ISSN 2250-0987
Prabakaran et al, UNIASCIT, Vol 2 (1), 2012, 207-215

ACCESS POINT ASSOCIATION IN MCMH WIRELESS LAN

Dr.M.Prabakaran

Assistant Professor,
Department of Computer Science,
Government Arts College, Ariyalur, Tamil Nadu, India.

Abstract: **In this paper, we present a routing and channel assignment protocol for Multi-Channel Multi-Hop (MCMH) Wireless LANs (WLANs). We view a multi-hop network as an extension to infrastructure networks, where a Mobile Node (MN) may connect to an AP using multi-hop wireless routes, via other MNs or WRs. MNs and WRs can select its operating channel dynamically through channel switching. In this, we propose a routing protocol that finds routes and assigns channels 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. Supporting nodes with single network interface is beneficial because having multiple interfaces can be costly for small and cheap devices. The protocol discovers many possible routes to multiple APs, possibly operating on different channels. Based on traffic and load information, each node selects the "best" route to an AP, and synchronizes its channel with the AP. With this behavior, the channel load is normalized, removing hot spots and improving channel utilization. The channel assignment assures every node has at least one route to an AP, 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 protocol with random channel assignment.**

I. INTRODUCTION

WLANs that are widely deployed for commercial use today are mostly single-hop infrastructure networks (WLANs). To access the Internet, a MN must be directly within range of an AP typically connected to the wired backbone network. Since the range of a single AP 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 nonoverlapping frequency channels. For example, IEEE 802.11b provides three non-overlapping channels (1, 6 and 11), and IEEE 802.11a has 12 channels available [5]. 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 [1], a MN in a hot-spot area can connect to an AP in the neighboring cell through another MN acting as a relay. Similarly, wireless mesh networks use WRs to cover a large service area without providing wired connectivity to a large number of APs [6]. In the multi-hop architecture, a node may potentially find multiple routes to different APs, 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 APs, WRs and MNs 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), 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 AP, 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 AP.

Estimating the traffic load accurately is critical in achieving channel load balancing and thus high channel utilization. In Section II, 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 and channel assignment protocol proposed in this paper addresses the issues mentioned above and achieves channel load balancing by dynamically assigning channels according to the current traffic condition. Channel assignment is done in a distributed manner, as each node selects its operating channel according to its observed load information. Simulation results show that our proposed protocol successfully adapts to the changing traffic conditions and balances load among channels to achieve high channel utilization. Thus, the contributions of this paper are the followings:

- A metric for estimating the current traffic load and a method for selecting the best route based on the load information.
- A routing and channel assignment protocol to achieve high performance in MCMH WLANs with nodes equipped with single network interface.

The rest of this paper is organized as follows. In Section II, we explain the MCMH 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 III. In Section IV, we report the results from simulations performed to evaluate the effectiveness of our proposed protocol. In Section V, we review previous work that is relevant to our work in this paper. Finally, we conclude with directions for future research in Section VI.

II. MULTI-CHANNEL MULTI-HOP WLANS

A MCMH WLAN of interest in this paper can be considered an extension to infrastructure networks, allowing nodes to connect with an AP via multiple hops. 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 AP directly, but they are connected via multiple wireless hops. Note that a “node” can be a MN or a WR. MNs are end-user devices, and WRs are simple routers with only wireless interfaces, and they act as intermediate nodes to relay packets.

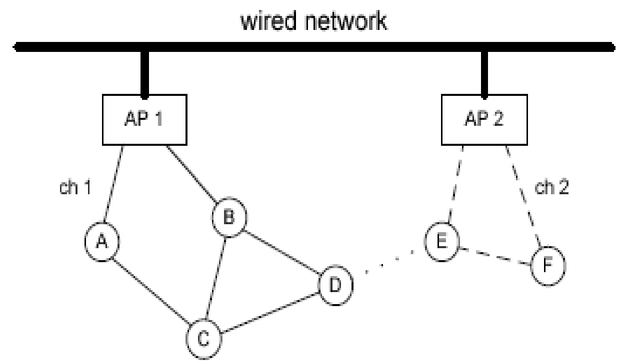


Fig. 1. An illustration of a MCMH WLAN. Solid lines are links on channel 1, and dashed lines are links on channel 2. The dotted line indicates a potential link, if node D and E were on the same channel.

