Multipath Selection and Channel Assignment in Wireless Mesh Networks

Soo-young Jang and Chae Y. Lee

Dept. of Industrial and Systems Engineering, KAIST, 373-1 Kusung-dong, Taejon, Korea
Tel: +82-42-350-5916, FAX: +82-42-350-3110
Email: {pz1004, chae}@kaist.ac.kr

Abstract In wireless networks, it is very important to optimize the number of channels, due to the limit on the number of usable channels in a given network. In addition, multimedia services with high QoS requirements with respect to throughput and delay have recently become popular. To satisfy these requirements, it has become important to find a way of providing multipath transmission.

A channel assignment algorithm is presented that minimizes the number of required channels while satisfying the throughput requirements of source-destination pairs in multichannel, multiradio, multirate wireless mesh networks. A mathematical model is proposed that considers interference effect, link capacity, and throughput requirements. A novel channel assignment algorithm is developed that takes into account multipath selection, channel reusability, link capacity sharing, and global optimization. The performance of the algorithm is compared with that of CPLEX, using 24 network scenarios. The maximum gap between the CPLEX solutions and those of the proposed algorithm is, on average, only 4.8%.

Keywords unicast, channel assignment, multipath transmission, QoS requirement , wireless Mesh networks

1. Introduction

A Wireless Mesh Network [1, 2] is a fully connected wireless multihop network whose nodes are all connected to each other via multiple hops, as shown in Figure 1. The network is made up of access points (APs), wireless mesh routers (WMRs), and mobile nodes (MNs) and supports two types of transmission: (i) infrastructureless multihop transmission via communication devices, such as the widely known Ad Hoc Networks and (ii) infrastructure-based transmission via WMRs and APs. As a result, the network can benefit, not only from the efficiency of multihop transmissions, but also from the stability of using APs and WMRs. APs also allow communication between wired Internet and wireless devices.

To date, a number of notable papers on wireless mesh networks have been published, on such matters as power control, channel assignment, routing mechanisms, network connectivity, and QoS issues. Among these topics, efficient channel assignment and QoS issues are regarded as being of major importance. The channel assignment problem can be seen from two points of view. One is minimizing the number of required channels and the other is fully utilizing the given amount of channels. QoS issues recently have become the subject of particular interest, due to the increase in multimedia streaming services such as IPTV/VOD services. Multimedia streaming services have various and high QoS requirements, according to the content or desired service qualities. The QoS requirements need to be satisfied in order to provide seamless service. Providing multipath transmission is one way of satisfying these requirements [3, 4, 6, 15, 16, 23]. Benefits of multipath transmissions are fault tolerance, load balancing, bandwidth aggregation, reduced delay, etc. [23]. Among them, two major benefits of applying multipath transmission to multimedia traffics are bandwidth aggregation and reduced delay. First, when links suffer bad channel condition and restricted bandwidth, single path may not satisfy the high throughput requirement of the multimedia traffics. In such a case, multipath transmission – splitting data through multiple different paths – can help to satisfy the throughput requirement. Second, when a path is broken, source-destination pair cannot communicate until the alternative path is found, which increases the end-to-end delay. However, this multipath technique tends to increase the number of

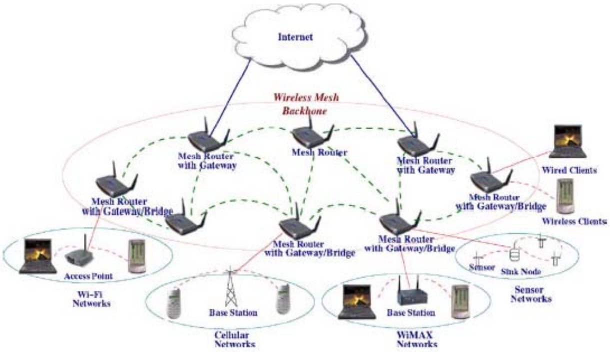


Figure 1 Wireless mesh networks [1]

required channels in the network. That being so, for multipath transmission, deciding the number of multipaths between a source-destination pair is one of the most important issues. The term *given multipath* in the paper refers to the multipath routes formed by any existing multipath routing algorithm. Thus *multipath selection* refers to the selection of a combination of given multipaths for flow streams. It does not mean the formation of multipath routes as used in the other papers.

The channel assignment problem of minimizing the number of required channels while satisfying the throughput requirements of source-destination pairs is a well-known NP-hard problem [4]. In this paper, we present a way of solving this problem, taking into account the throughput requirements of source-destination pairs and other network-related constraints in the channel assignment problem. Our solution to the problem has notable implications. For instance, it can be applied to multimedia services. As noted above, multimedia services have various QoS requirements. If all the multimedia services are being provided by the network, it is important to allocate the necessary resources to each source-destination pair according to its QoS requirement, so that the network resources can be used efficiently. For the defined problem, we propose a mathematically formulated model using binary integer programming and a novel channel assignment algorithm.

The contributions of this paper are that (i) we model the multipath selection and channel assignment problem mathematically, considering constraints such as link capacity, throughput requirements, and interference effect and (ii) propose a multipath selection and channel assignment algorithm that takes multipath selection, channel reusability, and global optimization into account.

The remainder of this paper is organized as follows. Section 2 discusses related work on channel assignment and multipath routing. Section 3 presents the system model, including assumptions, interference, and network flow models, that is used throughout the paper. Section 4 describes the channel assignment problem in detail. Section 5 introduces the proposed channel assignment algorithm. Section 6 describes the simulation and discusses the results. Section 7 concludes.

2. Related Work

2.1. Channel Assignment

Research on the channel assignment issue focuses on either distributed or centralized algorithms. In distributed algorithms, nodes use the local link-state information (usually within a two-hop or three-hop range) and allocate channels in a distributed manner [7, 9, 10, 18, 20, 21]. The distributed algorithms focus more on how to obtain m-hop neighbor information, how to solve hidden node problems, and how to handle changes of local information rather than the optimization problem itself. Centralized algorithms use the global link states and traffic information to centrally allocate channels to the networks [4, 5, 17, 18, 19]. Centralized algorithms are likely to yield a better solution than a distributed algorithm, because they use the full network information, rather than local link-state information and focus more on optimizing the network resources and performances. However, they require heavy overhead to maintain the network topology and the global link-state information when the network size is big. In [4], the authors simply assign channels to links with heavy traffic first, while in [5], an algorithm to optimize throughput considering the fairness constraints is suggested. However, neither of these algorithms takes into account the network interference. In [17], the authors consider interference between routers in the mesh networks and also, the mesh network and other co-located wireless networks, while in [18], the authors try to minimize interference considering the number of network interface card limitation of each node. In [19], the authors consider multi-rate nature of the wireless network. But none of the papers mentioned above consider multipath transmission between source-destination pairs and link capacity sharing among several flows.

2.2. Multipath Transmission and Selection

The multipath transmission issue can be addressed in two ways: the implementation of backup paths and concurrent paths, according to how the multipath information is used.

The backup paths method uses multipath information as a supplement path [22]. When the current path is broken, due to obstacles or the mobility of the source, destination, or relay nodes, the source-destination pair uses a supplementary path for transmission. This can reduce the end-to-end delay caused by searching for a new route while communicating. However, if the current path is sufficiently reliable and does not break during the transmission, the reserved resource in the supplement path is wasted.

