set of available channels in the network. Different SUs may have access to different available channels depending on the channels' conditions at the location of SUs. Moreover, a group of SUs may observe the same channel appear or disappear at the same time, which is dependent on the licensed PUs location and activities.

Definition 1: Secondary Users Group g_i: A group of SUs that can transmit to and/or receive from each other, over a common channel and within one hop. The index, i, is the group number. This group forms a clique in the network graph.

As an example, consider the three channels network shown in Fig. 1, in which 10 SUs are grouped into seven groups (g_1-g_7). Each node belonging to a certain group can transmit to and/or receive from any other node inside this group within one-hop transmission.

Definition 2: Shared Primary User Risk Group (SPURG): A group of one or more SUs groups that operate on the same channel that is licensed to a PU, and share a risk caused by the PU when the PU becomes active. Once this PU starts transmitting over the shared channel, all SUs belonging to this group will be blocked from using the shared channel. Transmissions by SUs in this group cause interference to the receiving PUs that operate on the shared channel.

In Fig.2, g_1 and g_2 operate on channel 1, and they share the same risk once PU_1 becomes active. Transmissions by SUs in g_1 and g_2 (H, F and D) cause interference to PU_1's reception. Hence, g_1 and g_2 are considered within the same SPURG, SPURG 1. Once PU_1 becomes inactive on channel 1, any SUs in g_1 or g_2 will have the ability to access this channel. We can see that PU_1 activities have no effect on other groups, which operate on channels other than channel 1, and SUs belong to these groups will not cause interference to PU_1.

A. Converting the Network to a Directed Graph

Assume that the communication links between A and B and between F and H in Fig. 1 are in one direction, and the rest of the communication links are in two directions. Also, SPURG 1 consists of g1 and g2, where each other SPURG consists of one group. The costs of links in each group are the same and are assigned based on the cost of leasing the channel, and assume that the cost values range from 1 to 5. This cost can also correspond to other metrics such as the delay. Each pair of SUs inside one group are interconnected to each other with one link if the link between them is in one direction, and two links if the link between them is bidirectional. Two sets, V, E, are used to represent the directed graph G(V, E), where V is a set of SUs nodes, and E is a set of links. Set L_i is a set that consists of all links between SUs belonging to g_i. Each link $x = (u, v) \in E$, where u & $v \in V$, is assigned three values, as shown in Fig. 3, representing SPURG, L_i set that the link belongs to, and the cost of the link. Accordingly, the network in Fig. 1 can be converted to a directed graph as shown in Fig. 3.

III. PROTECTION MODEL

Our proposed algorithms protect multiple multicast sessions against one PU interruption at a time. Once a PU becomes active, all SUs belonging to the groups that share the same risk of this PU will be blocked from using the interrupted channel and they need to release the channel to the PU. In other words, all SUs belonging to the same SPURG will be blocked from using their interrupted common channel once the corresponding PU of this SPURG becomes active on that channel. When SUs are blocked from using a common channel, all links between them become unavailable until the PU leaves the channel. PU interruption causes failures to all links in the groups belonging to the SPURG affected by this PU.

Table 1: Notations

Symbol	Meaning
G(V, E)	Network graph, where each link $x = (u, v)$ ∈E, and u, v ∈V .
S_k	Source node for kth session.
d_{ki}	Destination number i for kth session.
M_k	Multicast request for kth session, where M_k =(S_k, $\{d_{k1}, d_{k2}, \ldots, d_{km}\}$), m ≤max number of destinations.
PT_k	kth primary multicast tree.
BT_kx	Backup multicast tree for protecting link x in PT_k.
P	Union of all links used in primary trees.
B	Union of all links used in backup trees and not used in primary trees.
SPURG	Shared Primary User Risk Group.
g_i	Group of all secondary users that can transmit to and/or receive from each other, over a common channel and within one hop.
L_i	Set of all links that interconnect secondary users within one group, g_i.
$Path_k$	A path starts from source node, and traverses all destination nodes in session k.

