Maximizing Multicast Call Acceptance Rate in Multi-Channel Multi-Interface Wireless Mesh Networks

Hon Sun Chiu and Kwan L. Yeung

***Abstract*—In this paper, we consider the problem of constructing bandwidth-guaranteed multicast tree in multi-channel multi-interface wireless mesh networks. We focus on the scenario of dynamic multicast call arrival, where each call has a specific bandwidth requirement. A call is accepted if a multicast tree with sufficient bandwidth on each link can be constructed. Intuitively, if the carried load on both the most-heavily loaded channel and the most-heavily loaded node is minimized, the traffic load in the network will be balanced. If the network load is balanced, more room will be available for accommodating future calls. This would maximize the call acceptance rate in the network. With the above notion of load balancing in mind, an Integer Linear Programming (ILP) formulation is formulated for constructing bandwidth-guaranteed tree. We show that the above problem is NP-hard, and an efficient heuristic algorithm called Largest Coverage Shortest-Path First (LC-SPF) is devised. Simulation results show that LC-SPF yields comparable call acceptance rate as the ILP formulation, but with much shorter running time.**

***Index Terms*—Wireless mesh network, bandwidth-guaranteed multicast tree, load balancing.**

I. Introduction

Wireless Mesh Networks (WMNs) have emerged as a practical solution for the wireless extension of the broadband Internet. A WMN consists of stationary wireless mesh routers, which are connected to one another in a multi-hop manner to form a wireless backbone. End user mobile devices can connect to the wireless backbone via some mesh routers within their transmission range. With the increasing computational power of mobile devices in recent years, many multicast applications (e.g., video streaming) for wired networks are adopted by wireless networks [1], where efficient multicast algorithms are urgently needed to regulate the increased multicast traffic in the wireless backbone.

Unlike the wired networks, the broadcasting nature of wireless transmissions allows all neighbors to receive the same copy of data with the source node only transmitting once. This is known as wireless broadcast advantage (WBA) [2]. On the other hand, the interference induced by a transmission can suppress the transmissions on other wireless links within its interference range, which severely limits the capacity of a wireless network. To improve the network capacity, mesh routers of WMNs can be equipped with multiple network interface cards to allow parallel transmissions over multiple orthogonal channels [3], [4]. This provides a new dimension to network resources management, and makes the multicast routing in WMNs more challenging.

In this paper, we consider a wireless mesh network with dynamic multicast call arrivals. We envision that multicast video streaming will be a killer application for WMNs. In order to ensure video quality, each streaming session should be provisioned with sufficient network bandwidth. Accordingly, each multicast call is characterized by a specific bandwidth requirement. A call is accepted if a multicast distribution tree can be established for connecting the source node with all the receiving nodes, and with sufficient bandwidth reserved on each link. In order to maximize the call acceptance rate, each multicast tree must be judicially constructed. In this paper, the problem of constructing bandwidth-guaranteed multicast tree is studied. Intuitively, if the carried load on both the most-heavily loaded channel and the most-heavily loaded node is minimized, the traffic load in the network will be balanced and more room will be available for accommodating future calls. This would maximize the call acceptance rate in the network. With the above notion of load balancing in mind, an Integer Linear Programming (ILP) formulation is first formulated for constructing bandwidth-guaranteed tree. But solving ILP is time consuming. An efficient heuristic algorithm called Largest Coverage Shortest-Path First (LC-SPF) is thus devised.

In the next section, we review the related work on multicast tree construction and scheduling. In Section III, we state the network model used in our design and show that the problem of constructing bandwidth-guaranteed tree is NP-hard. In Section IV, an ILP is formulated for finding bandwidth-guaranteed broadcast and multicast trees. In Section V, an efficient heuristic algorithm (LC-SPF) for constructing multicast tree is designed. To facilitate conflict-free transmissions, a simple scheduling algorithm for allocating transmission time slots to each mobile user is introduced in Section VI. We compare LC-SPF with ILP formulations in Section VII. Finally, we conclude this paper in Section VIII.

II. Related Work

Multicast routing and scheduling [4]–[27] has been extensively studied under the context of both mobile ad hoc networks (MANETs) and wireless mesh networks (WMNs).

Manuscript received September 1, 2009; revised December 11, 2009 and March 3, 2009; accepted April 20, 2010. The associate editor coordinating the review of this paper and accepting it for publication was D. Tarchi.

The authors are with the University of Hong Kong, Dept. of EEE, CYC Bldg, HKU, Pokfulam Rd., Hong Kong (e-mail: {hschiu, kyeung}@eee.hku.hk).

Digital Object Identifier 10.1109/TCOMM.2010.062310.091323

While minimizing the energy consumption of a multicast tree is critical in MANETs [5], [6], it is usually not the primary consideration in WMNs where abundant power supply is available. In WMNs, on the other hand, the mesh nodes are usually at fixed locations with constant power supply, which allow WMNs to support more advanced services. Many multicast protocols have been proposed for WMNs with different objectives [7]–[26]. In [4], [8], [21]–[25], the objective is to maximize the tree bandwidth, which can be achieved by employing network coding and transmission power control [21], setting up path to each receiver via high-bandwidth links [22], or minimizing the mutual interference by proper scheduling [23] and/or using multiple channels [4], [8], [24], [25]. In this paper, we consider each call with a specific bandwidth requirement. As a result, reserving extra bandwidth is a waste of network resources.

Internet access is considered in [13], [16]–[19], where a single spanning tree is constructed with the root at the gateway node. In [13], multiple multicast sessions are taken into consideration. With the information of all sessions given, the algorithm first constructs a spanning tree rooted at the gateway, then prunes the tree to accommodate all sessions. A multicast tree can also be constructed to maximize reliability [14], [15], minimize latency [11], [12], the number of transmissions [9], [10], or the number of channels required [20].

Notably, all multicast protocols above [4], [8]–[25] aim at constructing a single tree. Admission control of real-time multicast call arrivals has not been addressed. Although multiple multicast trees are constructed in [7], they are used to carry the multiple descriptions of a video stream of a single multicast session. For unicast transmissions, some call admission control schemes have been proposed [27], [28], where existing calls may be re-routed in order to accept a new call. Although re-routing for unicast calls could be feasible, it may not be true for multicast calls due to the amount of processing and signaling overhead involved.

To the best of our knowledge, the only related work on multicast call admission control is [26]. In [26], while establishing a tree upon a call arrival, heuristic algorithms are designed to exploit the link-rate diversity for maximizing the residual channel bandwidth in the network. Although exploiting link-rate diversity is an interesting topic, it is not clear how they can be extended to a multi-channel and multi-interface network. In this paper, we aim at constructing a bandwidth-guaranteed multicast tree upon each call arrival in a multi-channel multi-interface WMN, and without re-routing any existing calls.

Constructing a bandwidth-guaranteed tree involves both routing and scheduling. Routing determines the tree topology and scheduling maps the required transmission time onto the frame structure of each tree link (please refer to Section VI). The joint optimization of routing and scheduling is shown to be NP-hard [23]. A practical approach [4], [11], [13], [23], [25], [26] is to solve the problem in two decoupled steps, where the tree is constructed first and the scheduling is done next. In this paper, we follow the same decoupled approach while considering a multi-channel multi-interface network. Specifically, time fractions that a tree link should be active are determined together with multicast tree construction as that in [4], [13], [25], [26].

