A Joint Routing and Scheduling Algorithm for Efficient Broadcast in Wireless Mesh Networks

Hon Sun Chiu and Kwan L. Yeung
Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong, PRC.
Tel: (852) 2857-8493   Fax: (852) 2559-8738
Email: {hschiu,kyeung}@eee.hku.hk

Abstract – With the increasing popularity of wireless mesh networks (WMNs), broadcasting traffic (e.g. IP-TV) will contribute a large portion of network load. In this paper, we consider a multi-channel multi-interface WMN with real time broadcast call arrivals. Aiming at maximizing the call acceptance rate of the network, an efficient broadcast tree construction algorithm, called Schedule-based Greedy Expansion (S-Expand), is designed. Unlike the existing time fraction approach, which focuses on assigning time fractions to tree links to guarantee the existence of a feasible schedule, we follow the approach of joint routing and scheduling. The proposed S-Expand algorithm packs non-interfering transmissions to use the same time slots; this would allow more flexibility in accepting future calls. Simulation results show that S-Expand achieves higher call acceptance rate than the traditional time fraction approach.

Keywords – Joint routing and schedule, multiple channels, multiple interfaces, wireless mesh network, broadcast.

I. INTRODUCTION[1]

Wireless mesh networks (WMNs) are deployed as the last mile for extending the Internet connectivity for end mobile users. Many broadcast applications (e.g. IP-TV) designed for the Internet are extended to the wireless networks [1]. Due to the rather limited wireless network capacity, efficient broadcast algorithms are urgently needed to regulate this increased traffic in the wireless backbone.

A fundamental difference between wired and wireless networks is the nature of transmission. Wireless transmission is broadcast, which allows all neighbors to receive the same copy of data with the source node only transmitting once. This is usually referred to as wireless broadcast advantage (WBA) [2]. To further increase the performance, each wireless device can be equipped with multiple network interface cards (NICs), and have simultaneous transmissions over multiple orthogonal channels [3]. In such a multi-channel multi-interface WMN, two nodes can communicate with each other only if they are within each other's transmission range, and one of their NICs is tuned to the same channel.

In the literature, many broadcast protocols have been proposed. While energy efficiency is the main concern for broadcasting in mobile ad hoc network [4, 5], it is not the case for WMN. In contrast, mesh nodes are usually stationary and with abundant power supply, which allows WMN to support higher functionality. In [6, 7], low latency broadcast is achieved using multi-rate transmission links, where the idea is to assign a path with longer hop distance to use links with higher transmission rate. Widest spanning tree is investigated in [8],

where the width of a tree is determined by the bandwidth of the bottleneck tree link. Nevertheless, a common feature of the above protocols [4-8] is that they only consider a single broadcast session.

In [9-11], a more realistic scenario with real time broadcast call arrival is considered. Each broadcast call is characterized by a specific bandwidth requirement. A call is accepted if a broadcast tree can be constructed with sufficient bandwidth reserved on each tree link. Otherwise, the call is rejected. To maximize the broadcast call acceptance rate, call routing and scheduling must be carried out judiciously.

In WMNs, nodes are stationary and the network traffic is rather stable, the network performance can be enhanced by proper scheduling. In [8-11], a frame-based MAC protocol is assumed. Conflict-free transmission is ensured by requiring interfered links be active at different time slots. Accordingly, a time fraction approach is followed, where routing and scheduling are solved in two sequential phases [8-11]. In the routing phase, the portion of a frame (i.e. time fraction) for each transmission link to be active is derived in accordance with a sufficient condition that guarantees the existence of a feasible schedule. Integer linear programming (ILP) is usually adopted for its solution. In the scheduling phase, the frame structure (i.e. the number of slots per frame) is derived where the obtained time fractions are transformed into time slots and placed onto a frame for conflict-free transmission. The resulting schedule is then distributed to each node in the network. Based on it, a node tunes its interfaces to send/receive on a specific channel at the stipulated time slots in a frame. It can be seen that the time fraction approach above cannot fully benefit from spatial reuse, as non-interfering transmissions will be unnecessarily required to use different time slots. This depletes the channel resource quickly, and prevents the system from admitting more calls. (Please refer to the discussion in Section III.)

