

QoS Driven Dynamic Channel Assignment in Multi-Channel Multi-Radio Wireless Mesh Networks

(February 2010)

Bahador Bakhshi

Computer Engineering Department
Amirkabir University of Technology
Tehran, Iran
bbakhshi@aut.ac.ir

Antonio Capone

Dipartimento di Elettronica e Informazione
Politecnico di Milano
Milan, Italy
capone@elet.polimi.it

Siavash Khorsandi

Computer Engineering Department
Amirkabir University of Technology
Tehran, Iran
khorsand@aut.ac.ir

Abstract— Channel assignment in multi-channel multi-radio wireless mesh networks is a powerful management tool to exploit available resources efficiently. We study the problem of dynamic channel assignment for network resource adaptation, in presence of traffic with QoS constraints, to improve network performance in terms of demands acceptance rate. We propose an on-line dynamic algorithm for the problem. To the best of our knowledge, this is the first time that an on-line dynamic algorithm channel is proposed for the problem. We assume there is no information about future demands and path of each demand is given. The algorithm reassigns channels only when a demand cannot be accepted using current channel assignment. The new channel assignment is selected according to our proposed metric that aims at maximizing resource availability for upcoming demands. The number of channel reassignments required for accepting a demand is kept small considering the links in a vicinity of the routing path; consequently, the strategy is localized and does not affect whole network channel assignment. Comparison with others algorithms, including the optimal static channel assignment, shows that the proposed algorithm outperforms existing solutions and can efficiently use available channels even with a limited number of radios per node.

Keywords- *Wireless Mesh Networks; Dynamic Channel Assignment; Quality of Service (QoS); Online Algorithm*

I. INTRODUCTION

Multimedia services as an integral ingredient of broadband wireless mesh networks (WMN) require intensive resources and quality of service (QoS) support. Besides guaranteeing the end-to-end QoS requirements, network resources should be managed efficiently to improve network performance that is measured as the number of admitted traffic demands. Capacity of WMNs is shrunk by the interferences that arise from the shared nature of wireless media. Multi-channel multi-radio networking is a promising approach to reduce the interference and improve network capacity.

Channel assignment is the key issue that needs to be addressed in multi-channel multi-radio WMNs and that can be used as a resource management tool. Since, the expected available bandwidth on each link is determined by the level of interference, which depends on the assigned channel, in the

presence of traffic with QoS constraints, *the main problem is efficient channel assignment to increase acceptance rate of the traffic demands.*

There are two broad categories of channel assignment strategies: static and dynamic. In the former strategy, channel is assigned for a long period of time while in the latter, channels can be changed frequently over time according to needs. Static methods are oblivious to dynamics of network traffic. Consequently, they provide suboptimal network performance. On the other hand, the dynamic approaches aim to achieve a near optimal performance by adapting network resource according to the traffic patterns¹. The existing dynamic channel assignment methods attempt to optimize different metrics as measure of network performance. Some mechanisms consider elastic flows and try to maximize aggregate throughput of a given set of flows [1], [2], [3]. The mechanisms rerun channel assignment every time that traffic pattern changes. Another group aims at maximizing one-hop capacity of the network. These mechanisms find overloaded links periodically and update channel of the links [4], [5], [6]; but they cannot guarantee end-to-end throughput. None of the existent dynamic channel assignments considers end-to-end QoS requirements and tries to maximize admitted demands.

In this paper, we propose an on-line dynamic channel assignment for the problem. It is assumed that each QoS demand arrives at a particular time and requires a specific bandwidth through a given path. The demand is accepted iff the network can provide sufficient resource through the path to allocate the required bandwidth; otherwise it is rejected. The primary goal is reassigning channels in the network according to the QoS requirements of arriving demands in order to maximize acceptance rate. We assume channel assignment is part of the network management tool and therefore we consider a centralized approach based on a call admission control (CAC) server. The server controls admission of QoS demands and runs the channel assignment algorithm. We assume the server has a fairly accurate and complete view of the network. Our contributions to the problem are as follows.

This work was done while Bahador Bakhshi was a visiting researcher at Politecnico di Milano and was supported through funds provided by Iran Telecommunication Research Center (ITRC) and the Italian PRIN project SESAME.

¹ Fast switching is a special case of the dynamic approaches in which channels are changed per-packet. The method needs particular MAC protocol and is not considered in this paper.

First, we formulate this problem and identify the requirements of an efficient algorithm for it in Section 3. We explain that besides maximizing the number of accepted demands, the number of channel reassignment should be minimized. Moreover, accepting a demand must not affect resource availability of other links far away from the routing path of the demand.

Second, we design an on-line on-demand channel assignment algorithm based on the proposed design choices in Section 4. We describe our design choices for *reassignment strategy*, *channel selection metric*, *on-demand radio utilization*. We clarify how the choices improve efficiency of the algorithm by satisfying the aforementioned requirements. We integrate these choices in the QoS Driven Dynamic Channel Assignment (QDDCA) algorithm to dynamically manage channels in the network.

Finally, we compare the proposed algorithm in terms of acceptance rate of QoS demands with other both static and dynamic algorithms including the optimal static channel assignment. We also study the effect of various parameters including arrival rate, the number of available channel, and the number of radio per node on performance of the algorithms in Section 5.

The remaining of this paper is organized as following. The related work is reviewed in Section 2. Assumptions, used models, and problem formulation are presented in Section 3. In Section 4, after discussing about the proposed design choices, we present the QDDCA algorithm and finally analyze the computational complexity of the algorithm. Simulation results are presented in Section 5 and Section 6 concludes this paper.

II. RELATED WORK

The previous works on dynamic channel assignment in multi-channel multi-radio WMNs can almost be divided into three categories [7]: (i) approaches designed to mitigate external interference [8], [9], [10], (ii) methods that reruns channel assignment periodically [1], [2], [3], and (iii) solutions that attempt to reassign channels according to local load measurements [4], [5], [6].