TABLE I
DEFINITION OF VARIABLES USED IN ILP

x	Maximum channel utilization in the network
y	Minimum residual interface capacity in terms of time fraction
t_u	Integer variable; number of interfaces at node u
r_u	Binary variable; equals to 1 if node u is a root, 0 otherwise
e_{uv}	Binary variable; equals to 1 if link (u, v) is on-tree, 0 otherwise
s_u	Voltage value $(0 \leq s_u \leq 1)$ at node u for loop prevention
f_{uv}^k	Time fraction $(0 \leq f_{uv}^k \leq 1)$ for link (u, v) to be active in carrying data using channel k
n_{mv}^k	Time fraction $(0 \leq n_{mv}^k \leq 1)$ that node m transmits on channel k and interferes with the receiving at node v
Y_{uv}^k	Existing total time fraction for (u, v) to be active on channel k
B_u^k	Existing total time fraction for node u to transmit on channel k
F_T	Bandwidth required (in time fraction) by the new call
S	Denotes the root/source of the new call
σ	A small constant (σ = 0.0001 is adopted in Section VII)

III. System Model and Problem Statement

A. System Model

We consider a multi-channel wireless mesh network (WMN) with each node equipped with multiple interfaces. We denote a WMN by a connected graph $\boldsymbol{G = (V, E)}$, the set of orthogonal channels by $\boldsymbol{C}$, and the set of nodes within the interference range of node v, including node v itself, by $\boldsymbol{I_v}$. The number of interfaces at each node can be different, which is pre-determined and is given by $t_u \forall u \in \boldsymbol{V}$. The receiver conflict avoidance [29] interference model and a schedule-based MAC protocol are assumed. With the receiver conflict avoidance interference model, a receiver is clear to receive if there is only one (legitimate) transmission within its interference range. In a schedule-based MAC protocol, conflict-free transmission is ensured by assigning nodes within each other's interference range to either send on different channels, or send on the same channel but at different time slots (in a frame). To construct a frame-based schedule, we assign a time fraction $f_{uv}^k \in [0, 1]$ to each link (u, v) to represent the portion of a frame for node u to transmit data to node v using channel k. Then, a scheduling algorithm (in Section VI) is needed to transform the time fractions into time slots. We have to make sure f_{uv}^k is schedulable in the sense that all interfering nodes can transmit within a frame duration. It should be noted that the bandwidth of a link is directly proportional to the sum of allocated time fraction over all channels.

We represent each multicast call using a 3-tuple (S, $\boldsymbol{R}$, F_T), where S is the source of the multicast data (i.e., root of the multicast tree), $\boldsymbol{R}$ is the set of receivers, and F_T is the bandwidth requirement of the multicast session. Without loss of generality, F_T is expressed using time fractions (e.g., F_T = 0.1 is equivalent to the 10% bandwidth of a channel). Note that broadcast (unicast) is a special scenario of multicast, with all nodes (one node) in the network being the members of a multicast session. Hence, for a broadcast request, $\boldsymbol{R}$ can be omitted. We consider a WMN with some on-going calls in the network. The existing load on a link (u, v) using channel k is denoted by Y_{uv}^k, the total time fraction (of a time frame) that channel k on this link is active (for carrying on-going calls), and the existing load at node u on channel k is denoted by B_u^k, the total time fraction that node u transmits on channel k to some neighboring nodes.

B. Problem Statement

When a new multicast call arrives, our objective is to find the *best* multicast tree without interrupting/re-routing any on-going calls. A key issue is how to measure the quality of a multicast tree. Due to the broadcasting nature of wireless transmission, the channel condition is affected by other nodes' transmissions within the interference range of the concerned node. Thus in wireless networks, channel utilization differs depending on the “observing" node's location.

Definition 1 *The utilization of channel k as observed by node v is defined as the total time fractions assigned to all nodes (including node v) within node v's interference range for transmission using channel k.*

Let n^k_{mv} denote the interference generated by node m in channel k as observed by node v (as defined in Table I). Then the maximum channel utilization, denoted by x, is defined as

$$x = \max_{\forall v \in \boldsymbol{V}, \forall k \in \boldsymbol{C}} \left\{ \sum_{m \in \boldsymbol{I_v}} n^k_{mv} \right\} \tag{1}$$

In (1), the sum of n^k_{mv} for the set $\boldsymbol{I_v}$ (including node v itself) is equivalent to obtaining the total amount of transmission in channel k within the interference range of node v. Please note that node v's transmission is also counted as it contributes to the channel utilization. In other words, it is the utilization of channel k as observed by node v. According to Definition 1, different nodes may observe different utilizations for a given channel. The maximum/worst-case channel utilization (x) is the utilization of the most congested channel in the network. By having x minimized, the loading on different *channels* (as perceived by different nodes) will be more balanced. This leaves more room for accommodating future calls.

On the other hand, channels may have spare capacity but a node cannot accept additional calls if it does not have sufficient interfaces to handle the new call. To this end, we also want to maximize the residual interface capacity at a node. Let y denote the minimum residual interface capacity, and it is defined as

$$y = \min_{\forall v \in \boldsymbol{V}, \forall (v,w) \in \boldsymbol{E}} \left\{ t_v - \sum_{k \in \boldsymbol{C}} \sum_{(u,v) \in \boldsymbol{E}} (f^k_{uv} + Y^k_{uv}) - \sum_{k \in \boldsymbol{C}} (f^k_{uv} + B^k_v) \right\} \tag{2}$$

In (2), t_v is the number of interfaces at node v. The first and second summation terms in the braces represent node v's total ingress and egress traffic, respectively. By maximizing the residual capacity of the node with the minimum residual capacity, more load can be allocated to a *node* with higher capacity.

Obviously, both x and y are critical in determining the quality of a multicast tree. We thus propose a joint objective function based on them, or $(x - \beta y)$. (Note that a linear function is required in formulating an ILP.) By minimizing $(x - \beta y)$ (which can be negative), the maximum channel utilization can be minimized, and the minimum residual interface capacity of a node can be maximized at the same time. The weighting factor β denotes the relative importance of x and y, and its value can be obtained empirically. Then, the problem of constructing a bandwidth-guaranteed multicast tree can be formulated as follows:

$$\text{Objective} \qquad \text{minimize } x - \beta y$$

$$\text{subject to} \quad \begin{cases} \text{Tree topology covers all receivers} \\ \text{Time fractions are schedulable} \\ \text{Interference constraints are not violated} \\ \text{Interface constraints are not violated} \\ \text{On-going calls are not interrupted} \end{cases}$$

Theorem 1 *Channel assignment and routing for the joint objective function $(x - \beta y)$ is NP-hard.*

Proof: Consider a special case with the weighting factor $\beta = 0$. The problem is reduced to a *simpler* problem of only minimizing the maximum utilization of a channel (among all channels) by channel assignment and routing. Further assume that the multicast tree routing is given. We only focus on the channel assignment sub-problem for minimizing the maximum channel utilization. From Definition 1, channel utilization is location dependent and is defined as the total interference observed by the concerned node. Accordingly, minimizing the maximum channel utilization is equivalent to minimizing the interference in the wireless network.

