

# Joint Logical Topology Design, Interface Assignment, Channel Allocation, and Routing for Multi-Channel Wireless Mesh Networks

A. Hamed Mohsenian-Rad, *Student Member, IEEE*, and Vincent W.S. Wong, *Senior Member, IEEE*

**Abstract**—A multi-channel wireless mesh network (MC-WMN) consists of a number of stationary wireless routers, where each router is equipped with multiple network interface cards (NICs). Each NIC operates on a distinct frequency channel. Two neighboring routers establish a logical link if each one has an NIC operating on a common channel. Given the physical topology of the routers and other constraints, four important issues should be addressed in MC-WMNs: logical topology formation, interface assignment, channel allocation, and routing. Logical topology determines the set of logical links. Interface assignment decides how the logical links should be assigned to the NICs in each wireless router. Channel allocation selects the operating channel for each logical link. Finally, routing determines through which logical links the packets should be forwarded. In this paper, we mathematically formulate the logical topology design, interface assignment, channel allocation, and routing as a joint linear optimization problem. Our proposed MC-WMN architecture is called *TiMesh*. Extensive *ns-2* simulation experiments are conducted to evaluate the performance of *TiMesh* and compare it with two other MC-WMN architectures *Hyacinth* [1] and *CLICA* [2]. Simulation results show that *TiMesh* achieves higher aggregated network throughput and lower end-to-end delay than *Hyacinth* and *CLICA* for both TCP and UDP traffic. It also provides better fairness among different flows.

**Index Terms**—Wireless mesh networks, multi-channel, multi-interface, logical topology, interface assignment, channel allocation, routing, max-min fairness, linear optimization.

## I. INTRODUCTION

The next generation fixed wireless broadband networks are being increasingly deployed as wireless mesh networks (WMNs) in order to provide ubiquitous access to the Internet. Research and development of WMNs are motivated by several applications including broadband home networking, community and neighborhood networking, enterprise networking, and metropolitan area networking [3]. Recently, there have been several implementation studies on WMNs [4], [5]. Some vendors have also begun to offer products in this area [6], [7]. The IEEE has also set up a new task group 802.11s for mesh networking [8].

The aggregate capacity and performance of WMNs can be increased by the use of multiple channels [9]. A multi-channel wireless mesh network (MC-WMN) consists of a number of

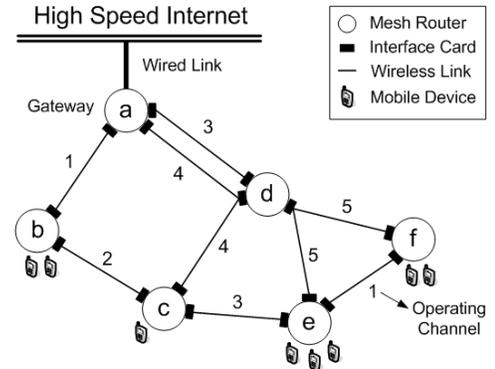


Fig. 1. An MC-WMN with six wireless mesh routers, five frequency channels, and three NICs per router. The number on each link indicates the operating channel number.

stationary wireless mesh routers, forming a *wireless backbone*. Each router is equipped with multiple network interface cards (NICs). Each interface operates on a distinct frequency channel in the IEEE 802.11a/b/g bands. The wireless mesh routers serve as access points (APs) for wireless mobile devices. Some of them also act as gateways to the Internet via high-speed wired links. Two neighboring routers can establish a logical link if each one has an interface operating on a common channel. There may be multiple logical links between some of the routers to increase the data transmission rate between them. A logical topology is comprised of the sets of routers and logical links. Fig. 1 shows an MC-WMN logical topology with six wireless mesh routers. Router *a* serves as the gateway to the wired high speed Internet. Each router has three NICs. Five frequency channels are being used in the network.

Within the IEEE 802.11 frequency bands, the number of available channels is limited. The 802.11b/g bands and the 802.11a band provide 3 and 12 orthogonal frequency channels, respectively. This implies that some logical links may be assigned the same channel. For example, in Fig. 1, both logical links (*a, b*) and (*f, e*) are assigned to channel number 1. In this case, interference will occur if these logical links are close to each other. Those two interfering links cannot be active simultaneously. In this paper, we define a logical link to be *active* if it is being used for packet transmission. A logical link is *idle* if it is not active.

In general, the number of available NICs is limited. In the experimental MC-WMN test-beds in [1] and [9], each mesh router is equipped with two NICs. Providing up to four NICs is also considered reasonable [1], [10]. A small number of

Manuscript received on June 1, 2006; revised on November 20, 2006 and April 20, 2007. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant number 261604-03. The review of this paper was coordinated by Dr. Giridhar Mandyam.

The authors are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada, e-mail: {hamed, vincentw}@ece.ubc.ca.

NICs implies that some logical links in a router may need to *share* an NIC to transmit and receive the data packets. For example, in Fig. 1, both logical links  $(d, f)$  and  $(d, e)$  share an NIC on router  $d$ . Thus, they are required to operate over the same frequency channel (i.e., channel number 5). When two logical links in a router share an NIC, they cannot be active simultaneously. It significantly reduces their *effective* capacity. The effective link capacity can be increased by removing some of the links from the logical topology. For example, by removing logical link  $(d, e)$  in Fig. 1, logical link  $(d, f)$  can be attached to an *exclusive* (not shared) NIC on router  $d$ . However, this may increase the number of hops through some of the routing paths (e.g., from router  $e$  to the gateway  $a$ ). In certain cases, the logical topology may not even be connected.

In summary, four issues should be addressed in MC-WMNs:

- 1) *Logical Topology Formation*: Given the physical topology, how many logical links (if any) should be assigned between a pair of neighboring routers?
- 2) *Interface Assignment*: Given the logical topology, how should the logical links be assigned to each NIC in a wireless mesh router?
- 3) *Channel Allocation*: Given the logical topology and interface assignment, how should a frequency channel be allocated on each logical link?
- 4) *Routing*: Given the logical topology, interface assignment, and channel allocation, through which logical links should the packets be forwarded?

In this paper, we formulate the logical topology formation, interface assignment, channel allocation, and routing as a joint linear optimization problem. We call our proposed MC-WMN architecture *TiMesh*. Our contributions are as follows:

- Our model formulation takes into account the number of available NICs in each wireless mesh router, the number of available frequency channels, the communication range and the interference range of the wireless mesh routers, and the expected traffic load between different source and destination pairs.
- Our model formulation allows having multiple logical links between the same pair of routers. This further increases the effective data transmission rate between the two routers.
- Our proposed algorithm guarantees the network connectivity. It also supports both *internal traffic* among the wireless routers and *external traffic* to the Internet.
- Simulation results show that *TiMesh* achieves higher aggregate throughput and lower end-to-end delay than the recently proposed *Hyacinth* [1] and *CLICA* [2] MC-WMN architectures.

The rest of the paper is organized as follows. We present related work in Section II. Our proposed joint design is described in Section III. Performance evaluation and comparison are given in Section IV. Conclusions are given in Section V.