IV. PROPOSED ALGORITHMS

We propose two algorithms for providing robust multicasting in cognitive radio networks. These algorithms are: 1) multicast sessions protection with link-sharing trees and 2) multicast sessions protection using rings. Multicast session request M_k is the kth multicast session, where 1 ≤ k ≤ n, and n is total number of multicast sessions. Each multicast session M_k is represented by a source node S_k and a set of m destinations (d_{k1}, d_{k2}, …, d_{km}). Given a multicast request M_k = (S_k, (d_{k1}, d_{k2}, …, d_{km})) and the directed graph *G(V, E)* of the cognitive radio networks, each of our proposed algorithms provisions multicast sessions that is protected against one PU interruption at a time. Table 1 describes the symbols used in our proposed algorithms for provisioning robust multiple multicasts in cognitive radio networks.

```
Algorithm 1: Constructing multicast tree
Input: G(V, E), A multicast request $M = (S, \{d_1, d_2, \ldots, d_m\})$ where $S$
       is a source node and $\{d_1, d_2, \ldots, d_m\}$ are destination nodes.
Output: Multicast tree.
1 for $k \leftarrow 1$ to $m$ do
2     Construct shortest path $P_k$ from source node $S$ to destination $d_k$ such
3     that no multiple links along the path belong to same set $L_i$.
4     for each link ($x$,y) along path $P_k$ do
5         If there is a link (y, $x$) $\in G(V, E)$ such that both ($x$,y) and ($y$,x)
6         belong to same set $L_i$, then remove ($y$,x) from G(V, E).
7         for each link ($x$,w) connected to node $x$ do
8             if ($x$,y) and ($x$,w) belong to the same set $L_i$, then set the cost
9             of link ($x$,w) to zero.
```

In our proposed algorithms, *Algorithm 1* is used to construct a multicast tree. Algorithm 1 approximates the optimal minimum Steiner tree in term of cost by using shortest paths tree. Line 1 constructs m paths from source node to all m destinations. In lines 2-3, the established path does not have multiple links belong to the same set L_i, and that because all nodes inside SUs group can be reached within one hop and without establishing multiple links. In lines 4-6, the link in the opposite direction is removed because wireless medium is already reserved for the link on the established path. In lines 7-8, the cost of the link (*x, w*) is set to zero since the SU has already reserved the wireless medium after establishing link (*x, y*). Algorithm 1 uses Dijkstra's algorithm to find the shortest path. Since the complexity of Dijkstra's algorithm is O(E log V), the complexity of Algorithm 1 is O(m E log V + E^2).

A. Protecting Multiple Multicast Sessions with Link-Sharing

We propose an algorithm that generates a primary tree for each multicast session, and protects each link in the primary tree by a backup tree. The protection method is reactive where the backup protection trees for all links in each primary tree are calculated in advance, and the traffic is rerouted to the backup tree once the corresponding link fails.

Multicast sessions protection with link-sharing algorithm allows link sharing inside one session, and also between different sessions. Once a link in a primary tree fails, then each backup tree can share all non-failed links in the primary tree in the same session and use them in constructing the backup tree. Moreover, it is possible that non-failed links in any backup tree in any session is used also in constructing a backup tree for another link. This leads to reducing the cost of multicast sessions and using the resources more efficiently. To protect a given link *x*, then any link in primary or backup trees in the same session or in backup trees in other sessions cannot be used in protecting link x if its SPURG is the same as SPURG of link *x*. The reason for eliminating the use of links with SPURG which is the same as SPURG of the link to be protected is that all of them belong to the same SPURG, and hence, they will fail together once the corresponding PU becomes active.

Multicast sessions protection with link-sharing algorithm is shown in Algorithm 2. Algorithm 2 starts with initializing different sets as shown in line 1. Lines 3-5 are used to construct a primary tree for the kth session and update the graph and set P. The lines from 6 to 25 show the required steps used to construct a backup tree for the link that needs to be protected. In backup tree construction, Algorithm 3 takes into consideration all links in current and previous sessions that can be shared, in addition to remaining links in the original graph.