In the literature, it has been shown that minimizing the interference in wireless network by channel assignment is actually solving the Max *K*-cut problem in the corresponding conflict graph [30]. Since the Max *K*-cut problem is NP-hard [31], so is the concerned channel assignment problem. Jointly solving the routing sub-problem and with $\beta > 0$ in the joint objective function $(x - \beta y)$ adds another dimension of complexity to the problem. Therefore, our joint optimization problem is also NP-hard. ■

IV. ILP Formulation

A. Bandwidth-Guaranteed Broadcast Tree

To simplify our discussion, we first consider the construction of a bandwidth-guaranteed broadcast tree, which is a multicast session involving all nodes in the network. When a broadcast call arrives at node S (then S is the tree root) and with bandwidth requirement F_T, a broadcast tree is to be built to carry this call. Based on the variables defined in Table I, the ILP for bandwidth-guaranteed broadcast tree construction is formulated below.

$$\text{Objective} \qquad \text{minimize } \{x - \beta y\} \tag{3}$$

subject to

$$r_{\mathrm{S}} = 1 \qquad \mathrm{S} \in \boldsymbol{V} \tag{4}$$

$$\textstyle\sum_{u \in \boldsymbol{V}} r_u = 1 \tag{5}$$

$$\textstyle\sum_{(u,v) \in \boldsymbol{E}} e_{uv} = 1 - r_v \qquad \forall v \in \boldsymbol{V} \tag{6}$$

$$e_{uv} + e_{vu} \leq 1 \qquad \forall (u,v) \in \boldsymbol{E} \qquad (7)$$

$$s_v - s_u \geq \sigma e_{uv} - (1 - e_{uv}) \qquad \forall (u,v) \in \boldsymbol{E} \qquad (8)$$

$$f_{uv}^k \leq e_{uv} \qquad \forall (u,v) \in \boldsymbol{E}, \forall k \in \boldsymbol{C} \qquad (9)$$

$$\textstyle\sum_{k \in \boldsymbol{C}} f_{uv}^k \geq e_{uv} \cdot F_T \qquad \forall (u,v) \in \boldsymbol{E} \qquad (10)$$

$$f_{uv}^k - f_{uw}^k \leq 2 - e_{uv} - e_{uw} \qquad \forall (u,v),(u,w) \in \boldsymbol{E}, \forall k \in \boldsymbol{C} \qquad (11)$$

$$\textstyle\sum_{k \in \boldsymbol{C}} \sum_{(u,v) \in \boldsymbol{E}} (f_{uv}^k + Y_{uv}^k) + \sum_{k \in \boldsymbol{C}} (f_{vw}^k + B_v^k) \leq t_v \qquad \forall v \in \boldsymbol{V}, \forall (v,w) \in \boldsymbol{E} \qquad (12)$$

$$n_{mv}^k \geq f_{mn}^k + B_m^k \qquad \forall v \in \boldsymbol{V}, \forall m \in \boldsymbol{I_v}, \forall k \in \boldsymbol{C}, \forall (m,n) \in \boldsymbol{E} \qquad (13)$$

$$f_{uv}^k + B_u^k + \sum_{\substack{m \in \boldsymbol{I_v} \\ m \neq u}} n_{mv}^k \leq 1 \qquad \forall (u,v) \in \boldsymbol{E}, \forall k \in \boldsymbol{C} \qquad (14)$$

$$x \geq \textstyle\sum_{m \in \boldsymbol{I_v}} n_{mv}^k \qquad \forall v \in \boldsymbol{V}, \forall k \in \boldsymbol{C} \qquad (15)$$

$$y \leq t_v - \textstyle\sum_{k \in \boldsymbol{C}} \sum_{(u,v) \in \boldsymbol{E}} (f_{uv}^k + Y_{uv}^k) - \sum_{k \in \boldsymbol{C}} (f_{vw}^k + B_v^k) \qquad \forall v \in \boldsymbol{V}, \forall (v,w) \in \boldsymbol{E} \qquad (16)$$

In the objective function (3), x is the maximum channel utilization and y is the minimum residual interface capacity. x and y are identified by constraints (15) and (16), respectively. Constraints (4)-(8) ensure that the resulting topology is a broadcast tree, where (4) and (5) specify that the tree has only one root at S, (6) assigns an ingress link to every non-root node, (7) indicates that the tree links are unidirectional, and (8) prevents the formation of routing loop. To carry the new call, time fractions on different channels are assigned to tree links according to (9)-(14). Constraints (9) and (10) ensure that time fractions are assigned to the tree links only, with the total amount not less than the requested time fraction F_T. Constraint (11) describes the property of wireless broadcast advantage (WBA), (12) prevents oversubscription of the interfaces at a node, (13) defines the interference generated by node m on channel k as observed by node v. Note that due to WBA, all outgoing links are treated as a single transmission. Constraint (14) states the sufficient condition for the assigned time fractions to be schedulable.

Theorem 2 *The obtained time fractions are schedulable if constraints (12) and (14) are satisfied.*

Proof: In a slotted system, the time fraction represents the portion of a frame. We can always find an integer M such that all $f_{uv}^k M$ are integers, which also give the number of time slots that link (u,v) is active. Indeed, M is the frame duration in slots. From (12) and (14), we have, for any channel k,

$$\sum_{k \in \boldsymbol{C}} \sum_{(u,v) \in \boldsymbol{E}} (f_{uv}^k + Y_{uv}^k) M + \sum_{k \in \boldsymbol{C}} (f_{vw}^k + B_v^k) M \leq t_v M \qquad (17)$$

$$(f_{uv}^k + B_u^k) M + \sum_{m \in \boldsymbol{I_v}, m \neq u} n_{mv}^k M \leq M \qquad (18)$$

In other words, (17) shows that at most $t_v M$ time slots are required for node v to transmit and receive in different channels. Note that the t_v interfaces on node v can operate simultaneously in different channels, thus the total time slots required is at most M time slots, i.e., within one frame. In addition, (18) shows that the number of time slots required by the link (u,v) and all interfering sources is at most M time slots, which is again within one frame. Thus we can conclude that the obtained time fractions are schedulable. ■

When a broadcast call arrives, the above ILP is solved. If a broadcast tree is found, the call is accepted. The running variables Y_{uv}^k and B_u^k are then updated to reflect that the call is accepted. Otherwise, the call is rejected. For an accepted call, a scheduling algorithm (in Section VI) is then employed to assign the time fractions into a frame for conflict-free transmissions.

B. Bandwidth-Guaranteed Multicast Tree

We next generalize the ILP for broadcast tree construction to multicast. In particular, we only need to replace the broadcast tree topology constraints (4)-(8) by the following multicast tree constraints:

$$r_{\text{S}} = 1 \qquad \text{S} \in \boldsymbol{V} \qquad (19)$$

$$\textstyle\sum_{u \in \boldsymbol{V}} r_u = 1 \qquad (20)$$

$$\textstyle\sum_{(u,v) \in \boldsymbol{E}} e_{uv} = 1 \qquad \forall v \in \boldsymbol{R} \qquad (21)$$

$$\textstyle\sum_{(u,v) \in \boldsymbol{E}} e_{uv} \leq 1 - r_v \qquad \forall v \in \boldsymbol{V} \setminus \boldsymbol{R} \qquad (22)$$

$$\textstyle\sum_{(u,v) \in \boldsymbol{E}} e_{uv} \geq e_{vw} - r_v \qquad \forall (v,w) \in \boldsymbol{E}, v \in \boldsymbol{V} \setminus \boldsymbol{R} \qquad (23)$$

$$\textstyle\sum_{w \in \boldsymbol{V}} e_{vw} \geq e_{uv} \qquad \forall (u,v) \in \boldsymbol{E}, v \in \boldsymbol{V} \setminus \boldsymbol{R} \qquad (24)$$

$$e_{uv} + e_{vu} \leq 1 \qquad \forall (u,v) \in \boldsymbol{E} \qquad (25)$$

$$s_v - s_u \geq \sigma e_{uv} - (1 - e_{uv}) \qquad \forall (u,v) \in \boldsymbol{E} \qquad (26)$$

Constraints (19) and (20) specify that there is only one root at node S. Constraint (21) ensures that there is only one ingress link to each receiver of a multicast group. Constraint (22) describes that it is not necessary to have an ingress link for non-member nodes (i.e., neither root nor receiver). According to (23), if there is an outgoing link from any non-member node, then there must be an ingress link to it. From (24), there should not be any ingress link to a non-member node if it is not assigned with an outgoing link. Finally, (25) indicates that the tree links are unidirectional, and (26) prevents the formation of routing loop.