The concurrent path method transmits flows concurrently along the multiple paths [3, 4, 6, 11]. There are two ways of implementing this method according to whether the same flows or different flows are transmitted through the multipath. The same flows are generally used for transmissions where reliability is the critical issue, such as when sending emergency signals [11]. However, using the same flows can generate excessive inefficiency in the networks. Different flows are mostly used to increase the throughput of the networks [3, 4, 6]. To date, most research on the concurrent path method has only considered the use of the same number of multipath for all source-destination pairs in the network. In this case, even though the QoS requirement of some source-destination pairs can be achieved using only a single path, these pairs use more resource than a single path, which causes inefficiency in the network.

Many studies on channel assignment have dealt with throughput optimization, but few have considered the various throughput requirements of source-destination pairs. Further, to the authors' knowledge, no reported study has considered a multipath selection scheme, which uses the given multipath information selectively for communication in order to minimize the number of required channels. If the throughput requirement is high, the source-destination pair is likely to use two or more paths for transmission in order to distribute its high throughput requirement across multiple paths. However, if the throughput requirement is low, the pair is likely to use only a single path for transmission. Using the number of multipath dynamically according to such demands, network resources can be used efficiently.

3. System Model and Assumptions

3.1. Assumptions

We assume that the multipath information for each source-destination pair is given. However, a source-destination pair does not have to use all the given multipath. It can selectively use a portion of the multipath information, in order to minimize the number of required channels. A pair can flow through a multipath simultaneously to increase the throughput of a source-destination pair. In this case, the sum of the flow rate of its multipaths is the flow rate of the pair. All devices, such as APs, WMRs, and MNs are equipped with several network interface cards (NICs). There are no collisions in the networks, and the link capacity varies according to the transmission range.

3.2. Interference Model

We define the interference set $I(i, j)$ as the links in the interference range of a link (i, j), where (i, j) denotes a directional link from i to j. The interference effect means that a link (i, j) and its interference set $I(i, j)$ cannot transmit simultaneously with the same channel. In Figure 2, the interference set $I(1, 2)$ is the link set $\{(2, 7), (3, 4)\}$ and by the interference effect, links $(1, 2)$, $(2, 7)$ and $(3, 4)$ cannot use the same channel. However, links $(1, 2)$ and $(5, 6)$ can use the same channel for transmission.

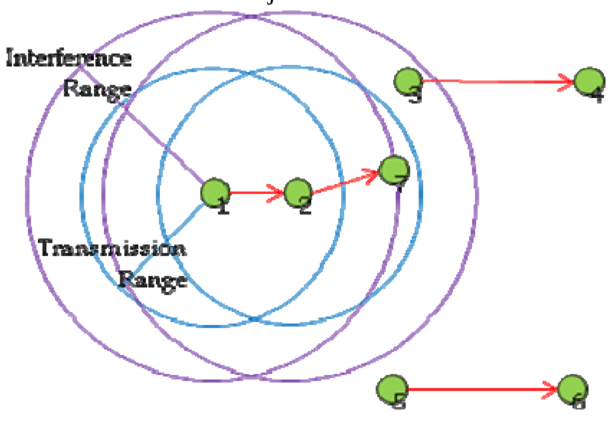


Figure 2 Interference model

3.3. Network Flow Model

The flow can be divided and flow along multipaths simultaneously, which makes it possible to increase the throughput between the source-destination pair without using two or more channels in one link. By exploiting this characteristic, we can vary the ways in which the throughput requirement of the source-destination pair can be satisfied. We can use additional channels in a bottleneck link or use additional multipaths.

3.4. Routing Model

We assume that when a source sends a data packet to a destination, the entire path information is included in the packet header, as is the case with the dynamic source routing (DSR) algorithm [8, 11, 12]. The flow simply follows the route that the source node has decided and cannot be merged or divided at the relay nodes.

4. Problem Formulation

4.1. Notations

A network is represented by a directed graph $G = (V, E)$, where V represent a set of vertices that includes APs, WMRs, and MNs, and E represents the transmission links. C represents the set of all the possible channels in the network. The following notation is used in the mixed integer linear programming (MILP) formulation:

- R_k: Throughput requirement of the source-destination pair k
- C_{ij}: Channel capacity of link (i, j)

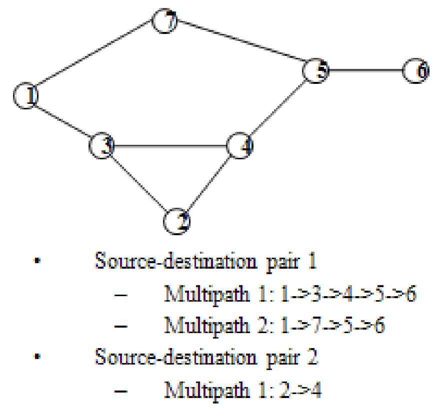


Figure 3 Sample network for P_k^m

- f_{ijkm}: Data rate of the m^{th} path of the source-destination pair k using link (i, j)
- r_k^m: The end-to-end data rate of the m^{th} path of the source-destination pair k (The minimum data rate among the links that constitute the m^{th} path of the source-destination pair k is the end-to-end data rate of the m^{th} path of the source-destination pair k)
- x^c_{ij}: a binary variable to represent the occupancy of channel c in a link (i, j)
- w^c: a binary variable to represent the occupancy of channel c in the network
- P_k^m: a set of links constituting the m^{th} path of the source-destination pair k

For the sample network given in Figure 3, we have the following paths:
$P_{11} = \{(1, 3), (3, 4), (4, 5), (5, 6)\}$
$P_{12} = \{(1, 7), (7, 5), (5, 6)\}$
$P_{21} = \{(2, 4)\}$
$P_{22} = \{(2, 3), (3, 4)\}$

4.2. Objective and Constraints

The objective is to minimize the number of required channels in the network:

$$Min \sum_{c\in C} w^c \qquad (1)$$

We consider the following five constraints:

- Throughput requirement
- Flow conservation equation
- Link capacity
- Network's channel occupancy
- Interference effect

The first constraint is the throughput requirement. The end-to-end data rate of all source-destination pairs should be higher than their own throughput requirements.

$$\sum_{m\in M_k} r_k^m \ge R_k,\ for\ all\ k \in K \qquad (2)$$

The second constraint is the flow conservation equation. The incoming and outgoing data rate for every node should be conserved. For each source-destination pair, the outgoing rate of the source is r_k^m, the incoming rate of the destination is r_k^m, and the difference between the incoming and outgoing rate of the relay nodes is 0. Relay nodes only relay the incoming flows to the next nodes.

$$\sum_{\{j|(i,j)\in P_k^m\}} f_{ijkm} - \sum_{\{j|(j,i)\in P_k^m\}} f_{jikm} = \begin{cases} +r_k^m,\ if\ i = s_k, & for\ all\ m\in M_k\ and\ k\in K \\ -r_k^m,\ if\ i = d_k, & for\ all\ m\in M_k\ and\ k\in K \\ 0,\quad otherwises, & for\ all\ m\in M_k\ and\ k\in K \end{cases} \qquad (3)$$

The third constraint is the link capacity. The sum of the rates of all flows that flow through a link (i, j) should be less than the link capacity of the link (i, j). The capacity of a link (i, j) is the product of the number of occupied channels in the link (i, j) and the maximum transmission rate of a channel, which is decided by the distance between nodes i and j.

$$\sum_{k\in K}\sum_{m\in M_k} f_{ijkm} \le C_{ij}\sum_{c\in C} x_{ij}^c,\ for\ all\ (i,j)\in E \qquad (4)$$

The fourth constraint is the network's channel occupancy. This constraint checks whether or not channel c is used in the network.

$$x_{ij}^c \le w^c,\ for\ all\ (i,j)\in E\ and\ c\in C \qquad (5)$$

The fifth constraint is the interference effect. The same channel cannot be used in a link (i, j) and its interference set $I(i, j)$. If channel c is occupied by the link (i, j), none of the links in the interference range of the link (i, j) can use channel c for transmission.

$$x_{ij}^c + x_{i'j'}^c \le 1,$$
$$for\ all\ (i',j')\in I(i,j),\ (i,j)\in E\ and\ c\in C \qquad (6)$$

4.3. Formulation of the Defined Multipath Selection and Channel Assignment

Given the foregoing, the defined multipath selection and channel assignment problem can be formulated as MILP1.