In line 7, the original graph, primary tree in current session and the set of all backup trees are copied, and the copies will be used in the following steps to avoid sharing the links that should not be shared. The link that needs to be protected is removed from *PT_k'* in order to avoid using it in the backup tree for that link, as shown in line 8. The steps shown in the lines 9-11 are used to remove the links that cannot be shared from the copy of the primary tree. The same procedure is applied for the links belonging to the copy of backup trees set *B'* or *G(E,V)'* as shown in lines 12-17. Lines 18-20 prevent any two links with the same SPURG value, and belonging to different sessions from sharing links in their backup trees.

All remaining links in sets *PT_k'* and *B'* can be shared and set to zero as shown in line 21. Lines 22-23 shows that the remaining links in *G(E,V)'*, *PT_k'* and *B'* can be used to construct *BT_k x* by running Algorithm 1 on all links in the union (*G(E,V)'* ∪ *PT_k'* ∪ *B'*). All links used in *BT_kx* are reserved by removing *BT_k x*'s links from *G(E,V)* as shown in line 24. In line 25, set *B* is updated by including all links used in backup tree *BT_k x* but not used in primary tree *PT_k'.* The reason for that is to prevent any backup tree from using a link belonging to a primary tree in another session. All links within one L_i set can be used by only one session since the wireless medium between the nodes in g_i will be entirely reserved for multicast session k's transmission. Therefore, all remaining links in any set L_i, such that session k used a link or more from L_i, will be removed, as shown in lines 26-28. Algorithm 2 uses Algorithm 1 to construct the multicast trees, hence, the complexity of Algorithm 2 is O(n^2 E^2 + nmE^2 log V + nE^3).

```
Algorithm 2: Multicast sessions protection with link-sharing
Input: G(V, E), n multicast request $M_k = (S_k, \{d_{k1}, d_{k2}, \ldots, d_{km}\})$
       where (1≤ $k$ ≤n), source node $S_k$ and destination nodes
       $\{d_{k1}, d_{k2}, \ldots, d_{km}\}$ id numbers for n sessions.
Output: Primary $PT_k$ and protection $BT_k x$ trees for n multicast sessions
1  Set $G(V,E)^{'} = \phi$, $P = \phi$, $B = \phi$, $B^{'} = \phi$ and $PT_k{}^{'} = \phi$, (1≤ $k$ ≤n).
2  for $k \leftarrow 1$ to $n$ do
3      Construct k$^{th}$ primary tree $PT_k$ by running Algorithm 1 on $G(V,E)$.
4      Update G(V, E) by removing all links used in $PT_k$.
5      Set $P = P \cup PT_k$.
6      for each link $x \in PT_k$ do
7          Set $G(V,E)^{'}$ = G(V, E), $PT_k{}^{'} = PT_k$ and $B^{'} = B$
8          Remove $x$ from $PT_k{}^{'}$
9          for each link $y \in PT_k{}^{'}$ do
10             if SPURG of $x$ is same as SPURG of $y$, then remove $y$ from
11             $PT_k{}^{'}$.
12         for each link $u \in B^{'}$ do
13             if SPURG of $x$ is same as SPURG of $u$, then remove $u$ from
14             $B^{'}$.
15         for each link $q \in G(V,E)^{'}$ do
16             if SPURG of x is same as SPURG of $q$, then remove $q$ from
17             $G(V,E)^{'}$.
18         for each primary tree $PT_j \in$(P - $PT_k$), $j \in$[1, k-1] do
19             if $\exists$ link $w \in PT_j$ s.t SPURG of $w$ is same as SPURG of x,
20             then Remove all links in $B^{'}$ used to protect link $w$.
21         Set the cost of all remaining links in $PT_k{}^{'}$ and $B^{'}$ to zero.
22         Construct back up tree $BT_k x$ to protect link x in k$^{th}$ session by
23         running Algorithm 1 on $(G(V,E)^{'} \cup PT_k{}^{'} \cup B^{'})$.
24         Remove all links used in $BT_k x$ from G(V, E).
25         Set $B = B \cup BT_k x \setminus (BT_k x \cap PT_k)$.
26     Update G(V, E) by removing all links belonging to $L_i$, such that the
27     primary or protection trees used a link or more from $L_i$, for all
28     possible number i.
```