In the first category, it is assumed that there is an external source of interference, e.g. coexisting network, and there is not any cooperation between the WMN and the external source. In these approaches, nodes in the network measure the interference periodically and switch to the least interfered channel if level of the interference in current channel exceeds a threshold. This approach is used in [8] to find the least interfered default channel. Similarly, authors in [9] used this approach to find a good channel as the default channel in each cluster. A game theory based solution was proposed in [10] to minimize external interference in the network. Although minimizing the external interference improves network performance, this category does not explicitly take into consideration the network traffic and its dynamics.

The previous works in the second category are load-aware channel assignment; wherein, channel is assigned to a link

according to the load of the link. Raniwala et al in their pioneer work [1] proposed a centralized joint channel assignment and routing algorithm. The channel assignment algorithm considers high load link first. The joint channel assignment and routing algorithm proceeds in an iterative fashion. In [2], Rad et al proposed a joint channel assignment and congestion control scheme. They assumed the path of the flows is given and developed an ILP model to maximize aggregate throughput. They employed exhaustive search for the channel assignment problem, yet it is applicable only for small networks. Authors in [3] also studied congestion control and channel assignment problem. An iterative approach was proposed. In each iteration, the congestion control problem is solved and the solution result is feed into channel assignment algorithm to select a new assignment. The major drawbacks of these schemes are the requirement of global information and the possibility of many channels reassignments in every iteration. In these approaches, the channel assignment algorithm reruns every time the traffic pattern changes. It needs global network information and may change all already assigned channels that leads to a significant overhead for updating entire network.

Schemes proposed in the third category are localized reassignment approaches in which each node periodically checks load of its links. In the case of detecting an overloaded link, channels are reassigned. In algorithm proposed in [4], each node estimates the usage status of all channels within its neighborhood. When it finds a channel with a lower usage, it changes to that channel. In [4], a tree topology was proposed and each node can change channel of the links between itself and its children. In proposed solution in [5], each node has two types of interfaces: Receive and Transmit. The channel of the Transmit interface is changed per-packet but the channel of the Receive interface is changed according to the local measured load. The most closely related work to this paper is [6]. Authors considered a general topology and assumed routing path of each flow is given. Each node measures load of its interfaces and if the load exceeds a predefined threshold it select the best channel for the link. The best channel is specified by the Channel Cost Metric (CCM). The authors extended their work in [11] and integrated power control with the channel assignment, which is beyond the scope of this paper. Although these schemes have not the drawbacks of the second category, they do not consider the end-to-end throughput explicitly. These schemes attempt to improve the one-hop capacity of the network but cannot guarantee the end-to-end requirement of flows, which is the main constraint in supporting quality of service.

III. SYSTEM MODEL AND PROBLEM STATEMENT

In this section, first, we describe the assumptions and used models. Then, the problem considered in this paper is formulated. Notations used through the paper are denoted in table I.

A. Assumptions

We consider 802.11 based multi-channel multi-radio wireless mesh networks. In the network, all nodes are static,

TABLE I. NOTATIONS

V	Set of nodes, $ V = n$
E	Set of edges, $ E = m$
(u, v)	Link $(u, v) \in E$
I_{uv}	Interference set of link (u, v)
c_{uv}	Physical channel capacity of link (u, v)
x_{uv}	Flow on link (u, v)
TR	Transmission range
IR	Interference range, $IR = TR \times q$ and $q > 1$
κ	The number of orthogonal channels in the network
(s, d, x, p, t_a, t_h)	QoS demand from s to d through path p , needs bandwidth x , arrives at t_a and hold-time is t_h
Ψ	Channel assignment

have multiple radios and all radios have a same transmission range TR and a same interference range IR . The RTS/CTS mechanism is enabled. There are κ orthogonal channels in the network. Each link can transmit only in one channel at each given time.

B. Network Model

Network is modeled by a digraph $G = (V, E, \Psi, C)$, where the V is a set of n vertices and E is a set of m edges. $\forall u \in V$ corresponds to a node in the network. Suppose $d(u, v)$ is the Euclidean distance between u and v . For a given pair of nodes u and v , there is a $(u, v) \in E$ iff $d(u, v) \leq TR$ and both ends of this link, nodes u and v , have at least a radio tuned to a common channel. Ψ is the current channel assignment. $C = [c_{uv}]_{1 \times m}$ where c_{uv} is the physical capacity of (u, v) under channel assignment Ψ . Detailed measurements and analysis of WMNs in [12] showed that PHY layer is stable and predictable; and, the link abstraction is a valid assumption. Therefore, we use the abstract model and assume that the physical channel capacity does not vary over time similar to the other previous works [2], [3], [13].

C. Interference Model

We use the *protocol model* for interference [14]. Using this model in conjunction with assuming RTS/CTS sequence yields that links (a, b) and (u, v) interfere with each other if a same channel is assigned to the links and if sender or receiver of one of them is in interference range of sender or receiver of the other; more specifically, $d(u, a) \leq IR$ or $d(b, u) \leq IR$ or $d(a, v) \leq IR$ or $d(b, v) \leq IR$ [13], [15]. I_{uv} is a set of links that interfere with (u, v) . Note that by definition (i) $(u, v) \in I_{uv}$, (ii) $(u, v) \in I_{ab}$ iff $(a, b) \in I_{uv}$, and (iii) I_{uv} corresponds to neighbors of (u, v) in the link interference/contention graph.

D. Available Bandwidth Model

Here, we use the *row constraint* to compute available bandwidth [16]. The constraint is a sufficient condition for feasibility of bandwidth allocation. Let x_{uv} is the flow on link (u, v) . The row constraint for feasibility of bandwidth

allocation imposes $\sum_{(a,b) \in I_{uv}} x_{ab} \leq c_{uv} \quad \forall (u, v) \in E$. Based on the constraint, we define available bandwidth of a link as following.

Definition 1. Available Link Bandwidth of (u, v) is

$$ALB(u, v) = c_{uv} - \sum_{(a,b) \in I_{uv}} x_{ab}.$$

To satisfy QoS requirement of the flows in the network we need $ALB(u, v) > 0 \quad \forall (u, v) \in E$.

