Load Balancing Routing in Multi-Channel Hybrid Wireless Networks with Single Network Interface

Jungmin So  Nitin H. Vaidya

Coordinated Science Laboratory, University of Illinois at Urbana-Champaign
Email: {jso1, nhv}@uiuc.edu

Abstract

A hybrid wireless network is an extension to an infrastructure network, where a mobile host may connect to an access point using multi-hop wireless routes, via other mobile hosts. The access points are configured to operate on one of multiple available channels. Mobile hosts and wireless routers can select its operating channel dynamically through channel switching. In this environment, we propose a routing protocol that finds routes to balance load among channels while maintaining connectivity. The protocol works with nodes equipped with a single network interface, which distinguishes our work with other multi-channel routing protocols that require multiple interfaces per node. The protocol discovers multiple routes to multiple access points, possibly operating on different channels. Based on traffic load information, each node selects the “best” route to an access point, and synchronizes its channel with the access point. With this behavior, the channel load is balanced, removing hot spots and improving channel utilization. The protocol assures every node has at least one route to an access point, where all intermediate nodes are operating on the same channel. Our simulation results show that the proposed protocol successfully adapts to changing traffic conditions and improves performance over a single-channel protocol and a multi-channel protocol with no load balancing.

1 Introduction

Wireless networks that are widely deployed for commercial use today are mostly single-hop infrastructure networks (wireless LANs). To access the Internet, a mobile host must be directly within range of an access point (AP) typically connected to the wired backbone network. Since the range of a single access point is limited, multiple APs are deployed to cover a large area. To reduce interference, neighboring APs are usually configured to operate on different frequency channels, since most wireless standards provide multiple non-overlapping frequency channels. For example, IEEE 802.11b provides three non-overlapping channels (1, 6 and 11), and IEEE 802.11a has 12 channels available [6].

There are several limitations to the single-hop infrastructure network architecture. First, it cannot handle unbalanced traffic load efficiently. In typical scenarios such as airports, traffic load is often spatially unbalanced. Places near gates can become hot spots when people are waiting for the plane. Second, for a large area, it can be expensive to deploy a large number of wireline-connected APs to cover the entire region.

Recently, researchers have proposed ideas to overcome these two limitations using multi-hop networking. For example, in [2], a mobile host in a hot-spot area can connect to an AP in the neighboring cell through another mobile host acting as a relay. Similarly, wireless mesh networks use wireless routers to cover a large service area without providing wired connectivity to a large number of APs [1].

In the multi-hop architecture, a node may potentially find multiple routes to different access points, possibly operating on different channels. Thus, each node should select the “best” route where it can achieve the best service quality. Since routes are on different channels, channel assignment is done with the route selection.

In general, nodes may have a single network interface, or multiple interfaces. Some routing protocols work only when every node has multiple interfaces [7, 11], but we assume that all nodes including access points, wireless routers and mobile hosts are equipped with a single network interface. The benefit of designing protocols that support nodes with a single network interface is that the protocol can be used for small, low-cost devices (e.g. cell phones, PDAs), connected to the wired backbone network. Since the range of a single access point is limited, multiple APs are deployed to cover a large area. To reduce interference, neighboring APs are usually configured to operate on different frequency channels, since most wireless standards provide multiple non-overlapping frequency channels. For example, IEEE 802.11b provides three non-overlapping channels (1, 6 and 11), and IEEE 802.11a has 12 channels available [6].
where equipping multiple interfaces can be costly.

With a single network interface, a node can only operate on one channel at a time. A node can switch its operating channel, but at the cost of channel switching delay. Due to the channel switching delay, frequent channel switching is expensive and thus not suggested. So in this paper, we consider a route as valid only if all nodes in the path are on the same channel.

To maximize channel utilization, the channels should be assigned so that traffic load is equally balanced among channels. However, the channel assignment problem is not trivial due to the following issues. First, the traffic load varies over time and is not known a priori. Second, the traffic load for a certain node depends on the number of hops from the node to its associated access point, because it determines how many times a packet is transmitted in order to achieve end-to-end throughput. Finally, channels should be assigned with the constraint that every node should have at least one route to an access point.

Estimating the traffic load accurately is critical in achieving channel load balancing and thus high channel utilization. In Section 2, we argue that traffic load observed locally by each node does not accurately reflect the actual load, and thus cannot be used as a base for selecting routes. Instead, load information should also be obtained from the APs. Also, when the load is measured at the AP, number of hops to the destination should be considered. Finally, when a node selects its primary route, local load information must be used to avoid route oscillation. We propose a new method for estimating the traffic load and selecting the best route according to the load information.

