A Low Complexity Distributed Algorithm for Computing Minimum-Depth Multicast Trees in Wireless Networks

A. Sinan Akyurek, Elif Uysal-Biyikoglu

◆

Abstract—This paper presents a wireless multicast tree construction algorithm, SWIM (Source-initiated WIreless Multicast). SWIM forms one shared tree from source(s) to the multicast destinations; yet, as a side product it creates a multicast mesh structure by maintaining alternative branches at every tree node, thus providing robustness to link failures. This makes it suitable for both ad-hoc networks and access networks with multiple gateways. It is proved that SWIM is fully distributed, with a worst case complexity (for multicast) upper-bounded by $O(N^3)$, and average complexity of only $O(N^2)$. SWIM constructs a tree on which each multicast destination has the minimum possible depth (number of hops from the nearest source). In terms of minimizing the number of forwarding nodes (NFN), SWIM is optimal for unicast. Its average NFN in the broadcast and multicast cases is compared with practical algorithms targeting low NFN reported in the literature. In both multicast and unicast, SWIM performs competitively in terms of NFN with the previous solutions, while having smaller maximum depth, and consequently low delay.

Index Terms—wireless multicast, multicast tree, minimum depth, wireless broadcast problem, greedy set cover, mesh network, number of forwarding nodes.

1 INTRODUCTION

In multihop wireless networks such as mobile ad-hoc networks and mesh networks, multicast sessions from a source node, gateway (or set of gateways) to a group of destinations, often occur. Such sessions are sometimes delay sensitive (in the case of, for example, near-real-time applications.) Rather than sending the data along unicast routes to each destination, it is desirable, primarily to avoid burdening the network with unnecessary transmissions, to construct a multicast route [1]. A good multicast tree will make efficient use of the bandwidth- or energy-constrained wireless links [2][3], avoid bottlenecks at multicast source nodes and maximize energy-efficiency by keeping the *number of forwarding nodes* (NFN) low, while maintaining maintaining a sufficiently low delay for each destination. In order to keep delay low, it is of interest to make the depth of each destination on the multicast tree small.

The problem of satisfying these requirements is defined as the Wireless Multicast Tree problem: finding a routing tree with the minimum number of transmissions needed to reach all destinations [4]: The tree needs to dominate, but not necessarily include, all destination nodes.

The Wireless Multicast Tree problem (WMT) is an NP-complete problem [5]. To see this, it suffices to show that any instance of the Steiner Tree problem can be converted to an instance of WMT in polynomial time: simply add dummy neighbor nodes to each of the nodes to be covered by the Steiner tree (except one, the source), and then solve the wireless multicast problem for the specified source and with the set of dummy nodes as multicast group. For a detailed survey on Steiner Tree Problem, refer to [6], [7].

In this paper, we approach the wireless multicast problem with two goals: The main goal is to obtain a multicast *tree* with minimum depth, while keeping NFN as low as possible, in the single-source case. The secondary goal is to obtain a multicast *mesh*, for suitability to the multiple-source scenario and also robustness to link failures and mobility.

The reason for the second goal is our conviction that the multiple-source scenario is important, especially with the emergence of wireless mesh networks (WMN) as access networks for widespread wireless networking, with all the self-organization, self-configuration and self-healing properties of this architecture [8], [9]. In the mesh network scenario, outside access (access to larger networks) is provided by *several gateway nodes* in the network. Clearly, it may be advantageous for different network nodes to be accessing different gateways depending on their respective proximity to these gateways. In this case, insisting on a tree can lead to a poor solution; rather, a multiple-source routing graph needs to be considered.

Combining both goals, the problem considered in this paper involves a node (or a set of nodes) acting as a source to a group of other nodes, the *multicast group*, that have requested the same data. The vision is to relieve the bottlenecks forming at sources [2], [10], and be bandwidth- and energy-efficient.

The main contributions of this paper are the following:

- A distributed and low-complexity routing algorithm based on a well-known Greedy Set Cover is given.

- SWIM is depth-optimal by construction in both single and multiple-source cases and in terms of NFN, SWIM is optimal in the unicast case, and exhibits good average performance in multicast and broadcast scenarios.
- Finally, a method is developed for creating alternative routing trees which are desirable especially in the case of high mobility [11] and in multimedia streaming [12] for reliable transmission.

The rest of this paper is organized as follows: In the next section, we discuss related work reported in the literature. In section 3 our problem setup is made precise. Section 4 introduces the proposed algorithm SWIM. The correctness and complexity analyses of SWIM are presented in Section 5. In Section 6, performance of SWIM is explored through extensive simulations and compared with several algorithms from the literature. In section 7 the generation of alternative paths is explained. Section 8 presents concluding remarks and outlines future directions.