## II. RELATED WORK

Several topology formation, interface assignment, channel allocation, and routing algorithms have been recently proposed for MC-WMNs. Raniwala *et al.* [1] proposed a logical tree

topology for MC-WMNs, called *Hyacinth*. The tree construction mechanism is similar to the IEEE 802.1D spanning tree formation. The gateways are the roots. Each router uses an up-NIC to exclusively connect to its parent and uses several (probably shared) down-NICs to connect to its children. Each parent router provides the Internet connectivity to its children routers. As a result, each wireless mesh router can access the Internet through the shortest available routing path. In the *Hyacinth* architecture, each router allocates the channels that are less used by its neighboring routers to its down-NICs.

Marina *et al.* [2] proposed the *Connected Low Interference Channel Assignment (CLICA)* algorithm for MC-WMNs. The interference among logical links is modeled by a weighted conflict graph. The weight of the edge between two vertices in the weighted conflict graph indicates the extent of interference between their corresponding logical links using the protocol interference model [11]. The proposed heuristic algorithm determines the logical links and assigns their channels permanently so that the average interference weight in the resultant conflict graph is minimized while the network connectivity is maintained. In [12], the channels are allocated so as to minimize the maximum number of interfering links within each neighborhood, subject to the connectivity constraint of the logical topology graph. A bandwidth-aware routing algorithm is also proposed to facilitate the path finding operation.

For the frequency channel allocation, various centralized and distributed algorithms have recently been proposed for MC-WMNs. The centralized schemes (e.g., [10], [13]) require a network controller to collect the topology information and assign the channels. In the distributed schemes (e.g., [14], [15]), some of the mesh routers are responsible for channel assignment for a subset of interfaces. The channel allocation algorithms in MC-WMNs can also be classified as static and dynamic. The static algorithms (e.g., [16], [17]) assign a channel to each network interface permanently, while dynamic algorithms (e.g., [1], [18]) allow each network interface to change its channel in some short or long intervals. Unlike the static algorithms, the dynamic channel allocation requires a coordination mechanism to ensure that the sending and the receiving routers/NICs use the same frequency channel at the same time. The coordination overhead can be noticeable and reduce the network performance significantly [17].

Various joint designs for MC-WMNs have also been proposed. The joint channel allocation and routing problems are studied in [10], [19], [20]. A joint routing and NIC assignment scheme is proposed in [21]. In [22], we formulate the joint channel assignment and congestion control as a utility maximization problem. The model is extended to incorporate network interface assignment and media access control in [23].

## III. JOINT LOGICAL TOPOLOGY DESIGN, INTERFACE ASSIGNMENT, CHANNEL ALLOCATION, AND ROUTING

In this section, we formulate the logical topology formation, interface assignment, channel allocation, and routing as a joint linear optimization problem. For the rest of this paper, the terms routers and nodes are used interchangeably.

### A. Problem Formulation

We first model an MC-WMN by a *physical topology graph*  $G(N, E)$  where  $N$  denotes the set of all vertices and  $E$  denotes the set of all unidirectional edges. Each vertex  $n \in N$  represents a stationary wireless mesh router. For simplicity, we assume that  $n = \{1, 2, \dots, |N|\}$ . For any two nodes  $m, n \in N$ , if node  $n$  is within the *communication range* of node  $m$ , then there is an edge or link from node  $m$  to node  $n$  in set  $E$ . The link from node  $m$  to node  $n$  is denoted by  $e_{mn} \in E$ . We assume the connectivity to be symmetric. That is, link  $e_{mn} \in E$  if and only if  $e_{nm} \in E$ . Each wireless mesh router is equipped with  $I$  network interface cards. There are  $C$  orthogonal frequency channels available.

For any two nodes  $m$  and  $n$  such that  $e_{mn} \in E$ , and any frequency channel  $i \in \{1, \dots, C\}$ , we define a *link channel allocation variable*  $x_{mn}^i$ . In the logical topology, if node  $m$  communicates with node  $n$  over the  $i^{\text{th}}$  frequency channel, then  $x_{mn}^i$  is equal to 1; otherwise, it is equal to zero. For the MC-WMN in Fig. 1 with  $C = 5$ , we have  $x_{ab}^1 = 1$  and  $x_{ab}^2 = x_{ab}^3 = x_{ab}^4 = x_{ab}^5 = 0$ . In general, two nodes may communicate with each other over multiple distinct frequency channels. For example, in Fig. 1, we have  $x_{ad}^3 = x_{ad}^4 = 1$  and  $x_{ad}^1 = x_{ad}^2 = x_{ad}^5 = 0$ .

To establish the logical links, nodes  $m$  and  $n$  should assign the same frequency channels to communicate with each other. This requires that,

$$x_{mn}^i = x_{nm}^i, \quad \forall m, n \in N, e_{mn} \in E, \forall i = 1, \dots, C. \quad (1)$$

The link channel allocation variables implicitly provide the required information to create the logical topology. Due to traffic and interference constraints, it is possible that there is a link between nodes  $m$  and  $n$  in the physical topology graph (i.e.,  $e_{mn} \in E$ ), but there is no logical link between them in the logical topology. In that case, we have  $x_{mn}^i = 0$  for all  $i = 1, \dots, C$ . Note that we allow multiple logical links between the same pair of nodes in the logical topology. They operate independently over distinct frequency channels and can significantly increase the effective capacity between two neighboring nodes.

For any node  $m \in N$  and any channel  $i \in \{1, \dots, C\}$ , we define  $y_m^i$  to be as follows:

$$y_m^i = \begin{cases} 1, & \text{if } \exists n \in N \text{ and } e_{mn} \in E, \text{ such that } x_{mn}^i = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

We refer to  $y_m^i$  as the *node channel allocation variable* corresponding to node  $m$  and channel  $i$ . For node  $a$  in Fig. 1 with  $C = 5$  and  $I = 3$ , we have  $y_a^1 = y_a^3 = y_a^4 = 1$  and  $y_a^2 = y_a^5 = 0$ . From (2),  $\sum_{i=1}^C y_m^i$  indicates the total number of channels that are being used by node  $m$  to establish logical links with its neighboring nodes. Since each NIC operates on a distinct frequency channel,  $\sum_{i=1}^C y_m^i$  cannot be larger than the total number of available NICs on node  $m$ . That is,

$$\sum_{i=1}^C y_m^i \leq I, \quad \forall m \in N. \quad (3)$$

The link and node channel allocation variables implicitly provide the required information for interface assignment. For

example, given  $y_a^1 = y_a^3 = y_a^4 = 1$ , we assign channel 1 to the first NIC, channel 3 to the second NIC, and channel 4 to the third NIC of node  $a$ . Since  $x_{ab}^1 = x_{ad}^3 = x_{ad}^4 = 1$ , node  $a$  uses its first NIC to communicate with node  $b$  and its second and third NICs to communicate with node  $d$ .