V. Heuristic Algorithm

Solving the ILPs for tree construction is time consuming, and is therefore not suitable for real-time call admission. In this section, two efficient heuristic algorithms are designed, one for solving the bandwidth-guaranteed broadcast tree, and the other for multicast tree. In the following, we first describe a heuristic algorithm (greedy expansion) for the bandwidth-guaranteed broadcast tree construction. Then we modify it to support multicast, and we call the resulting algorithm largest coverage shortest-path first (LC-SPF). It should be noted that broadcast is a special case of multicast. So a LC-SPF can also be used to construct a broadcast tree. But owing to the additional constraints in constructing a multicast tree, greedy expansion is more efficient.

A. Greedy Expansion for Broadcasting

To construct a broadcast tree that spans all nodes in the network, we start from the source node (i.e., the root of the tree). Then we (greedily) find the most *efficient* forwarding nodes recursively until all nodes in the network are attached to the tree. The efficiency of a forwarding node is determined by its potential in reducing the channel and interface utilization in the network. From a local perspective, a transmission using the least channel and interface resources is preferred. From a global perspective, due to WBA, selecting a transmission that covers the most uncovered (not-yet-on-tree) nodes minimizes the total number of transmissions required to cover all nodes in the network, consequently reducing the global channel utilization. With the above consideration in mind, an efficient forwarding node can be identified as follows.

Among all the *non-forwarding* on-tree nodes, we first identify a set of candidate nodes that have the largest coverage. A non-forwarding on-tree node is a node that is already added to the tree, but has not been selected for forwarding traffic to other not-yet-on-tree nodes. The coverage of (non-forwarding) node v, denoted by $N(v)$, is defined as the number of not-yet-on-tree nodes that node v can cover. We only focus on non-forwarding nodes because if a node has already become a forwarding node (for the current request), all its neighbors must have been covered and are already on the tree. If there are multiple candidates, we rank the efficiency of each candidate node by its cost $U(v)$, where

$$U(v) = x' - \beta y' \tag{27}$$

where x' is the *local* maximum channel utilization and y' is *local* minimum residual interface capacity as observed by node v (when v is selected as a forwarding node). Then the node with the least $U(v)$ will be selected as a forwarding node, provided that constraints (12) and (14) are not violated. If there is a tie, the winner is randomly selected.

It is worthwhile in noting the differences between the *global* $(x - \beta y)$ in (3) and the *local* $(x' - \beta y')$ in (27). Consider a situation that part of the network is heavily loaded, and we are selecting a forwarding node among the nodes outside that particular region. Then the value of $(x - \beta y)$ remains unchanged no matter which node is selected. Accordingly, we cannot differentiate the efficiency among individual candidate nodes based on $(x - \beta y)$. On the other hand, $(x' - \beta y')$ varies based on the local view of the selected node. For this reason, our heuristic algorithm aims at minimizing $U(v) = (x' - \beta y')$, the local cost of selecting node v as a forwarding node.

Our greedy expansion algorithm operates as follows. When a broadcast call arrives at node S with a bandwidth request F_T, a simple screening test is performed to determine if any *node* (interface) in the network will be oversubscribed. If yes, the call is rejected right away. Otherwise, a broadcast tree rooted at S is to be built by adding one forwarding node to the tree at a time. Let $\boldsymbol{Z}$ be the set of covered nodes. At the beginning, $\boldsymbol{Z}$ only contains the root S, and S is not assigned with any transmission at this time. In each round of iteration, a forwarding node with the highest efficiency is identified from $\boldsymbol{Z}$. According to Definition 1, different nodes may observe different utilizations of a channel. As illustrated in Fig. 1, if node v is selected as the forwarding node and assigned to transmit in channel k, the channel utilizations of k, as seen by nodes A, B and C, will be increased. To accept a call, we have to ensure that the channel utilization, as seen by *any* node within the interference range of v, to be less than the normalized channel capacity one. If no such node can be found, then the call is rejected.

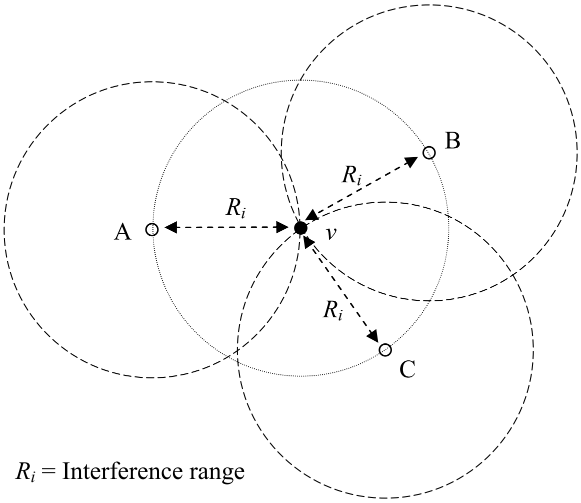


Fig. 1. Region affected by node v's transmission.

When a forwarding node is identified, it is assigned with time fraction for transmission in certain channel(s). In order to minimize the channel utilization, we add the requested load (F_T) to the least utilized channel first. If the selected channel does not have enough capacity, then the outstanding demand will be allocated to the second least utilized channel. When assigning a time fraction to a node, constraints (12) and (14) should not be violated.

All newly covered nodes are then added to $\boldsymbol{Z}$. The channel and interface utilization are updated according to the allocated time fractions. The algorithm continues to find the next forwarding node until all nodes are covered. The pseudo code of the greedy expansion algorithm is detailed in Fig. 2.

B. Largest Coverage Shortest-Path First for Multicasting

Unlike the greedy expansion for broadcast tree, forwarding nodes should only be added to the multicast tree if they extend the tree towards one or more receivers in the receiver set $\boldsymbol{R}$. Intuitively, to minimize channel utilization, the number of transmissions should be minimized. It is interesting to see that a pure shortest-path tree cannot achieve this objective. As shown in Fig. 3, a pure shortest-path tree is constructed for a five-member multicast group, with source S and receivers R1, R2, R3 and R4. We can see that the resulting multicast tree consists of six forwarding nodes, each identified by a double-circle in Fig. 3. In contrast, the multicast tree in Fig. 4 only requires five forwarding nodes. We can see that the major deficiency of constructing a pure shortest path tree is that it cannot fully benefit from WBA.