Note that finding a channel assignment for optimal performance even when the multiple paths are given is NP-hard [4]. This implies that the solution time of any known exact algorithm increases exponentially as the size of the problem increases, which means that any known exact algorithm is inappropriate for a real-world solution to the problem.

Therefore, we suggest multipath selection and channel assignment algorithm that will solve the problem and that is suitable for use in real-world applications, in a reasonable amount of time, with satisfying results.

MILP1:

$$Min \sum_{c \in C} w^c$$

$$subject \;\; to$$

$$\sum_{m \in M_k} r_k^m \geq R_k, \qquad \qquad for \; all \; k \in K$$

$$\sum_{\{j|(i,j) \in P_k^m\}} f_{ijkm} - \sum_{\{j|(j,i) \in P_k^m\}} f_{jikm} = \begin{cases} +r_k^m, & if \;\; i = s_k, \quad for \; all \; m \in M_k \; and \; k \in K \\ -r_k^m, & if \;\; i = d_k, \quad for \; all \; m \in M_k \; and \; k \in K \\ 0, & otherwises, \quad for \; all \; m \in M_k \; and \; k \in K \end{cases}$$

$$\sum_{k \in K} \sum_{m \in M_k} f_{ijkm} \leq C_{ij} \sum_{c \in C} x_{ij}^c, \qquad \qquad for \; all \; (i, j) \in E$$

$$x_{ij}^c \leq w^c, \qquad \qquad for \; all \; (i, j) \in E \; and \; c \in C$$

$$x_{ij}^c + x_{i'j'}^c \leq 1, \qquad \qquad for \; all \; (i', j') \in I(i, j), \; (i, j) \in E \; and \;\; c \in C$$

$$x_{ij}^c \in \{0,1\}, \qquad \qquad for \; all \; (i, j) \in E \; and \; c \in C$$

$$w^c \in \{0,1\}, \qquad \qquad for \; all \; c \in C$$

5. Algorithm

The proposed algorithm considers four important aspects to satisfy the objective of minimizing the number of required channels of the networks: multipath selection, link capacity sharing, channel reusability, and global optimization. The algorithm consists of Multipath Selection, Channel Allocation and Channel Rearrangement, as shown in Figure 4.

5.1. Multipath Selection and Channel Allocation

We first assume *k*-shortest paths are given for each source-destination pair. We then select a subset of the k-shortest paths that requires least number of channels.

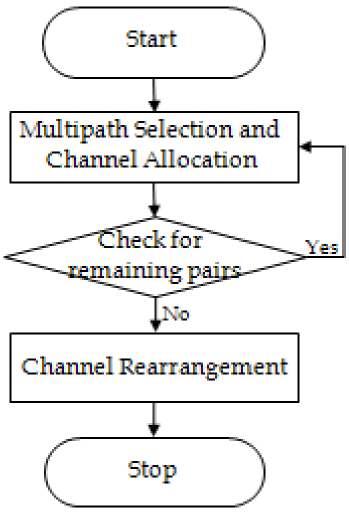


Figure 4 Channel assignment algorithm

The number of required channels for each source and destination pair is determined by preassigning a channel in each link with the following reuse policy.

(1) Reuse the channel previously allocated to the link by other source-destination pairs. This case is possible only when the link has some extra bandwidth for current source-destination pair.

(2) Reuse the channel used out of its interference range of the link

The interference is not considered for initial channel assignment. However, as the iterations proceed, we take into account the information about channels and data rates that were previously allocated by other flows.

At each iteration, when all source-destination pairs are examined, we select the source-destination pair and its multipaths that give the least number of required channels.

Algorithm. Multipath Selection and Channel Allocation

Input: Network graph G(V, E), Set of source-destination pair K, Set of subset of multipaths of source-destination pair k M_k
Output: Channel allocation information E_c, Data rate allocation information E_d

Repeat
  For each pair $k \in$ K **do**
    For each subset of multipath $m \in M_k$ **do**
      Calculate the number of required channels of *m* referencing E_c, E_d and G(V,E).
    End for
    Select *m* that has the least number of required

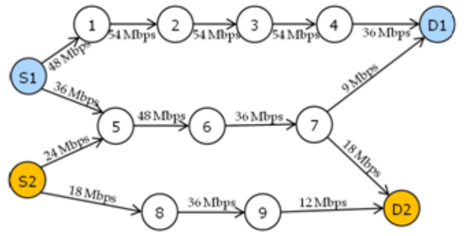
channels as the multipath of the pair *k*.
End for

Select the pair *k* that has the least number of required channels among set K.
Allocate channels and a data rate to the multipath of pair *k*
Update E_c, E_d
Delete the pair *k* from the set K.

Until set K is empty

When calculating the number of required channels for multipaths of a source-destination pair, we consider the information about previously allocated channels and data rates. This allows us to use two important properties: (i) link capacity sharing by which we can reduce the number of required channels by using the same link between two or more flows; and (ii) channel reusability, by which we can reuse the same channel without affecting the number of required channels as long as we avoid the interference range of a channel. For example, if channel c is allocated to link (*i*, *j*), then using channel c in the link (*i'*, *j'*), which is outside the interference range of link (*i*, *j*), would not affect the total number of channels that the network requires. In addition, using the remaining capacities of the links to which channels and data rate were previously allocated by other flows would not affect the total number of channels that are required. So, in both cases, it is efficient to use channel c in the link (*i'*, *j'*) and the path that contains the links. The channel allocation process utilizes these properties. Hence, we can conclude that the channel allocation process promotes the sharing of links or the reusing of channels, thereby reducing the total number of required channels.

5.2. Channel Rearrangement



	S - D pair1	S - D pair2
Throughput Requirement	1 Mbps	0.7 Mbps
Multipath Information	P1 (S1 - 1 - 2 - 3 - 4 - D1)	P1 (S2 - 5 - 6 - 7 - D2)
	P2 (S1 - 5 - 6 - 7 - D1)	P2 (S2 - 8 - 9 - D2)

Figure 5 Sample network for the algorithm

At the start of the channel rearrangement process, the channels are preliminarily allocated to the network as in the previous section. Since the preliminary allocation is performed based on each path connecting source-destination pair, the allocation may inefficiently increase the number of required channels. To solve this problem, we reallocate channels to the links by modifying the Welsh and Powell's algorithm [13] as follows.