*Lemma 1:* The desired correspondence in (2) is obtained by having  $y_m^i$  be a *continuous real variable* for all nodes  $m \in N$  and all channels  $i \in \{1, \dots, C\}$  and also requiring that:

$$0 \leq y_m^i \leq \sum_{n \in N, e_{mn} \in E} x_{mn}^i \quad (4a)$$

$$x_{mn}^i \leq y_m^i \leq 1, \quad \forall n \in N, e_{mn} \in E. \quad (4b)$$

*Proof:* Assume that node  $m$  is assigned to communicate with  $K$  neighboring nodes over channel  $i$ . Thus, constraint (4a) can be written as  $0 \leq y_m^i \leq K$ . If  $K = 0$ , then constraints (4a) and (4b) become  $0 \leq y_m^i \leq 0$  and  $0 \leq y_m^i \leq 1$ , respectively. This implies that  $y_m^i = 0$ . On the other hand, if  $K > 0$  (i.e., if  $K \geq 1$ ), then constraints (4a) and (4b) become  $0 \leq y_m^i \leq K$  and  $1 \leq y_m^i \leq 1$ , respectively. This implies that  $y_m^i = 1$ . ■

### B. Effective Capacity

Let  $c^0$  denote the *nominal* link-layer data rate in the corresponding 802.11 standard (e.g., 54 Mbps in 802.11a). Also let  $0 \leq c_{mn}^i \leq c^0$  denote the effective capacity of the logical link  $(m, n)$  in the direction from node  $m$  to node  $n$  over frequency channel  $i$ . We have:

$$c_{mn}^i \leq x_{mn}^i c^0, \quad \forall m, n \in N, e_{mn} \in E, \forall i = 1, \dots, C. \quad (5)$$

From (5), if node  $m$  does not allocate channel  $i$  to communicate with node  $n$  (i.e.,  $x_{mn}^i = 0$ ), then node  $m$  cannot transmit any packet to node  $n$  over channel  $i$  (i.e.,  $c_{mn}^i = 0$ ).

For any two nodes  $m$  and  $n$  such that  $e_{mn} \in E$ , we define a set of potential interfering links  $F_{mn} \subset E$ .  $F_{mn}$  includes all  $e_{pq} \in E$  such that nodes  $p$  or  $q$  (or both) are within the *interference range* of nodes  $m$  or  $n$  (or both). Note that we always have  $e_{nm} \in F_{mn}$ . Considering the IEEE 802.11 based RTS-CTS-DATA-ACK model, we have [10], [24]:

$$\frac{c_{mn}^i}{c^0} + \sum_{p, q, e_{pq} \in F_{mn}} \frac{c_{pq}^i}{c^0} \leq 1, \quad \forall m, n \in N, e_{mn} \in E, \forall i = 1, \dots, C \quad (6)$$

where  $c_{mn}^i/c^0$  denotes the fraction of time that logical link  $(m, n)$  can be active in the direction from node  $m$  to node  $n$  over frequency channel  $i$ .

### C. Total Flows on a Logical Link

For efficient network planning, a statistical model for network traffic needs to be available. Let  $\gamma^{sd}$  denote the expected traffic rate to be delivered between source and destination pair  $(s, d)$ , where  $s, d \in N$ . We assume that the information  $\gamma^{sd}$  for all source and destination pairs is given. For any source and destination pair  $(s, d)$ , any nodes  $m, n \in N$  such that  $e_{mn} \in E$ , and any channel  $i \in \{1, \dots, C\}$ , we define a binary *routing variable*  $a_{mn,i}^{sd}$ . The variable  $a_{mn,i}^{sd}$  is equal to 1 if the traffic from source  $s$  to destination  $d$  is being routed via link  $(m, n)$  in the direction from node  $m$  to node  $n$  over channel

$i$ , and is equal to 0 otherwise. Note that  $a_{mn,i}^{sd} \neq a_{nm,i}^{sd}$  in general. Multiple links between a pair of nodes can provide more than one path between them. Since each of the multiple links is operating over a distinct channel, packets that are forwarded on different links experience different latencies. Thus, if packets that belong to the same flow use parallel links between a pair of neighboring nodes, this can cause packets to arrive out of order. To avoid this issue, only one of the available logical links between each pair of neighboring nodes is used to route packets of each flow. That is,

$$\sum_{i=1}^C a_{mn,i}^{sd} \leq 1, \quad \forall s, d, m, n \in N, e_{mn} \in E. \quad (7)$$

Let  $\lambda_{mn}^i$  denote the aggregate traffic from all source and destination pairs that is routed over logical link  $(m, n)$  in the direction from node  $m$  to node  $n$  over channel  $i$ . We have,

$$\lambda_{mn}^i = \sum_{s,d \in N} a_{mn,i}^{sd} \gamma^{sd}, \quad \forall m, n \in N, e_{mn} \in E, \quad \forall i = 1, \dots, C. \quad (8)$$

The aggregate traffic  $\lambda_{mn}^i$  cannot be more than the effective capacity  $c_{mn}^i$  for all nodes  $m, n \in N$  such that  $e_{mn} \in E$ , and all channels  $i \in \{1, \dots, C\}$ . Consider the following constraint:

$$\lambda_{mn}^i \leq \Lambda c_{mn}^i, \quad \forall m, n \in N, e_{mn} \in E, \quad \forall i = 1, \dots, C. \quad (9)$$

where  $\Lambda \leq 1$  is a positive parameter. From (9), the parameter  $\Lambda$  imposes an upper bound on the expected *link utilization*  $\lambda_{mn}^i/c_{mn}^i$ . The higher the link utilization, the higher the queuing delay [25]. In the current Internet, an access link is considered to be overloaded when its average utilization is greater than 80% [26], [27]. Thus, we set  $\Lambda = 0.8$ .

#### D. Flow Conservation at Each Node

The flow conservation requires that for  $s, d, m \in N$ ,

$$\sum_{\substack{n \in N, \\ e_{mn} \in E}} \sum_{i=1}^C a_{mn,i}^{sd} \gamma^{sd} - \sum_{\substack{n \in N, \\ e_{nm} \in E}} \sum_{i=1}^C a_{nm,i}^{sd} \gamma^{sd} = \begin{cases} \gamma^{sd}, & \text{if } s = m, \\ -\gamma^{sd}, & \text{if } d = m, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

In (10), the term on the left-hand side is the *net flow* out of node  $m$  for the flow from source  $s$  to destination  $d$ . The net flow is the difference between the outgoing flow and the incoming flow. The term on the right-hand side is equal to 0 if node  $m$  is neither the source nor the destination for that specific flow. If node  $m$  is the source (i.e.,  $s = m$ ), then the net flow is equal to  $\gamma^{sd}$ . If node  $m$  is the destination (i.e.,  $d = m$ ), then the net flow is equal to  $-\gamma^{sd}$ . Note that both sides in (10) can be divided by the common factor  $\gamma^{sd}$ .

Constraints (10) also guarantees that there is at least one routing path available between each source and destination pair  $(s, d)$ . In practice, all nodes have some traffic to or from the Internet; thus we can make the valid assumption that  $\gamma^{sd} > 0$ , if either  $s$  or  $d$  is a gateway node. The constraint in (10) and the aforementioned assumption guarantee that the obtained topology is connected. That is, there is neither an isolated node nor an isolated group of nodes.

#### E. Feasible Region and the Objective Function

Given the expected traffic demand  $\gamma$  and the network resources  $C$ ,  $I$ , and  $c^0$ , the constraints in (1)-(10) form the feasible region for all logical topologies that can properly support the expected traffic demand  $\gamma$ . The feasible region could be empty. We can enlarge the feasible region by choosing a higher value for parameter  $\Lambda$ ; however, even for  $\Lambda = 1$ , the feasible region can still become empty if and only if the network resources cannot support the expected traffic demand. In that case, we need to either increase the available resources or limit the traffic demand.