In this paper, we target at the real time broadcast call arrival scenario as that in [9-11] and propose a centralized algorithm, called Schedule-based Greedy Expansion (S-Expand), to carry out call routing and scheduling jointly (i.e. in a single phase). We assume an *N*-slot frame structure is given in advance. The bandwidth requirement of a broadcast call is measured by the number of time slots required per frame. When a call arrives at a node, the node becomes the root of a broadcast tree to be constructed. In our proposed S-Expand algorithm, starting from the root, we greedily and iteratively add the next best forwarding node to the tree, until all nodes are on-tree. Whenever a forwarding node is identified, idle time slots in the frame are immediately assigned/reserved to carry the call. To maximize the call acceptance rate, we always pack non-

interfering transmissions to use the same slots; this can provide more flexibility in accepting future calls.

The rest of the paper is organized as follows. We first introduce the system model in the next section. Then the deficiency of the existing time fraction approach is discussed in Section III. In Section IV, our Schedule-based Greedy Expansion algorithm is introduced and we compare its performance with a time fraction based algorithm in Section V. Finally, we conclude the paper in Section VI.

II. SYSTEM MODEL AND ASSUMPTIONS

We consider a wireless mesh network (WMN) with multiple orthogonal channels. Each mesh node is equipped with multiple network interface cards (NICs), where the number of NICs a node has can be different and is assumed to be not more than the number of channels. A NIC can tune to use different channels and the channel switching latency is assumed to be small. The transmission range and interference range are assumed to be fixed and known in advance. Wireless links are unidirectional, where (u, v) and (v, u) are considered as different links.

The *receiver conflict avoidance* interference model [12] is adopted, which only requires the receiver to be clear for receiving. We assume a frame-based MAC protocol with N slots per frame. Conflict-free transmission is ensured by requiring interfered links be active at different time slots. With conflict-free transmission, we assume perfect delivery of each packet and thus acknowledgement is not required.

We assume that broadcast calls arrive in real time, with each call characterized by a source node and bandwidth requirement in terms of number of slots per frame. The broadcast tree constructed for this new call should not affect/interrupt any on-going calls. Otherwise, the new call is rejected. For a call-based system, the network load changes only upon a new call arrival or an existing call departure. Call admission control is carried out based on the required bandwidth of the new call, and the existing call-based network resources utilization. With such stable network condition, the overhead involved in getting the network resources utilization statistic is minimal.

III. SOME OBSERVATIONS ON TIME FRACTION APPROACH

The time fraction approach [8-11] consists of two separate phases. In the routing phase, the portion of a frame (i.e. time fraction) for each transmission link to be active is determined. In the scheduling phase, the frame structure (i.e. the number of slots per frame) is derived. The obtained time fractions are transformed into time slots and placed onto a frame to form a schedule for conflict-free transmission. In order to guarantee that a feasible schedule always exists in the scheduling phase, a sufficient condition must be satisfied in the routing phase. Let f_{uv}^k denote the time fraction for link (u, v) to be active for transmission on channel k, I_v denote the set of nodes that locate within the interference range of node v and n_{mv}^k denote the interference (in terms of time fraction) caused by node m on channel k as observed by node v. Then the sufficient condition for a feasible schedule is that, for any node v in channel k,

$$f_{uv}^k + \sum_{m \in I_v, m \neq u} n_{mv}^k \leq 1. \qquad (1)$$

In other words, the sufficient condition states that, for any
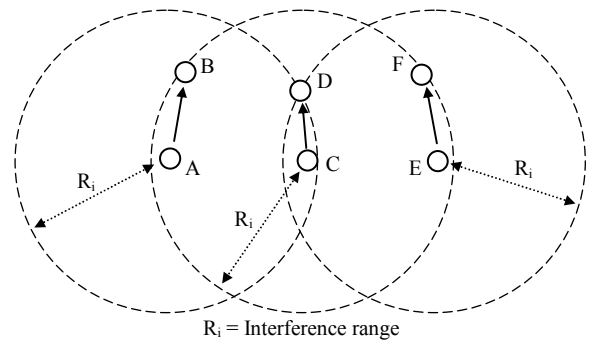

R$_i$ = Interference range

Fig. 1. Scenario with three sender-receiver pairs

node v, all interfering transmissions must have a total load less than the normalized channel capacity. If it is satisfied, the interfering transmissions can (simply) take turns to be active within a frame duration. The associated scheduling in the second phase for ensuring conflict-free transmission is thus straightforward. Notably, in [9, 10], load balancing among channels is carried out in the routing phase to further improve the call acceptance performance. In doing so, a call may be split to use different channels on a transmission link.