2 RELATED WORK

Interest in wireless multicast has risen rapidly in the last decade [13]. A number of multicast routing algorithms have been developed, focusing on different priorities such as low latency, energy efficiency, and so on [14]. A notable multicast tree formation algorithm - for both weighted and not-weighted graphs- appears in [4]. This solution is based on merging optimum unicast routes and pruning the resulting subgraph. A different approach is presented in [15]: The objective is to select the minimum number of nodes in the network that are ”on” (and keep others turned “off”,) while keeping a communication path from the source to the destinations, by utilizing information about the geographic position of the nodes in the network. An MST is calculated on the final state to further reduce the number of “on” nodes. Another notable multicast routing heuristic presented in [16], relies on clustering and a certain MAC protocol [17].

There is a richer literature on wireless *broadcast*, which is a special case of the multicast problem. Much of the recent work on broadcast has considered energy efficiency, and power control [18], [19], [15]. Iterative Maximum-Branch Minimization (IMBM), proposed in [20], constructs an iterative mechanism for reducing power in a source-initiated wireless broadcast tree with the objective of minimizing the total required power. Another iterative method with an energy-efficiency objective is presented in [21], where integer programming has been used.

NP-hardness of the minimum energy broadcast problem in metric space was proved in [22], and later in [23] it was shown under more general conditions that power-optimal broadcast is NP-complete. Wieselthier et al. [18] proposed several broadcast tree heuristics: Broadcast Incremental Power (BIP), BliMST, and BLU. It is worth noting that multicast versions of these, namely, MIP, MLU, MLiMST, have also been proposed. These algorithms have some commonalities with the algorithm proposed in this paper: worst case complexity of $O(N^3)$, optimality in the unicast case, and containing a sweeping operation to remove unnecessary transmissions. However, not all of these are distributed, and moreover, their closeness to the optimal tree in terms of the number of transmissions, in other words, Number of Forwarding Nodes (NFN), was not studied. Moreover, as these algorithms are based on power control, which is out of the scope of our treatment, they are not directly comparable with our solution. More recently, an algorithm specifically developed for voice multicasting with the purpose of minimizing NFN was proposed in [24]; however, this protocol does not address the multiple source case.

This paper will study NFN in addition to tree depth, as simple and well-defined, if somewhat abstract, measures that are related to concrete real-life performance measures. The relation between NFN and other measures is evident: minimizing NFN ideally results in minimizing the transmit power dissipation. Minimizing depth ideally minimizes the maximum forwarding delay. Realistic performance metrics, such as Average Delay and Maximum Delay are also studied through real-time event based ns3 simulation environment. In addition, issues such as resilience to link failures, distributed implementation, messaging overhead, and computational complexity will be addressed as primary concerns in the development of SWIM, which is intended for real-life implementation.

While a full experimental comparison including realistic channel models and packet arrival models requires a separate study, one can readily make a conceptual comparison between two well-known protocols and our proposal:

1) MAODV forms a “shared tree”, that is, one tree connecting the source(s) with the multicast destinations, without explicitly optimizing tree depth or NFN by taking advantage of the Wireless Multicast Advantage.
2) ODMRP, on the other hand, forms a mesh rather than a tree: redundant routes are kept for reliable transmission in case of a link failure. SWIM contains the redundant route property of ODMRP though the maintenance of alternative routes.
3) SWIM not only finds alternative routes from source(s) to destinations, but also finds alternative routes from the intermediate forwarding nodes to the destinations. This implies that SWIM finds at least as many alternative routes as ODMRP does, implying increased reliability.

Simulations have been done to compare MAODV and SWIM. It should be noted that the simulations in this paper are on a static network model and hence are not conclusive as to which is a better solution under mobility.

3 THE PROBLEM

We consider a general multi hop wireless network model with N nodes (including any gateway nodes, if exist.) A multicast session will be initiated by a source (or set of sources). The set of multicast destinations, M, will be referred to as the "multicast group". Our objective is to create routing paths from the given set of sources to the multicast group, so that while each destination is reached with the absolute *minimum number of transmissions*, the total number of transmissions (NFN) is made as low as possible. We assume that there is a given set of links between nodes, forming a connected graph, and that each node knows about its one-hop neighbors on this graph.