Accordingly, instead of adding the shortest-path corresponds to each receiver to the tree directly, we propose to add *the largest coverage shortest-path first* (LC-SPF), where

```
Greedy Expansion Algorithm

Input: $G(V, E)$, channel $C$, on-going calls, new call = {source S, load $F_T$}
Output: Broadcast tree $T$ rooted at S for load $F_T$.

$Z$ = {S}, $T = \phi$;

for $v \in V$ do                                   /* Pre-check */
    if (no available interfaces to receive $F_T$) then
        return $T = \phi$;
    end
end

while ($V \setminus Z \neq \phi$) do                   /* Construct Tree */
    candidate_node = $\phi$;
    max_node_cover = $\max_{\forall v \in Z,\, v \text{ is not transmitting}} \{N(v)\}$;

    for $v \in Z$ && $v$ is not transmitting do
        if ($N(v)$ = = max_node_cover) then
            candidate_node = candidate_node $\cup$ {$v$};
        end
    end

    forward_node = $\phi$;
    min_node_utilization = $\min_{\forall v \in candidate} \{U(v)\}$;

    for $v \in$ candidate_node do
        if ($U(v)$ = = min_node_utilization) then
            forward_node = forward_node $\cup$ {$v$};
        end
    end

    if (forward_node = = $\phi$) then
        return $T = \phi$;                          /* call rejected */
    else
        randomly select $v \in$ forward_node;
        allocate time fractions to $v$;
        $T = T \cup$ {$v$, {all $u \in V \setminus Z$ covered by $v$}};
        $Z = Z \cup$ {$v$, {all $u \in V \setminus Z$ covered by $v$}};
        update channel and interface utilization;
    end
end

return $T$;
```

Fig. 2. Greedy expansion algorithm for broadcast tree construction.

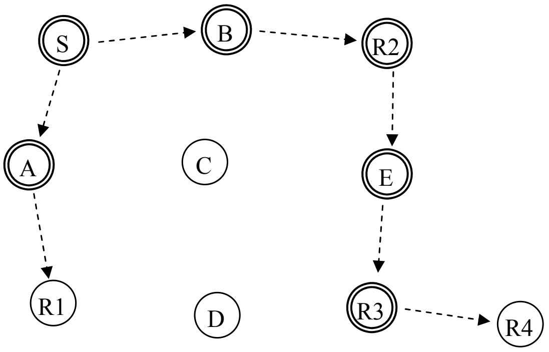


Fig. 3. Pure shortest-path tree based on hop count.

the coverage of a path is defined as the number of not-yet-on-tree receivers that can be covered by any nodes along the path. In the following, we illustrate the operations of our LC-SPF algorithm by the example in Fig. 4. When a multicast call arrives at node S with receivers R1, R2, R3, R4 and a bandwidth requirement F_T, a simple screening test is performed to determine if any multicast group member(s) will be oversubscribed. If yes, the call is rejected. Otherwise, a multicast tree rooted at S will be built.

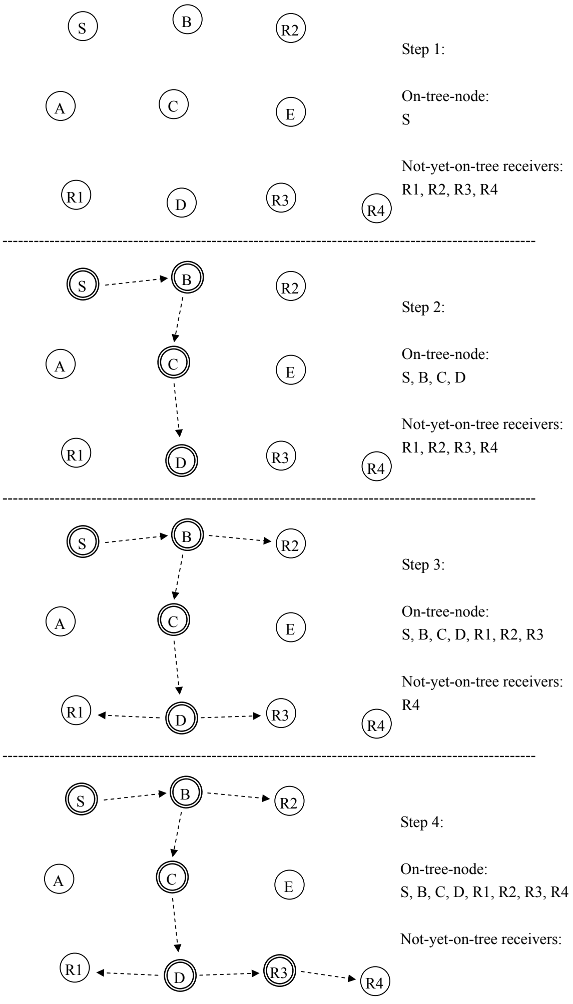


Fig. 4. Largest coverage shortest-path first.

In Step 1 of Fig. 4, we have node S as an on-tree node, while R1, R2, R3 and R4 are not-yet-on-tree receivers. Instead of finding the most efficient (forwarding) node as in the greedy expansion algorithm, LC-SPF iteratively adds the most efficient path to the tree until all receivers are attached. To reduce the complexity involved in trying all possible shortest-paths in the network, we first identify the largest coverage node(s). The coverage of a node, denoted by $N(.)$, is measured by the number of (not-yet-on-tree) receivers within its transmission range. The coverage of each node in Step 1 is N(S)=0, N(A)=1 (covering R1), N(B)=1 (covering R2), N(C)=0, N(D)=2 (covering R1 and R3), N(E)=2 (covering R2 and R3), N(R1)=1 (covering R1), N(R2)=1 (covering R2) and N(R3)=2 (covering R3 and R4). Then the three largest coverage nodes are D, E and R3, which form the candidate set.

In Step 2, the most efficient shortest-path corresponds to each largest coverage node is identified. Without loss of generality, we consider candidate node D first. There are three possible shortest-paths from the tree (S only) to node D: P1_D = S-A-R1-D, P2_D = S A-C-D, and P3_D = S-B-C-D. Among them, we select the one with the largest coverage (thus the most efficient path). In this case, $N(\text{P1}_\text{D})$=2 (covering R1 and R3), $N(\text{P2}_\text{D})$=2 (covering R1 and R3), $N(\text{P3}_\text{D})$=3 (covering

Largest Coverage Shortest-Path First Algorithm

Input: $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$, channel $\boldsymbol{C}$, on-going calls, new call = {S, $\boldsymbol{R}$, F_T}
Output: Broadcast tree $\boldsymbol{T}$ rooted at S for load F_T.

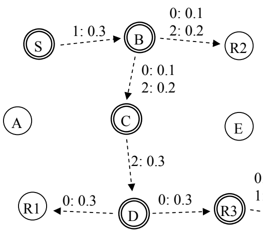
```
Z = {S}, T = φ;
for v ∈ R do                                        /* Pre-check */
    if (no available interfaces to receive $F_T$) then
        return T = φ;
    end
end
while ( $R - (R \cap Z) \neq \phi$ ) do             /* Construct Tree */
    candidate_path = φ;
    max_node_cover = $\max_{\forall v \in V \text{ and } v \text{ is not transmitting}} \{N(v)\}$;

    for v ∈ V && v is not transmitting do
        if (N(v) = = max_node_cover) then
            p = shortest-path corresponds to node v with max N(p)
            candidate_path = candidate_path ∪ {p};
        end
    end

    efficient_path = φ;
    min_global_utilization = $\min_{\forall p \in candidate\_path} \{U(p)\}$;

    for p ∈ candidate_path do
        if (U(p) = = min_global_utilization) then
            efficient_path = efficient_path ∪ {p};
        end
    end

    if (efficient_path = = φ) then
        return T = φ;                               /* call rejected */
    else
        randomly select p ∈ efficient_path;
        allocate time fraction $TF_k$ to v along p ∀k ∈ C;
        T = T ∪ {{all v∈p}, {all v ∈ $R - (R \cap Z)$ covered by p}};
        Z = Z ∪ {{all v∈p}, {all v ∈ $R - (R \cap Z)$ covered by p}};
        update channel and interface utilization;
    end
end
return T;
```

Fig. 5. Largest coverage shortest-path first algorithm.