But in general, condition (1) is too conservative, and easily causes network under-utilization. Consider a simple unicast scenario in Fig. 1. There are three sender-receiver pairs A→B, C→D and E→F using the same channel k. To have a feasible transmission schedule, from node D's point of view, $f_{AB}^k + f_{CD}^k + f_{EF}^k \leq 1$. Hence, each transmission can on average be assigned with a time fraction of 1/3. However, A→B and E→F are in fact outside each other's interference range, they can be active at the same time.

If routing and scheduling are jointly considered, the channel utilization can be improved. Assume that we have a frame of 10 slots, and each connection request asking for 5 slots (equivalent to a time fraction requirement of 0.5). Without loss of generality, assume connection A→B sets up first and occupies the first 5 slots of the frame. Then the second call for connection C→D arrives. Since the first 5 slots of the frame are occupied, C→D selects the last 5 slots to use. When the third call for connection E→F arrives, from F's its point of view, the first 5 slots are idle, while the last 5 slots are occupied by C→D. So the first 5 slots are used to admit the third call. Now, all three call requests (with 0.5 time fraction each) are successful. But with the time fraction approach, the third call will be rejected in the routing phase due to the violation of the sufficient condition in (1), because from D's point of view, admitting the third call would cause $f_{AB}^k + f_{CD}^k + f_{EF}^k = 1.5 > 1$.

In the above example, it is worthwhile to note that if E→F sets up its route prior to the C→D connection, then E and F will observe an idle frame and E→F can be set up using any 5 slots in the frame. If E→F chooses to occupy the last 5 slots, then as observed by C and D, the frame is fully occupied and the C→D connection will be rejected. We can see that to maximize the call acceptance rate, particularly for broadcast calls, we need an efficient joint routing and scheduling algorithm to avoid such *inefficient slot assignment*.

Another point to note is that in the time fraction approach, whenever the time fractions for a new call are found, it cannot be simply transformed and added to the existing frame. Instead,

it has to re-generate the entire frame because the frame size with the new time fractions may change. That is to say, when a new call is admitted, the transmission pattern of the existing calls will be affected. On the other hand, an algorithm based on the joint routing and scheduling approach directly allocates slots for each transmission. Existing calls will not be affected by any new call arrival.

IV. JOINT ROUTING AND SCHEDULING DESIGN

In this section, we propose an efficient joint routing and scheduling algorithm, called Schedule-based Greedy Expansion (S-Expand), for broadcast call admission. As its name implied, starting from the root, we greedily and iteratively add the next best forwarding node to the tree, and at the same time assign/reserve idle time slots in the frame for carrying the transmission. S-Expand continues to run until all nodes are on-tree or no feasible forwarding node/idle slot can be identified. In the latter case, the call will be rejected.

In each iteration, the best forwarding node is identified as the on-tree node with the largest coverage. The coverage of a node is measured by the number of not-yet-on-tree nodes within its transmission range. By always selecting a node with the largest coverage, the total number of transmissions (i.e. forwarding nodes) required in establishing the broadcast tree can be minimized. Accordingly, the interference induced to other transmissions/calls in the network can also be minimized.

If there are multiple on-tree nodes with the same largest coverage, we select the node that yields the best load balancing performance. The load balancing performance in a WMN consists of two components, channel-based load balancing and node-based load balancing. Let $C(u)$ be the number of slots to be occupied on the least utilized channel as observed by node u if it is selected as a forwarding node. Intuitively, selecting a forwarding node u with the minimum $C(u)$ helps to balance the loading on different channels in the network. On the other hand, channels can have spare capacity for a new call but a node may not have sufficient NIC capacity to carry the call. To this end, we also want to balance the spare NIC capacity at a node. Let $N(u)$ be the *expected* spare NIC capacity of node u if it becomes a forwarding node. To balance the spare NIC capacity at each node, we prefer the node with the largest $N(u)$.

