Routing and Channel Assignment in Multi-Channel Multi-Hop Wireless Networks with Single-NIC Devices

Jungmin So[+] Nitin H. Vaidya[*]

Department of Computer Science[+],
Department of Electrical and Computer Engineering[*],
Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign
Email: {jso1, nhv}@uiuc.edu

Technical Report

December 2004

***Abstract*—In this paper, we present a routing and channel assignment protocol for multi-channel multi-hop wireless networks. We consider a multi-hop network, where a mobile host may connect to an access point using multi-hop wireless routes, via other mobile hosts or wireless routers. Also, we consider a multi-channel network where multiple non-overlapping (orthogonal) channels are available, and each host or router can dynamically select a channel to improve performance. In this environment, we propose a multi-channel routing protocol that works with nodes (mobile hosts or wireless routers) equipped with a single NIC (network interface card). Supporting single NIC devices is beneficial because having multiple network interface can be costly for small and cheap devices.**

Using the proposed protocol, nodes discover multiple routes to multiple access points possibly operating on different channels. Based on the traffic load information, each node selects the "best" route to an access point, and stays on the channel where the access point is on. With this behavior, the protocol balances load across the channels, thus removing hot spots and improving channel utilization. The channel assignment does not cause network partitions, assuring that if a path exists from a node to an access point, the node finds a route to an access point, where all the intermediate nodes and the destination are operating on the same channel.

Our simulation results 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

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, as in Figure 1.

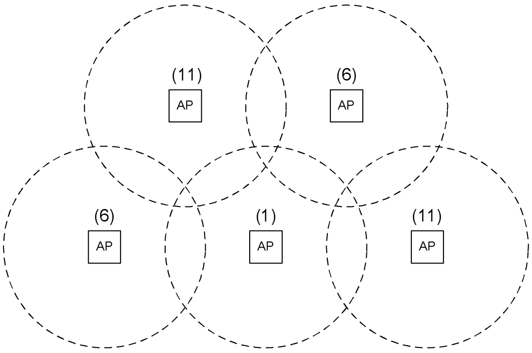


Fig. 1. An example deployment of access points. Numbers in parentheses indicate operating channels.

To reduce interference, neighboring APs are usually configured to operate on different frequency channels. For example, IEEE 802.11b standard for wireless LANs [1] provides three non-overlapping channels (1, 6 and 11), where communication can take place simultaneously without interfering each other. In Figure 1, APs are assigned channels so that neighboring APs operate on different channels.

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 not balanced among cells. Places near gates can become *hot spots* when people wait for the plane to start boarding. Second, for a large area, it can be expensive to deploy a large number of backbone-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 [3].

In the multi-hop architectures, a node may find multiple routes to different access points, potentially operating on different channels. Thus each node must select the "best" route where it can achieve the best service quality. Since routes are on different channels, selecting a route also means selecting which channel the node should stay on. We assume that all access points, wireless routers and mobile hosts are equipped with a single network interface card (NIC). With a single NIC, 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*. The minimum channel switching delay reported is 80 μs [4], which implies that per-packet 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 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 multi-channel multi-hop wireless networks with nodes equipped with single NIC.

The rest of this paper is organized as follows. In Section II, 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 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 WIRELESS NETWORKS

A multi-channel multi-hop wireless network of interest in this paper can be considered an extension to infrastructure networks, allowing nodes to connect with an access point via multiple hops. An example network is illustrated in Figure 2. 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 C and D, if their channels match each other.

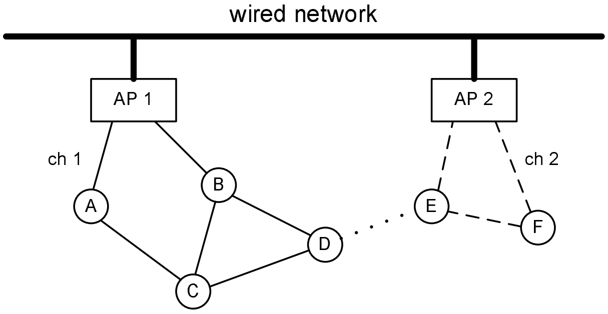


Fig. 2. An illustration of a multi-channel multi-hop wireless network. Solid lines are links on channel 1, and dashed lines are links on channel 2. The dotted line indicates a potential link, if node C and D were on the same channel.

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.

Note that a "node" can be a mobile host or a wireless router. Mobile hosts are end-user devices, and wireless routers are simple routers with only wireless interfaces, and they act as intermediate nodes to relay packets. Wireless routers are always willing to relay packets, whereas mobile hosts may or may not volunteer to relay packets of other mobile hosts. In the proposed protocol described in Section III, mobile hosts 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 2, 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. 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 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 or becomes congested. For example, in Figure 2, 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 mobile hosts. 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, although there is no limit on the number of hops the protocol supports. With this assumption, we do not need to consider simultaneous transmission within the route tree due to spatial reuse. Similar assumptions are made in other works [5], [6]. 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. With this assumption, balancing load among route trees leads to balancing load across channels. 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 2 again. Currently node D is connected to AP1 via node B. Node D has another route to AP2 via node C, but it is not used presently. Suppose each node exchanges their 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.