R1, R2 and R3), and therefore S-B-C-D is selected. Similarly, the most efficient path for E is found to be S-A-C-E and that for R3 is S-B-C-D-R3. Let $U(p) = x - \beta y$ denote the cost of adding path p. The path with the least $U(p)$ will be selected. Note that x and y are global values. This is because each iteration of selecting the most efficient shortest-path affects a larger area (i.e., more nodes are involved) than that of greedy expansion.

To determine $U(p)$, we tentatively allocate transmission time fractions to each node along path p as if they were added to the tree, provided that constraints (12) and (14) are not violated. If any node along the selected path has become a forwarding node in some previous iteration, due to WBA, no further time fractions should be assigned to it. In Step 2 of Fig. 4, suppose path S-B-C-D yields the smallest $U(p)$, then the nodes along the path (B, C, D) are added to the on-tree node set $\boldsymbol{Z}$, and the traversed nodes (S, B, C, D) are assigned to transmit for a time fraction of TF_k.

Then in Step 3 of Fig. 4, LC-SPF examines the not-yet-on-tree receivers covered by the newly assigned transmissions (i.e., R1, R2, R3), and adds them to the on-tree node set $\boldsymbol{Z}$. At this point, there is still one not-yet-on-tree receiver R4. The algorithm proceeds to find the next largest coverage shortest-path until all receivers are attached to the multicast tree. In Step 4, R4 is attached to the tree via the largest coverage shortest-path R3-R4. When all receivers are attached, the algorithm terminates. The pseudo code of LC-SPF is detailed in Fig. 5.



	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	5	6	7	8	9
Ch 0	B→C,R2	D→R1,R3	D→R1,R3	D→R1,R3	R3→R4					
Ch 1	S→B	S→B	S→B	R3→R4	R3→R4					
Ch 2	C→D	B→C,R2	B→C,R2	C→D	C→D					

Fig. 6. A possible schedule for the multicast tree.

VI. Scheduling Algorithm

Both ILP formulations in Section IV and heuristic algorithms in Section V aim at assigning time fractions to each link/channel, while ensuring a feasible transmission schedule exists. Time fractions only describe the portion of frame for active transmissions, collision may occur if the transmissions are not properly scheduled. In this section, based on the time fractions obtained (where a feasible schedule is already guaranteed), we introduce a simple algorithm for generating a conflict-free transmission schedule.

We first find a frame of M time slots such that $f_{uv}^k M$ is an integer for all link (u, v) and channel k. $f_{uv}^k M$ also gives the number of time slots required by link (u, v) to transmit in channel k. Then start from an arbitrary node u, we allocate time slots according to the assigned time fractions. Due to WBA, f_{uv}^k of different v of the same call are carried out by a single transmission, and thus we only assign the time slots once. One interface on each u and v is marked as occupied in the assigned time slots. When all time fractions of node u are allocated to the frame, we can proceed to the next node. It should be noted that the number of concurrent transmissions in each slot should not exceed the number of interfaces a node has.

A simple example is given by Fig. 6, where each node has 2 interfaces and a multicast tree rooted at A is built with bandwidth requirement 0.3 (in terms of time fraction). The time fraction f_{uv}^k is presented with the format "$k : f_{uv}^k$" on link (u, v), and the bandwidth of a link is the sum of all f_{uv}^k of that link. Based on the time fractions given in Fig. 6, we can find M = 10 time slots. Since a frame schedule is periodic, we can start from any node. Without loss of generality, we consider root node S first. The given time fraction shows that it requires 3 time slots in channel 1 for sending to B. Thus, slots 0, 1 and 2 of channel 1 are assigned to the S→B transmission. We then consider node B. Due to WBA, B can transmit to C and R2 at the same time, with 1 time slot in channel 0 and 2 time slots in channel 2. Then time slot 0 of channel 0 is allocated to this transmission. For the transmission in channel 2, although slot 0 of channel 2 is idle, B cannot use it because the two interfaces of B are fully occupied (sending in channel 0 and receiving in channel 1). Therefore B has to use slots 1 and 2, the next idle slot and on an idle interface. We then consider node C, which requires three time slots in channel 2. Slot 0 of channel 2 can be allocated, but not slots 1 and 2. Although C has an idle interface in these two slots, it interferes (and is interfered by) B's transmission. Therefore, slots 3 and 4 are assigned to the C→D transmission. Similarly, transmissions of D and R3 are scheduled as shown in Fig. 6. Note that in this example, the resulting schedule requires only five time slots because the bandwidth requirement is only 0.3.

VII. Performance Evaluations

In this section, we compare the performance of finding bandwidth-guaranteed broadcast/multicast trees using ILPs and heuristic algorithms. The transmission range and interference range of each node are set to 250m and 500m, respectively. Due to the static topology of WMN, $\boldsymbol{I_v}$ the set of interfering nodes with respect to node v is uniquely determined by the transmission and interference range of each node. The number of orthogonal channels varies from 3 to 12, and the number of interfaces per node is 3 unless otherwise specified. For each network topology, 200 multicast requests for different group sizes (denoted by m) are randomly generated, each with a bandwidth requirement of 0.01 (in terms of time fraction). One of the group members is randomly selected as the source node. When m equals to the network size, it becomes a broadcast request. We assume each simulation starts with an idle system, i.e., the initial values of B_u^k and Y_{uv}^k are 0. We use call acceptance rate as our performance metric, where each data point in Figs. 7-11 is the average over 20 independent samples.

A. Studying the Weighting Factor β

We first investigate the impact of the weighting factor β, which determines the importance of interface utilization in the objective functions $(x - \beta y)$ and $(x' - \beta y')$. By setting β to 0 and 1, we can determine whether taking the residual interface capacity into account or not. To eliminate the impact of different multicast group sizes, we consider broadcast calls. Without loss of generality, the broadcast ILP (B-ILP) and greedy expansion (GE) are used for tree construction. Fig. 7 shows the call acceptance rate in a 4×5 grid network with 12 orthogonal channels, where the distance between adjacent nodes in the same row or column is 200m. Each node is randomly equipped with one to five interfaces. For comparison, a grid network with one interface per node is also simulated. In Fig. 7, “random" stands for randomly equipping one to five interfaces to each node, while “1 nic" (network interface card) represents the scenario of having one interface per node.

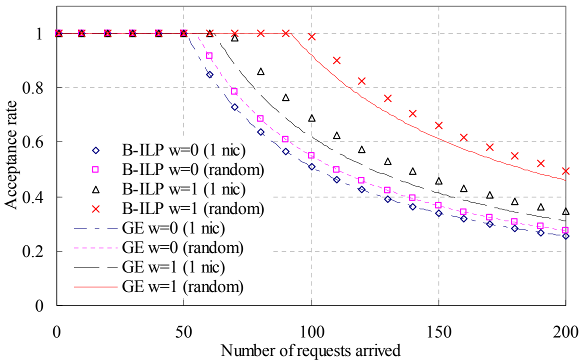


Fig. 7. Impact of residual interface capacity (nic = network interface card).