To jointly consider both channel-based and node-based load balancing performance, a cost function of selecting node u as the forwarding node is defined as

$$U(u) = C(u) - N(u). \qquad (2)$$

In our S-Expand algorithm, among all the largest coverage nodes, we select the forwarding node as the one with the minimum $U(u)$ (which can be negative). To fine tune the performance, we can add a weighting factor β to $N(u)$ to reflect the relative importance of the two components. Extensive simulation results suggest that $\beta = 1$ yields good performance. As such, β is dropped from (2).

We describe our Schedule-based Greedy Expansion (S-Expand) algorithm in more details below. When a broadcast call arrives at node S (then S becomes the root of the tree) with bandwidth requirement of T time slots, S-Expand first checks whether any node in the network does not have enough NIC capacity to receive the broadcast call. If yes, the call is rejected. Otherwise, a broadcast tree rooted at S is to be built.

Schedule-based Greedy Expansion Algorithm

Input: Node set V, link set E, channel set C, new call={source S, T slots}
Output: Broadcast tree BT root at S with each tree link active for T slots.

$Z = \{S\}, \quad BT = \phi$;

for $v \in V$ **do**　　/* Pre-check */
　if (not enough NICs capacity) **then**
　　return $BT = \phi$; 　/* call rejected */
　end
end

while ($V \setminus Z \neq \phi$) **do**　　/* Construct Tree */
　for $v \in Z$ **do**
　　calculate coverage(v);
　　calculate $U(v) = C(v) - N(v)$;
　end
　$\text{coverage}_{max} = \max_{v \in Z} \text{coverage}(v)$;
　$U_{min} = \min_{v \in Z, \text{coverage}(v) = \text{coverage}_{max}} U(v)$;
　select node *best_N* with coverage$_{max}$ and U_{min} as forwarding node;
　if (allocate T time slots = fail) **then**
　　return $BT = \phi$; 　/* call rejected */
　else
　　$BT = BT \cup (best\_N \rightarrow \{\text{all } v \notin Z \text{ covered by } best\_N\})$
　　$Z = Z \cup best\_N \cup \{\text{all } v \notin Z \text{ covered by } best\_N\}$;
　end
end
return BT;

Fig. 2. Pseudo code of the Schedule-based Greedy Expansion algorithm

---

Let Z be the set of on-tree nodes. In the beginning, Z only contains the root S. In each iteration, the largest coverage node in Z is selected as the next forwarding node. If there are multiple nodes with the same largest coverage, the node with the minimum cost $U(u)$ is selected. Then the required time slots are assigned for its transmission by packing them to the lowest numbered slots in the frame. Specifically, we first identify the channel with the least utilization $C(u)$. Then the idle slots, as seen by the sender u and all the receivers under its coverage, are identified. If the number of idle slots is larger than the required number of slots, we assign the lowest numbered idle slots to it. Otherwise, the call is rejected. When assigning an idle time slot, it must ensure that both the sender and all receivers have a spare NIC for this transmission in that particular slot. By always selecting the lowest numbered available slots of the least utilized channel, the inefficient slot assignment problem mentioned in Section III can be avoided. We continue to find the next forwarding node until all nodes are attached to the broadcast tree. The pseudo code of the S-Expand algorithm is detailed in Fig. 2.

To illustrate the operation of S-Expand, let us consider an example in Fig. 3. The network in Fig. 3 supports two orthogonal channels and an 8-slot frame structure. Each node is equipped with 2 NICs. When a broadcast call with a bandwidth requirement of two time slots arrives at node S, we use S-Expand to construct a broadcast tree. In the beginning, Z only contains S and therefore S is selected as the first forwarding node. To allocate the required number of time slots, S-Expand selects the least utilized channel (i.e. the channel with the minimum $C(u)$), and assigns two idle slots as seen by the sender and all covered receivers (i.e. nodes S, A and D). In this case, slots 0 and 1 of channel 0 are selected and reserved for the call. The nodes covered by S (i.e. A and D) are attached to the tree and are added to the set Z. Then S-Expand continues