WRs are always willing to relay packets, whereas MNs may or may not volunteer to relay packets of other MNs. In the proposed protocol described in Section III, MNs that are not willing to relay packets of other hosts do not participate in the protocol to send HELLO messages or reply to SCAN messages (details explained later). Coming back to Figure 1, consider node D. It 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 including current traffic load on the channel and the quality of links on the route affected by environmental factors. 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. 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 subsection.

A. Estimating traffic load

Before discussing how to estimate traffic load, we state our assumptions. First, although a node may have multiple routes to the AP, 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 or becomes congested. 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 *primary route*. 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 MNs. The proposed protocol supports uplink traffic, 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 for simultaneous transmissions within a route tree due to simultaneous transmissions.

When the depth of a route tree becomes large, spatial reuse must be taken into account in the load estimation. It will be address 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. Thus, we convert channel load balancing problem into balancing load among route trees. In Section VI, we revisit these assumptions and discuss problems that arise if these assumptions do not hold, and suggest ways to address the issues. To discuss how to estimate traffic load, we refer to Figure 1 again. Currently node D is connected to AP1 via node B. D has another route to AP2 via node E, but it is not used presently. Suppose each node exchanges its traffic load information via control messages (the protocol details are explained later).

So D obtains load information from B, C, and E. What would be the metric that nodes should use to communicate the load information? First, each node can measure the number of bytes it has received or forwarded during a recent time window. For example, during last 10 seconds, the average traffic load that node B has received or forwarded traffic is 500 Kbps, and the average traffic load that node E has received or forwarded is 100 Kbps. Does this information suggest that node D should switch to channel 2 and join AP2 route tree? The answer is no, because E does not know if it is receiving 100 Kbps because that is the total load on the channel, or it is only receiving 100 Kbps of traffic because AP is busy forwarding traffic to other nodes. So locally measured load cannot be used as basis for selecting routes.

Another metric that can be used is the load measured at the AP. Since all the traffic destined to the nodes associated with the AP goes through the AP, it can accurately measure the load on its route tree. We assume that the bandwidth of the wired backbone that the APs are connected to is much larger than the bandwidth of wireless links. Suppose AP1 observes that during last 10 seconds, it has forwarded 2Mbps of traffic. Also, AP2 has forwarded 1Mbps of traffic. If D obtains this information, D knows that AP2 has a lighter load than AP1. However, the AP-measured load is still not an accurate measure that can be used in selecting routes. Consider the scenario in Figure 2. Currently, D is associated with AP1 on channel 1, via node B. Suppose AP1 has 2Mbps of traffic destined for node A, and AP2 has 1Mbps of traffic destined for node F. If node D obtains this information, does this suggest that node D should switch to channel 2 and connect to AP2? The answer is no. Since each packet needs to be transmitted three times to reach node F, the actual load on the route tree is 3Mbps instead of 1Mbps (recall that due to small depth of the tree, two transmissions in the same route tree are assumed to interfere with each other). So it is better for D to stay on channel 1.