The routing protocol proposed in this paper addresses the issues mentioned above and achieves channel load balancing by dynamically selecting routes to an AP according to the current traffic condition in a distributed manner. Simulation results show that our proposed protocol successfully adapts to the changing traffic conditions and balances load among channels to achieve high channel utilization.

The rest of this paper is organized as follows. In Section 2, we explain the multi-channel multi-hop network architecture and discuss methods for estimating traffic load and selecting routes in this environment. After that, we describe our proposed routing and channel assignment protocol in Section 3. In Section 4, we report the results from simulations performed to evaluate the effectiveness of our proposed protocol. In Section 5, we review previous work that is relevant to our work in this paper. Finally, we conclude with directions for future research in Section 6.

2 Multi-Channel Hybrid Wireless Networks

The network of interest in this paper can be considered an extension to infrastructure networks, allowing nodes to connect with an access point via other nodes. An example network is illustrated in Figure 1. In the figure, solid lines indicate links on channel 1, and dashed lines indicate links on channel 2. The dotted line indicates that there is a potential link between D and E, if their channels match each other. In this example, nodes A, B, C, and D are associated with AP1 on channel 1, and nodes E and F are associated with AP2 on channel 2. Nodes C and D cannot reach an access point directly, but they are connected via multiple wireless hops.

![Figure 1. An example hybrid wireless network. Solid lines are links on channel 1, dashed lines are links on channel 2, and the dotted line indicates a potential link, if D and E were on the same channel.](image-url)

In the figure, node D is currently on channel 1, and is associated with AP1. However, if D switches its channel to channel 2, it can associate with AP2 via node E. Once D associates with AP2, node B and C can also connect to AP2 via D and E.

Since node D has two potential routes it can use, it must choose the route where it can achieve a better quality of service. The quality of service at a node is affected by current traffic load on the channel and the quality of links on the route. In this paper we mainly focus on the traffic load when selecting routes. Considering link quality as a factor in load metric can improve the accuracy of the metric, but it is outside the scope of this paper and left as a future work.

In Figure 1, node D chooses the route with less traffic load. In order to do that, D must know the load on its current channel as well as other channels. Thus, we discuss how to estimate traffic load in the following.

2.1 Estimating traffic load

Before discussing how to estimate traffic load, we state our assumptions. First, although a node may have multiple routes to the access point, only one route is used at any given time, and other routes are maintained for backup so that they can be used when the primary route fails. For example, in Figure 1, node D only uses the route through node B to connect to the wired network (this route is called the
The primary routes of nodes associated with the same AP form a route tree, rooted at the AP. Second, we assume that most of the traffic is downlink traffic (e.g., accessing web data), sent from AP to mobile hosts. The proposed protocol supports uplink traffic as well, but the load estimation is based on the downlink traffic. Third, we assume that APs are placed dense enough that most routes are short in terms of number of hops, such as 3 or 4 hops (similar assumptions are made in other works [9, 16]). Thus, there is little chance spatial reuse within a route tree. When the depth of a route tree becomes large, spatial reuse must be taken into account in the load estimation. It will be addressed in the future work. Finally, we assume that as in single-hop infrastructure networks, neighboring APs are typically assigned different channels. So it is unlikely that a node finds short routes to two different APs that are on the same channel.

To discuss how to estimate traffic load, we refer to Figure 1 again. In the figure, node D can associate with AP1 on channel 1 or AP2 on channel 2. It is better for node D to connect to the route tree with less amount of load. There are several ways of estimating the load of a route tree, as described in the following.

**Locally measured load:** Each node measures the amount of traffic it has received or forwarded during a recent time window, and exchange the information with neighbors. For example, during last 10 seconds, the average traffic load that node B has received or forwarded traffic is 500Kbps, and the average traffic load that node E has received or forwarded is 100Kbps. But using this information, D cannot tell if E is receiving 100Kbps because that is the total load on the AP2-tree, or E is only receiving 100Kbps because AP2 is busy forwarding traffic to other nodes.

**AP-measured load:** Since all traffic go through the AP, the load of the route tree can be estimated as the load at the AP. For example, during last 10 seconds, AP1 observes that it has forwarded 2Mbps of traffic, and AP2 observes that it has forwarded 3Mbps of traffic. D can use this information as a basis for selecting a route tree. However, this metric is not enough in estimating the load of a route tree, because it does not consider how many hops a packet needs to travel. If all traffic of AP1 was for node C, then every packet has to be transmitted twice, which means the actual load on AP1-tree is 4Mbps.