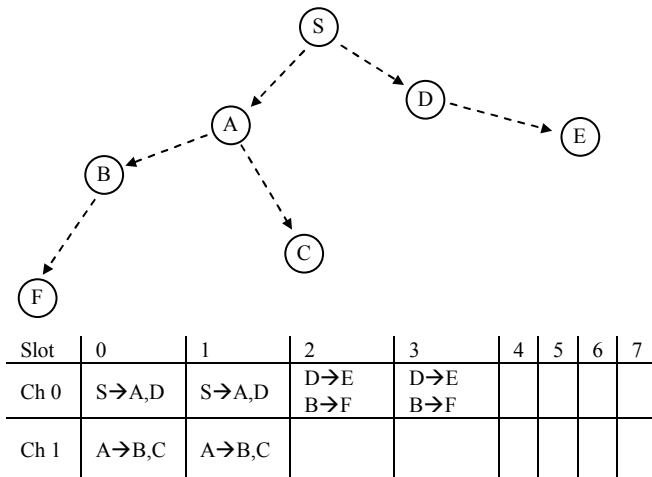| Slot	0	1	2	3	4	5	6	7
Ch 0	S→A,D	S→A,D	D→E B→F	D→E B→F				
Ch 1	A→B,C	A→B,C						

Fig. 3. A broadcast call rooted at S, with a bandwidth requirement of 2 time slots on each 8-slot frame.

to find the next forwarding node.

When there are multiple nodes in Z, we first identify the set of candidate nodes. In this case, A covers B and C, while D can only cover E. (Note that S has zero coverage in this round, as all its neighbors are on-tree.) So node A is our only candidate node, and slots 0 and 1 of channel 1 are reserved for its transmission (to both B and C). Nodes B and C are then added to Z. At this moment, the two NICs of A are busy in receiving on channel 0 and sending on channel 1 at both slots 0 and 1. Note that we have packed existing calls to use the lowest numbered time slots first.

Continue with the example, we have two candidate nodes, B and D, in Z for selecting as the next forwarding node. Since both nodes have the same coverage of 1 node, we select the one with the minimum cost $U(u)$ as defined in (2). If there is a tie, we randomly pick up one as the winner. Suppose $U(D) < U(B)$, then node D is selected as the next forwarding node, slots 2 and 3 of channel 0 are reserved, and node E is added to Z.

Finally, node B is selected. Since D is outside the interference range of B, from both B and F's point of view, slots 2 and 3 of channel 0 are idle. Therefore, B→F can *reuse* these slots. When node F is added to Z, all nodes are attached to the broadcast tree. The broadcast tree rooted at S with bandwidth requirement of 2 slots is thus constructed.

V. PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of our Schedule-based Greedy Expansion (S-Expand) algorithm with different network settings, and compare it with the time fraction based algorithm in [10]. Note that the time fraction algorithm in [9] only considers the load balancing among channels, while the one in [11] focuses on multi-rate transmission with a single channel. The algorithm in [10] is selected because it uses multiple channels, and considers both channel-based and node-based load balancing. Both grid topology and random topology are used in our simulations. Grid network consists of 4x5 nodes, and the grid length is set to 200m. For random networks, 50 nodes are randomly placed within the simulated area, where 1000x1000m^2 is used to simulate a dense network, and 2000x2000m^2 for a sparse
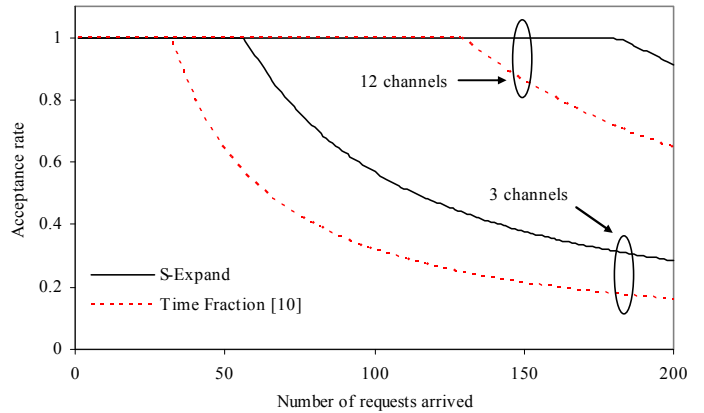

Fig. 4. Performance in 4x5 grid network with 3 NICs/node

network. The transmission range and interference range are set to 250m and 500m, respectively. The number of orthogonal channels and the number of NICs per node vary as specified later on. For our S-Expand, we adopt a frame with size of 100 slots. In each simulation run, 200 broadcast calls will arrive sequentially, each with a randomly selected root/source and a bandwidth requirement of 1 time slot (which is equivalent to a time fraction of 0.01 when the algorithm in [10] is used). We use acceptance rate as our performance measure, where each data point in Figs. 4-7 is obtained by averaging over 20 independent runs.