From constraint (9), the difference  $(\Lambda c_{mn}^i - \lambda_{mn}^i)$  is always non-negative. As  $\lambda_{mn}^i$  approaches  $\Lambda c_{mn}^i$ , the difference  $(\Lambda c_{mn}^i - \lambda_{mn}^i)$  tends to 0 and the corresponding logical link becomes more prone to congestion. Let  $\delta_{\min}$  denote the *minimum* difference  $(\Lambda c_{mn}^i - \lambda_{mn}^i)$  across all channels and all links that exist in the logical topology. That is,

$$\delta_{\min} = \min_{\substack{m, n \in N, e_{mn} \in E, \\ i \in \{1, \dots, C\}, x_{mn}^i = 1}} (\Lambda c_{mn}^i - \lambda_{mn}^i). \quad (11)$$

Note that  $\delta_{\min}$  corresponds to the most congested (i.e., the bottleneck) logical link across the network. Our objective is to maximize the variable  $\delta_{\min}$ . It can be achieved by decreasing the aggregate traffic load or increasing the effective capacity (or both) on the network's bottleneck link. The former implies *load balancing*: balancing the traffic load among different logical links using proper logical topology formation and routing schemes; while the latter implies *congestion-aware capacity planning*: providing higher effective capacity for more congested logical links using proper logical topology formation, interface assignment, and channel allocation schemes. Load balancing is shown to be a proper objective for joint topology control and routing algorithms in optical networks [28]. Congestion aware capacity planning is also proposed for cross-layer congestion control designs in wireless ad-hoc [29] and mesh networks [22].

Maximizing  $\delta_{\min}$  can also be justified in terms of providing fairness among the existing logical links. In fact, it leads to achieving *max-min* or *bottleneck-optimal* fairness [25], [30], [31]. The system is fair in the sense that all the logical links will experience similar level of congestion.

*Lemma 2*: The desired correspondence in (11) can be obtained by requiring that:

$$\delta_{\min} \leq (\Lambda c_{mn}^i - \lambda_{mn}^i) + \Lambda c^0 (1 - x_{mn}^i), \quad \forall m, n \in N, e_{mn} \in E, \quad \forall i = 1, \dots, C. \quad (12)$$

*Proof*: If there exists a logical link between nodes  $m$  and  $n$  over channel  $i$  (i.e.,  $x_{mn}^i = 1$ ), then  $\Lambda c^0 (1 - x_{mn}^i) = 0$  and constraint (12) simply becomes  $\delta_{\min} \leq (\Lambda c_{mn}^i - \lambda_{mn}^i)$ . On the other hand, if there is no logical link between nodes  $m$  and  $n$  over channel  $i$  (i.e.,  $x_{mn}^i = 0$ ), then  $\Lambda c^0 (1 - x_{mn}^i) = \Lambda c^0$ . From eqs. (5) and (9) we also have  $(\Lambda c_{mn}^i - \lambda_{mn}^i) = 0$ . Thus, constraint (12) becomes  $\delta_{\min} \leq \Lambda c^0$ . Note that  $\Lambda c^0$  is always an upper bound for variable  $\delta_{\min}$ . ■

## F. Hop Count Constraint

Load balancing avoids highly loaded links and prevents congestion; however, it may lead to assigning long routing paths. For each source and destination pair  $(s, d)$ , the hop count along the assigned routing path is obtained as  $\sum_{m,n \in N, e_{mn} \in E} \sum_{i=1}^C a_{mn,i}^{sd}$ . Let  $h_G^{sd}$  denote the hop count for the *minimum* hop path between source and destination pair  $(s, d)$  in the physical topology graph  $G(N, E)$ . The ratio  $\sum_{m,n \in N, e_{mn} \in E} \sum_{i=1}^C a_{mn,i}^{sd} / h_G^{sd}$  is always greater than or equal to 1, and is defined as the *stretch factor* for the routing path from the source node  $s$  to the destination node  $d$ . We define the hop count constraint to be as follows:

$$\sum_{m,n \in N, e_{mn} \in E} \sum_{i=1}^C a_{mn,i}^{sd} \leq \Gamma h_G^{sd}, \quad \forall s, d \in N. \quad (13)$$

where  $\Gamma \geq 1$  is a tunable parameter to set an upper bound on the routing stretch factor. Note that there is always a *trade off* between load balancing and shortest path routing [32]. This trade off can be controlled by using the tunable parameter  $\Gamma$ . By assigning  $\Gamma = 1$ , the routing part of the algorithm becomes the shortest path routing. By assigning  $\Gamma = \infty$ , the hop-count constraint (13) is relaxed. In general, the greater the tunable parameter  $\Gamma$ , the larger the feasible region.

## G. Optimization Problem

We now summarize our joint problem formulation. Given the physical topology graph  $G(N, E)$  and the parameters  $C, I, \Gamma, \Lambda, c^0, F_{mn}, \gamma^{sd}, h_G^{sd}$ , for all nodes  $m, n, s, d \in N$ ,

$$\begin{aligned} & \textbf{maximize} \quad \delta_{\min} \\ & x, y, c, a, \lambda, \delta_{\min} \\ & \textbf{subject to} \\ & x_{mn}^i = x_{nm}^i, \\ & x_{mn}^i \leq y_m^i, \\ & y_m^i \leq \sum_{n \in N, e_{mn} \in E} x_{mn}^i, \\ & \sum_{i=1}^C y_m^i \leq I, \\ & c_{mn}^i \leq x_{mn}^i c^0, \\ & c_{mn}^i + \sum_{p,q, e_{pq} \in F_{mn}} c_{pq}^i \leq c^0, \\ & \sum_{\substack{n \in N, \\ e_{mn} \in E}} \sum_{i=1}^C a_{mn,i}^{sd} - \sum_{\substack{n \in N, \\ e_{nm} \in E}} \sum_{i=1}^C a_{nm,i}^{sd} = \begin{cases} 1, & \text{if } s = m, \\ -1, & \text{if } d = m, \\ 0, & \text{otherwise,} \end{cases} \\ & \sum_{i=1}^C a_{mn,i}^{sd} \leq 1, \\ & \lambda_{mn}^i = \sum_{s,d \in N} a_{mn,i}^{sd} \gamma^{sd}, \\ & \lambda_{mn}^i \leq \Lambda c_{mn}^i, \\ & \delta_{\min} \leq (\Lambda c_{mn}^i - \lambda_{mn}^i) + \Lambda c^0 (1 - x_{mn}^i), \\ & \sum_{m,n \in N, e_{mn} \in E} \sum_{i=1}^C a_{mn,i}^{sd} \leq \Gamma h_G^{sd}, \end{aligned}$$

where

$$\begin{aligned} & x_{mn}^i, a_{mn,i}^{sd} \in \{0, 1\}, \quad y_m^i, c_{mn}^i, \lambda_{mn}^i, \delta_{\min} \geq 0, \\ & y_m^i \leq 1, \quad c_{mn}^i \leq c^0, \\ & \forall m, n, s, d \in N, \quad e_{mn} \in E, \quad \forall i = 1, \dots, C. \end{aligned} \quad (14)$$

Let  $W$  denote the number of source and destination pairs. Also let  $|N|$  and  $|E|$  denote the cardinality of sets  $N$  and  $E$ , respectively. The linear mixed-integer problem (14) has  $|E|C(1+W)$  integer variables and  $C(2|E| + |N|) + 1$  real variables. It also has  $1.5|E|C + |N|W$  equality and  $|E|(5C + W) + |N|(C+1) + W$  inequality constraints.