E. Problem Statement

The problem studied in this paper is to maximize network performance, which is measured in term of number of accepted QoS demands, using an efficient channel assignment. In the problem, there is a set of QoS demands $\{(s, d, x, p, t_a, t_h)\}$. Demand (s, d, x, p, t_a, t_h) arrives at time t_a , it requires bandwidth x from node s to node d through the path p and if it is accepted, it will leave the network after t_h time. A demand is accepted iff allocating the required bandwidth through the given path does not violate capacity constraints of any link; in the other words, $ALB'(u, v) > 0 \quad \forall (u, v) \in E$ where $ALB'(u, v)$ is the available bandwidth of (u, v) after allocating bandwidth x through path p . In this case, path p is named as a *feasible path*. In the problem, it is assumed that there is not any information about a demand before its arrival time and we are not allowed to reroute the admitted demands. The primary goal is that (re)assign channels such that the number of the accepted demands is maximized.

Besides the primary goal, the channel (re)assignment strategy should satisfy two other requirements. First, the number of channel reassignments should be minimized. This is necessary to reduce the overhead of the algorithm and amount of the signaling traffic used to update channel assignment in the network. Second, the solution should also be a localized in two senses: information and impact. The *information locality* implies that only the local information is used for channel reassignment; changing channel of a link to a new one should be carried out according to the information of the other links in a vicinity, e.g. IR or $2IR$ range, of the link. *Impact locality* means that reassignment of channel of a link must not propagate in whole network and should not influence resource availability of other links far away from the link. Note that satisfying the information locality does not necessarily guarantee the impact locality because changing channel of a link may triggers many other reassignment in the network due to the channel dependency and limited number of radios, which is know as the ripple effect [4].

IV. PROPOSED SOLUTION

In this section, we first clarify our design choices in the proposed solution. Then, we explain how the choices help us to achieve the goals mentioned in the previous section, and finally we present QDDCA algorithm and computational complexity analysis of it.

A. Design Choices

1) On-demand Channel Reassignment

We use on-demand strategy for channel reassignment. We greedily accept given demands and reassign channels if a demand cannot be accepted under current channel assignment. When demand (s, d, x, p, t_a, t_b) cannot be accepted under current channel assignment, it means that the path p is not a feasible path. In the other words, allocating the required bandwidth x through the path p violates capacity constraint of at least one link, $\exists (u, v) \in E$ st. $ALB^1(u, v) < 0$. The link that its capacity constraint is violated is named as *violated link*, which is the key concept in our proposed solution.

The main body of the on-demand algorithm is as follows. For a given demand, we check feasibility of the path. If the path is feasible, the demand is accepted; otherwise, we find the violated links and change their channel to resolve the violation. If it is not possible to resolve the violations, we have to reject the demand. The new channel for a violated link is the best feasible channel. Satisfying feasibility and finding the best channel are explained in the following.

Note that the violated links are not necessarily in the path of routing; even, it is possible that none of the links in the path is violated while there are some other violated links in the network. To illustrate this issue, consider Fig. 1. Assume that all links are assigned to a same channel with capacity 100. In this figure, interference range of nodes b and g are shown by dashed circles. Therefore, $I_{de} = \{ (a, b), (b, c), (d, e), (f, g) \}$. Two flows, one from d to e and another from f to g , are already admitted and now there is a new demand from a to c . If the required bandwidth 20 is allocated on links (a, b) and (b, c) the out-of-path link (d, e) is violated whereas the in-path links (a, b) and (b, c) are not. Because, capacity constraint of links (a, b) and (b, c) are satisfied, $x_{ab} + x_{bc} + x_{de} < 100$, but the constraint of (d, e) is not, $x_{ab} + x_{bc} + x_{de} + x_{fg} > 100$.

2) Feasibility Satisfaction

To admit a demand, the given path should be feasible. As mentioned, if the path is not feasible under current channel assignment, we reassign channels to make the path feasible. Selecting a new channel for a violated link (u, v) is subject to two types of constrains: radio constraint in nodes u and v and capacity constraint.

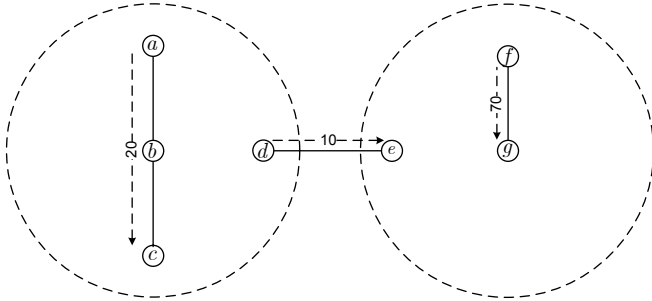


Figure 1. Topology for illustrating out-of-path violated link. Interference ranges and flows are shown by dashed circles and dashed arrows respectively. A same channel is assigned to all links. New flow from a to c violates capacity constraint of (d, e) .

The radio constraint in a node u is satisfied if at least one of the following conditions holds: (i) one of the radios of u is already tuned to the new channel, or (ii) there is a free radio in node u , or (iii) the old channel assigned to link (u, v) can be replaced by the new channel. Note that to avoid the ripple effect [4] the last condition holds if (u, v) is the only link that uses the old channel in node u .

The capacity constraint imposes that switching to the new channel should not violate capacity constraint of the other links already assigned to the channel. According to these constraints, we define two types of channels as follows.

Definition 2. Candidate channel for (u, v) is the channel that satisfies the radio constraint in nodes u and v .

Definition 3. Valid channel for (u, v) is the *candidate channel* that satisfies the capacity constraint.

3) Best Channel Selection

Resource availability is the main parameter that affects admission of demands. Selecting a new *valid channel* for a violated link should provide additional resources for the upcoming demands. We use $ALB(u, v) / |I_{uv}|$ to measure the resource of the network that is defined as following.