The other metric we can use 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 3. 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 interfere with each other). So it is better for D to stay on channel 1.

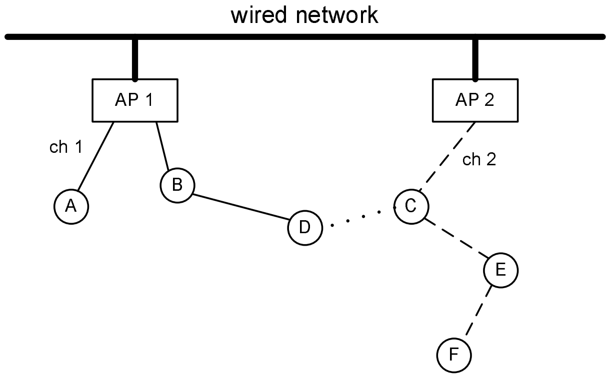


Fig. 3. An example network scenario. This example indicates that number of hops must be considered in measuring the load. The solid lines are links on channel 1, and the dashed lines are links on channel 2.

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 4. 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 2, 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

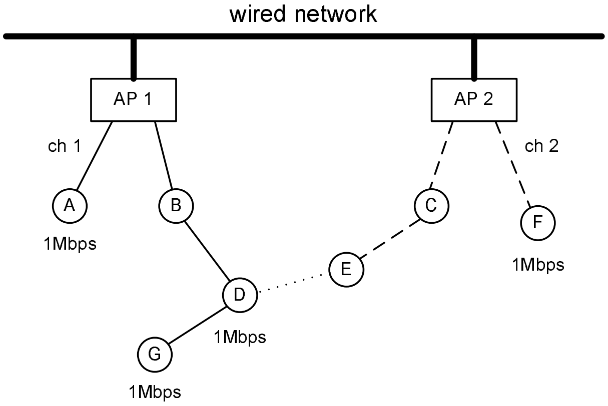


Fig. 4. An example network scenario. This example indicates that subtree load must be considered when selecting the best route.

switch channels, all its descendants in the route tree must also switch channels. But if D and G switches together, load of AP2-tree becomes 8Mbps, and thus D may decide to stay on channel 1.

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. 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 is 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 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:

- 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.

A. Route establishment

When a node is turned on, it must first discover a route to an access point. For this purpose, the node performs *”active scanning” on all channels*. Consider the scenario in Figure 5. 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 5(a). Now node B joins the network as in Figure 5(b). Initially, node B selects a random channel, and starts scanning by broadcasting a SCAN message on the channel. The SCAN message contains the address of the sender. 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.

(a) before

(b) after

Fig. 5. A simple network scenario with two access points and two nodes.

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 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.

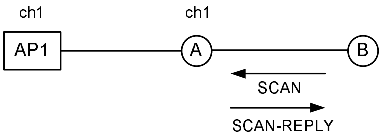
After scanning all channels, node B selects its *primary route* by choosing one of its neighbor as its *parent* node. Among all the routes received, B selects the route with the minimum load according to the weighted-load metric explained in Section II. If there is a tie, the one with the minimum number 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. The reverse path from the AP to the node is established by *association* process. Node B sends an ASSOCIATION message along its primary route to its associated AP, and all intermediate nodes and the AP set up a path to node B. The scanning and association process is illustrated in Figure 6, where node B is establishing a route with AP1.

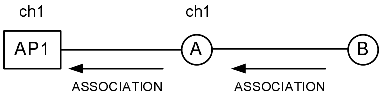
The route table that each node maintains is similar to that of AODV [7], with some changes in the route entry. An example route table is shown in Figure 7.

In the topology shown in Figure 7(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

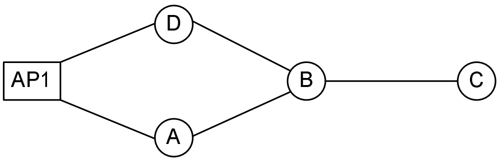


(a) scanning



(b) association

Fig. 6. An illustration of scanning and association process.



(a) Network topology

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	

(b) The route table of node B

Fig. 7. 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.

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 switches 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 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 discussed what metric to use for load information. We concluded that the *weighted load* was 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 7(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 8.

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. 8. Route table of AP 1 in the scenario shown in 7(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 7(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. Since a node with a single-NIC can only listen on one channel at a time, a node cannot monitor other channels to find the traffic condition on other channels. To exchange routes and load information, HELLO messages are used. Periodically, each AP transmits a HELLO message which includes the load information measured using weighted-load metric. As the scanning process previously explained, HELLO messages are sent on all channels, one at a time. When the 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.