B. *Protecting Multiple Multicast Sessions using Rings*

Provisioning and protecting a multicast session can be done using only one ring. It was proposed to use rings to protect multicast sessions in optical networking [7]. However, the nature of wireless networking and cognitive radio networks in particular necessitates the use of rings in different ways. Protecting a multicast session using a ring requires that each consecutive node along the ring can send to and receive from each other. PU interruption causes failures to all links along the ring with SPURG affected by this PU. Therefore, it is required to provision a ring that is able to reach all destinations even with presence of one PU interruption at a time.

We propose an algorithm that supports multiple multicast sessions over cognitive radio networks using rings as shown in Algorithm 3. This algorithm is proactive where each destination SU receives at least one copy of the multicast message even if the PU interrupts the SUs. Therefore, this proposed algorithm has the advantage of protecting the multicast tree immediately when a failure happens and without using backup trees. Each multicast session is generated and protected using a ring. This ring is established in such a way that it starts with the source node, traverses all destinations and ends at the source node to create a ring. Under normal network operation, source node S forwards it message to both nodes connected to it, then, each node that receives a message forwards it to the next node connected to it along the ring until one node receives two copies of the message. If a node receives two copies of a message from two neighboring nodes along the ring, then it will stop forwarding the message since it can conclude that all other destinations have received the message.

Suppose that PU_1 starts transmitting, then all links with a SPURG corresponding to PU_1 will fail as shown in Fig. 4. Although both links between S and d1 fail when PU_1 becomes active, node d1 is still reachable by the source node using the path S-n3-d2-n2-d1. As a result of that, cognitive radio networks in this case can recover from the failure without rerouting the traffic to another backup tree.

Line 1 Algorithm 3 initializes graph *G(E,V)'* and *$Path_k$'* , which will be used to temporarily modify the original graph and *$Path_k$* in the following steps. In line 2, a loop starts constructing multicast sessions using ring structure. In lines 3-5, the original directed graph is copied to graph *G(E,V)'*, source node is set to the S_k and destination nodes list D is created. Then, shortest path from source node to closest destination in list D is created using Algorithm 1, as shown in lines 6-8. Lines 9-11 are used to make each path ,from a source node to a destination or from a destination to a destination, be SPURG disjoint. After establishing the path from source node to the first destination, this destination will be set as a source node, as shown in line 12. In line 13 and 14, all steps from line 6 to line12 will be repeated until all destinations are reached. Once all destinations are reached, a path starting from S_k and traversing all destination nodes is created, which is called *$Path_k$*.

In lines 15-16, another path is established from S_k to the last reached destination in order to create a ring. If the ring is always created by connecting the last reached destination node in *$Path_k$* to the source node, then this may lead to block the session if it is not possible to find a path connecting these two nodes. Even if there are enough resources to establish a path from last destination in *$Path_k$* to source node, it is possible to find another ring that has a lower cost, and does not directly connect the last destination to the source node. Hence, the procedures in line 17-26 are used to find a ring that minimizes the cost while considering closing the ring from a node in *$Path_k$* other than the last reached destination. Fig. 5 shows two different ways, which may lead to different costs, to create a ring for protecting the multicast session.

In line 27, the ring with the lowest cost is selected to generate and protect the multicast session. Finally, the original graph is updated in lines 28-30 as described in lines 26-28 of Algorithm 2, and also by removing all used links in the selected ring. Algorithm 4 uses Algorithm 1 to construct all paths between source and destination nodes, and between any two nodes along the ring. Since Algorithm 1 is used with m=1 in this case, the complexity of Algorithm 1 is O(E log V + E^2).