The steps in the algorithm are as follows:

1) Sort the links in decreasing order of the size of their interference sets and initially leave every link with no channel.

2) Traverse the links in order, assigning channel 1 to the first link that has not yet received an allocation, provided that it does not yet have a neighbor with channel 1 allocated to it.

3) Repeat this process with channels 2, 3, etc. until channels have been allocated to all links.

5.3 Time Complexity of the Algorithm

Parameters that affect the time complexity of the proposed algorithm are the number of channels c, the number of given multipaths m, and the number of pairs p.

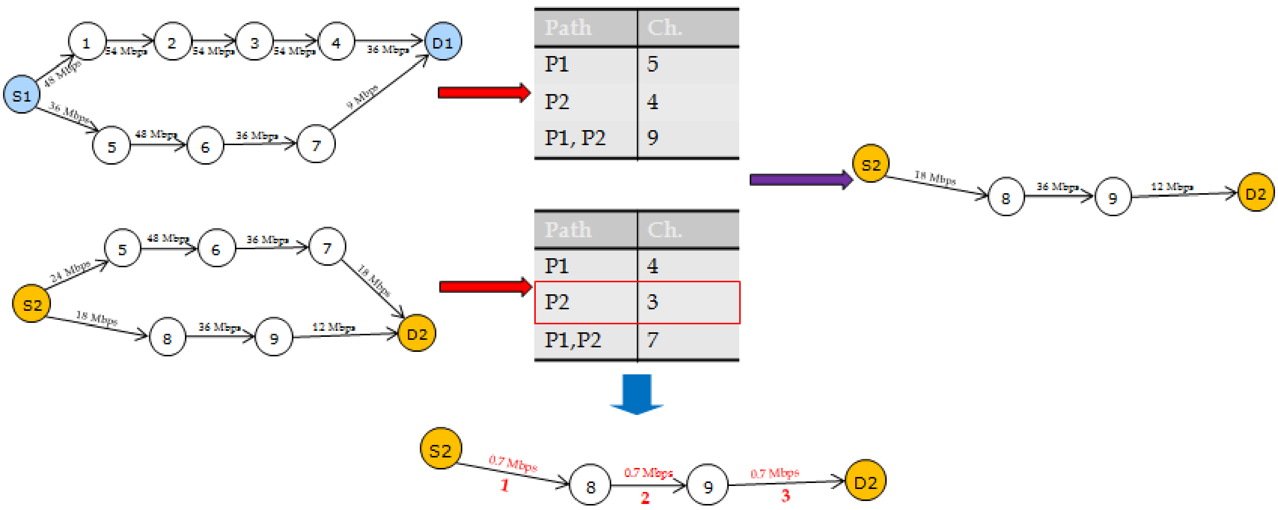


Figure 6 Multipath Selection and Channel allocation (the first iteration)

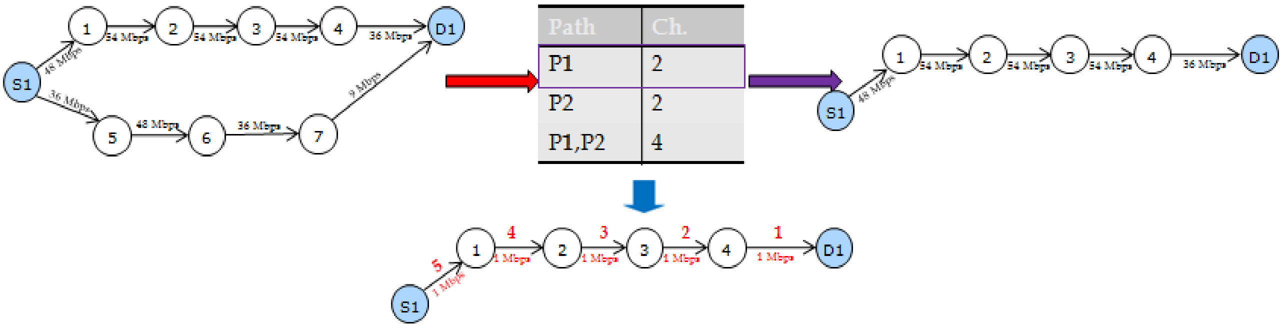


Figure 7 Multipath Selection and Channel allocation (the second iteration)

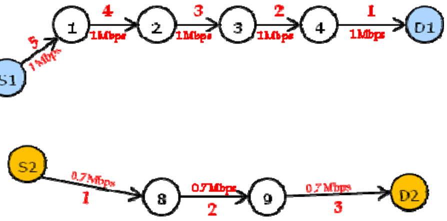


Figure 8 Result of channel allocation

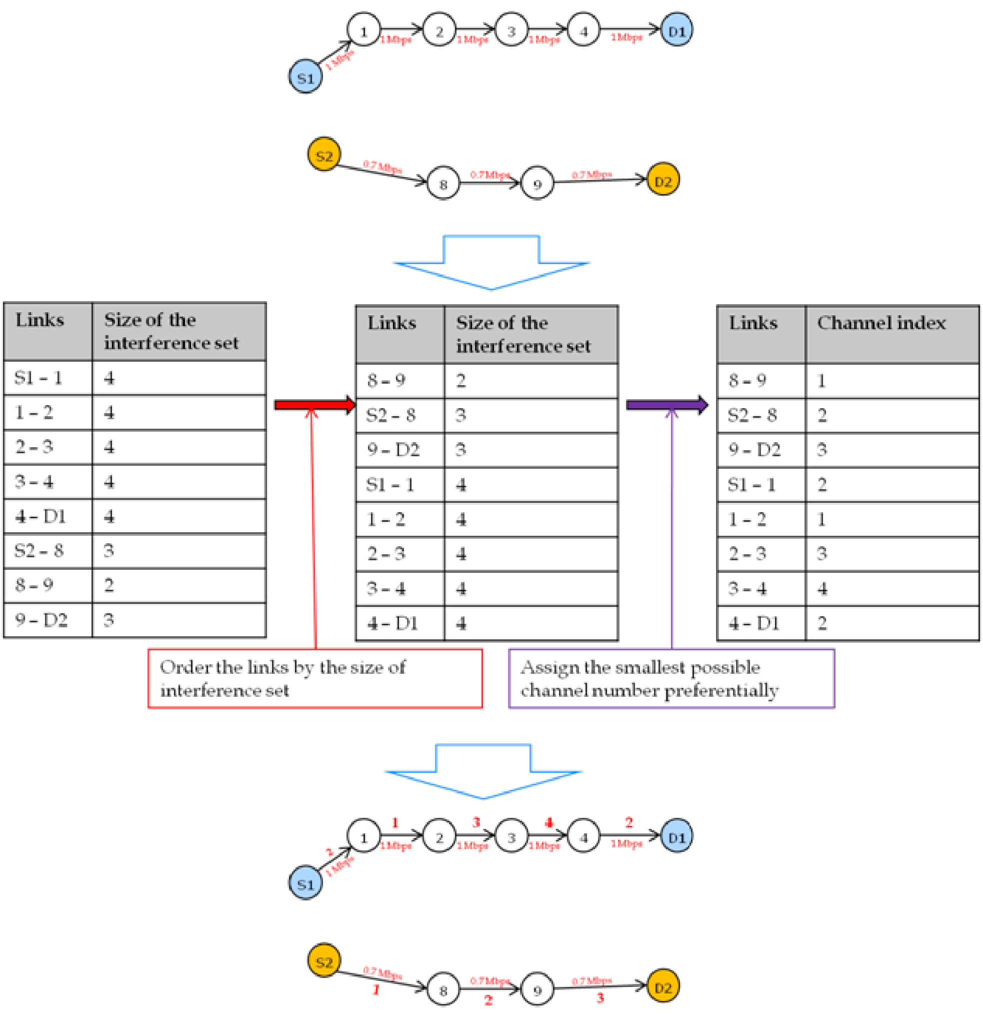


Figure 9 Channel rearrangement

Considering basic operations of the algorithm, the complexity of the Multipath Selection and Channel Allocation is given by
$O(p*(p*(2^m*(m*(p*m)*c)+2^m)+p)) = O(c*p^3*m^2*2^m)$