A node switches its channels while sending HELLO messages, becoming deaf to the transmissions on its original channel. However, the duration of sending HELLO messages on all channels is small, around a few milliseconds per channel. When node finishes sending HELLO messages, it waits for a short delay before resuming to transmit data packets, so that its child nodes can finish sending HELLO messages and return to their original channel.

We call the period for sending HELLO messages P_{hello}. P_{hello} must not be long enough to reduce 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].

The HELLO message is used for two purposes: update load information and discover backup routes to other APs on other channels. When a node sends a HELLO message, it includes the following information.

- The address of AP that the node is currently associated with
- Number of hops to the AP
- Load of the node's route tree

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. For ease of understanding, consider the following scenario in Figure 9.

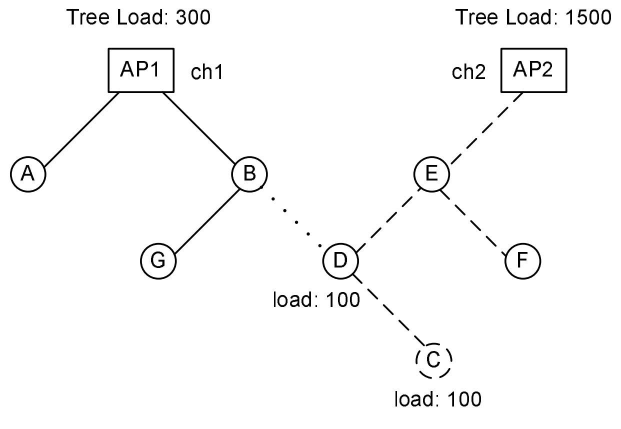


Fig. 9. An example network scenario to illustrate the process of obtaining route information and selecting primary route based on load information.

In Figure 9, 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. 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 2, 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 10.

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. 10. Route table of node D in the scenario shown in 9.

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 9, 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*. In the above example, node D has to compare the current load of AP2-tree and the load of AP1-tree considering the load when node D joins AP1-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 subtree moves to the new route tree. So the load information should be computed correspondingly.

For example, in Figure 9, 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:

$$L_{AP1'} = L_{AP1} + \sum_{i} (h_{iAP1} \times l_i) \tag{2}$$

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, i is the node in the subtree rooted at node D, h_{iAP1} 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:

$$L_{AP1'} = 300 + (100 \times 2 + 100 \times 3) = 800 \tag{3}$$

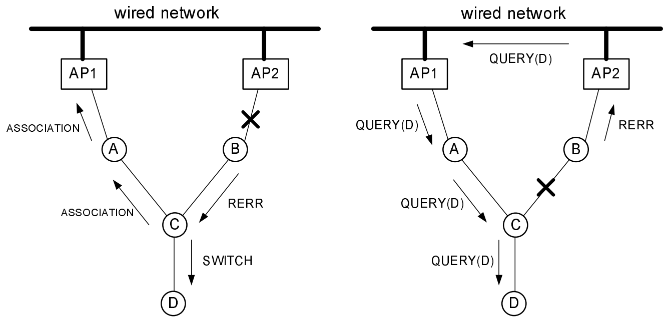
Since it is still smaller than the current load of AP2-tree, node D can decide to switch channels so that it can join AP1.

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 T_{switch} and we use $T_{switch} = 10$ seconds in the simulations. T_{switch} parameter should be longer than the $2P_{hello}$, so that nodes can make decisions based on up-to-date information. Also, to avoid route oscillation due to multiple nodes switching at the same time, each node chooses a random duration larger than T_{switch} before switching.

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 update 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 AP1 through wired backbone network that node D has left AP1. All children of node D go through the same process to associate with the new AP.

C. Route recovery

Due to mobility or node failures, the primary route may fail. The route recovery process of the proposed protocol is similar to route recovery process of ad-hoc routing protocols such as AODV. When the route is broken, the node which observes the failure informs the source node using RERR (Route Error) message. The source node initiates the recovery process. For example, consider the scenario in Figure 11.



(a) Mobile node is the source. (b) AP is the source.

Fig. 11. Route recovery process of the proposed protocol.

In Figure 11(a), node C was trying to send a packet to AP2, but B could not forward the packet to AP2. Then B sends an RERR message back to C. If node C does not have any backup route, then it starts the scanning process again. If C has a backup route as in this example, C selects the route, and switches to the corresponding channel. After that, C sends an ASSOCIATION message on the new route. If it has child nodes, it sends SWITCH message to its children so that they can switch their primary routes too (the switch process is similar to that discussed earlier).