Definition 4. Resource of the network under channel assignment Ψ is $R(G, \Psi) = \sum_{(u, v) \in E} (ALB(u, v) / |I_{uv}|)$.

Using definition 4, the best channel is the one that maximize $R(G, \bar{\Psi})$ where $\bar{\Psi}$ is the new channel assignment. To find the best channel, it is not necessary to recomputed $R(G, \bar{\Psi})$ because changing channel (u, v) only affects the links in the interference range of the link. Let I_{uv} and \bar{I}_{uv} are interference sets of (u, v) under channel assignment Ψ and $\bar{\Psi}$; and $ALB(u, v)$ and $\bar{ALB}(u, v)$ are the available bandwidth of the link under channel assignment Ψ and $\bar{\Psi}$ respectively. It is easy to show that

$$R(G, \bar{\Psi}) = R(G, \Psi) + \sum_{(a, b) \in I_{uv} \cup \bar{I}_{uv}} \frac{\bar{ALB}(a, b)}{|\bar{I}_{ab}|} - \sum_{(a, b) \in I_{uv} \cup \bar{I}_{uv}} \frac{ALB(a, b)}{|I_{ab}|}. \quad (1)$$

In (1), the second term is the aggregate resource of the links in I_{uv} and \bar{I}_{uv} after changing channel of link (u, v) and the third term is the aggregate resource of the links before changing the channel. Equation (1) implies that we need to compute the difference between these two aggregate resources. The best channel is the one that provides the maximum improvement. Note that this computation only needs the information of the links in I_{uv} and \bar{I}_{uv} which is local.

4) Group Channel Change

The aforementioned explained procedure to resolve violations focuses on the violated link and attempts to find the best valid channel for the link. There are situations that there is not any valid channel for the violated link, but changing channel of other links in the interference set of the violated link resolves the violation. An example is shown in Fig. 2. Assume

that there are only 2 available channels in frequency spectrum and physical channel capacity is 100. In this figure, interference ranges of nodes c , d , and f are shown by dashed circles, flows and their required bandwidth are shown by dashed arrows, and links and the assigned channel are shown by solid lines. There are four already admitted flows in the network: (i) from a to b , (ii) from e to f , (iii) from g to h , and (iv) from k to l . Allocating required bandwidth 30 for the new demand from c to d violates capacity constraint of link (c, d) ; $x_{cd} + x_{ef} + x_{gh} + x_{kl} > 100$. In this case, there is not any valid channel for the violated link because both channels 1 and 2 are already overloaded in interference range of (c, d) . But, if we change channel of links (e, f) and (g, h) to channel 1, it resolves the violation.

This strategy has a side effect, channel reassignments to resolve violation of a link may affect available bandwidth of other links beyond the interference range of the link; e.g. in Fig. 2, resolving violation of (c, d) affects $ALB(i, j)$. To control the side effect and maintain the *impact locality*, we propose a group channel change procedure that limits the channel reassignments for demand (s, d, x, p, t_a, t_h) in $2IR$ range of path p . The procedure is as follows. Suppose there is not any valid channel for the violated link. We distinguish between the in-path violated links and the out-of-path ones. If the violated link (u, v) is out-of-path, that means it is in IR range of path p , we change channel of links $(a, b) \in I_{uv}$, which are in $2IR$ range of path, one-by-one. Changing channel of (a, b) reduces the number of interfering links with (u, v) so increases $ALB(u, v)$. We change the channels until the violation is resolved or there is not any channel change possibility. When the violated link (u, v) is in-path, we have more opportunities. We can move violation from (u, v) to other links $(a, b) \in I_{uv}$ and try to resolve the new violations as follows. For each *candidate channel* of link (u, v) , we assign the channel to the link. Since the channel is not a valid channel, this channel change creates a new set of violated links in IR range of (u, v) . Now, we attempt to resolve the new violated links as the original violated links.

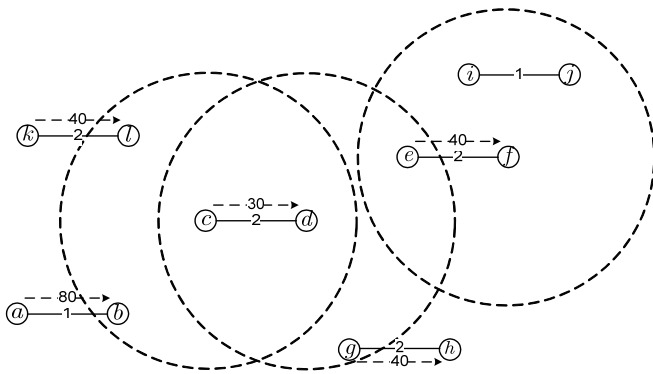


Figure 2. Topology for illustrating group channel change. Interference ranges and flows are shown by dashed circles and dashed arrows respectively. Links and assigned channels are shown by solid lines. Changing channel of (e, f) and (g, h) to channel 1 resolves violated link (c, d) .

Two points must be noted. First, some new violated links can be in the path p and there would not be any valid channel for them; in this case, repeating this strategy can create a loop. To avoid the loop, we treat all new violated links as an out-of-path link. Second, some new violated links are out-of-path and there would not be any valid channel for them; in this case, resolving their violation may reassign channel of another link in $2IR$ range of the path.

In summary, the proposed group channel change procedure at most changes channel of the links in $2IR$ range of routing path of a demand to accept it.

5) On-demand Resource Utilization and Initial Channel Assignment

The available channels in the frequency spectrum and radios per node are scarce resources in multi-channel multi-radio networks. To utilize the resources efficiently, we assign channel to a link only if the link is in the path of an admitted demand. When an admitted demand (s, d, x, p, t_a, t_h) leaves the network at time $t_a + t_h$, we check all links in the path p . If there is not any other path going through link (u, v) , we *remove the assigned channel* to the link. Then we check nodes u and v , if (u, v) is the only link that uses the channel in the node we free the radio tuned to the channel.