There are efficient commercial software (e.g., CPLEX [33]) to solve linear mixed-integer programs. Most of them use the branch-and-cut algorithm [34]. Problem (14) can easily be solved in practice for small-scale MC-WMNs. However, finding the optimal solutions are not trivial for large-scale networks. An alternative is to use some simple and efficient *metaheuristic* methods to find the sub-optimal solutions [35]. In this paper, we use the Iterated Local Search (ILS) [36] which is a simple and powerful metaheuristic algorithm. We will investigate the sub-optimality of the ILS algorithm in comparison with the optimal branch-and-cut algorithm in Section IV.

---

### Algorithm 1 Iterated Local Search

---

- 1: **set**  $K$  = the maximum number of iterations.
  - 2: **set**  $x_{mn}^1[1] = 1, \forall m, n \in N, e_{mn} \in E$ .
  - 3: **set**  $x_{mn}^2[1] = \dots = x_{mn}^I[1] = 0, \forall m, n \in N, e_{mn} \in E$ .
  - 4: **for**  $k = 1$  **to**  $K$  **do**
  - 5: Randomly choose  $p, q \in N$  such that  $e_{mn} \in E$ .
  - 6: **solve** problem (14) **subject to**

$$\begin{aligned} 0 & \leq a_{mn,i}^{sd} \leq 1, & \forall s, d, m, n \in N, e_{mn} \in E, \\ & \forall i \in \{1, \dots, C\} \\ x_{mn}^i & = x_{mn}^i[k], & \forall m, n \in N, e_{mn} \in E, m, n \notin \{p, q\}. \end{aligned}$$
  - 7: **set**  $x_{mn}^i[k+1] = x_{mn}^i, \forall m, n \in N, e_{mn} \in E, \forall i \in \{1, \dots, C\}$ .
  - 8: **end**
  - 9: **for** each source and destination pair  $(s, d)$  **do**
  - 10: **set**  $m = s$ .
  - 11: **while**  $m \neq d$  **do**
  - 12: **set**  $a_{mn,i}^{sd} = 1$  **where**  $\{i, n\} = \text{argmax}_{i,n} a_{mn,i}^{sd}$ .
  - 13: **set**  $m = n$ .
  - 14: **end**
  - 15: **for** all  $m, n \in N$  and all channels  $i \in \{1, \dots, C\}$  **do**
  - 16: **if**  $a_{mn,i}^{sd} \neq 1$ , **then set**  $a_{mn,i}^{sd} = 0$ .
  - 17: **end**
  - 18: **end**
- 

The pseudo-code for the proposed ILS algorithm is provided in Algorithm 1. In line 2, a fully connected single-channel logical topology is selected as the starting point. At each iteration, lines 4 and 5 are used to randomly select a pair of nodes  $p, q \in N$  (e.g., with probability that is proportional to the worst congestion status among the current logical links between them). From the two additional constraints in line 6, the integer constraint on the routing variable  $a$  is relaxed. Some variables are also set as constants in the current iteration. Therefore, the modified problem only has a few integer variables and can be solved easily. Given the sub-optimal topology formation, interface assignment and channel allocation solutions, the routing path from source  $s$  to destination  $d$  is assigned by traversing the logical topology

from source  $s$  to destination  $d$ , and by choosing the next hop based on the maximum observed value for routing variable  $a$  (lines 10-18). The intuitive justification is that if the relaxed  $a_{mn,i}^{sd}$  is close to 1, it indicates that it is better to forward the packets from source  $s$  to destination  $d$  on the logical link  $(m,n)$  over channel  $i$ . On the other hand, if the relaxed  $a_{mn,i}^{sd}$  is close to 0, it implies that it is better to avoid forwarding packets on logical link  $(m,n)$  over channel  $i$ .

#### IV. PERFORMANCE EVALUATION AND COMPARISON

In this section, we evaluate the performance of our proposed *TiMesh* MC-WMN architecture and compare it with the *Hyacinth* [1] and *CLICA* [2] architectures using *ns-2* simulations. We consider both UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) traffic. The *default* simulation model is as follows. The size of the network field is 1000 m  $\times$  800 m. Ten sample MC-WMNs are generated. Each MC-WMN consists of 30 routers. Four of them serve as gateways. The gateways are located at the four corners of the field. The communication and the interference ranges are 250 m and 450 m, respectively. Each router is equipped with three NICs. Six channels are available. Parameter  $\Gamma$  is set to two. The IEEE 802.11a standard with 54 Mbps data rate is being used. In each topology, there are 30 flows: 15 flows are internal, and 15 flows are external. For each internal flow, two non-gateway nodes are randomly selected to be the source and destination nodes. Each external flow is established between a randomly selected node and a gateway. For UDP traffic, the packet size is 1000 bytes and the transmission rate is 500 kbps. For TCP traffic, the packet size is 1020 bytes and the transmission rate is set by the TCP Vegas. The simulation time is 300 sec.

For UDP traffic, the performance metrics include: 1) *packet delivery ratio*: the total number of packets received by all destinations divided by the total number of packets transmitted by all sources; 2) *end-to-end delay*: the time takes for a packet to traverse the network from a source to a destination. For TCP traffic, the metrics include: 1) *aggregate throughput*: the total number of correctly received packets (in bits) at the destinations divided by the simulation time; 2) *round-trip time*:

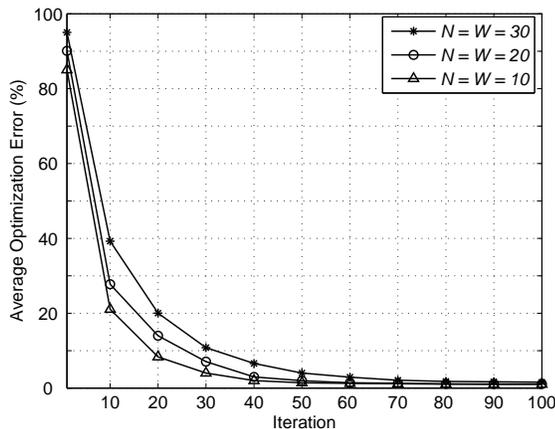


Fig. 2. Average optimization error versus the iteration number using the ILS algorithm when the number of nodes/flows vary from 10 to 30.