From Fig. 7, we can see that if residual interface capacity is not considered (i.e., β = 0), the network can only accept 50 calls. Since each call requires 0.01 time fraction, a node with single interface can accept at most 100 calls. However, a forwarding node needs to share its interface capacity in both receiving and sending, which is equivalent to doubling the loading for each call. Once the interface capacity of these nodes is fully occupied, no more call can be accepted. On the other hand, if we take residual interface capacity into consideration (i.e., β = 1), the forwarding nodes will be selected from those nodes with more residual interface capacity. This allows the network to accept more calls. When the number of interfaces per node is random, although there are nodes with single interface, the network can still accept about 100 calls. This is because the algorithm tends to not select the single-interface nodes as forwarding nodes.

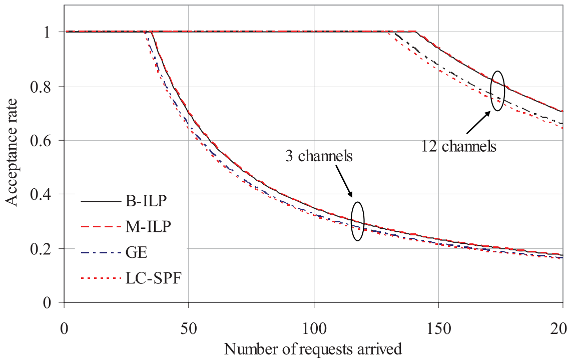
Since setting β = 1 allows the network to achieve higher call acceptance rate, in the following simulations β = 1 is used.

B. Accepting Broadcast Calls

Four broadcast tree construction algorithms are used for call admission, the broadcast tree ILP (B-ILP), multicast tree ILP (M-ILP), greedy expansion (GE) and largest coverage shortest-path first (LC-SPF). Fig. 8 shows the call acceptance rate and running time required in the same 4×5 grid network. The running time is obtained on a Pentium 4 3.2GHz machine with 512MB RAM, where “Total" stands for the total time of serving all 200 broadcast call requests, and “Tree" stands for the average time of building each tree (rejected requests are not counted).

From Fig. 8, as expected both B-ILP and M-ILP provide identical and highest call acceptance rate. This is because the two ILPs are empowered with the global network knowledge in tree construction. On the other hand, the two heuristic algorithms (only) carry out local optimization. The resulting network channel and interface utilization may be higher, and thus they accept fewer requests than ILPs. Although B-ILP and M-ILP provide the same acceptance rate, M-ILP requires longer running time as it has more constraints and variables. Nevertheless, as compared with the two heuristic algorithms, ILPs take much longer time to build a tree, e.g., 5-6 minutes for a network with 12 channels, which is not suitable for real-time call arrivals.

Focusing on the two heuristic algorithms, GE tends to accept more calls than LC-SPF. This is due to the difference between the tree construction procedures. In GE, the most efficient *node* is selected from the on-tree nodes for



	3 channels		12 channels	
	Total	Tree	Total	Tree
B-ILP	363.687s	8.652s	13h3m5s	332.314s
M-ILP	408.187s	9.384s	13h35m49.36s	350.596s
GE	188ms	1ms	391ms	2.857ms
LC-SPF	1.031s	8.063ms	1.891s	13.7ms

Fig. 8. Acceptance rate and running time of accepting broadcast calls in grid network.

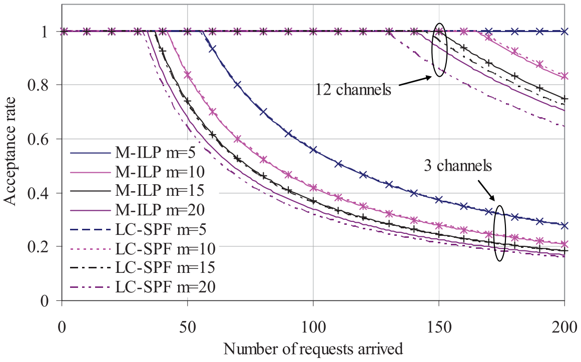


Fig. 9. Mutlticast in 4×5 grid network with 3 interfaces per node.

forwarding, where the resulting transmissions are more diverse and the resource utilization is thus more evenly distributed. In contrast, LC-SPF selects the largest coverage *shortest-path*. The flexibility of selecting individual forwarding nodes decreases, which lowers the efficiency on utilizing the multiple channels and interfaces. Also, since path-based selection is considered, LC-SPF requires longer running time than GE.

C. Accepting Multicast Calls

Focusing on the performance of multicast tree construction, two algorithms, M-ILP and LC-SPF, are simulated under the same 4×5 grid network. Fig. 9 shows the resulting call acceptance rate in admitting multicast calls with different group sizes (m). Obviously, more calls can be accepted by having more channels in the network. With 12 orthogonal channels, the grid network can accept all 200 multicast calls with m = 5. As expected, the acceptance rate decreases with the increasing multicast group size, because more channels and interface resources are consumed for carrying a call with larger m.

Then, we consider the dense and sparse 50-node random networks. Figs. 10 and 11 give the corresponding call acceptance rate performance. Since it takes time to solve the ILP for 50-node random networks, only LC-SPF performance is shown.

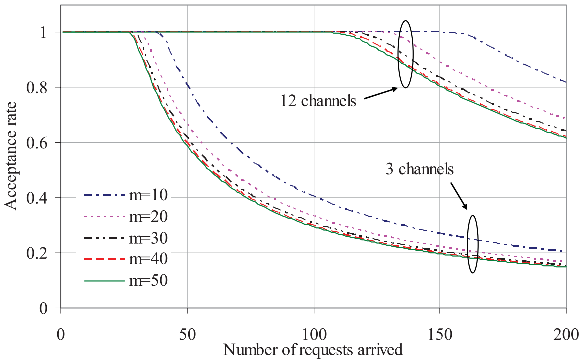


Fig. 10. Multicast in dense 50-node random network with 3 interfaces/node.

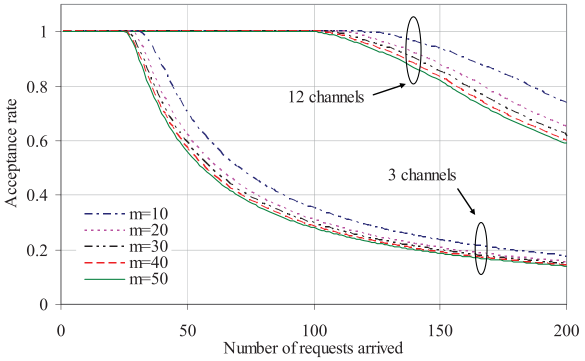


Fig. 11. Multicast in sparse 50-node random network with 3 interfaces/node.

We study the performance of LC-SPF algorithm in randomly generated networks, where 50 nodes are randomly placed in a given area. For a sparse network, an area of 1000×1000m^2 is used. For a dense network, an area of 2000×2000m^2 is used. Figs. 10 and 11 show the call acceptance rates in dense and sparse networks respectively. We can see that dense networks yield higher acceptance rate. When the multicast group size m=10, dense networks accept 39.3% and 22.3% more calls than the sparse ones using 3 channels and 12 channels, respectively. The reason is that in a dense network, more nodes can be covered by a transmission. Due to WBA, fewer transmissions are required for multicasting to all the group members. Besides, a sparse network has fewer choices of shortest-paths. As a result, multiple calls may pass through the same (or part of) shortest-path, causing hotspots.