In Figure 11(b), AP2 is the sender of a data packet, and node D is the destination. After the link between B and C breaks, AP2 is informed that the route is broken. Since APs do not maintain multiple routes, AP2 broadcasts the QUERY message on its route tree to look for node D. If node D receives the query message, it selects primary route (as discussed before in route establishment process) and sends an ASSOCIATION message along the route. If the AP cannot find the destination node in its own route tree, it asks neighbor APs through the wired link to look for node D. In this case AP1 finds node D. Then AP1 tells AP2 that it has found node D, and have node D associate with AP1.

IV. PERFORMANCE EVALUATION

We have performed simulations to evaluate the performance of the proposed protocol. In this section, we report and discuss the results.

There are two main design goals for the proposed protocol. First, the protocol should allow every node in the network to find a route to at least one AP, if such a route exists. To avoid fast channel switching, we consider a route as valid only if all nodes on the path are on the same channel. Second, within the constraint that every node should have at least one route to an AP, the routing protocol should adapt to changing traffic conditions on channels and balance load among them to maximize channel utilization.

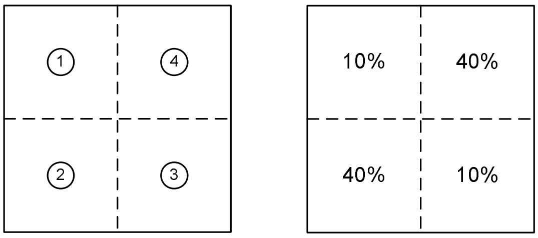
To see how well the proposed protocol utilizes available bandwidth in available channels, we compare our protocol with two other protocols. The first one is AODV [7], which is a single channel protocol. When running AODV, all nodes including APs are assigned the same channel. The second one is a multi-channel protocol, but each node selects routes based on the number of hops and there is no load balancing. We call this protocol “MCP” (Multi-Channel Protocol). With MCP, APs in a crowded area will have a correspondingly large number of nodes in its route tree, and APs in other areas will have small number of nodes associated with them. We call our proposed protocol as “MCP-LB” (Multi-Channel Protocol with Load Balancing), to distinguish with the other two protocol.

The goal of performance evaluation is to see how well the proposed protocol, MCP-LB, meets the design goal. In the following, we first describe our simulation setup, and then report the results.

A. Simulation Setup

We have used ns-2 simulator [8] with wireless extensions for our simulations. The simulation area is a 1000m × 1000m square, where 64 nodes are randomly placed. The transmission range of each node is 250 m, and the channel bit rate is 2 Mbps. Each node uses IEEE 802.11 DCF for medium access control.

Unless otherwise specified, 4 APs are placed at the center of 4 quadrants, as in Figure 12(a). There are 4 orthogonal channels total, and each AP operates on a different channel. Among 64 nodes in the area, 16 nodes are randomly picked as destination nodes that receive traffic from the wired network. Constant bit rate (CBR) traffic comes from the wired network through the AP and the AP forwards the traffic to the destination node. The size of each packet is 512 bytes. To create unbalanced traffic load in the area, we have picked the destination nodes using the distribution shown in Figure 12(b).



(a) Placement of APs

(b) Placement of destination nodes

Fig. 12. Placement of APs and destination nodes.

For the protocol parameters, we have set duration T, which is the duration of time window used by an AP to measure the load, as 10 seconds. Also, we have set the HELLO period, P_{hello}, to be 3 seconds and the minimum amount of time, T_{switch}, to be 10 seconds. These parameters were explained in Section III.

Finally, the simulation time for each simulation is 400 seconds, and each data point in the graphs are a result of 10 runs, except for Figure 17 and Figure 18.

B. Results

In the first simulation, we measured the aggregate throughput of the three protocols, varying the total network traffic. The total traffic load is divided equally among flows. So if the total load is 4Mbps, the rate of each flow is 250Kbps. The results are shown in Figure 13.

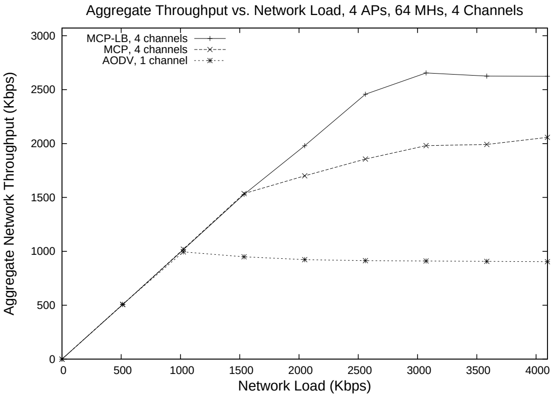


Fig. 13. Aggregate Network Throughput Varying Network Load.