Our first set of simulations aims at observing the performance difference in the 4x5 grid network with 3 NICs/node. The call acceptance rate performance is shown in Fig. 4, where "S-Expand" stands for our Schedule-based Greedy Expansion algorithm, and "Time Fraction [10]" refers to the time fraction based algorithm in [10]. As expected, using more channels allows the network to accept more calls. In both 3 channels and 12 channels situations, S-Expand achieves higher acceptance rate. This is because of the effectiveness of joint routing and scheduling design in utilizing the channel resource. From this result, we can also see that the 3 NICs are not fully utilized with 3 channels in the network, as higher acceptance rate can be obtained if 12 channels are provided.

In the time fraction approach, time fractions are assigned to tree links while not violating the sufficient condition in (1). However, as discussed in Section III, it suffers from poor spatial reuse, where some non-interfering transmissions are allocated into different time slots, and thus depleting the channel resource faster. On the other hand, S-Expand follows the joint routing and scheduling approach, where the schedule is determined hand-in-hand with the routing process. Non-interfering transmissions are allocated into the same slots, where more spare slots are available for accepting future calls.

The second set of simulations aims at investigating the performance of different approaches with different numbers of NICs equipped on each node. To minimize the limitation on the number of channels, 12 orthogonal channels are assumed. Fig. 5 shows the acceptance rate performance with 1 NIC/node to 5 NICs/node. Obviously, equipping more NICs on a node allows higher data forwarding capability. The call acceptance rate increases with the number of NICs/node until all channels are fully occupied.
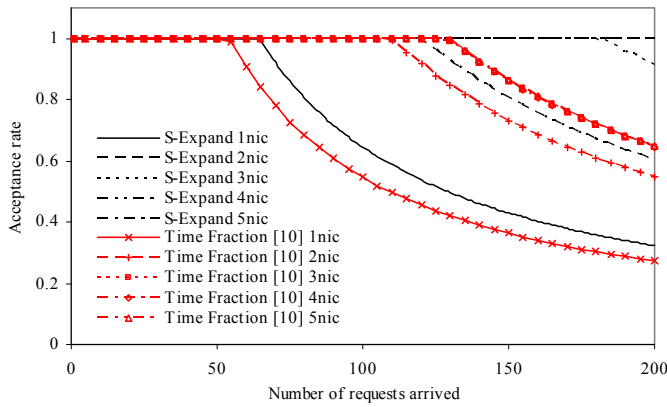
Fig. 5. Performance in 4x5 grid network with 12 channels

Fig. 6. Performance in dense 50-node random network with 3 NICs/node

Fig. 7. Performance in sparse 50-node random network with 3 NICs/node

From Fig. 5, we can see that the time fraction approach depletes the channel resource more quickly than the joint routing and scheduling approach. It can only accept up to 129 calls with 3 NICs/node. At this point, all 12 channels are fully occupied, and no improvement can be observed even if more NICs are provided. On the other hand, with 3 NICs/node, S-Expand accepts 177 broadcast calls. At this point, the channels have not been fully utilized yet. S-Expand can even accept all the 200 calls with 4 NICs/node.

Then, we compare the performance using random networks. With 3 NICs/node, the call acceptance rate performances in dense and sparse networks are given by Figs. 6 and 7, respectively. Again, with joint routing and scheduling design, S-Expand achieves higher acceptance rate in both scenarios.

Nevertheless, we can see that dense networks enable a higher call acceptance rate than the sparse networks. This is because in dense networks, more nodes can be covered by a transmission, and thus fewer transmissions are required to reach all nodes in the network. In contrast, transmission in a sparse network tends to be less efficient as it covers fewer nodes. More transmissions are required and thus more resources are consumed by each call.