The main advantage of this strategy is increasing probability of free radios in nodes. Providing free radios in nodes significantly improves the probability of accepting demands. Suppose (u, v) is a violated link and both nodes u and v have a free radio; in this case, the *candidate channel* set for the link contains all available channels, which means there would be at least one *valid channel* for the link with high probability.

To implement the strategy in our solution, we use virtual channel 0. To remove the assigned channel of a link we assign the channel 0 to the link. Assigning the channel to a link does not consume any radio. The physical channel capacity is 0. So, routing any demand along a link in channel 0 makes the link violated. Furthermore, interference set of a link in channel 0 contains only the link but not other links. In addition, in the *initial channel assignment*, when there is not any load, all links are assigned to channel 0.

B. Achieving Design Goals

As stated before, the objectives of the channel assignment algorithm for the problem are (i) maximizing the number of admitted demands while (ii) minimizing the number of channel reassignments and (iii) maintaining locality. The proposed design choices help us to accomplish the goals. Table II shows the relation between the design choices and the goals.

The number of admitted demands is boosted by the best channel selection, on-demand resource utilization, and group channel change strategies. Best channel selection provides more resources for the upcoming demands. On-demand resource utilization frees channels and radios for the upcoming demands and the group channel change offers more opportunity to resolve violations.

TABLE II. RELATION BETWEEN DESIGN CHOICES AND OBJECTIVES OF QoS DRIVEN DYNAMIC CHANNEL ASSIGNMENT

Choice Goal	On-demand Channel Reassignment	Best Channel Selection	On-demand Resource Utilization	Group Channel Change
Maximize # of admitted demands		√	√	√
Minimize # of channel reassignments	√	√		
Locality	√	√		√

The number of channel reassignments is kept small by on-demand reassignment and best channel selection. Because the on-demand strategy reassigns channels if it is required and the best channel selection metric provides more resources for the upcoming demands, which means more demands can be admitted without channel reassignment.

The proposed solution is localized. On-demand channel reassignment only focuses on the violated links that are in at most IR range of the path. Selecting the best channel needs only information of the other links in the interference range of the violated links. Finally, in the worst case, group channel change limits channel reassignments in $2IR$ range of the path.

C. QDDCA Algorithm

The aforementioned design choices are integrated in QoS Driven Dynamic Channel Assignment (QDDCA) algorithm. Pseudo-code of the algorithm is shown in Fig. 3. For a given demand (s, d, x, p, t_a, t_h) , QDDCA checks feasibility of the bandwidth allocation. If the path is not feasible then it finds violated links and calls *ResolveViolation*. For each violated link, *ResolveViolation* first tries to resolve the violation using *LinkChannelChange* in line 3 and if it is not successful, *GroupChannelChange* is invoked in line 5. Since changing channel of a link can resolve violation of multiple links, after each successful resolve the remaining violated links are rechecked in line 7. *LinkChannelChange* finds set of valid links and if there is any one, assigns the best one to the violated link.

In the group channel change, as mentioned before, we distinguish between in-path and out-of-path violated links. *GroupChannelChange* in line 1 checks that if the link is out-of-path or is created by the *GroupChannelChange* itself. If at least one of these conditions holds, it changes channel of links in interference set of the violated link in lines 2 and 3. It continues changing the channels until the link becomes unviolated or there is not any unvisited link in the interference range. If the conditions in line 1 do not hold, it checks each candidate channel for the violated link, in lines 5 – 9, by (i) assigning the candidate channel to the link, (ii) finding new violated links in the new channel, and (iii) attempting to resolve the new violations. This search is finished as soon as it finds a feasible candidate channel.

D. Worst Case Complexity Analysis

The worst case running time of QDDCA algorithm is the case that all links in the path p are violated and *LinkChannelChange* cannot resolve the violations. In this case,

for each link, we have to call *GroupChannelChange* wherein lines 5-9 run and *ResolveViolation* is recalled. In the worst case, *LinkChannelChange* again cannot resolve the new violations and we have to call *GroupChannelChange* again. However, in this case, lines 2-3 run that breaks the recursive function calls. To analyze the worst case, we use notations denoted in table III. Let \tilde{I} and \tilde{r} are respectively size of the largest interference set and the maximum number of radios per node. $O(LCC) = O(\kappa(\tilde{r} + \tilde{I}))$ since there are κ channels and we need to check radio and capacity constraint per channel. $O(GCC_1) = O(LCC)\tilde{I} = O(\tilde{I}\kappa(\tilde{r} + \tilde{I}))$. $O(GCC_2) = \kappa\tilde{r} + \kappa(\tilde{I} + \tilde{I}(O(LCC) + O(GCC_1)))$ since radio constraint must be check for κ channels and at most there would be \tilde{I} new violated links which *ResolveViolation* is called for. So, $O(GCC_2) = O(\kappa^2\tilde{I}^2(\tilde{r} + \tilde{I}))$. The length of path can be at most n , so $O(RV) = n(O(LCC) + O(GCC_2)) = O(n\kappa^2\tilde{I}^2(\tilde{r} + \tilde{I}))$ and finally $O(QDDCA) = n\tilde{I} + O(RV) = O(n\kappa^2\tilde{I}^2(\tilde{r} + \tilde{I}))$.