The complexity of Channel Rearrangement becomes $O((p*m)^2+c*(p*m)*(p*m)) = O(c*p^2*m^2)$

From the above, the time complexity of the proposed algorithm is $O(c*p^3*m^2*2^m)$. It shows that the complexity depends mostly on (i) the number of multipaths and (ii) the number of source-destination pairs.

5.4. Algorithm Process with a Sample Network

We now examine how the algorithm works by following an example given in Figure 5:

1) Overview

The sample network in Figure 5 has two source-destination pairs. Each pair has two given multipaths (P1 and P2) and a throughput requirement. We can predict that the multipath selection and channel allocation process needs two iterations to allocate channels and data rates to two source-destination pairs, one for each pair.

The algorithm follows the three processes:

1. Multipath Selection and Channel Allocation (the first iteration)
2. Multipath Selection and Channel Allocation (the second iteration)
3. Channel Rearrangement

2) Process

1. Multipath Selection and Channel Allocation (the first iteration)

We choose the P2 of the source-destination pair 2 that has the least number of required channels and allocate channels 1, 2, and 3 and a data rate 0.7 Mbps, which is the throughput requirement of source-destination pair 2.

Given that there exists a source-destination pair to which channels and data rate are not assigned, continue with a second iteration.

2. Multipath Selection and Channel Allocation (the second iteration)

Given that there is only one pair left, we only have to decide the route that requires the least number of channels. In the first iteration, channels 1, 2, and 3 are already assigned to P2 of source-destination pair 2. Considering channel reusability, using channel 1, 2, and 3 again in source-destination pair 1 does not affect the required number of channels. So the number of channels that are required for P1 and P2 of source-destination pair 1 change; the result is shown below. We select P1 of source-destination pair 1 and allocate channels 1, 2, 3, 4, and 5 and a data rate 1 Mbps, which is the throughput requirement of source-destination pair 1.

Given that the channel and data rate of all pairs are allocated, we go to the channel rearrangement process.

3. Channel Rearrangement

We delete all of the information about the preliminary allocation of channels, and reallocate the smallest possible channel index among the possible channel set in order of the size of interference sets.

We can see that the number of required channels has decreased to 4 from 5 after channel rearrangement.

6. Simulation

The simulation is performed using the following set up. The network topology consists of one AP, wireless mesh routers (which comprised 20% of the nodes in the network), and mobile nodes. The AP is placed in the center of the network space. The wireless mesh routers and mobile nodes are distributed randomly over a terrain of 700 X 700 meters. The transmission rates are given in Table 1. The maximum transmission range is 90 m and the interference range is 180 m [5].

We use the k-shortest path algorithm to find the k-multipath information between every source-destination pair. All procedures are run on a Pentium IV-2.8 GHz PC with 1024Mbytes of memory. In the simulation, we first compare our algorithm with nearly optimal solutions that are obtained by CPLEX[14] to solve the MILP formulation presented in Section 4. CPLEX is an optimization software package that can solve linear programming, quadratic programming, integer programming, and mixed integer linear programming problems (MILP). CPLEX applies the branch and cut technique for solving MILP problems. The number of branches in the branch and cut technique increases exponentially as the number of binary decision variables increases. Thus, we adopt a timeout stopping rule to ensure the search to terminate in a reasonable time. For example, in the case of 50 nodes, 10 source-destination pairs and 2 multipaths, the number of binary decision

Transmission rate (Mbps)	Transmission range(m)
6	90
9	77
12	69
18	60
24	45
36	37
48	32
54	30

Table 1 Transmission rate vs. transmission range

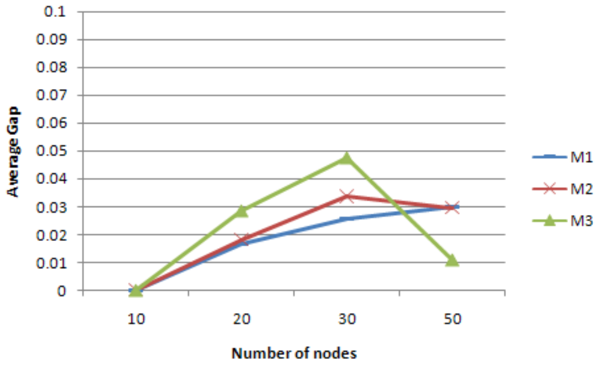


Figure 10 Number of nodes vs. average gap (pairs = 0.2* nodes)

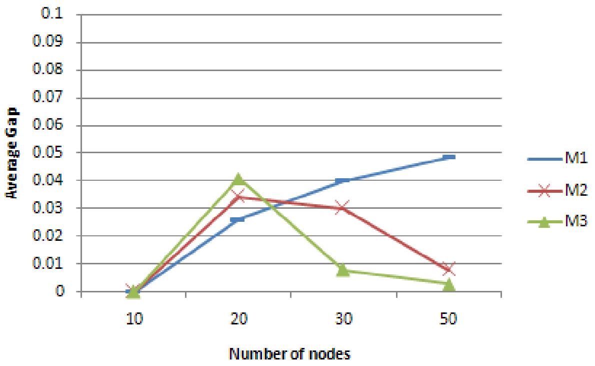


Figure 11 Number of nodes vs. average gap (pairs = 0.4 * nodes)

variables is about 500, and it generates huge number of branches. Without the timeout stopping rule, the process might continue several more hours until the computer runs out of memory without any progress in the solution. 10,000 seconds is a long enough CPU time for our problem instances before the computer runs out of memory. In most cases, CPLEX solves channel allocation problems and yields optimal solutions. However, as the problems become more complex, CPLEX fails to solve them within the time limit of 10,000 seconds, due to the exponentially increasing number of branches that are generated during the search process. In this case, we compare our algorithm with the best feasible solutions that CPLEX obtained within the time limit.

We test our algorithm with 24 network scenarios and compare with CPLEX solutions. Table 2 shows the simulation results. Network scenarios are varied in terms of the number of nodes, the number of source-destination pairs, and the number of given multipaths. The numbers of nodes considered are 10, 20, 30, and 50. The number of source-destination pairs is set at 20% and 40% of the number of nodes. The tests are performed with 1, 2, and 3 multipaths. The throughput requirements of the source-destination pairs have a uniform distribution between [0.5, 2.0] Mbps. For each network scenario, five problems are randomly generated and compared with the CPLEX solutions.