**AP-measured weighted load:** The load is measured at the AP, but the load is weighted according to the hop distance from the AP to the destination. We use this metric for load measurement in our protocol. The details of how the load is measured at the AP and how the load information is distributed is explained in Section 3. Next we discuss how a node should select routes based on this load information.

### 2.2 Selecting the route with minimum load

Suppose a node obtains load information on all its potential routes to destinations. When does a node decide to switch channels and join another route tree? A node cannot freely switch channels because it might have child nodes in the route tree. Consider the scenario in Figure 2. Initially node D is associated with AP1, and so is node G. Suppose AP1 has 1Mbps of traffic for node A, 1Mbps for node D and 1Mbps for node G. Also, AP2 has 1Mbps for node F. Node D has to decide whether it should switch to channel 2.

![Figure 2. An example network scenario. This example indicates that subtree load must be considered when selecting the best route.](image)

Using the weighted load metric, the load of AP1-tree (the route tree rooted at AP1) is 6Mbps (1Mbps for A, 2Mbps for D, and 3Mbps for G), and the load of AP2-tree is 1Mbps. If only node D switches to channel 1, the load of AP1-tree will become 4Mbps, and the load of AP2-tree will become 4Mbps (1Mbps for G and 3Mbps for D). So this suggests that D should switch to channel 2. However, it will lead to node G being disconnected from the network. So when D decides to switch channels, all its descendants in the route tree must also switch channels (G might have another route to the AP on channel 1, but D does not know this). But if D and G switch together, load of AP2-tree becomes 8Mbps, and thus D should decide not to switch. This example indicates that a node D decides to move from AP1-tree to AP2-tree only when the current load of AP1-tree is larger than the current load of AP2-tree plus the load of the subtree rooted at node D weighted according to the number of hops in the AP2-tree.

### 3 Proposed Routing Protocol

In this section, we describe our routing and channel assignment protocol in detail. As mentioned earlier, we as-
assume that all nodes in the network communicate via access points, and not with each other directly. Whenever two mobile nodes need to communicate, they can use their routes through APs. So it is enough that each node maintains at least one route to an access point, and routes to all the descendant nodes in the route tree. The AP must maintain routes to all the nodes associated with the AP.

The routing protocol must answer the following questions: (1) How are the routes established? (2) How are the routes maintained and updated? (3) How are the routes recovered after failures? In the following, we describe how the proposed protocol addresses these issues. Due to space limitations, we only describe the route establishment and route maintenance. Also, we have omitted some details. Fully described details of the protocol including route recovery scheme are in [14].

3.1 Route establishment

When a node is turned on, it must first discover a route to an access point. For this purpose, the node performs an "active scanning" on all channels. Consider the scenario in Figure 3. There are two APs, operating on channel 1 and channel 2, respectively. Before node B joins the network, node A is already in the network, associated with AP1 on channel 1. Now node B joins the network. Initially, node B selects a random channel, and starts scanning by broadcasting a SCAN message on the channel. After sending the SCAN message, node B waits on the channel for some time to collect reply packets and then moves on to the next channel, and eventually scans all channels in a round-robin manner.

Figure 3. An example network scenario. Node B joins the network after node A.

Access points and nodes that are already associated with an access point can reply to the SCAN message by unicasting a REPLY message back to the sender, node B in our example. The REPLY message contains the address of the replier, the address of the AP that the replier is associated with, and the number of hops to the AP. In the above scenario, node A replies to SCAN on channel 1, and AP2 replies on channel 2. Since there can be multiple neighbors replying on a channel, nodes wait for a random delay before sending the REPLY message.

After scanning all channels, node B selects its primary route by choosing one of its neighbor as its parent node. Among all the routes received, B selects the route with the minimum load according to the weighted load metric explained in Section 2. If there is a tie, the one with the minimum number of hops is chosen. In the above example, B selects AP2 as its parent node. Once a node selects its primary route, the path from the node to the AP is established. Then, node B sends an ASSOCIATION message using its primary route, so that a reverse path is set up from the AP to node B.

The route table that each node maintains is similar to that of AODV [10], with some changes in the route entry. An example route table is shown in Figure 4.