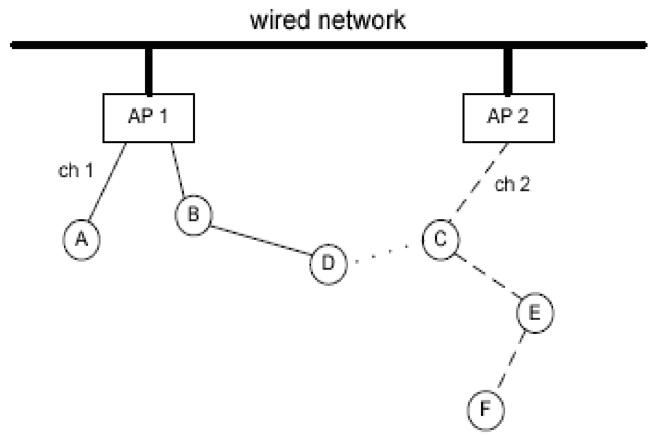


Fig. 2. An example network scenario.
This example indicates that number of hops must be considered in measuring the load. The solid lines are links on channel 1, and the dashed lines are links on channel 2.

This example indicates that the load should be weighted according to the number of hops to the destination. We call this new metric the *weighted-load* metric, we use this metric for load measurement in this paper. The specific details of how the load is measured at the AP and how the load information is distributed is explained in Section III. Next we discuss how a node should select routes based on this load information.

B. 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? This subsection discusses this issue. A node cannot freely switch channels because it might have child nodes in the route tree. Consider the scenario in Figure 3. 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. If node D obtains this information, should node D switch to channel 2? Using the weighted load metric, the load of AP1-tree (the route tree rooted at AP1) is 6 Mbps (1 Mbps for A, 2 Mbps for D, and 3 Mbps for G), and the load of AP2-tree is 1Mbps. If only node D can switch to channel 1, the load of AP1-tree will become 4 Mbps, and the load of AP2-tree will become 4 Mbps (1 Mbps for G and 3 Mbps 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 (node G might have another route to the AP on channel 1, but D does not know this). But if D and G switches together, load of AP2-tree becomes 8Mbps, and thus D may decide to stay on channel 1.

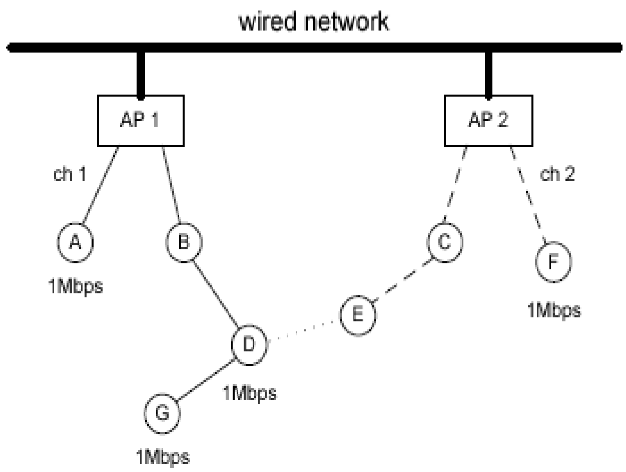


Fig. 3. An example network scenario. This example indicates that sub tree load must be considered when selecting the best route.

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 sub tree rooted at node D weighted according to the number of hops in the AP2-tree. If the current load and the load after D moves is equal, tie is broken using number of hops from D to the AP. A node may decide to switch its primary route within the tree (i.e. without switching channels or associating with another AP). This happens when the primary route has larger hops from the AP than the alternative route. Then the weighted load after the node switches its primary route will be smaller than the current load. So the weighted load metric prefers routes with smaller hops. Formal descriptions of how a node selects its primary route are presented in Section III.

III. PROPOSED ROUTING AND CHANNEL ASSIGNMENT PROTOCOL

In this section, we describe our routing and channel assignment protocol in detail. As mentioned earlier, we assume that all nodes in the network communicate via APs, 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 AP, 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:

- How are the routes established?
- How are the routes maintained and updated?
- How are the routes recovered after failures?

In the following subsections, we describe how the proposed protocol addresses these issues. Due to space limitations, we only describe the route establishment and route maintenance here. Also, we have omitted some details. For fully described details of the protocol including route recovery scheme, refer to [14].