	Throughput requirement (Mbps)	Node size	Ratio of number of pairs to node size (%)
Figure 7	0.5 ~ 2	10 ~ 50	20
Figure 8	0.5 ~ 2	10 ~ 30	40
Figure 9	0.5 ~ 2	20	20 ~ 60

Table 3 Throughput requirement variation test

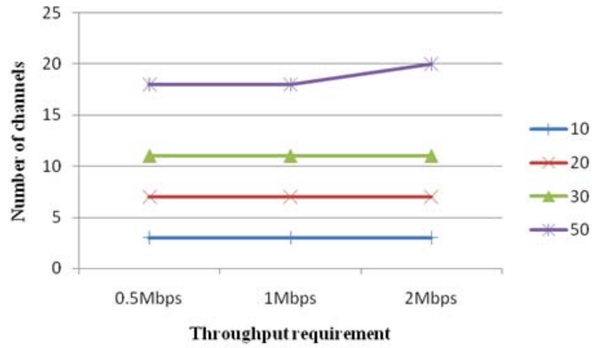


Figure 12 Effect of variation in throughput requirement (pairs = 0.2 * nodes)

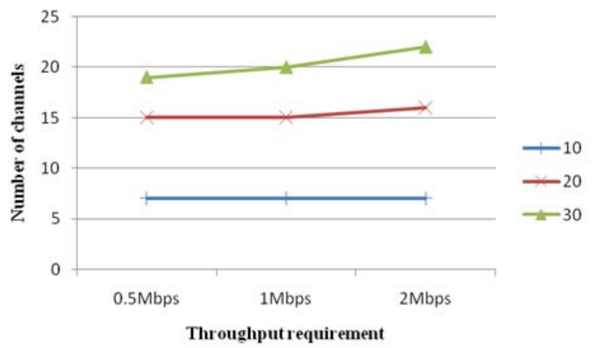


Figure 13 Effect of variation in throughput requirement (pairs = 0.4 * nodes)

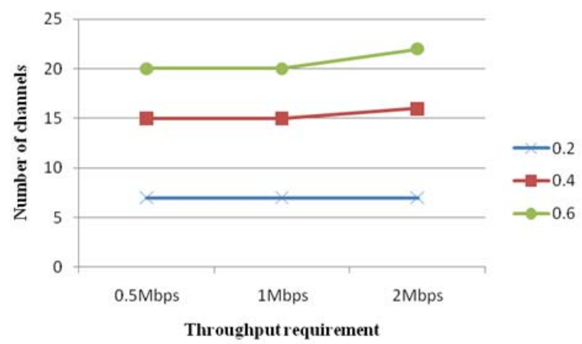


Figure 14 Effect of variation in throughput requirement (20 nodes)

Figure 10 and 11 show the average gap between the algorithm and CPLEX. M1, M2, and M3 indicate the number of given multipaths (1, 2, and 3, respectively). In Figure 10, the number of source-destination pairs is 20% of the number of nodes, while in Figure 11 it is 40%. The outstanding performance of the algorithm can be seen from these figures. Compared with CPLEX solutions,

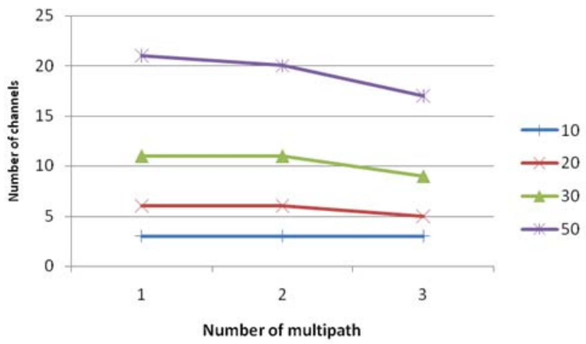


Figure 15 Effect of the number of given multipaths (pairs = 0.2 * nodes)

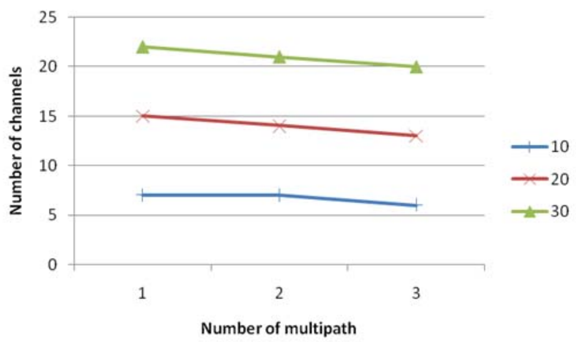


Figure 16 Effect of the number of given multipaths (pairs = 0.4 * nodes)

when the number of pairs is 20% of the number of nodes, the maximum average gap is 4.7%, whereas when the number of pairs is 40% of the number of nodes, the worst average gap is 4.8%.

From Figures 10 and 11, it is clear that the average gap increases as the number of nodes increases for single-path cases. However, the average gap decreases as the number of nodes increases for two and three given multipaths. For 30 and 50 nodes, the average gap for M1 is smaller than that for M2 and M3 in Figure 10, whereas in Figure 11 the average gap for M1 is greater than that for M2 and M3.

The reason for these results is the relationship between the complexity of the problems and the solutions yielded by CPLEX. CPLEX mostly fails to yield an optimal solution within the time limit when the problem becomes complex.

Next, we test the algorithm by varying the simulation environment with respect to throughput requirement and the number of given multipaths between source-destination pairs. To determine the effect of variation in throughput requirement, we keep other simulation parameters constant and vary the throughput requirements of source-destination pairs among 0.5, 1, and 2 Mbps. To determine the effect of the number of given multipaths, we keep other simulation parameters constant and vary the number of given multipath of source-destination pairs among 1, 2, and 3.

Figures 12, 13, and 14 represent the effect of different throughput requirements on the number of channels for different numbers of nodes and pairs, while keeping other parameters fixed. Table 3 shows the test conditions of Figures 12, 13 and 14. For example, in Figure 12, the throughput requirement is varied from 0.5 Mbps to 2 Mbps and the number of nodes is varied from 10 to 50, while the ratio of the number of pairs to the number of nodes is fixed at 1:5. From Figures 12, 13, and 14, we can see that the number of channels increases only slightly as the throughput, number of nodes, and number of pairs increase. This is because our algorithm employs link sharing and channel reuse. The flows in the network tend to share or reuse the same channel frequently, in order to reduce the total usage of channels in the network.

Figures 15 and 16 show the effect of the number of given multipaths on the number of channels. In Figure 15, the number of given multipaths varies from 1 to 3, the number of nodes varies from 10 to 50, and the ratio of the number of pairs to the number of nodes is 1:5. In Figure 16, the number of given multipaths varies from 1 to 3, the number of nodes varies from 10 to 30, and the ratio of the number of pairs to the number of nodes is 2:5.