![Figure 4. An example route table and its corresponding network topology. Node B has two routes to AP1, and a route to node C. Between the two routes to AP1, the route via node A is selected as the primary route.](image)

The fields in the route entries that are not in the route entries of AODV are type, channel, load and path. The type indicates the node type of the destination: whether it is an AP, or a mobile node. Among routes to APs, one route is selected as primary route, which has “PRIM” under the type field. The channel indicates which channel the route uses. The load field will be explained later. Finally, instead of sequence numbers used in AODV, the entire path information is recorded in the route entry to prevent route loops when nodes update their routes.

3.2 Route management and updates

Managing and updating routes is the most important part of our proposed protocol. Once the primary route has been established, each node collects load information for its own route tree and other route trees. Based on this information, the node may switch to the route tree with minimum load so that it can obtain the highest quality of service possible. First, we describe how the load is measured at the APs.
Next, we explain how the load information is collected by the nodes. Finally, we present the process of route update.

3.2.1 Measuring load

In Section 2, we have decided to use the weighted load metric. Here we present the detailed description of how the load information is collected and distributed. Note that the protocol performs load balancing based on the downlink traffic, because we assume that downlink traffic is much more dominant than uplink traffic. Although not considered in estimating load, the protocol supports uplink traffic as well as downlink traffic.

Each AP remembers the amount of traffic it has received during past $T$ seconds. In the simulations, we have used 10 seconds as $T$. The packets counted as traffic are the ones that are from wired network to a node in the route tree rooted at the AP. Since the AP knows the destination, it records the amount of traffic per destination. For example, let us consider Figure 4(a) again. Suppose during last $T$ seconds, AP1 has received 100Kbps of load for node D, and 200Kbps of load for node B. The AP1 records this information in its route table as in Figure 5.

![Figure 5. Route table of AP 1 in the scenario shown in Figure 4(a).](image)

The weighted load metric indicates that the load for each destination should be weighted by the number of hops from AP to the destination node. So the weighted load of the route tree $L_1$ is computed as follows.

$$L_1 = \sum_i (h_i \times l_i)$$  \hspace{1cm} (1)

where $i$ is a node in the route tree rooted at the AP, $h_i$ is the number of hops, from AP to the node, and $l_i$ is the amount of traffic destined for the node. So in the above example, the total load of AP1-route tree is 500Kbps.

3.2.2 Distributing and collecting load information

In Section 2, we have discussed how a node selects a route tree. To make the decision, a node should obtain load information on its own route tree, other route trees, and the amount of traffic destined for the node itself and nodes in its subtree. To allow each node to obtain the load information of its subtree, the AP piggybacks the load information in the data packet. For example, in scenario shown in Figure 4(a), AP1 observes that 200Kbps of traffic has been received to be delivered to node B during last $T$ seconds. Then AP1 sends 200Kbps with the data packet along the route. The intermediate nodes and the destination node record the information on their route table. So in the example, node A records 200Kbps in the route entry that has node B as destination, and node B records 200Kbps as a separate variable name “LOAD” in its route table.

Now a node has to obtain information on other route trees. Periodically, each AP transmits a HELLO message which includes the load information measured using weighted load metric. Similar to the scanning process, HELLO messages are sent on all channels, one at a time. When nodes receive the HELLO message, they update their route table according to the information given in the message (as explained later). After that, only if the sender of the HELLO message is the next hop node in its primary route, the node forwards the HELLO message. Otherwise the packet is discarded. To avoid collision among nodes that transmit HELLO messages at the same time, each node waits for a short random delay before sending its HELLO message. In this manner, the HELLO messages are initiated by the APs and forwarded along the route tree. We call the period for sending HELLO messages $P_{hello}$. $P_{hello}$ must be long so that it does not induce much overhead on the network. In the simulations, we have used 3 seconds as the $P_{hello}$. To avoid synchronized HELLO period among APs, each AP randomly picks the next HELLO time between the range [1.5-4.5].

Since the HELLO messages are sent on all channels, a node can receive HELLO from all the neighbors including those on other route trees. When a node receives a HELLO message, it first checks whether the HELLO message carries a new route to an AP through the sender. If so, then the new route is recorded in the node’s route table. Then the node updates load information for the route tree of the sender. Consider the following scenario in Figure 6.