A. Route establishment

When a node is turned on, it must first discover a route to an AP. For this purpose, the node performs an *"active scanning" on all channels*. Consider the scenario in Figure 4. 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 (as shown in Figure 4(a). Now node B joins the network as in Figure 4(b). 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 responses and then moves on to the next channel and eventually scans all channels in a round-robin manner.

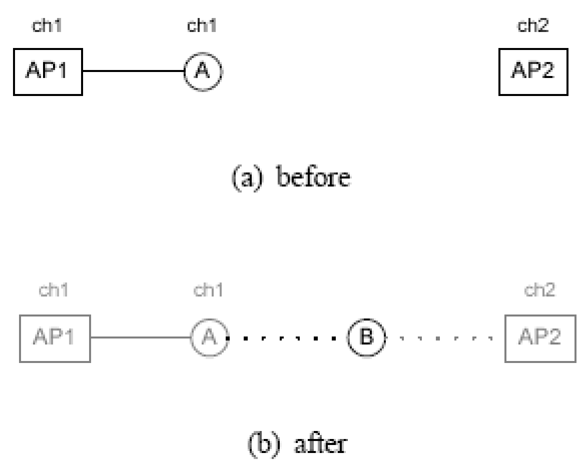
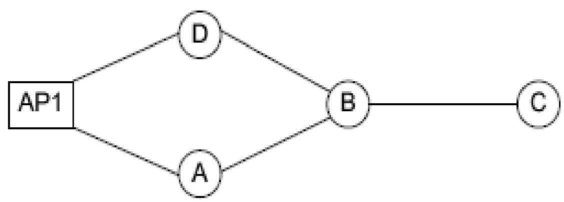


Fig. 4. A simple network scenario with two APs and two nodes.

APs, and nodes that are already associated with an AP 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 associate 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 neighbors as its *parent* node. Among all the routes received, B selects the route with the minimum load according to the weightedload metric explained in Section II. 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 the selected 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 5.



Route Table of B

dst	nexthop	hops	type	chan	load	path
AP1	A	2	PRIM	1	500	A AP1
AP1	D	2	AP	1	500	D AP1
C	C	1	MH	1	0	

Fig. 5. 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.

In the topology shown in Figure 5(a), node B has two routes to AP1, and a route to node C. Between the two routes to the AP, node B has chosen the route via node A as its primary route. 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.

B. 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.

1) Measuring load: In Section II, we have suggested to use the *weighted load* was the suitable measure. 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 the 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 second 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 5(a) again. Suppose during last *T* seconds, AP1 has received 100Kbps of load for node D, and 200 Kbps of load for node B. The AP1 records this information in its route table as in Figure 6.

Route Table of AP1

dst	nexthop	hops	type	chan	load	path
A	A	1	MH	1	0	
B	A	2	MH	1	200	
C	A	3	MH	1	0	
D	D	1	MH	1	100	

Fig. 6. Route table of AP 1 in the scenario shown in 5(a).

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.

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

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

2) Distributing and collecting load information: How a node makes decision on which route tree to stay on was explained in Section II. 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 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 5(a), AP1 observes that 200Kbps of traffic has been received to be delivered to node B during last *T* seconds. Then AP1 sends 200 Kbps with the data packet along the route. The intermediate node and the destination node records the information on their route table. So in the example, node A records 200 Kbps in the route entry that has node B as destination, and node B records 200 Kbps as a separate variable name "LOAD" in its route table. Now a node has to obtain information on 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 *Phello*. *Phello* must not be long enough to reduce overhead on the network. In the simulations, we have used 3 seconds as the *Phello*. 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 that the sender is on. Consider the following scenario in Figure 7.

In Figure 7, node D is initially associated with AP2 on channel 2. AP2 has observed that 100 Kbps of load is for node D and 100 Kbps 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 1500 Kbps. Now at some point of time AP1 starts HELLO process.