As shown in the graph, the throughput of the AODV is limited to around 1000 Kbps. For MCP and MCP-LB, MCP-LB achieves higher throughput than MCP. This is because the two channels that the APs in the crowded area use are congested, whereas the other two channels are under-utilized. Note that in the MCP, nodes select routes based on number of hops. Since the destination nodes are placed in a non-uniform distribution, 80% of the nodes are associated with two APs in the crowded area, and only 20% of the nodes are associated with the other two APs. In MCP-LB, some destination nodes join route trees of the APs in the low-density area. Since these nodes are connected via multiple hops, the actual throughput achieved is less than the maximum achievable throughput, which is approximately 4Mbps.

In the next simulation, we varied the number of channels to study its impact on the performance of the MCP-LB, the proposed protocol. For all scenarios, the number of AP is set to 8, and all APs are placed in the center of the area. For scenarios with channels less than 8, there are multiple APs on the same channel. 32 flows were generated for 32 different destination nodes. The result is shown in Figure 14.

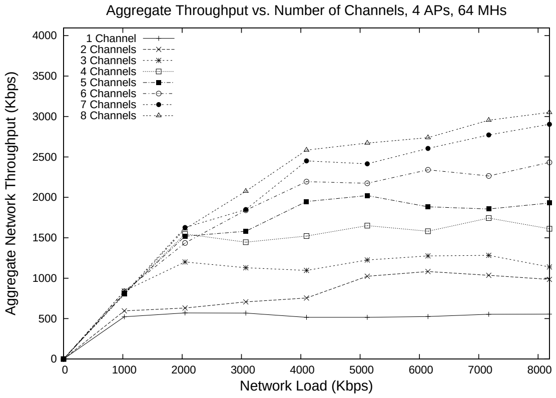


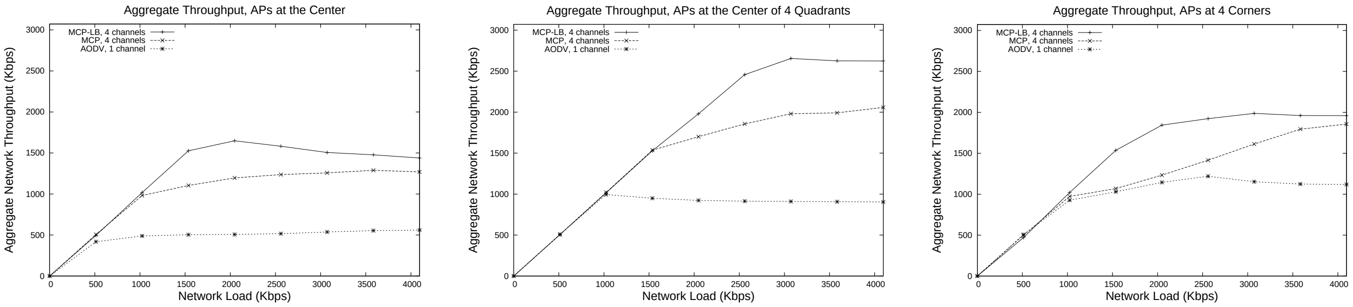
Fig. 14. Aggregate Network Throughput Varying Number of Channels.

As shown in the graph, the throughput increases as the number of channels increase, almost in a linear fashion. We can observe that even with 8 channels, the achieved throughput is around 3Mbps, when the total network load is 8Mbps. This is because the average number of hops from source to the AP is approximately 2 hops. If the APs are placed in uniform distribution, the achieved throughput would further increase.

This observation leads us to our next simulation. We have simulated three scenarios with exact same setting, including the placement of mobile nodes and selection of destination nodes. The only difference between three scenarios is the placement of APs. In the first scenario, the 4 APs were placed in the center. In the second scenario, the APs were placed in the center of 4 quadrants, as in Figure 12(a). In the third scenario, the APs were placed at the 4 corners of the simulation area. The result is shown in Figure 15.

Among the three scenarios, the throughput of MCP and MCP-LB are the highest in the second scenario, where the APs are placed in the center of 4 quadrants. This is because in the second scenario, the average number of hops is smaller, and the number of hops do not increase much even when a node moves to associate with an AP in another quadrant. When the APs are at the 4 corners, the benefit of load balancing is decreased because when nodes move to other route trees, the number of hops in the tree is very large so that the amount of throughput improvement is lowered.

This result indicates that the density of APs is critical in the performance of our proposed protocol. If the APs are placed too far away so that a node has to use 5 or 6 hops to connect with another AP, the benefit of load balancing is reduced. To achieve significant benefit from load balancing, the APs have



(a) center (b) middle (c) corner

Fig. 15. Aggregate Throughput varying Placement of APs.

to be placed dense enough so that a node can find multiple APs in the range of 2 or 3 hops.

The next simulation is performed to see the performance of the proposed protocol in relieving the congestion in hot-spots by redirecting nodes to other APs. To make an extreme hot-spot, all destination nodes were selected from nodes in the upper-right quadrant. In Figure 16(a), the aggregate throughput of MCP and MCP-LB are shown. Since node 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 16(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.