ALGORITHM: QDDCA

Input: $G, I, (s, d, x, p, t_a, t_h)$

1. Check allocating bandwidth x through path p
 2. **If** the path is feasible **then** return **Accept**
 3. **else**
 4. $VL \leftarrow$ Violated links
 5. *ResolveViolation* (VL)
 6. **if** violations are resolved **then** return **Accept**.
-

ALGORITHM: ResolveViolation

Input: G, I, V

1. **while** V is not empty **do**
 2. $(u, v) \leftarrow V[0]$
 3. *LinkChannelChange* $((u, v))$
 4. **if** violation is not resolved **then**
 5. *GroupChannelChange* $((u, v))$
 6. **if** violation is not resolved **then** return **Reject**
 7. **else** remove unviolated links from V
-

ALGORITHM: LinkChannelChange

Input: $G, I, (u, v)$

1. $VC \leftarrow$ Valid channels for (u, v)
 2. **if** VC is not empty **then**
 3. Update channel of (u, v) to the best channel
-

ALGORITHM: GroupChannelChange

Input: $G, I, (u, v)$

1. **If** (u, v) is out-of-path **or** $(u, v) \in NVL$ **then**
 2. **while** (u, v) is violated **and**
 3. there is unvisited $(a, b) \in I_{ab}$ **do**
 4. *LinkChannelChange* $((a, b))$
 5. **else**
 6. $CC \leftarrow$ Candidate channels for (u, v)
 7. **for** $\forall ch \in CC$ **and while** there is violation **do**
 8. Change channel (u, v) to ch
 9. $NVL \leftarrow$ New violated links
 10. *ResolveViolation* (NVL)
-

Figure 3. Pseudo-code of QDDCA algorithm.

TABLE III. NOTATION USED FOR COMPUTATIONAL COMPLEXITY ANALYSIS.

Notation	Complexity of Algorithm
$O(QDDCA)$	QDDCA
$O(RV)$	<i>ResolveViolation</i>
$O(LCC)$	<i>LinkChannelChange</i>
$O(GCC_1)$	Lines 2, 3 of <i>GroupChannelChange</i>
$O(GCC_2)$	Lines 5-9 of <i>GroupChannelChange</i>

It should be noted that the worst case occurs very rarely in practice. Our simulation results, table VI, show that the number of violated links per demand is much less than the number of nodes, n , and interestingly the number of channel reassignments per resolved violated link is less than 1 that means one channel reassignment resolves more than one violated link.

V. SIMULATION RESULTS

In this section, we present simulation results to evaluate the performance of QDDCA. First, the simulation parameters and other simulated algorithms are described. Then, we study effect of different parameters including arrival rate, the number of available channels, and the number of radios per node on performance of the algorithms.

A. Simulation Setup

We used a flow-level event-driven simulator. The simulator was developed in Java. Simulations were done in a random topology. Parameters of the topology are shown in table IV. In each run of the simulations, a set of random traffic with QoS constraint was used. Parameters of the traffic are shown in table V. Parameters in tables IV and V were the default values and were used in all simulations unless otherwise stated. The minimum-hop routing algorithm was used to find path of demands. As mentioned before, the admitted demands were not rerouted. Every result presented here, is the average obtained from 10 different traffic sets.

Six algorithms are compared in these simulations. There are two variations of QDDCA: QDDCA-LCC and QDDCA-GCC. QDDCA-LCC is the version in which group channel change is **not** used; in the other words, violations are resolved only by changing channel of the violated links. QDDCA-GCC is the full version. There are two static channel assignment algorithms: random and greedy minimum interference [17]. For these static algorithms, we applied the algorithms before loading the network and did not change channels later.

TABLE IV. PARAMETERS OF SIMULATION TOPOLOGY.

Parameter	Value
Area	$1000m \times 1000m$
Number of nodes	100
Radios per node	Uniform random variable in [2,5]
TR	150m
IR	350m
Number of channels	12
Physical Capacity	100Mbps

TABLE V. PARAMETERS OF SIMULATION TRAFFIC.

Parameter	Value
Number of demands	500
Arrival rate	Poisson random variable
Hold time	Exponential random variable, mean 10 min
Required bandwidth	Uniform random variable in [0, 20Mbps]

We also implemented dynamic channel assignment algorithm proposed in [6]. A few notes should be made about the implementation. First, in these simulations we assume (i) there is only one kind of radios in the network; (ii) all channels have a same physical channel capacity; (iii) all packets have a same length; (iv) collision and packet loss probabilities are negligible since the bandwidth allocation satisfies the row feasibility condition; and (v) overhead of MAC and PHY layers framing is negligible. Second, only one link exists between each node pair. Third, this method needs an initial channel assignment; here, we use the random and greedy algorithms as the initial channel assignment. Finally, the method needs a threshold to check load of radios, the value of the threshold is 0.6 in our simulations. To sum up, besides QDDCA, there are four other algorithms: Random-Static, Random-Dynamic, Greedy-Static, and Greedy-Dynamic. The dynamic versions are the integration of the static channel assignment and the method proposed in [6].

B. Effect of Demand Arrival Rate

Arrival rate of the QoS demands is the first parameter we investigated. Acceptance rate versus arrival rates are shown in Fig. 4. As seen in this figure, both versions of QDDCA outperform other static as well as dynamic approaches. Group channel change is an effective approach and improves acceptance rate of QDDCA-GCC up to 10% in comparison to QDDCA-LCC. Performance of the approach proposed in [6] depends on initial channel assignment; the Greedy-Dynamic has much better performance than Random-Dynamic. Another interesting point is that while Random-Dynamic has more acceptance rate than Radom-Static, but in most cases, Greedy-Static outperforms the Greedy-Dynamic especially in high-loads. It means that the decisions made by the dynamic algorithm have negative effects on acceptance of the upcoming demands.

C. Effect of Number of Available Channels

An efficient channel assignment algorithm should be able to exploit available channels. To compare the algorithms from this point of view, we conduct simulations with different number of channels. In these simulations, the arrival rate is equal to 5 demands per minute. Fig. 5 shows the performance of algorithms versus the number of available channels. This figure shows, besides the high acceptance rate of QDDCA, it has the maximum increase in acceptance rate per additional channel. More precisely, in this figure, average slope of the curves for QDDCA based, Greedy based, and Random based algorithms is respectively 0.041, 0.03, and 0.005. The higher value means more ability to exploit additional channels.