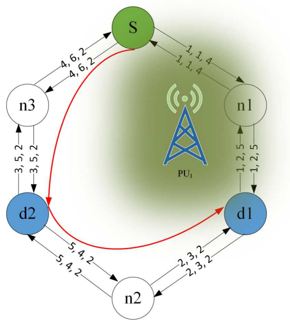

Fig. 4: Multicast session protection in case of primary user appearance.

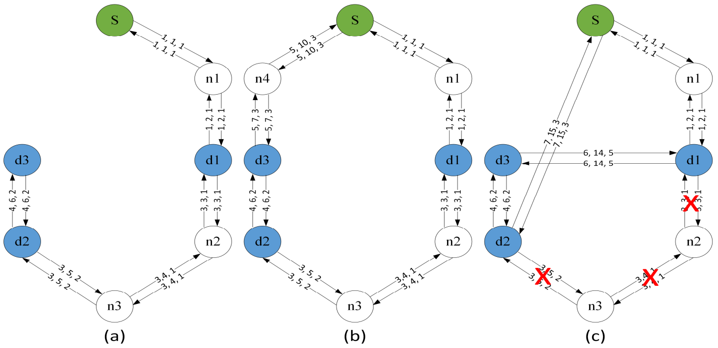

Fig. 5: Creating a ring for multicast protection. In (a): a path starts from source node and traverses all destination nodes, (b): creating a ring by connecting last destination to the source node, and (c): creating a ring, which has a lower cost, without connecting the last destination to the source node.

Accordingly, total time complexity for Algorithm 4 is O(nmE log V + nmE^2). Since in a connected graph, E ≥ V-1, and log V < V, then the complexity is O(nmE^2).

Algorithm 3: Multicast sessions protection using rings

```
Input: G(V, E), n multicast requests $M_k = (S_k, \{d_{k1}, d_{k2}, \ldots, d_{km}\})$
       where (1≤ $k$ ≤n), source node $S_k$ and destination nodes
       $\{d_{k1}, d_{k2}, \ldots, d_{km}\}$ id numbers for n sessions.
Output: n multicast rings for n multicast sessions.
1  Set $G(V,E)' = \phi$, $Path_k' = \phi$.
2  for $k \leftarrow 1$ to $n$ do
3      Set $G(V,E)'$ = G(V, E), source node = $S_k$.
4      Create a destination list D consists of all destination nodes
5      $\{d_{k1}, d_{k2}, \ldots, d_{km}\}$.
6      Find the shortest path in $G(V,E)'$ from source node to closest
7      unvisited destination node, $d_{kx}$ ∈D, using Algorithm 1, where
8      $M = (S_k, \{d_{kx}\})$.
9      for each link $e \in G(V,E)'$ do
10         if SPURG of any link along the path from $S_k$ to $d_{kx}$ is same as
11         SPURG of $e$, then remove $e$.
12     Set source node = $d_{kx}$.
13     Repeat the steps from line 6 to 12 until all destinations in list D are
14     reachable, and $Path_k$ is established.
15     Find the shortest path in $G(V,E)'$ from $S_k$ to last node in $Path_k$
16     to create a ring.
17     if $m \geq 3$ then
18         for $j \leftarrow 1$ to $m-2$ do
19             Set $Path_k' = Path_k$.
20             Remove all links along the path from $d_{kj}$ to $d_{k(j+1)}$ from
21             $Path_k'$.
22             Find the shortest path in $G(V,E)'$ from last node in
23             $Path_k'$ to node $d_{k_j}$ in $Path_k'$.
24             Find the shortest path in $G(V,E)'$ from node $d_{k(j+1)}$ in
25             $Path_k'$ to node $S_k$.
26             Create j$^{th}$ ring.
27     Selcet one ring, which has the lowest cost.
28     Update G(V, E) by: removing all links used in the selected ring, and
29     removing all links belong to $L_i$, such that the ring used a link or more
30     from $L_i$, for all possible number i.
```