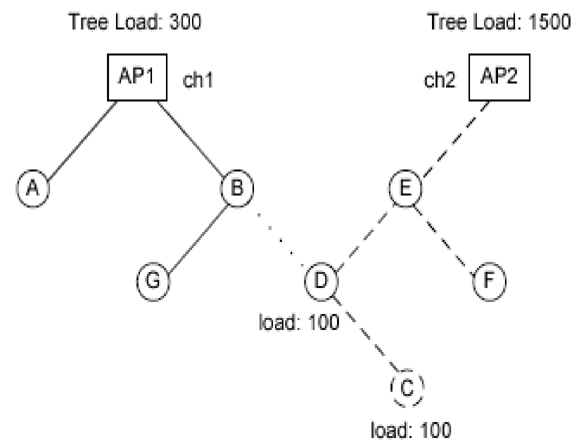


Fig. 7. An example network scenario to illustrate the process of obtaining route information and selecting primary route based on load information.

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 8.

Route Table of D Load: 100

dst	nexthop	hops	type	chan	load	path
AP2	E	2	PRIM	2	1500	E AP2
C	C	1	MH	2	100	
AP1	B	2	AP	1	300	B AP1

Fig. 8. Route table of node D in the scenario shown in 7.

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

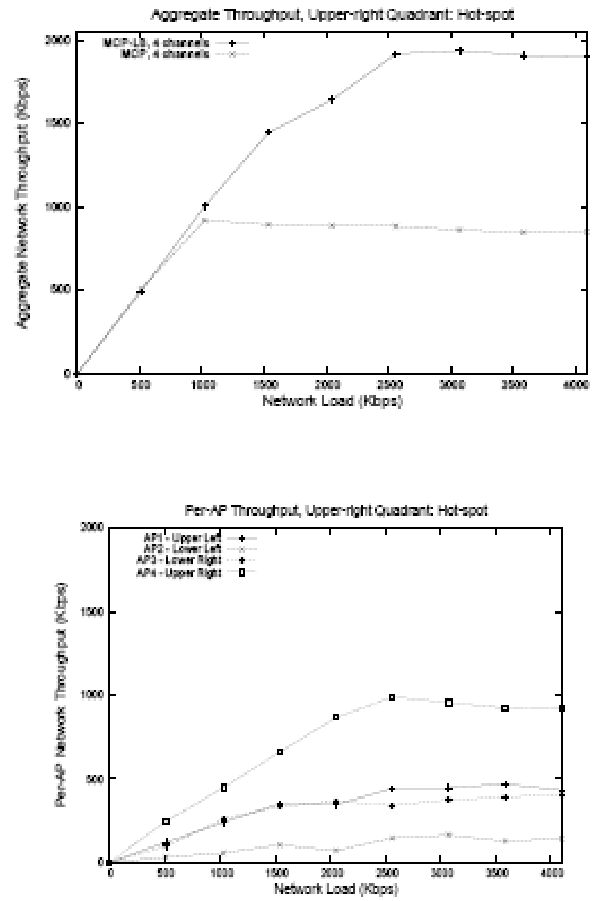
3) Switching route trees for load balancing: Once the necessary load information is obtained, nodes can decide whether to switch to other route trees. In the example shown in Figure 7, node D can switch its channel to channel 2 and re-associate with AP1, because it has a lower load. When making the decision, the node compares the current load of its route tree and the load of the other tree *when the node joins the tree*. 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 sub tree moves to the new route tree. So the load information should be computed correspondingly. For example, in Figure 7, suppose node D wants to decide if it should move to AP1. The current load of AP2-tree is 1500Kbps. The above example, the load of AP1-tree after the sub tree of D joins the tree is computed as 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 less traffic 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 decides to switch channel with confidence that it will balance the load among APs. The duration of time a node waits before it switches route trees is a tunable parameter. We denote it as *Tswitch* and we use *Tswitch* = 10 seconds in the simulations. Once node D decides to switch channels, it first sends a SWITCH message to all its child nodes, and 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 switches their channels and updates their route entry 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.