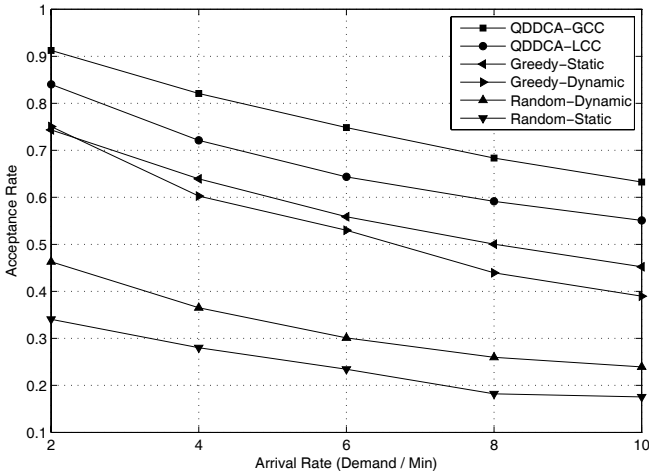


Figure 4. Acceptance rate versus arrival rate. Parameters of simulation are according to tables IV and V.

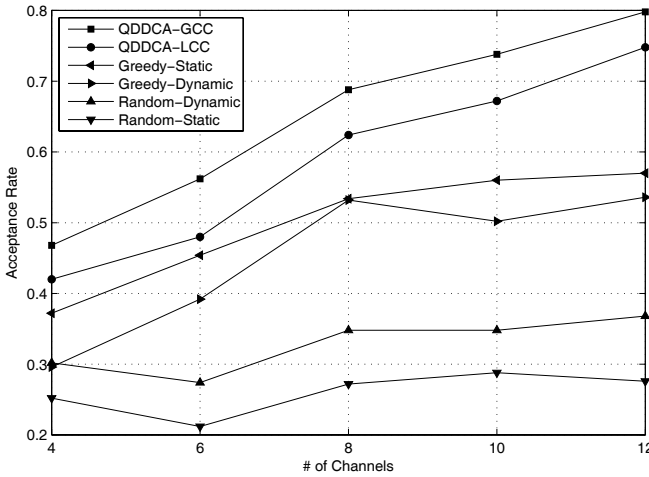


Figure 5. Acceptance rate versus the number of available channels. Arrival rate is 5 demands per minutes and other parameters are according to tables IV and V.

D. Effect of Number of Radios per Node

Radios in each node are scarce resources. An efficient algorithm should be capable to provide good performance even with a limited number of radios. To evaluate this capability of the algorithms, we compared their performance versus the numbers of radios per node, which is shown in Fig. 6. In these simulations, the arrival rate is equal to 5 demands per minute. Fig. 6 depicts that a limited number of radios per node, 4 or 5, is sufficient for QDDCA-GCC to achieve the maximum performance. This figure also indicates the number of radios is an important parameter in random based algorithms since providing more radios improves the performance of the algorithms.

Combining results in Fig. 5 and Fig. 6 indicates an important fact: providing many available channels with a very few radios per node is sufficient for QDDCA-GCC to achieve a good performance. This is situation of the real-life mesh

networks, e.g. there are 12 orthogonal channels in IEEE 802.11a, and each node usually has 2-4 radios in real networks.

E. Comparing with Optimal Static Channel Assignment

Here, we compare QDDCA against an Integer Linear Programming (ILP) based optimal static channel assignment. The ILP model was proposed in [17] and aims at minimizing total network interference. The model is the classic ILP model of the Max-K-Cut problem with additional constraint to satisfy the radio constraint². Since the problem is NP-Complete, the model is not solvable for large networks. In our experiments, the state-of-the-art solver CPLEX 11.0 [18] cannot solve the model for the 100-node simulation topology. Even the model cannot be solved for 50-node network in a reasonable time (less than 3 days). We present results in a 25-node network. In the network, nodes are scattered randomly in $750m \times 750m$ area, $TR = 200m$, and $IR = 400m$. The number of radios per node is a uniform random variable in $[2,5]$ and there are 10 channels.

Comparison between QDDCA-GCC, the optimal model, and Greedy-Static versus demand arrival rates is depicted in Fig. 7. As seen in this figure, whereas the optimal model has significantly better performance than Greedy-Static, about 15%, it is outperformed by the QDDCA-GCC algorithm. It must be noted that whereas the difference between acceptance rate of QDDCA-GCC and the optimal model is at most 5%, complexity of the model is much higher than QDDCA-GCC; the model is not practical for real-life large/medium networks as mentioned.

F. Operation Statistics

Here, we present statistics about operation of QDDCA. We are interested in average number of violated links per demand, percentage of resolved violated links, and average number of channel reassignment per resolved violated link. The statistics are shown in table VI.

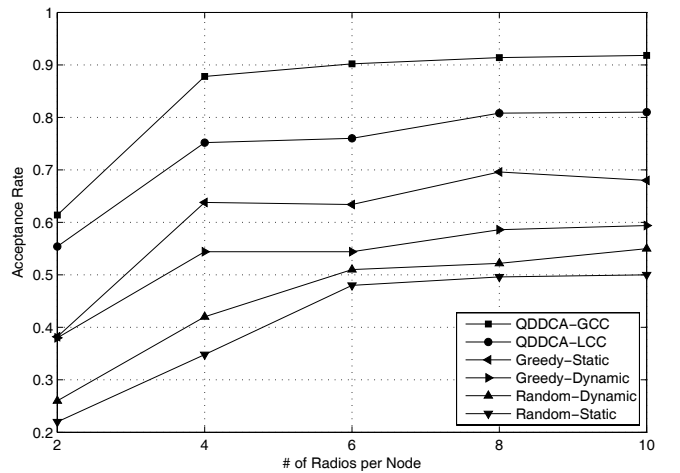


Figure 6. Acceptance rate versus the number radios per node. Arrival rate is 5 demands per minutes and other parameters are according to tables IV and V.

² We improved the model by adding two sets of additional valid inequalities which are not presented here due to space limitation.