In Figure 6, node D is initially associated with AP2 on channel 2. AP2 has observed that 100Kbps of load is for node D and 100Kbps of load is for node C. As the data packet is forwarded, D obtains load information of itself and node C. When AP2 broadcasts a HELLO message, D learns that the load of its route tree is 1500Kbps. Now at some point of time AP1 starts HELLO process. Node B receives the HELLO message and rebroadcasts it on all channels. When B transmits the HELLO packet on channel 2, node D receives the packet. Now D finds out that B is associated with AP1, and is 1 hop away from AP1. So D obtains a backup route to AP1 on channel 1, through B. In the HELLO message, B includes the load of its route tree, which is 300Kbps. So after receiving the HELLO packet
and updating its route table, the route table of node D looks like Figure 7.

![Figure 6. An example network scenario to illustrate the process of obtaining route information and selecting primary route based on load information.](image)

![Figure 7. Route table of node D in the scenario shown in Figure 6.](image)

Using this information, node D can now decide if it should switch to the other route tree.

### 3.2.3 Switching route trees for load balancing

Once the necessary load information is obtained, nodes can decide whether to switch to other route trees. In the example shown in Figure 6, node D can switch its channel to channel 2 and re-associate with AP1, because it has a lower load. When making the decision, the node compares the current load of its route tree and the load of the other tree after the node joins. Since node D has children in the route tree, it cannot just switch channels to join other trees, because the child nodes will lose connections with the AP. Instead, if D decides to switch channels, it should tell all its children to switch channels as well. Effectively, the whole subtree moves to the new route tree. So the load information should be computed correspondingly.

For example, in Figure 6, suppose node D wants to decide if it should move to AP1. The current load of AP2-tree is 1500Kbps. Now the load of the other tree should be computed as

\[
L_{AP1} = L_{AP1} + \sum_i (h_i \times l_i)
\]

where \(L_{AP1}\) is the load of AP1-tree after node D joins the tree, \(L_{AP1}\) is the load of AP1-tree before node D joins the tree, \(h_i\) is the node in the subtree rooted at node D, \(l_i\) is the number of hops from node \(i\) to AP1, and \(l_i\) is the load destined for node \(i\). In the above example, the load of AP1-tree after the subtree of D joins the tree is computed as

\[
L_{AP1} = 300 + (100 \times 2 + 100 \times 3) = 800.
\]

Since it is still smaller than the current load of AP2-tree, node D can decide to switch channels so that it can join AP1.

Even if node D observes that AP1 has lower load than AP2, it does not immediately move to AP1, because the decision can be based on out-of-date information. Also, reacting immediately can cause route oscillations, because multiple nodes can switch back and forth causing the traffic load to oscillate between two route trees. Instead, if node D observes that AP1 has lower load for sufficiently long time, it can decide to switch channel with confidence that it is based on up-to-date information. The duration of time a node waits before it switches route trees is a tunable parameter. We denote it as \(T_{switch}\) and we use \(T_{switch} = 10\) seconds in the simulations.

Once node D decides to switch channels, it first sends a SWITCH message to all its child nodes. The SWITCH message includes the new AP, number of hops from node D to the AP, and the new channel. The SWITCH packet is forwarded down the tree, and all children of node D switch their channels and update their route entries for the primary route. After sending the SWITCH packet, node D associates with the new AP by sending ASSOCIATION message on the new route. The ASSOCIATION message includes the previously associated AP, which is AP2 in this case. When AP1 receives the ASSOCIATION message, it informs AP2 through wired backbone network that node D has left AP2. All children of node D go through the same process to associate with the new AP.

### 4 Performance Evaluation

We have performed simulations using the ns-2 simulator [17] to evaluate the performance of the proposed protocol. Due to space limitations, we briefly report and discuss some results here. More extensive experimental results are presented in [14].

A simulation area of 1000m \(\times\) 1000m is divided into 4 quadrants, and APs are placed at the center of each quadrant. All AP are assigned different channels. 16 nodes were randomly placed in each quadrant, making a total of 64 nodes. To create an unbalanced traffic pattern, the 16 nodes in the northeast quadrant were selected as destinations that receive traffic from the AP. Under this scenario, we have compared two protocols. The first one which we call “MCP” assigns channels randomly and selects routes based on number of hops, without any consideration of traffic load. The second one is “MCP-LB”, which is the proposed protocol.
In Figure 8(a), the aggregate throughput of MCP and MCP-LB are shown. Since nodes associate with closest AP in the MCP, only one channel is used and other three channels are wasted. So the throughput is limited at 1Mbps. However, MCP-LB redirects nodes to other APs to improve performance. Figure 8(b) shows the throughput achieved per AP. AP4, which is placed in upper-right quadrant where all destination nodes are placed, achieves a throughput of 1Mbps, because all destination nodes are in one-hop range of the AP. For AP1 and AP3, the throughput is around 40% of AP4. This indicates that the average number of hops the nodes connecting to these APs is approximately 2.5. Finally the throughput of AP2 is the least among APs. Since AP2 is placed far away from AP4, nodes have to travel approximately 5 hops to communicate with AP2. Although the throughput of the APs is different, the proposed protocol regards this as balanced, because it uses the weighted load metric, multiplying number of hops to the actual load for a node.