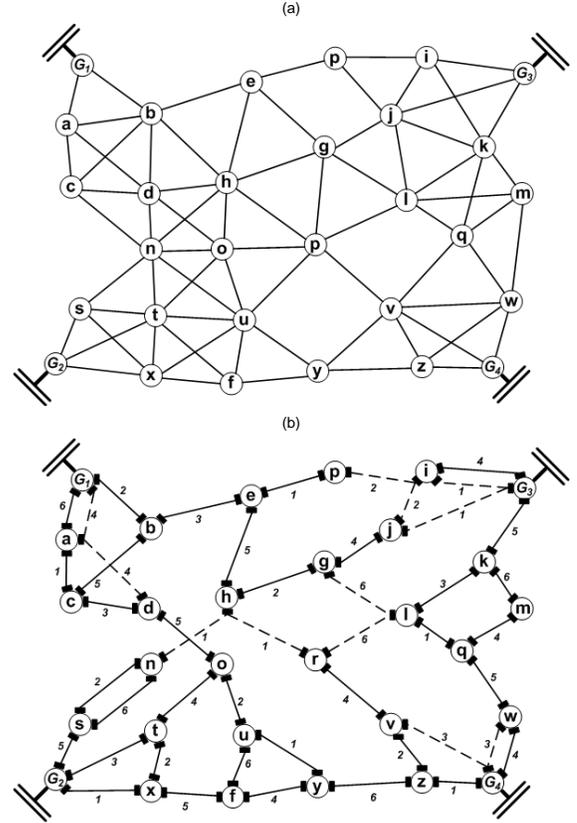


Fig. 3. A random topology with 30 routers. Each router is equipped with 3 NICs. (a) Physical topology, (b) Logical topology, interface assignment, and channel allocation. Solid lines are wireless links that use only exclusive (not shared) NICs. Dashed lines are the links that share an interface with some other links. The number on each link indicates the channel number.

the time delay between sending a TCP segment and receiving its acknowledgement.

#### A. Optimal and Sub-optimal Solutions

In this section, we compare the solutions for problem (14) obtained from the ILS algorithm with those obtained from the optimal branch-and-cut solver. Let  $\delta_{\min}^*$  denote the optimal value. Also let  $\delta_{\min}[k]$  denote the value obtained from the ILS algorithm after the  $k$ th iteration. We define  $(\delta_{\min}^* - \delta_{\min}[k]) / \delta_{\min}^*$  as the *optimization error*. Fig. 2 shows the average optimization error across all ten topologies versus the iteration number when the number of nodes and flows vary from 10 to 30. We can see that the average optimization error reduces when the number of iterations in ILS algorithm increases. After 50 iterations, the error is 1.4%, 1.9%, and 4% for 10, 20, and 30 nodes/flows, respectively. These results show that a near optimal solution can be achieved within limited number of iterations. For the results presented in the subsequent sections, the near optimal solutions of problem (14) are obtained from the ILS algorithm after 50 iterations.

#### B. Sample Logical Topology

Fig. 3(a) shows a sample physical topology. The corresponding logical topology, interface assignment, and channel allocation are shown in Fig. 3(b). For the physical topology

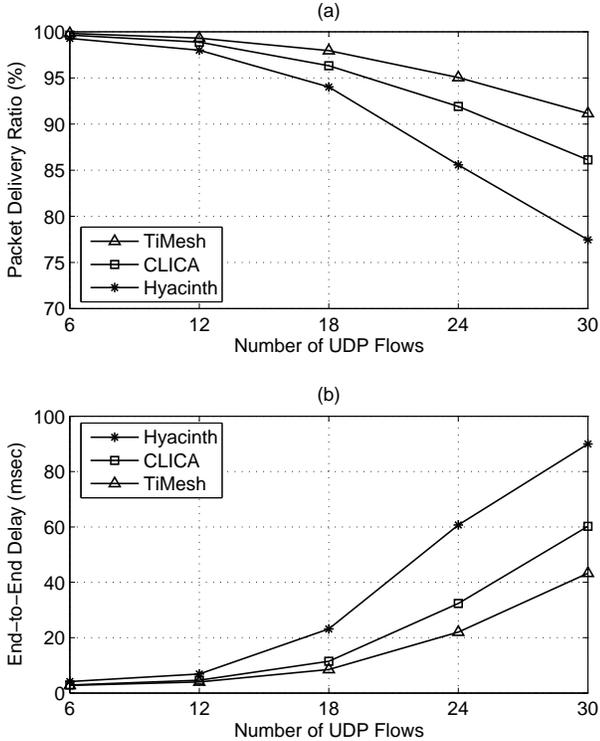


Fig. 4. Comparison between *TiMesh*, *Hyacinth*, and *CLICA* MC-WMN architectures in the presence of UDP traffic, (a) Packet delivery ratio, (b) End-to-end delay.

graph in Fig. 3(a), 72 pairs of neighboring nodes are within the communication range of each other. The logical topology in Fig. 3(b) includes at least one logical link between 43 pairs of neighboring nodes. There are two links between nodes  $G_1$  and  $a$ ,  $G_3$  and  $i$ ,  $G_4$  and  $w$ , as well as  $s$  and  $n$ . These links operate over distinct channels. In Fig. 3(b), there are 12 logical links that share an NIC with some others. The sharing of the logical links do not happen often as it reduces the corresponding link capacities. In the obtained routing solution, the 2-hop route  $\{o, t, G_2\}$  is replaced by the 3-hop route  $\{o, t, x, G_2\}$  to take the advantage of the unused logical link  $(t, x)$ .

### C. Impact of the Network Traffic

In this section, we first investigate the performance with UDP traffic. The number of active flows varies from 6 to 30. Fig. 4 shows the results of the packet delivery ratio and the average end-to-end delay. In this figure, each point is the average of measurements for all 10 simulated topologies. When the number of UDP flows increases, the network becomes congested. Since UDP does not have any congestion control mechanism, there is a reduction of the packet delivery ratio and an increase of the end-to-end delay. When there are 30 UDP flows, Fig. 4(a) shows that the packet delivery ratio obtained from *TiMesh* is 7% and 18% higher than *CLICA* and *Hyacinth*, respectively. Fig. 4(b) shows that the average end-to-end delay obtained from *TiMesh* is 28% and 52% lower than *CLICA* and *Hyacinth*, respectively.

Fig. 5 shows the results of the aggregated throughput and the average round-trip time when there are different number of

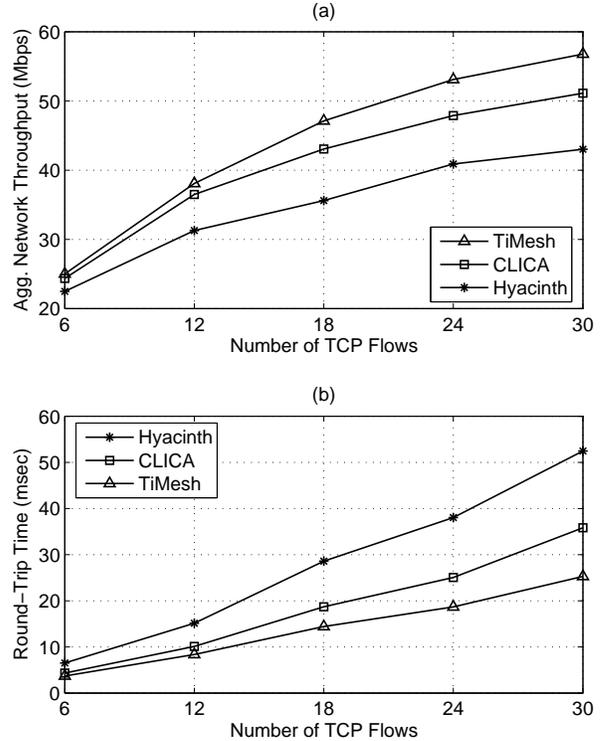


Fig. 5. Comparison between *TiMesh*, *Hyacinth*, and *CLICA* MC-WMN architectures in the presence of TCP traffic, (a) Aggregated throughput, (b) Average round-trip time.