IV. 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. For more extensive experimental results, refer to [14]. A simulation area of 1000m £ 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.

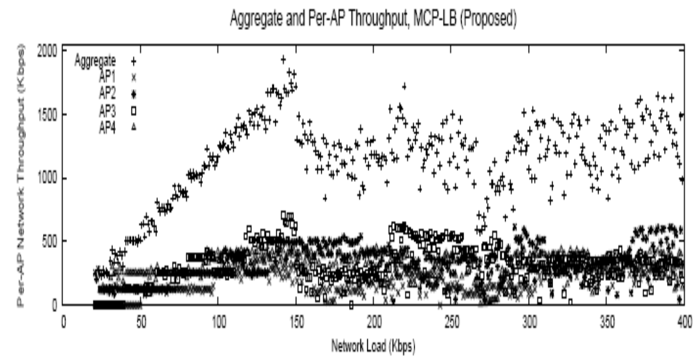


(b) Per-AP throughput

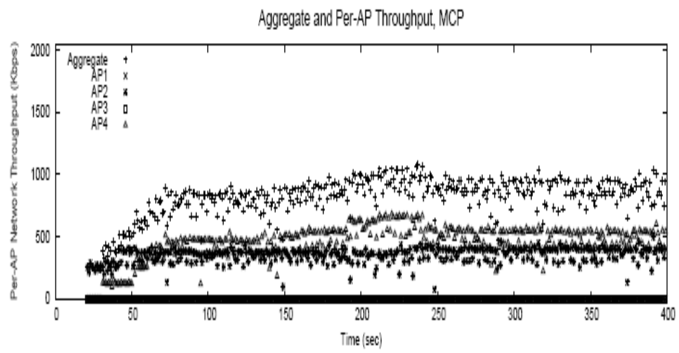
Fig. 9. Aggregate and Per-AP throughput for the scenario with a hot-spot.

In Figure 9(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 9(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 10. 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.



(a) MCP-LB (Proposed), 4 channels



(b) MCP, 4 channels

Fig. 10. Aggregate and per-AP throughput.

V. RELATED WORK

There has been vast amount of effort in the research community to improve performance of WLANs. 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 multichannel routing protocols and load balancing techniques, that are relevant to our work in this paper. More related works are discussed in [14]. Many routing protocols have been proposed for multi-hop networks, that supports only a single-channel [13]. Recently, routing protocols have been proposed for MCMH networks, that combine channel assignment and routing so that multiple channels can be utilized without changing the MAC layer protocol. Draves et al. [2] proposed a metric for route selection in multi-channel network. The metric, calledWeighted Cumulative Expected Transmission Time (WCETT), selects high quality routes considering bandwidth and loss rate of the link, and also the amount of interference on the channel.

This protocol 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 networks that works with nodes equipped with a single network interface. Since a node can only listen to one channel at a time, the protocol makes sure that when a route is established, all nodes in the path switch to the same channel. To allocate different channels to two flows that intersect with each other, the intersecting node becomes a “switching node”, which switches channels from time to time so that it can forward packets on both flows (see below for comparison with the proposed 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. Among multiple interfaces, each node maintains one interface on a fixed channel so that neighboring nodes know on which channel it should transmit to reach this node. The other interfaces are free to switch channels.

Raniwala et al. [12], [11] proposed a multi-channel routing protocol that also requires multiple interfaces per node. The paper addresses two main issues: neighbor-interface binding and interface-channel assignment. Since two neighboring nodes need to be on the same channel to communicate, these nodes need to have at least one interface that is on a common channel. Within this constraint, the protocol tries to assign channels to interfaces so that the load is balanced among channels. Our proposed protocol also assigns channels at the network layer, and is most similar to [15] and [11].