I. SIMULATION RESULTS

A. *Single Multicast Session*

In this section, we compare the number of paths that can be established to reach destination nodes using optimal solution in [4] with multicast sessions protection with link-sharing algorithm. These paths include the paths in the primary tree and the average number of paths in the backup trees for single multicast session. Also, we compare the number of links in the primary and backup tree used to establish the multicast session. In performance evaluation, we use the following parameters: 25 SUs, 10 SUs groups per each channel, one SUs groups, g_i, per each SPURG, one source node and 8 destination nodes. SUs with a common channel are assigned randomly to a SUs group, g_i. Each link in the network is assigned a cost which ranges from 1 to 5, where all links belonging to the same set L_i are assigned same cost. Then, random sessions with 8 destinations are generated over 100 random graphs.

Fig. 6 shows a comparison between the numbers of generated paths with respect to the number of available channels using the optimal solution in [4] and multicast session protection with link-sharing algorithm. The number of established paths increases as the number of available channels increases. It is shown that the number of generated paths using our proposed algorithm is close to the optimal solution when the number of available channels is small. As the number of available channels is increased to 7, the difference between the optimal solution and our proposed algorithm is approximately 2 paths.

The number of links used in single multicast session increases as the number of available channels increases as shown in Fig. 7. The number of used links in our proposed algorithm is lower than the number of links in the optimal solution in [4] since the number of generated paths in our proposed algorithm is lower. Also, link-sharing algorithm generates backup trees that share some links, and this leads to reducing the number of used links in the multicast session.

B. *Multiple Multicast Sessions*

We compare the performance of our proposed algorithms using two metrics: the number of sessions that can be accommodated in the network and the average cost of the multicast sessions. We modify some network parameters used in the simulation as follows. The network consists of 50 SUs and up to 36

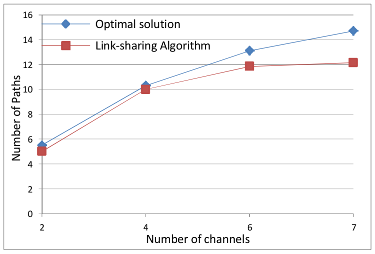

Fig. 6: Number of paths with respect to the number of channels.

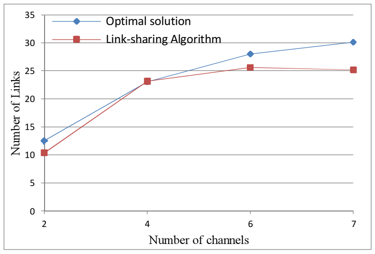

Fig. 7: Number of links with respect to the number of channels.

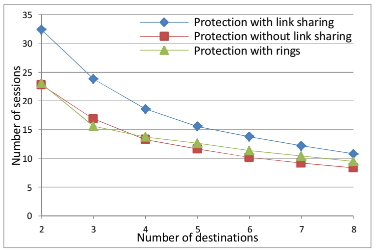

Fig.8: Number of multicast sessions in the network with respect to session size.

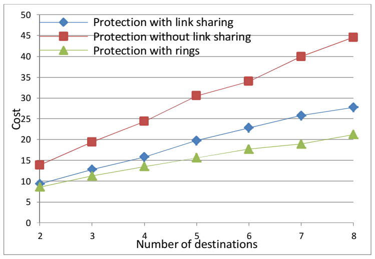

Fig. 9: The cost of the multicast session with respect to its size.

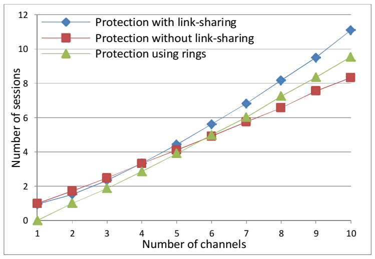

Fig. 10: Number of multicast sessions in the network with respect to the number of available channels.