VI. CONCLUSION

In this paper, we have considered a real time broadcast call arrival problem, and have presented a broadcast tree construction algorithm using joint routing and scheduling approach, called Schedule-based Greedy Expansion (S-Expand), for maximizing the call acceptance rate of a multi-channel multi-interface wireless mesh network. As its name, S-Expand greedily selects the most efficient forwarding node until all nodes are attached to the broadcast tree. Unlike existing time fraction designs, we do not consider the conservative sufficient condition for guaranteeing the existence of a feasible schedule, which forces some non-interfering transmissions to use different time slots. In contrast, S-Expand works on the frame directly, where the required time slots are allocated once the forwarding node is selected. By packing non-interfering transmissions to use the same time slots, higher flexibility is allowed for accepting future calls. We have compared our performance with an existing time fraction design, and the results have shown that S-Expand achieves higher call acceptance rate.
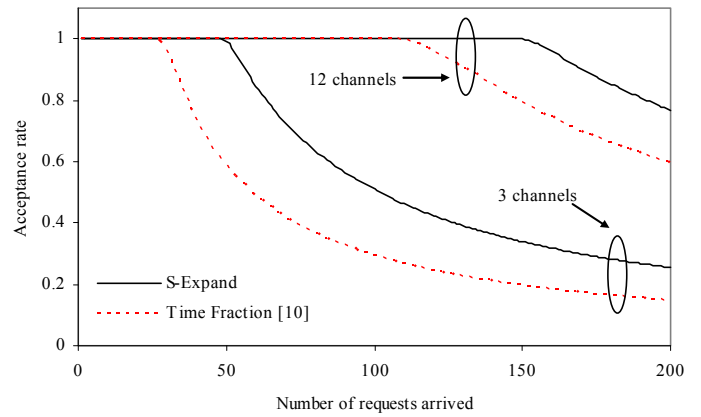
REFERENCES

[1] H. Gossain, C. D. M. Cordeiro, and D. P. Agrawal, "Multicast: Wired to Wireless," *IEEE Communications Magazine*, vol. 40, issue 6, pp. 116-123, Jun 2002.

[2] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Mobile Networks and Applications*, 7(6): 481-492, 2002.

[3] 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.

[4] Y.-Z. Huang, W. J. Liu, and K.-T. Feng, "Efficient Implementation of an Energy-Conserving Multicast Routing Protocol for Wireless Multihop Networks," *Proc. of IEEE RWS*, pp. 847-850, Jan 2008.

[5] W. Liang, "Approximate Minimum-Energy Multicasting in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 377-387, April 2006.

[6] J. Qadir, C. T. Chou, A. Misra, and J. G. Lim, "Localized Minimum-Latency Broadcasting in Multi-radio Multi-rate Wireless Mesh Networks," *Proc. of WoWMoM*, pp. 1-12, June. 2008.

[7] J. Qadir, C. T. Chou, A. Misra, "Exploiting Rate Diversity for Multicasting in Multi-Radio Wireless Mesh Networks," *Proc. of IEEE LCN*, pp. 287-294, 2006.

[8] H. S. Chiu, B. Wu, K. L. Yeung, and K.-S. Lui, "Widest Spanning Tree for Multi-Channel Multi-Interface Wireless Mesh Networks," *Proc. of IEEE WCNC*, pp. 2194-2199, 2008.

[9] H. S. Chiu, K. Yeung, and K.-S. Lui, "Maximizing Broadcast Load in Multi-Channel Multi-Interface Wireless Mesh Networks," *Proc. of IEEE Globecom*, pp. 1-5, 2008.

[10] H. S. Chiu, K. Yeung, and K.-S. Lui, "Bandwidth-Guaranteed Multicast in Multi-Channel Multi-Interface Wireless Mesh Networks," *Proc. of*

IEEE ICC, pp. 1-5, June 2009.

[11] C. T. Chou, B. H. Lui, and A. Misra, "Maximizing Broadcast and Multicast Trafic Load Through Link-Rate Diversity in Wireless Mesh Networks," *Proc. of IEEE WOWMOM*, 2007.

[12] H. S. Chiu, Kwan L. Yeung 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. on Wireless Communications*, vol. 8, issue 4, pp.1706-1715, April 2009.