Our protocol is similar to [15] in the sense that the protocol assumes a single network interface per node. However, [15] assumes no infrastructure, and supports on-demand route establishment between any two nodes in the network if they need to communicate. Instead, our proposed protocol optimizes for when an infrastructure exists and only the routes between APs and mobile nodes need to be maintained proactively. With the infrastructure, two mobile nodes can communicate if they are independently connected with an AP. As a result, our proposed protocol does not need nodes that switch channels frequently, which reduces channel switching overhead. The protocol in [15] tries to select a channel with minimum load, but since there is no proactive route management, the network cannot adapt to changes in the traffic condition on each channel. On the other hand, our proposed protocol can adapt to changing traffic conditions so that the load is balanced among channels. Also, our proposed protocol is similar to [11], because our protocol assumes existence of infrastructure, and maintains routes between mobile nodes and APs. Also, the goal of our protocol is to balance the load among channels, so that the channel utilization is maximized. The difference between our work and [11] is that our protocol does not require multiple interfaces per node, and the protocol uses a different metric for load estimation.

Finally, we review the load balancing techniques proposed for WLANs. Hsiao et al. [4] 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 protocols do not assume that the load information is known. Also, in [4], the AP directs nodes to switch to another tree. This is not possible if the AP does not have the neighbor information of all the nodes, because AP does not know what alternative routes the node can take if it decides to switch trees. In our protocol, each node independently decides whether it should switch to another tree. Hassanein et al. [3] 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. We argue in Section II that locally measured load may not reflect the actual load, and propose the *weighted-load* metric.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a routing and channel assignment protocol for MCMH 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 the traffic has to travel, and proposed a load metric that accounts for these factors. Using the proposed load metric, we have proposed a routing protocol that assigns channels dynamically to nodes to balance 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 channel load. Second, the protocol only considers traffic load and does not take into account the varying channel conditions due to other environmental factors. The load should be assigned accordingly so that the performance is maximized. 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. So in this case, a node may need to consider the combined load of the two trees as the channel load when it compares channel load to decide whether it should switch to another channel. 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

[1] A.Balachandran, P. Bahl, and G. M. Voelker. Hot-spot congestion relief in public-area wireless networks. In IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 2002.
[2] R.Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In ACM Mobicom, 2004.
[3] H.Hassanein and A.Zhou. Routing with load balancing in wireless ad hoc networks. In ACM Int'l workshop on modeling, analysis and simulation of wireless and mobile systems (MSWim), 2001.
[4] P.Hsiao, A.Hwang, H.Kung, and D.Vlah. Load balancing routing for wireless access networks. In IEEE INFOCOM, 2001.
[5] IEEE Standard for Wireless LAN-Medium Access Control and Physical Laye Specification, P802.11, 1999.
[6] M. N. Inc. Mesh networks technology overview, http://www.meshnetworks.com.
[7] P. Kyasanur and N. Vaidya. Routing and interface assignment in multi-channel multi-interface wireless networks. In IEEE WCNC, 2005.
[8] S.Lee and M.Gerla. Dynamic load-aware routing in ad hoc networks. In ICC, 2001.
[9] W. List and N. Vaidya. A routing protocol for k-hop networks. In IEEE WCNC, 2004.
[10] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 1999.
[11] A. Raniwala and T. cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In IEEE INFOCOM, 2005.
[12] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. Mobile Computing and Communications Review, 8(2):50–65, April 2004.
[13] E. Royer and C. Toh. A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications, 1999.
[14] J. So and N. H. Vaidya. Routing and channel assignment in multi-channel multi-hop wireless networks with single-nic devices. Technical report, University of Illinois at Urbana-Champaign, December 2004.
[15] J. So and N. H. Vaidya. A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver. Technical report, University of Illinois at Urbana-Champaign, October 2004.
[16] C. Tschudin, R. Gold, O. Rensfelt, and O. Wibling. Lunar - a lightweight underlay network ad-hoc routing protocol and implementation. In Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN), 2004.
[17] VINT Group. UCB/LBNL/VINT network simulator ns (version 2).