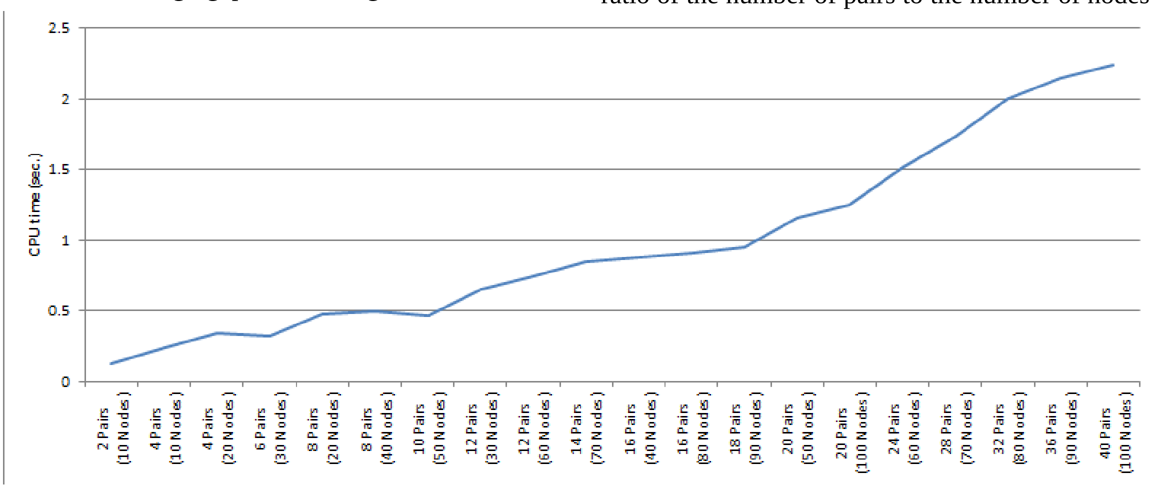


Figure 17 Time complexity of the algorithm (time vs. pairs)

From Figures 15 and 16, we can see that the number of channels decreases as the number of given multipaths increases. The network scenarios with three given multipaths provide more opportunity to choose routes that utilize link sharing and channel reusability than those with only one or two given multipaths. Note that we do not have to provide flows to all the given multipaths.

Figure 17 shows the time complexity of the algorithm for maximum 100 nodes. 20% and 40% of the node size are considered for the number of source-destination pairs with two given multipaths. From the figure, it is clear that the suggested algorithm is powerful to provide practical solution to the multipath selection and channel assignment in wireless mesh networks.

7. Conclusion

We have addressed the channel assignment problem of minimizing the number of channels while considering the throughput requirements of source-destination pairs in multichannel, multiradio, multirate wireless mesh networks. We formulated the channel assignment problem, which is a well-known NP-hard problem, through binary integer programming and proposed a novel algorithm for assigning channels. Specifically, the mathematical model considers such constraints as interference effect, link capacity, and throughput requirements and the algorithm takes two processes into account to obtain better solutions. First, we suggested a 'pair selection' process. When selecting a pair to allocate next, we refer to the previously allocated channel and data rate information to exploit the benefits of channel reusability and link capacity sharing. Second, we suggested a 'channel rearrangement' process. The allocation of one pair at a time iteratively may cause the solution to fall into a local optimum. Hence, in this process, we delete the previously allocated channel information and reallocate channels to the networks. From the analyses, it is evident that the number of channels decreases as the number of given multipaths increases and slowly increases as the throughput requirements of source-destination pairs increase. This is due to the factors that our algorithm considers. A greater number of given multipaths offers more opportunity to exploit the benefits of channel reusability and the sharing of link capacity. In addition, sharing link capacity alleviates the increment of throughput requirements of source-destination pairs.

The performance of the proposed algorithm was evaluated by running simulations of 24 different sets of network scenarios and comparing the results to the CPLEX solutions, which give nearly optimal values. CPLEX tends to fail to obtain optimal solutions when the problem sizes get bigger and, as a result, become complex. In these cases, we compared our results with the solutions that CPLEX obtained during 10,000 seconds. The simulation results verify that the proposed channel assignment algorithm is very effective.

References

1. Akyildiz, I.F., Wang, X., and Wang, W., (2005). Wireless mesh networks: a survey. *Computer Networks Journal (Elsevier).*
2. Bruno, R., Conti, M., and Gregori, E., (2005) Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, v.43(3), pp.123-131.
3. Sheriff, I., and Belding-Royer, E., (2006). Multipath Selection in Multi-radio Mesh Networks. *Broadnets 2006*, pp.1-11.
4. Raniwala, A., Gopalan, K., and Chiueh, T.-C., (2004). Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM Mobile Computing and Communications Review (MC2R)*, vol 8(2), pp.50-65.
5. Alicherry, M., Bhatia, R., and Li, L., (2006). Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Networks. *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 11, pp.1960-1971.
6. Tsai, J.W., and Moors, T., (2007). Interference-aware Multipath Selection for Reliable Routing in Wireless Mesh Networks. *Mobile Adhoc and Sensor Systems*, pp. 1-6.
7. Raniwala, A., and Chiueh, T., (2005). Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. *INFOCOM 2005*, Vol. 3, pp.2223 – 2234.
8. Johnson, D.B., and Maltz, D.A., (1996). Dynamic source routing in ad hoc networks. *Mobile Computing*, Vol.353.
9. Cidon, I., and Sidi, M., (1989). Distributed assignment algorithms for multihop packet radio networks. *IEEE Transactions on Computers*, vol.38, no.10, pp.1353-1361.
10. Ko, B.-J., Misra, V., Padhye, J., and Rubenstein, D., (2007). Distributed channel assignment in multi-radio 802.11 mesh networks. *IEEE Wireless Communications and Networking Conference*, pp.3978-3983.
11. Leung, R., Liu, J., Poon, E., Chan, A.L.C., and Li, B., (2001). MP-DSR: a QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks. *Proceedings of the 26th IEEE Annual Conference on Local Computer Networks*, pp.132-141.
12. Johnson, D.B., Maltz, D.A., and Broch, J., (2001). DSR: the dynamic source routing protocol for multihop wireless ad hoc networks. *Ad hoc networking*, pp.139-172.
13. Welsh, D.J.A., and Powell, M.B., (1967). An upper bound for the chromatic number of a graph and its

application to timetabling problems. *The Computer Journal* pp.85-86.
14. CPLEX 10.1, http://www.ilog.com/products/cplex
15. Marina, M.K., and Das, S.R., (2001). On-Demand Multi Path Distance Vector Routing in Ad Hoc Networks. *Proceedings of the Ninth International Conference on Network Protocols*, p.14-23.
16. Nasipuri, A., Casta, R., and Das, S.R., (2001). Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks. *Mobile Networks and Applications*, vol.6, no.4, pp.339-349.
17. Ramachandran, K.N., Belding, E.M., Almeroth, K.C., Buddhikot, M.M., (2007). Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. IEEE Infocom.
18. Prabhu, A., Gupta, H., Das, S., (2008). Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks. IEEE Transactions on Mobile Computing.
19. Lin, K.C.-J., Chou, C.-F., (2009). Exploiting multiple rates to maximize the throughput of wireless mesh networks. IEEE Transactions on Wireless Communications.
20. Bononi, L., Felice, M.D., Molinaro, A., Pizzi, S., (2009) A cross-layer architecture for efficient multi-hop communication in multi-channel multi-radio wireless mesh networks. IEEE SECON Workshops
21. Huang, R., Kim, S., Zhang, C., Fang, Y., (2009). Exploiting the capacity of multichannel multiradio wireless mesh networks. IEEE Transactions on Vehicular Technology.
22. Nasipuri, A., Das, S., (1999). On-demand multipath routing for mobile ad hoc networks. Computer Communication Networks.
23. Tsai, J., Moors, T., (2006). A Review of Multipath Routing Protocols: From Wireless Ad Hoc to Mesh Networks. Proc. ACoRN Early Career Researcher Workshop on Wireless Multihop Networking.