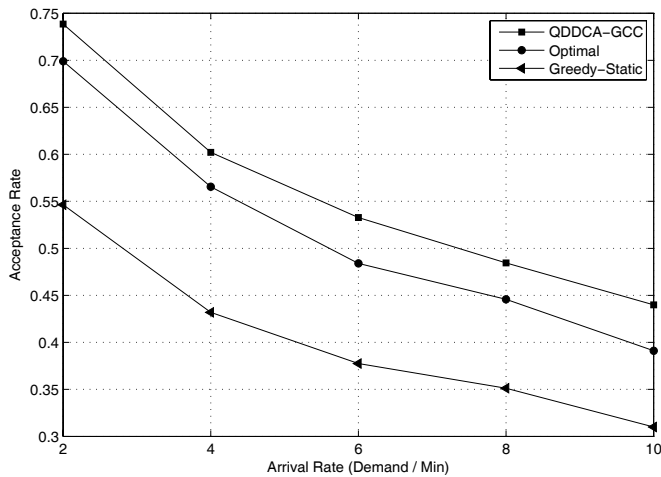


Figure 7. Acceptance rate versus demand arrival rate. Simulation topology is the 25-node random topology and traffic parameters are as table V.

TABLE VI. STATISTICS OF LCC AND GCC VERSION OF QDDCA. SIMULATION PARAMETERS ARE SHOWN IN TABLES IV AND V.

Parameter Arrival Rate	Violation / Demand		Resolved / Violation		Channel Change / Resolved	
	LCC	GCC	LCC	GCC	LCC	GCC
1	3.987	3.946	0.897	0.945	0.987	0.976
2	3.955	3.956	0.718	0.835	0.959	0.923
3	3.901	4.059	0.634	0.785	0.934	0.873
4	4.125	4.380	0.545	0.702	0.920	0.832
5	4.128	4.885	0.476	0.622	0.915	0.810

Violation per demand in QDDCA-GCC is greater than in QDDCA-LCC because the extra demands accepted by QDDCA-GCC consume some resources and consequently the upcoming demands generate more violated links. As expected from previous results QDDCA-GCC has more resolve per violation. Average channel change per resolved violation in QDDCA-GCC is unexpectedly less than QDDCA-LCC. The reason is that changing channel of a link in interference range of violated link, which is done by *GroupChannelChange* in QDDCA-GCC, not only resolves the violated link but also can resolve some other violated links.

VI. CONCLUSION

We studied the problem of performance optimization of multi-channel multi-radio wireless mesh networks in presence of traffic with QoS constraint. The performance is measured in terms of number of accepted demands. We proposed an on-line dynamic channel assignment to improve the performance by adapting network resources. The algorithm is localized, does not require prior information of the offered load, and aims to keep the number of channel reassignments small. We developed the algorithm based on on-demand channel assignment, best channel selection, and group channel change ideas. In demand arrival time, the algorithm detects the *violated links* and tries to find the best channel for them. If it is not possible, due to the radio and capacity constraints, it changes channel of a group of links. Our simulation results show that the efficient

algorithm that adapts channel assignment according to the traffic requirement can increase network performance up to 23% in comparison to static approaches and even it outperforms optimal static channel assignment. Here, we assumed the path of demands is given. In the future work, we plan to investigate the joint QoS routing and dynamic channel assignment problem.

- [1] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, April 2004.
- [2] A. H. M. Rad and V. W. S. Wong, "Joint optimal channel assignment and congestion control for multi-channel wireless mesh networks," *IEEE ICC*, 2006.
- [3] A. Giannoulis, T. Salonidis, and E. Knightly, "Congestion control and channel assignment in multi-radio wireless mesh networks," *IEEE SECON*, 2008.
- [4] A. Rainwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," *IEEE INFOCOM*, 2005.
- [5] V. Bhandari and N. Vaidya, "Channel and interface management in a heterogeneous multi-Channel multi-Radio wireless network," Technical Report, University of Illinois at Urbana-Champaign, Mar. 2009.
- [6] H. Wu, F. Yang, K. Tan, J. Chen, Q. Zhang, and Z. Zhang, "Distributed channel assignment and routing in multi-radio multi-channel multi-hop wireless networks," *IEEE JSAC special issue on multi-hop wireless mesh networks*, vol. 24, no. 11, pp. 1972–1983, Nov. 2006.
- [7] J. Crichigno, M.Y. Wu, and W. Shu, "Protocols and architectures for channel assignment in wireless mesh networks," *Ad Hoc Networks*, vol. 6, no. 7, pp. 1051-1077, Sep. 2008.
- [8] K. Ramachandran, E.M. Belding, K. Almeroth, and M.M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," *IEEE INFOCOM*, 2006.
- [9] A. Naveed, S.S. Kanhere, and S.K. Jha, "Topology control and channel assignment in multi-radio multi-channel wireless mesh networks," *IEEE MASS*, 2007.
- [10] J.G.L.C.T Chou, and S. Jha, "Non-cooperative coexistence of co-located independent wireless mesh networks," *IEEE MASS*, 2007.
- [11] L. Chen, Q. Zhang, M. Li, and W. Jia, "Joint topology control and routing in IEEE 802.11-based multiradio multichannel mesh networks," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3123–3136, Sep. 2007.
- [12] B. Raman, K. Chebrolu, D. Gokhale, and S. Sen, "On the feasibility of the link abstraction in wireless mesh networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 528 - 541, Apr. 2009.
- [13] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and qos routing in multi-channel wireless mesh networks," *ACM MobiHoc*, 2005.
- [14] P. Gupta, P.R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388 - 404, Mar. 2000.
- [15] M. Alicherry, R. Bhatia, Li Erran Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 24, no. 11, pp. 1960 - 1971, Nov. 2006.
- [16] R. Gupta, J. Musacchio, and J. Walrand, "Sufficient rate constraints for qos flows in ad-hoc networks," *Ad Hoc Networks*, vol. 5, no. 4, pp. 429 - 443, May 2007.
- [17] A. P. Subramanian, H. Gupta, S. R. Das, "Minimum-interference channel assignment in multi-radio wireless mesh networks," *IEEE Transaction on Mobile Computing*, vol. 7, no. 12, pp. 1459 - 1473, Nov. 2008.
- [18] ILOG CPLEX methetical programming optimizers. <http://www.ilog.com/products/cplex>