In our final simulation, we studied the adaptive behavior of our proposed protocol. During 400 seconds of simulation time, we simulated 32 flows, one flow starting at every 10 seconds. We plotted aggregate and per-AP throughput for MCP-LB and MCP. To create hot-spots, destination nodes were only selected from upper-right and lower-left quadrant. The result is shown in Figure 17. 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 throughput than MCP.

In addition to the aggregate and per-AP throughput, we also plotted the weighted load at each AP to see how the proposed protocol balances the load among APs using the weighted load metric. The result is shown in Figure 18. This graph shows how the proposed protocol tries to balance the weighted load among APs, in the changing traffic conditions.

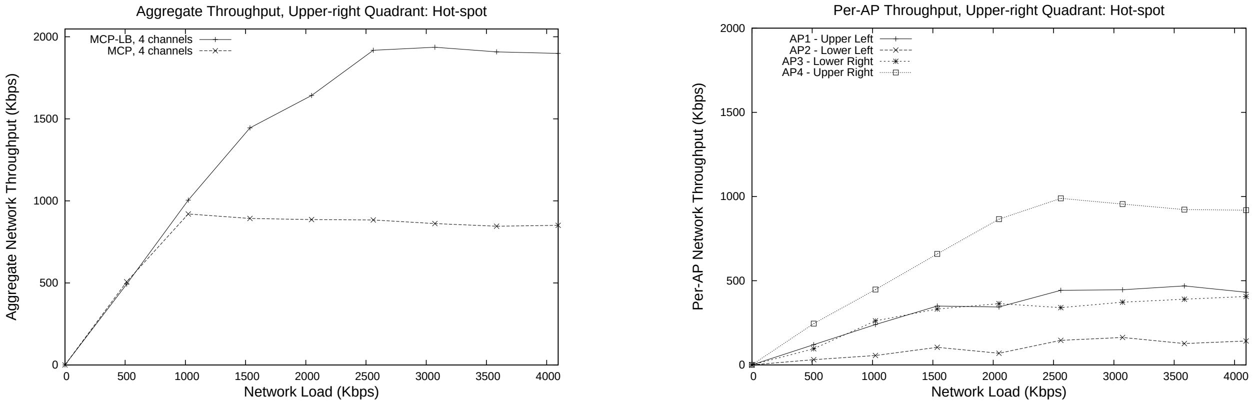
In conclusion, the proposed protocol successfully utilizes available bandwidth in available channels, by balancing the load among APs. So it achieves significant improvements over MCP, when the traffic load is unbalanced in the area.

V. 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 protocols and load balancing techniques, that are relevant to our work in this paper.

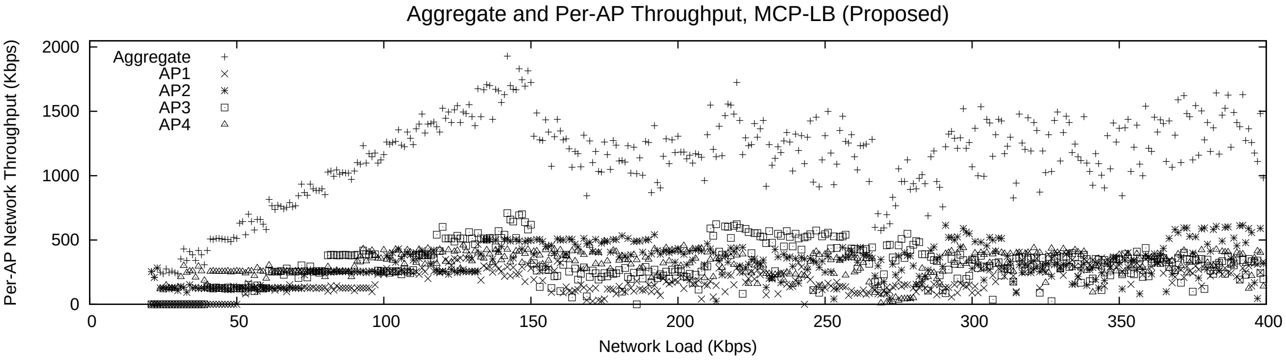
There are several MAC protocols proposed that support multiple channels. Wu et al. [9] proposed a protocol that requires two NICs per node, one for data communication and one for exchange of control messages. The channel for data communication is negotiated on the dedicated control channel, where every node is listening on. Then nodes switch their data channels accordingly. So et al. [10] proposed a multi-channel MAC protocol that requires only a single NIC per node. Instead of having a dedicated control channel, the protocol relies on temporal synchronization to have the nodes negotiate channels at a synchronized time window. Bahl et al. [4], proposed a protocol that works with single NIC and does not require synchronization. Each node switches channels according to a pseudo-random sequence, and it is guaranteed that the channels of any two nodes overlap periodically, so that they can communicate in while their channels overlap.