Problem	10 Nodes						20 Nodes						30 Nodes						50 Nodes					
	2 Pairs			4 Pairs			4 Pairs			8 Pairs			6 Pairs			12 Pairs			10 Pairs			20 Pairs		
	Algorithm	CPLEX	GAP	Algorithm	CPLEX	GAP	Algorithm	CPLEX	GAP	Algorithm	CPLEX	GAP	Algorithm	CPLEX	GAP	Algorithm	CPLEX	GAP	Algorithm	CPLEX	GAP	Algorithm	CPLEX	GAP
1	3 (0.12)	3 (0.02)	0	6 (0.22)	6 (0.01)	0	10 (0.24)	10 (0.17)	0	16 (0.42)	16 (0.19)	0	11 (0.30)	11 (0.09)	0	20 (0.55)	19 (0.36)	0.05	21 (0.46)	20 (0.33)	0.0476	34 (1.22)	31 (1.20)	0.08824
2	4 (0.11)	4 (0.01)	0	11 (0.24)	11 (0.16)	0	8 (0.29)	8 (0.06)	0	14 (0.31)	13 (0.22)	0.07143	16 (0.33)	15 (0.25)	0.0625	18 (0.56)	18 (0.25)	0	18 (0.59)	17 (0.25)	0.0556	29 (1.23)	28 (1.11)	0.0345
3	4 (0.11)	4 (0.01)	0	7 (0.21)	7 (0.03)	0	7 (0.15)	7 (0.05)	0	17 (0.35)	16 (0.23)	0.05882	13 (0.32)	13 (0.17)	0	26 (0.69)	24 (0.67)	0.0769	21 (0.48)	20 (0.45)	0.0476	33 (1.27)	32 (1.39)	0.0303
4	4 (0.12)	4 (0.02)	0	8 (0.22)	8 (0.05)	0	12 (0.22)	11 (0.11)	0.0833	12 (0.28)	12 (0.16)	0	12 (0.28)	12 (0.13)	0	27 (1.01)	25 (0.69)	0.0741	16 (0.49)	16 (0.35)	0	35 (1.38)	34 (2.45)	0.0286
5	3 (0.09)	3 (0.01)	0	7 (0.19)	7 (0.03)	0	10 (0.21)	10 (0.11)	0	13 (0.29)	13 (0.11)	0	15 (0.26)	14 (0.24)	0.0667	20 (1.05)	20 (0.58)	0	13 (1.02)	13 (0.41)	0	33 (1.59)	31 (2.63)	0.06061
Average	3.6	3.6	0	7.8	7.8	0	9.4	9.2	0.01666	14.4	14	0.02605	13.4	13	0.02584	22.2	21.2	0.0402	18	17.2	0.03016	32.8	31.2	0.04845
6	4 (0.11)	4 (0.13)	0	6 (0.22)	6 (0.44)	0	6 (0.34)	6 (0.41)	0	12 (0.23)	12 (1.51)	0	13 (0.35)	12 (7.41)	0.0769	21 (0.51)	20 (–)	0.0476	18 (0.43)	17 (–)	0.05556	24 (1.09)	24 (–)	0
7	3 (0.09)	3 (0.05)	0	7 (0.19)	7 (0.50)	0	6 (0.24)	6 (0.20)	0	7 (0.29)	7 (0.61)	0	8 (0.37)	8 (1.45)	0	22 (0.52)	22 (–)	0	19 (0.42)	17 (9.47)	0.10526	25 (1.35)	24 (–)	0.04
8	3 (0.12)	3 (0.08)	0	5 (0.25)	5 (0.25)	0	9 (0.38)	9 (0.84)	0	13 (0.43)	13 (2.08)	0	11 (0.35)	10 (0.94)	0.09091	18 (0.50)	18 (–)	0	20 (0.45)	19 (–)	0.05	28 (1.04)	28 (–)	0
9	5 (0.22)	5 (0.11)	0	6 (0.24)	6 (0.39)	0	11 (0.44)	10 (0.64)	0.0909	10 (0.68)	9 (4.11)	0.1	11 (0.24)	11 (0.56)	0	18 (0.58)	17 (8.23)	0.05556	15 (0.52)	15 (–)	0	30 (1.20)	31 (–)	−0.0333
10	2 (0.10)	2 (0.05)	0	8 (0.29)	7 (0.72)	0	11 (0.29)	11 (1.95)	0	14 (0.78)	13 (1.58)	0.07143	12 (0.30)	12 (1.97)	0	21 (0.53)	20 (683.14)	0.04762	16 (0.51)	17 (–)	−0.0625	31 (1.09)	30 (–)	0.0323
Average	3.4	3.4	0	6.4	6.4	0	8.6	8.4	0.01818	11.2	10.8	0.03429	11	10.6	0.03356	20	19.4	0.03015	17.6	17	0.02966	27.6	27.4	0.0078
11	3 (0.09)	3 (0.22)	0	7 (0.20)	7 (0.81)	0	5 (0.12)	5 (0.41)	0	14 (0.38)	13 (6.22)	0.0714	7 (0.37)	7 (2.31)	0	16 (0.59)	17 (–)	−0.0625	14 (0.48)	15 (–)	−0.0714	27 (1.12)	27 (–)	0
12	2 (0.11)	2 (0.09)	0	7 (0.19)	7 (0.33)	0	9 (0.34)	9 (5.97)	0	10 (0.57)	10 (4.56)	0	13 (0.38)	13 (–)	0	17 (0.45)	17 (–)	0	12 (1.15)	12 (–)	0	23 (1.24)	23 (–)	0
13	4 (0.20)	4 (0.28)	0	4 (0.27)	4 (0.36)	0	7 (0.28)	6 (0.50)	0.1429	11 (0.68)	11 (4.02)	0	13 (0.24)	11 (2.84)	0.15385	18 (1.02)	19 (–)	−0.0556	12 (0.43)	12 (–)	0	34 (1.07)	35 (–)	−0.0294
14	6 (0.15)	6 (0.11)	0	5 (0.22)	5 (0.63)	0	9 (0.35)	9 (1.06)	0	11 (0.45)	11 (39.00)	0	12 (0.36)	11 (3.56)	0.08333	15 (1.13)	14 (1114.22)	0.0667	18 (0.51)	17 (–)	0.0556	26 (1.09)	26 (–)	0
15	4 (0.12)	4 (0.31)	0	6 (0.23)	6 (0.95)	0	6 (0.21)	6 (1.00)	0	15 (0.59)	13 (3.70)	0.1333	10 (0.31)	10 (21.64)	0	22 (0.51)	20 (6.13)	0.09091	14 (0.48)	13 (–)	0.07143	23 (1.26)	22 (–)	0.0435
Average	3.8	3.8	0	5.8	5.8	0	7.2	7	0.02858	12.2	11.6	0.04094	11	10.4	0.04744	17.6	17.4	0.0079	14	13.8	0.01112	26.6	26.6	0.00282

Table 2 Simulation result

GAP = (Algorithm - CPLEX)/Algorithm

The number in parentheses represents the CPU seconds

The hyphen in the parentheses represents that CPLEX was terminated by the time limit of 10,000 seconds.