TCP flows established in the network. In this figure, each point is the average of measurements for all 10 simulated topologies. If only a few flows are established (e.g., less than 6 flows), the *TiMesh*, *CLICA*, and *Hyacinth* architectures achieve almost the same performance. By increasing the number of flows, the network becomes congested and the round-trip time increases significantly. When there are 30 flows, Fig. 5(a) shows that the aggregated throughput obtained from *TiMesh* is 11% and 32% higher than *CLICA* and *Hyacinth*, respectively. Fig. 5(b) shows that the average round-trip time obtained from *TiMesh* is 29% and 52% lower than *CLICA* and *Hyacinth*, respectively.

The better performance of *TiMesh* can be explained based on the features of the three architectures. Unlike *Hyacinth* that concentrates the traffic on long routing paths with highly loaded links (especially the links connected to the gateways), *TiMesh* distributes and balances the traffic among different links. It also assigns shorter routing paths. *TiMesh* has two distinct advantages when it is compared to *CLICA*. The logical topology created by *TiMesh* depends on the expected traffic demand. *TiMesh* also allows having multiple links between the same pair of routers. This further increases the effective data transmission rate between two neighboring routers.

### D. Impact of the Number of NICs and Channels

In this section, we compare the performance by varying the number of interfaces and the available channels. Thirty TCP traffic flows are generated in the network. Results for the aggregate throughput and average round-trip time are shown in Table I. We can see that both *TiMesh* and *CLICA*

improve the performance significantly when the number of channels is increased from 6 to 9. The performance gain is less for *Hyacinth* as it has fewer logical links in its topology and cannot efficiently use the available resources. When the number of NICs increases from 3 to 4, the aggregate network throughput increases by 36%, 29%, and 37% for *TiMesh*, *CLICA*, and *Hyacinth*, respectively. The average round-trip time also decreases by 92%, 17%, and 48%, respectively. The observed high performance gain for *TiMesh* is due to the fact that it uses the extra available NICs to assign multiple logical links between the same pair of routers.

### E. Impact of the Tunable Parameter $\Gamma$

To investigate the impact of the tunable parameter  $\Gamma$  in *TiMesh*, 30 UDP flows are generated. Let  $\tau$  denote the average stretch factor across all flows in the network. That is,

$$\tau = \frac{1}{W} \sum_{s,d \in N, \gamma^{sd} \neq 0} \sum_{m,n \in N, e_{mn} \in E} \sum_{i=1}^C \frac{a_{mn,i}^{sd}}{h_G^{sd}}.$$

From (13) we have,  $\tau \leq \Gamma$ . Fig. 6 shows the average packet delivery ratio and the average stretch factors when  $\Gamma$  is equal to 1, 1.5, 2, 2.5, 3, and  $\infty$ . The network performance degrades slightly when the shortest path routing is used (i.e.,  $\Gamma = 1$ ). This is expected as it reduces the load balancing aspect of the proposed algorithm; however, the algorithm still assigns channels according to the congestion level on each logical link. Note that even when the hop count constraint is relaxed (i.e.,  $\Gamma = \infty$ ), the average stretch factor  $\tau$  is equal to 1.19. It indicates that *TiMesh* still avoids assigning long routing paths. This is due to the fact that constraints (11) and (12) aim to maximize the available capacity of the bottleneck link.

### F. Fairness

Recall from Section III-E that *TiMesh* can achieve *max-min* fairness among logical links. To quantitatively measure the fairness that is attained among different flows, we let  $\Psi_{\text{PDR}}$  and  $\Psi_{\text{EED}}$  denote Jain's fairness indices [37] for packet delivery ratio and end-to-end delay, respectively. We have,

$$\Psi_{\text{PDR}} = \frac{(\sum_{s,d \in N, \gamma^{sd} \neq 0} \text{PDR}(s,d))^2}{W \sum_{s,d \in N, \gamma^{sd} \neq 0} \text{PDR}(s,d)^2},$$

where  $\text{PDR}(s,d)$  denotes the packet delivery ratio for the flow from source  $s$  to destination  $d$ . Fairness index  $\Psi_{\text{EED}}$  can be expressed similarly. The measured  $\Psi_{\text{PDR}}$  and  $\Psi_{\text{EED}}$  for *TiMesh*, *CLICA*, and *Hyacinth* architectures are shown in Table II. We can see that *TiMesh* offers better fairness among the flows. The lower fairness indices in *CLICA* and *Hyacinth* are the result of

TABLE I  
IMPACT OF VARYING THE NUMBER OF NICs AND CHANNELS

Architecture	Throughput (Mbps)			Round-Trip Time (msec)		
	$I=3, C=6$	$I=4, C=9$	$I=4, C=12$	$I=3, C=6$	$I=4, C=9$	$I=4, C=12$
<i>TiMesh</i>	57.9	68.3	92.0	25.5	21.8	11.6
<i>CLICA</i>	51.7	61.5	79.1	36.8	32.7	24.4
<i>Hyacinth</i>	43.1	48.2	66.4	52.7	49.9	41.0

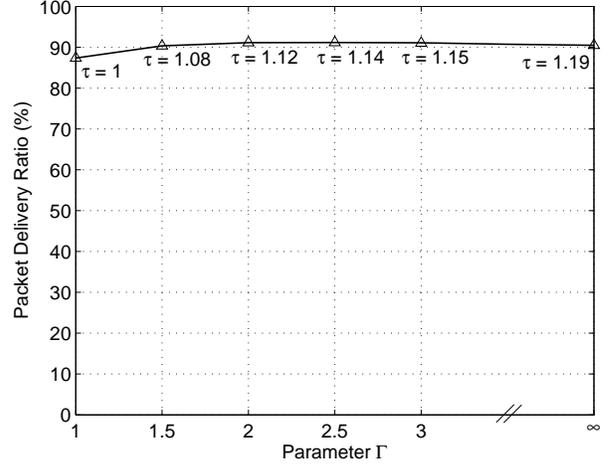


Fig. 6. The average packet delivery ratio versus different values for parameter  $\Gamma$ . The corresponding average stretch factor  $\tau$  is provided at each point.

TABLE II  
ACHIEVED FAIRNESS AMONG DIFFERENT FLOWS

Architecture	Fairness Index	
	Packet Delivery Ratio ( $\Psi_{\text{PDR}}$ )	End-to-End Delay ( $\Psi_{\text{EED}}$ )
<i>TiMesh</i>	0.914	0.903
<i>CLICA</i>	0.867	0.836
<i>Hyacinth</i>	0.712	0.678

several highly congested bottlenecks. Those flows that traverse the bottleneck links experience higher delays and more packet loss compared to the rest of the flows. This is also the case for *Hyacinth* where the links connected to the gateways are the bottlenecks. The proposed topology control, routing, interface assignment, and channel allocation algorithms in *TiMesh* manage to avoid different links experiencing different congestion levels. Thus, different flows that are traversing different links achieve similar performance.

### V. CONCLUSIONS

In this paper, we proposed the *TiMesh* MC-WMN architecture by formulating the logical topology design, interface assignment, channel allocation, and routing as a joint linear mixed-integer optimization problem. Our model formulation takes into account the number of available NICs in routers, the number of available orthogonal frequency channels, expected traffic load between different source and destination pairs, and the effective capacity of the logical links. The proposed scheme balances the load among logical links and provides higher effective capacity for the bottleneck link(s). We conducted extensive *ns-2* simulation experiments to evaluate our algorithm and compared it with *Hyacinth* and *CLICA* MC-WMN architectures. Simulation results show that our proposed *TiMesh* architecture provides a higher aggregated network throughput and a lower end-to-end delay for both TCP and UDP traffic. The available NICs and frequency channels are also better utilized. The *TiMesh* also offers better fairness among different flows.