Many routing protocols have been proposed for multi-hop networks, that supports only a single-channel [11]. 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. [12] proposed a metric for route selection in multi-channel network. The metric, called Weighted 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. [13] propose a routing protocol for multi-channel networks that works with nodes equipped with a single NIC. 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

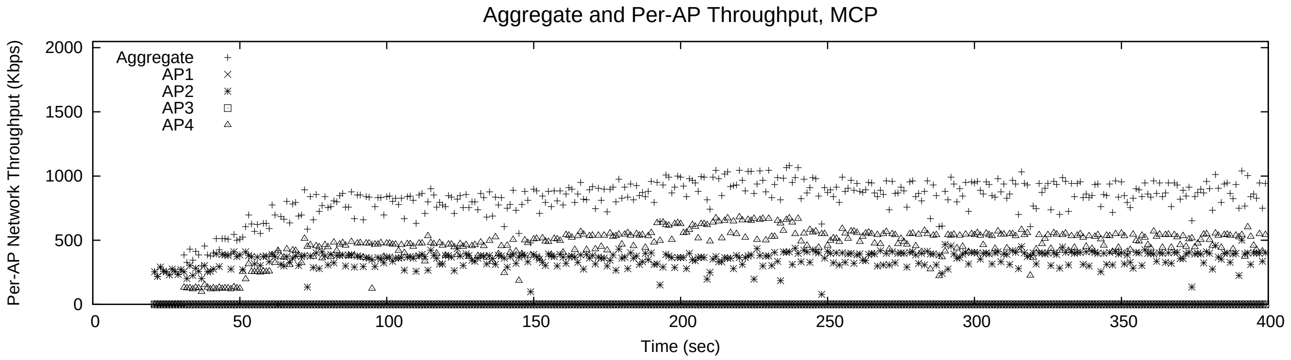


(a) Aggregate throughput (b) Per-AP throughput

Fig. 16. Aggregate and Per-AP throughput for the scenario with a hot-spot.



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

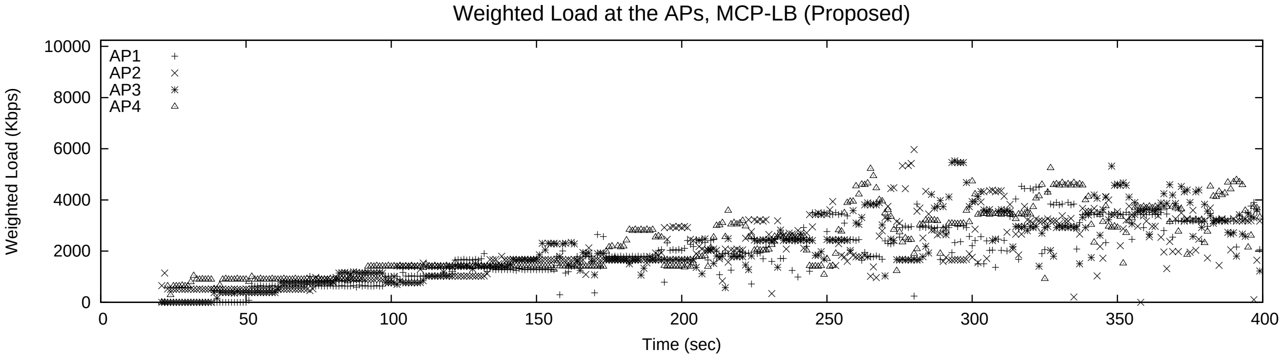


(b) MCP, 4 channels

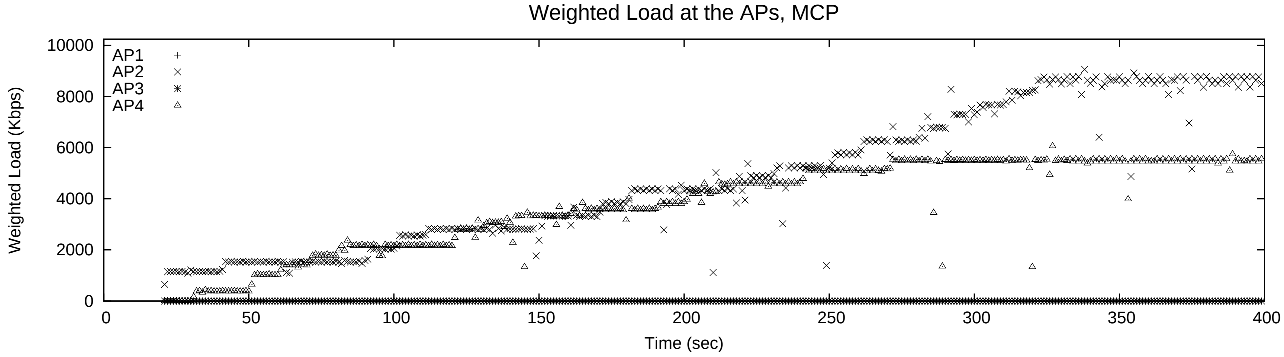
Fig. 17. Aggregate and per-AP throughput.