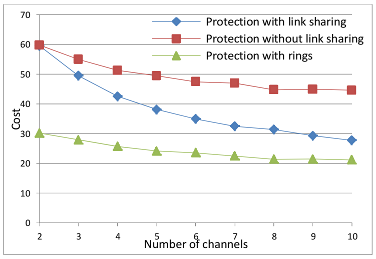

Fig. 11: The cost of the multicast session with respect to the number of available channels.

SUs groups. Each SUs group, g_i, consists of up to 5 SUs, and each SPURG consists of up to three SUs groups.

We compare the number of multicast sessions that can be accommodated in the network for our proposed algorithms as shown in Fig. 8, when the number of available channels is 10. The number of sessions decreases as the number of destinations increases for all three algorithms. It is shown that multicast sessions protection using link-sharing algorithm achieves the highest number of sessions. Fig. 9 shows the average cost per session with respect to session size. As the number of destinations increases, the cost per session increases since a higher number of links is generally required. Multicast sessions protection using rings achieves the lowest cost, and the reason is that it does not require protecting every link by a protection tree compared with the other two approaches.

We study the effect of increasing the number of channels to the performance of the algorithms when the number of destination is 8. It is shown in Fig. 10 that increasing the number of available channels for SUs increases the number of sessions that can be accommodated in the network. Therefore, a higher number of protected sessions can be accommodated in the network when the number of available channels increases.

Multicast sessions protection with link-sharing algorithm achieves the highest number of sessions that can be accommodated in the network. It is shown in Fig. 11 that the multicast session cost decreases as the number of available channels increases. Multicast session protection using rings generates protected sessions with the lowest cost compared to the other two approaches. Sharing some links within one session or between different sessions allows multicast session protection with link-sharing to outperform multicast session protection without link-sharing.

II. CONCLUSION

In this paper, we propose two algorithms that provision and protect multiple multicast sessions in cognitive radio networks. The goals of our proposed algorithms are: provisioning multiple multicast sessions in cognitive radio networks, protecting all multicast sessions against one active PU at a time, increasing the number of sessions that can be accommodated in the network and minimizing the cost of the multicast sessions. We show that the number of accommodated sessions in the network increases and the cost of multicast sessions decreases when the number of available channels increases or the session size becomes smaller. Multicast sessions protection with link-sharing can support the highest number of multicast sessions in the network compared with the other algorithm. However, multicast sessions protection using ring can generate multicast sessions with the lowest cost compared with multicast sessions protection with link-sharing.

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,” Comput. Netw., vol. 50, no.13, pp. 2127–2159, May 2006.

[2] Y. Shi Gao, Y. T. Hou, H. D. Sherali, and H. Zhou, ”Multicast Communications in Multi-Hop Cognitive Radio Networks,” IEEE Journal on Selected Areas in Communications, vol.29, no.4, pp.784-793, Apr. 2011.

[3] H. M. Almasaeid and A. E. Kamal, “Exploiting multichannel diversity for cooperative multicast in cognitive radio mesh networks,” IEEE/ACM Transactions on Networking, to appear, 2014.

[4] S. H. Alnabelsi, A. E. Kamal, " Resilient Multicast Routing in CRNs Using a Multilayer Hypergraph Approach ", IEEE International Conference on Communications (ICC), 2013, pp. 2910-2915.

[5] D. Hu , S. Mao , Y. Hou and J. Reed "Scalable video multicast in cognitive radio networks", IEEE J. Select. Areas Commun., vol. 28, no. 3, pp.334 -344, 2010.

[6] Zhihui Shu, Yi Qian, Yaoqing Yang and H. Sharif. “Channel Allocation and Multicast Routing in Cognitive Radio Networks,” Wireless Communications and Networking Conference (WCNC), pp. 1703-1708, 2013.

[7] N. Singhal and B. Mukherjee, “Protecting multicast sessions in WDM optical mesh networks,” J. Lightwave Technol., vol. 21, no. 4, pp. 884–892, 2003.