For future work, we plan to study the effect of using directional antenna to reduce the co-channel interference between

some of the neighboring links in MC-WMNs. We also plan to extend our model by including the use of partially overlapping channels in the framework.

#### ACKNOWLEDGMENT

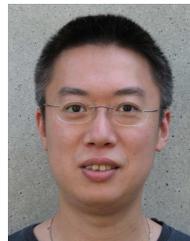
The authors would like to thank the anonymous reviewers for their helpful comments which improved the presentation of this paper.

#### REFERENCES

- [1] A. Raniwala and T. Chiuah, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc. of IEEE Infocom'05*, Miami, FL, March 2005.
- [2] M. Marina and S. Das, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," in *Proc. of Broadnet'05*, Boston, MA, Oct. 2005.
- [3] I. Akyildiz, X. Wang, and X. Wang, "Wireless mesh networks: A survey," *Elsevier Journal on Computer Networks*, vol. 47, pp. 445–487, March 2005.
- [4] "MIT Roofnet," <http://pdos.csail.mit.edu/roofnet/doku.php>.
- [5] "Microsoft Mesh Networking," <http://research.microsoft.com/mesh/>.
- [6] "Tropos Networks," <http://www.tropos.com>.
- [7] "BelAir Networks," <http://www.belairnetworks.com>.
- [8] "Status of Project IEEE 802.11s ESS Mesh Networking," [http://grouper.ieee.org/groups/802/11/Reports/tgs\\_update.htm](http://grouper.ieee.org/groups/802/11/Reports/tgs_update.htm), January 2007.
- [9] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering wireless systems with multiple radios," *ACM Computer Communication Review*, vol. 34, pp. 39–46, Oct. 2004.
- [10] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proc. of ACM MobiCom'05*, Cologne, Germany, Sept. 2005.
- [11] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proc. of ACM MobiCom'03*, San Diego, CA, Sept. 2003.
- [12] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," in *Proc. of ACM MobiHoc'05*, Urbana-Champaign, IL, May 2005.
- [13] A. Raniwala, K. Gopalan, and T. Chiuah, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM Mobile Computing and Communications Review*, vol. 8, pp. 50–65, April 2004.
- [14] A. Brezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks - a partitioning approach," in *Proc. of ACM MobiCom'06*, Los Angeles, CA, Sept. 2006.
- [15] B. Ramin, "Channel allocation in 802.11-based mesh networks," in *Proc. of IEEE Infocom'06*, Barcelona, Spain, April 2006.
- [16] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. of ACM MobiCom'04*, Philadelphia, PA, Sept. 2004.
- [17] A. K. Das, R. Vijayakumar, and S. Ray, "Static channel assignment in multi-radio multi-channel 802.11 wireless mesh networks," in *Proc. of IEEE Globecom*, San Francisco, CA, November 2006.
- [18] R. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. of IEEE Infocom'06*, Barcelona, Spain, April 2006.
- [19] X. Meng, K. Tan, and Q. Zhang, "Joint routing and channel assignment in multi-radio wireless mesh networks," in *Proc. of IEEE ICC*, Istanbul, Turkey, June 2006.
- [20] Y. Y. Chen, S. C. Liu, and C. Chen, "Channel assignment and routing for multi-channel wireless mesh networks using simulated annealing," in *Proc. of IEEE Globecom*, San Francisco, CA, November 2006.
- [21] P. Kyasanur and N. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Proc. of IEEE WCNC'05*, New Orleans, LA, March 2005.
- [22] A. H. Mohsenian Rad and V. W. S. Wong, "Joint optimal channel assignment and congestion control in multi-radio wireless mesh networks," in *Proc. of IEEE ICC'06*, Istanbul, Turkey, June 2006.
- [23] —, "Joint optimal channel allocation, interface assignment, and MAC design for multi-channel wireless mesh networks," in *Proc. of IEEE Infocom'07*, Anchorage, AK, May 2007.
- [24] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proc. of ACM MobiCom'05*, Cologne, Germany, Sept. 2005.
- [25] D. P. Bertsekas and R. Gallager, *Data Communications*, 2nd ed. Prentice Hall, 1992.
- [26] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An approach to alleviate link overload as observed on an IP backbone," in *Proc. of IEEE Infocom'03*, San Francisco, CA, April 2003.
- [27] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiron, F. Tobagi, and C. Diot, "Analysis of measured single-hop delay from an operational backbone network," in *Proc. of IEEE Infocom'02*, New York, NY, June 2002.
- [28] A. Narula-Tam and E. Modiano, "Dynamic load balancing in WDM packet networks with and without wavelength constraints," *IEEE J. on Selected Areas in Communications*, vol. 18, pp. 1972–1979, Oct. 2000.
- [29] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. on Selected Areas in Communications*, vol. 23, pp. 104 – 116, Jan. 2005.
- [30] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. on Networking*, vol. 8, pp. 556–567, Oct. 2000.
- [31] J. M. Jaffe, "Bottleneck flow control," *IEEE Trans. on Communications*, vol. 29, pp. 954–962, July 1981.
- [32] J. Gao and L. Zhang, "Tradeoffs between stretch factor and load balancing ratio in routing on growth restricted graphs," in *Proc. of ACM Symposium on Principles of Distributed Computing*, St. Johns, Canada, July 2005.
- [33] "ILOG CPLEX," <http://www.ilog.com/products/cplex/>, 2006.
- [34] A. Lucena and J. E. Beasley, "Branch and cut algorithms," in *Advanced in Linear and Integer Programming*, J. E. Beasley, Ed. Oxford University Press, 1996.
- [35] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, pp. 268–308, Sept. 2003.
- [36] H. R. Lourenco, O. Martin, and T. Stutzle, "Iterated local search," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds. Kluwer Academic Publishers, 2002, pp. 321–353.
- [37] R. Jain, W. Hawe, and D. Chiu, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report DEC-TR-301, Sept. 1984.



**Amir-Hamed Mohsenian-Rad** is a Ph.D. candidate in electrical engineering at the University of British Columbia (UBC), Vancouver, BC. He received his B.Sc. degree from Amir-Kabir University of Technology, Tehran, Iran (2002) and his M.Sc. degree from Sharif University of Technology, Tehran, Iran (2004), both in electrical engineering. From March to July 2007, he was also a visiting scholar at Princeton University, Princeton, NJ. Mr. Mohsenian-Rad has been granted the UBC Graduate Fellowship as well as the Pacific Century Graduate Scholarship from the British Columbia Provincial Government. His research interests are in the area of optimization theory and its applications in computer communications and wireless networking.



**Vincent W.S. Wong** received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. He is currently an Associate Professor in the Department of Electrical and Computer Engineering at UBC. His research interests are in resource and mobility management for wireless mesh networks, wireless sensor networks, and heterogeneous wireless networks. Dr. Wong is an associate editor of the IEEE Transactions on Vehicular Technology. He serves as TPC member in various conferences, including the IEEE International Conference on Communications (ICC) and Globecom. He is a senior member of the IEEE and a member of the ACM.