To study the adaptive behavior of our proposed protocol, we have simulated a scenario with dynamically changing traffic pattern. During 400 seconds of simulation time, we simulated 32 flows, one flow starting at every 10 seconds. The destination nodes were only selected from upper-right and lower-left quadrant. We plotted aggregate and per-AP throughput for MCP-LB and MCP. The result is shown in Figure 9. Comparing the two protocols, we can see that the proposed protocol utilizes all 4 APs by redirecting nodes to other APs, whereas with MCP, throughput of two APs are kept at zero. As a result, MCP-LB achieves significantly higher aggregate throughput than MCP.

5 Related Work

There has been vast amount of effort in the research community to improve performance of wireless networks. One research direction that has gained increasing attention recently is to utilize multiple channels to improve network performance. In this section, we review and summarize the previous work on multi-channel routing protocols and load balancing techniques, that are relevant to our work in this paper.

Many routing protocols have been proposed for multi-hop networks, that supports only a single-channel [13]. Recently, routing protocols have been proposed for multi-channel multi-hop networks, that combine channel assignment and routing so that multiple channels can be utilized without changing the MAC layer protocol. Draves et al. [3] proposed a metric for route selection in multi-channel network, which assumes that each node has the number of interfaces equal to the number of available channels. So et al. [15] proposed a routing protocol for multi-channel ad hoc networks that works with nodes equipped with a single network interface. Since this protocol is proposed for ad hoc networks, the route discovery and maintenance is more complicated compared to our protocol. Kyasanur et al. [7] proposed a routing protocol that requires multiple network interfaces per node, where the number of interfaces does not need to equal the number of available channels. Raniwala et al. [12, 11] proposed a multi-channel routing protocol that also requires multiple interfaces per node.

Finally, we review the load balancing techniques proposed for wireless networks. Hsiao et al. [5] at el. proposed a load balancing algorithm for wireless access networks. The protocol builds a backbone tree rooted at the APs, similar to our proposed protocol. However, the protocol assumes that each node knows its load and the load information is reported to the AP. Our protocol do not assume that the load information is known. Hassanein et al. [4] proposes to use as the number of “active” paths in the neighborhood as the load metric. Also, Lee et al. [8] use the number of packets buffered in its interfaces as the load metric.
6 Conclusion and Future Work

In this paper, we have proposed a routing protocol for multi-channel hybrid networks that works for nodes equipped with a single network interface. The protocol ensures that every node in the network has at least one route to an AP, while allowing nodes to switch channels to associate with an AP with minimum load. We have argued that load should be estimated considering what is observed at the AP and also the number of hops to the destination. Using the weighted load metric, we have proposed a routing protocol that balances load among channels. The simulation results show that our proposed protocol can successfully reduce congestion in hot-spots and avoid wasting channel bandwidth due to unbalanced traffic load.

Our proposed protocol has several limitations. First, the proposed protocol only considers downlink traffic when measuring load. If there is significant amount of uplink traffic, it will result in incorrect measurement and unbalance in the channel load. Second, the protocol only considers traffic load and does not take into account the varying channel conditions due to other environmental factors. Third, the protocol assumes that neighboring APs are assigned different channels, so balancing load among APs can be converted to balancing the channel load. Although it is true that neighboring APs are unlikely to be assigned the same channel, it may not be necessarily true. If two route trees that are close by are on the same channel, the load balancing method of the proposed protocol will result in higher load in this channel. In this case, a node may need to consider the combined load of the two trees as the channel load. Finally, the load metric proposed in this paper assumes that only one transmission take place at a time in the same route tree. This is not true if the depth of a route tree becomes large. To consider this, the weighted load metric can be changed so that instead of multiplying the amount of traffic with the number of hops the traffic needs to travel, it can use a different coefficient so that the possibility of spatial reuse is considered. All of these issues are directions for our future research.

References