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. [14] proposed a routing protocol that requires multiple network interfaces per node, 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. [15], [16] 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



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



(b) MCP, 4 channels

Fig. 18. Weighted load at each AP over time.

layer, and is most similar to [13] and [16]. Our protocol is similar to [13] in the sense that the protocol assumes a single NIC per node. However, [13] 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 [13] 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 [16], because our protocol assumes existence of infrastructure, and maintains routes between mobile nodes and access points. Also, the goal of our protocol is to balance the load among channels, so that the channel utilization is maximized. There are several differences between [16] and our work. Our protocol does not require multiple interfaces per node. Supporting nodes with single interface can be beneficial because equipping multiple network interface can be costly for small and cheap devices. Also, we use a different metric for estimating load in the route tree.

Finally, we review the load balancing techniques proposed for wireless networks. Hsiao et al. [17] 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. Also, in [17], 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. [18] proposes to use as the number of “active” paths in the neighborhood as the load metric. Also, Lee et al. [19] 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 multi-channel multi-hop networks that works for nodes equipped with a single NIC. 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 locally measured load may not reflect actual load, because the node does not know whether the low traffic indicates congestion near the AP, or the requested traffic was low in the first place. Thus, we proposed a load metric that considers number of hops from AP to the destination.

Using the proposed load metric, the load information is distributed via control messages, and each node can independently decide to switch channels so that it can join a route tree with minimum load. As a whole, the network adapts to changing traffic conditions and 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 channel load. Each node can measure the amount of traffic generated at the node itself, but it is expensive to have every node report to the AP periodically so that the AP can update the load information considering uplink traffic too. A node can locally advertise its load to its neighbors, but it is not only the neighbors that are affected by this load. Nodes in the other parts of the route tree are also affected by this load. Second, the protocol only considers traffic load and does not take into account the varying channel conditions due to other environmental factors. For example, one channel may have higher packet loss than other channels. Also, channel conditions can vary in different regions. 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 lead 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 route tree has nodes that are large number of hops away from the AP. Then simultaneous transmissions can take place at the same time. 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 limitations are directions for our future research.

REFERENCES

[1] *IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specification, P802.11*, 1999.

[2] Anand Balachandran, Paramvir Bahl, and Geoffrey M. Voelker, “Hot-spot congestion relief in public-area wireless networks,” in *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 2002.

[3] Mesh Networks Inc., “Mesh networks technology overview, http://www.meshnetworks.com,” .

[4] Paramvir Bahl, Ranveer Chandra, and John Dunagan, “Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks,” in *ACM Mobicom*, 2004.

[5] William List and Nitin Vaidya, “A routing protocol for k-hop networks,” in *IEEE WCNC*, 2004.

[6] Christian Tschudin, Richard Gold, Olof Rensfelt, and Oskar Wibling, “Lunar - a lightweight underlay network ad-hoc routing protocol and implementation,” in *Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN)*, 2004.

[7] Charles E. Perkins and Elizabeth M. Royer, “Ad-hoc on-demand distance vector routing,” in *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1999.

[8] VINT Group, “UCB/LBNL/VINT network simulator ns (version 2),” .

[9] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, and Jang-Ping Sheu, “A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks,” in *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN)*, 2000.

[10] Jungmin So and Nitin H. Vaidya, “Multi-channel mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver,” in *ACM Mobihoc*, 2004.

[11] E. Royer and C. Toh, “A review of current routing protocols for ad hoc mobile wireless networks,” *IEEE Personal Communications*, April 1999.

[12] Richard Draves, Jitendra Padhye, and Brian Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *ACM Mobicom*, 2004.

[13] Jungmin So and Nitin H. Vaidya, “A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver,” Tech. Rep., University of Illinois at Urbana-Champaign, October 2004.

[14] Pradeep Kyasanur and Nitin Vaidya, “Routing and interface assignment in multi-channel multi-interface wireless networks,” in *IEEE WCNC*, 2005.

[15] Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh, “Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks,” *Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, April 2004.

[16] ”Ashish Raniwala and Tzi cker Chiueh”, “Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network,” in *IEEE INFOCOM*, 2005, http://www.ecsl.cs.sunysb.edu/multichannel.

[17] P. Hsiao, A. Hwang, H. Kung, and D. Vlah, “Load-balancing routing for wireless access networks,” in *IEEE INFOCOM*, 2001.

[18] 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.

[19] S.J. Lee and M. Gerla, “Dynamic load-aware routing in ad hoc networks,” in *ICC*, 2001.