VIII. Conclusion

In this paper, we have investigated the real-time multicast call arrivals in multi-channel multi-interface wireless mesh networks and discussed the bandwidth-guaranteed multicast tree construction problem. To maximize the call acceptance rate of the network, the multicast tree must be constructed judiciously with load balancing in mind. We first formulated the bandwidth-guaranteed multicast tree construction as an integer linear programming (ILP) problem, where channel assignment, routing and scheduling are jointly considered. By minimizing the carried load on both the most-heavily loaded channel and the most-heavily loaded node, the network load can be balanced, which in turn allows the network to accept more future calls. Since the above joint optimization problem is NP-hard, solving ILP can be time-consuming. To this end, an efficient multicast algorithm called largest coverage shortest-path first (LC-SPF) was proposed. Simulation results revealed that LC-SPF achieves comparable performance as the ILP, but with much shorter running time.

References

[1] H. Gossain, C. D. M. Cordeiro, and D. P. Agrawal, "Multicast: wired to wireless," *IEEE Commun. Mag.*, vol. 40, no. 6, pp. 116-123, June 2002.

[2] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks Applications*, vol. 7, no. 6, pp. 481-492, 2002.

[3] H. S. Chiu, K. Yueng, and K.-S. Lui, "J-CAR: an efficient joint channel assignment and routing protocol for IEEE 802.11-based multi-channel multi-interface mobile ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1706-1715, Apr. 2009.

[4] H. S. Chiu, B. Wu, K. L. Yeung, and K.-S. Lui, "Widest spanning tree for multi-channel multi-interface wireless mesh networks," in *Proc. IEEE WCNC*, Mar.-Apr. 2008, pp. 2194-2199.

[5] Y.-Z. Huang, W. J. Liu, and K.-T. Feng, "Efficient implementation of an energy-conserving multicast routing protocol for wireless multihop networks," in *Proc. IEEE RWS*, Jan. 2008, pp. 847-850.

[6] W. Liang, "Approximate minimum-energy multicasting in wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 4, pp. 377-387, Apr. 2006.

[7] W. Wei and A. Zakhor, "Multiple tree video multicast over wireless ad hoc networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 1, pp. 2-15, Jan. 2007.

[8] Z. Yin, Z. Li, and M. Chen, "A novel channel assignment algorithm for multicast in multi-radio wireless mesh networks," in *Proc. IEEE ISCC*, July 2007, pp. 283-288.

[9] P. M. Ruiz and A. F. Gomez-Skarmeta, "Approximating optimal multicast trees in wireless multihop networks," in *Proc. IEEE ISCC*, June 2005, pp. 686-691.

[10] U. T. Nguyen and J. Xu, "Multicast routing in wireless mesh networks: minimum cost trees or shortest path trees?" *IEEE Commun. Mag.,* pp. 72-77, Nov. 2007.

[11] J. Qadir, C. T. Chou, and A. Misra, "Exploiting rate diversity for multicasting in multi-radio wireless mesh networks," in *Proc. IEEE LCN*, 2006, pp. 287-294.

[12] J. Qadir, C. T. Chou, A. Misra, and J. G. Lim, "Localized minimum-latency broadcasting in multi-radio multi-rate wireless mesh networks," in *Proc. WoWMoM*, June 2008, pp. 1-12.

[13] B. Wang, M. Mutka, and E. Torng, "Optimization based rate allocation and scheduling in TDMA based wireless mesh networks," in *Proc. IEEE ICNP*, Oct. 2008, pp. 147-156.

[14] B.-S. Kim, S. W. Kim, and F. Ishmanov, "Reliable wireless multicasting with minimum overheads in OFDM-based WLANs," in *Proc. IEEE ICC*, May 2008, pp. 2519-2523.

[15] K. L. Yeung and H. L. Wong, "Caching policy design and cache allocation in active reliable multicast," *Computer Networks*, vol. 43, no. 2, pp. 177-193, Oct. 2003.

[16] J. He, J. Chen, and S. H. G. Chan, "Extending WLAN coverage using infrastructureless access points," in *Proc. IEEE HPSR*, 2005, pp. 162-166.

[17] M. Alicherry, R. Bhatia, and L. (Erran) Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 1960-1971, Nov. 2006.

[18] J. So and N. H. Vaidya, "Load balancing routing in multi-channel hybrid wireless networks with single network interface," in *Proc. IEEE QShine*, Aug. 2005, p. 8.

[19] A. Raniwala and T.-C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc. IEEE Infocom*, Mar. 2005, vol. 3, pp. 2223-2234.

[20] K. Han, Y. Li, Q. Guo, and M. Xiao, "Broadcast routing and channel selection in multi-radio wireless mesh networks," in *Proc. IEEE WCNC*, Mar. 2008, pp. 2188-2193.

[21] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multihop multicast in wireless mesh networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 2092-2103, Nov. 2006.

[22] S. Roy, D. Koutsonikolas, S. Das, and Y. C. Hu, "High-throughput multicast routing metrics in wireless mesh networks," in *Proc. IEEE ICDCS*, 2006, p. 48.

[23] X.-Y. Li and Y. Wu, "Maximizing throughput multicast routing for wireless networks," in *Proc. IEEE WASA*, Aug. 2007, pp. 3-10.

[24] G. Zeng, B. Wang, Y. Ding, L. Xiao, and M. Mutka, "Multicast algorithms for multi-channel wireless mesh networks," in *Proc. IEEE ICNP*, 2007.

[25] H. S. Chiu, K. L. Yeung, and K.-S. Lui, "Interface placement in constructing widest spanning tree for multi-channel multi-interface wireless mesh networks," in *Proc. IEEE WCNC*, Apr. 2009, pp. 1-5.

[26] C. T. Chou, B. H. Liu, and A. Misra, "Maximizing broadcast and multicast traffic load through link-rate diversity in wireless mesh networks," in *Proc. IEEE WoWMoM*, June 2007, pp. 1-12.

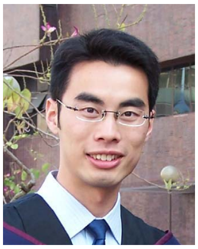
[27] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proc. ACM MobiCom*, Sep. 2003, pp. 66-80.

[28] S. M. Das, D. Koutsonikolas, and Y. C. Hu, "Practical service provisioning for wireless meshes," in *Proc. ACM CoNEXT*, Dec. 2007.

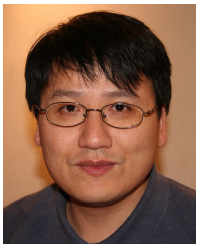
[29] M. Kodialam and T. Nandagopal, "The effect of interference on the capacity of multi-hop wireless networks," on *Proc. IEEE Symp. Infor. Theory*, July 2004, p. 472.

[30] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 12, pp. 1459-1473, Dec. 2008.

[31] A. Frieze and M. Jerrum, "Improved approximation algorithms for MAX k-CUT and MAX BISECTION," *Algorimica*, 1997.



Hon Sun Chiu received his B.Eng. degree in information engineering in 2002, and the M.Phil. and Ph.D. degrees in electrical and electronic engineering (major in computer networks) in 2004 and 2009, respectively, all from The University of Hong Kong. His research interests include routing, resource allocation and management, MAC layer protocol design, cross-layer optimization and security in both wired and wireless networks.



Kwan L. Yeung received his B.Eng. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong in 1992 and 1995, respectively. He joined the Department of Electrical and Electronic Engineering, The University of Hong Kong, in July 2000, where he is currently an associate professor. His research interests include next-generation Internet, active queue management, packet switch/router design, all-optical networks, and wireless data networks.