4 THE ALGORITHM SWIM

We can now describe the algorithm SWIM (named after "Source initiated WIreless Multicast") in some detail. As introduced in Section 1, the main idea of SWIM is to first create a connected directed graph (a union of trees) rooted at the source(s), which dominates the multicast group, and then to prune this graph to obtain a single tree using a Greedy set cover algorithm. Hence, SWIM works in two phases: (I) routing graph generation, and (II) pruning.

Phase (I) of SWIM could be skipped when hopcount information to the source(s) for all neighbors are already stored by nodes, for example, in proactively created and maintained routing tables of a distance-vector type unicast routing algorithm running in the background. Hence, when SWIM is run on top of such a unicast routing algorithm, it could start directly from Phase (II).

SWIM only depends on basic neighbor discovery, MAC broadcast and a correctly operating link layer that guarantees packet delivery as long as a link is operational. Troubleshooting when a link is broken will also be described as part of the routing protocol, for completeness. We assume that an ARQ mechanism is used to ensure the delivery of control packets.

4.1 Definitions

We now pin down the notation to be used to explain SWIM's operation. In the table in Fig. 1, local definitions (about information kept by an individual node), and system-wide definitions are separately listed.

Local Definitions:	**System Definitions:**
P : Parent Node Set	P_i : Parent Node Set of node i
S : Sibling Node Set	S_i : Sibling Node Set of node i
C : Child Node Set	C_i : Child Node Set of node i
N : Neighbor Node Set	N_i : Neighbor Node Set of node i
M_Seen : Set of Multicast nodes seen	h(i) = d : node i is d hops away from the nearest source

Fig. 1. Notation used to describe SWIM.

4.2 SWIM Phase I: Tree generation

This phase is initiated by the source[1]. The source broadcasts the message "$h(source) = 0$". Neighbors of the source, upon reception, record their distance as "$h(node) = 1$" and send it as a message to their own neighbors. During this leveling, sequence numbers are checked at every node for packets received from each neighbor to avoid loops and usage of outdated information.

In general, a node will receive distance messages from each neighbor, and will record its own distance to the source as the minimum of these. As the distance-setting process starts at the source itself and is based on hop count, it is expected to progress in "levels".

Ensuring Correct Distributed Operation: To ensure correct distributed operation, sequence numbers are used in the implementation of SWIM: older messages received from a given neighbor are discarded in favor of up-to-date ones. Similarly, if, because of excessive delays, a node receives such an update from a neighbor after it has already moved on to the next state, it goes back to the previous state. If a total link failure happens, the node at the child-end of it will inform neighbors of this through the neighbor discovery procedure. All these nodes will reset themselves to h-msg state.

Neighbor Hierarchy: Each node will categorize its immediate neighbors into three groups: neighbors belonging to the immediately higher level (neighbors that are one hop further from the source) are recorded as "children", those one hop closer to the source as "parents" and those at the same level as "siblings" (see Fig. 2.)

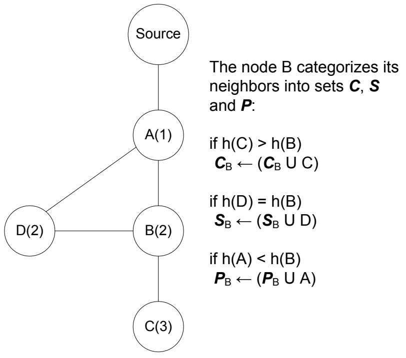

Fig. 2. Parent, child, and sibling designations of a node in the single source case.

Multiple sources: In the multiple source case, all sources start the process independently. Nodes that receive distance messages simply choose the smallest distance value received so far. Note that, nodes can simply keep one distance value for themselves, regardless of which

---

[1] We view the algorithm as source-initiated without loss of generality. When the session request actually originates at a client, this request can be conveyed to the source along a unicast route, following which the source initiates a multicast session as described here.

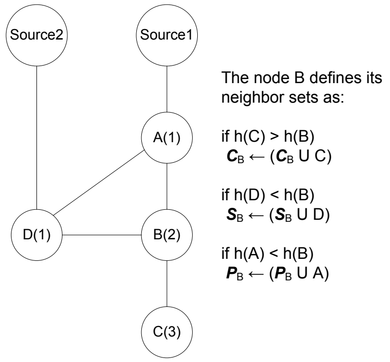

Fig. 3. Parent, child, and sibling designations made by a node in the multiple source case. Distance, $h(\cdot)$ is defined as the distance to the nearest source. Note that, $h(\cdot)$ does not carry source ID information. This does not cause a problem, as all nodes see their parent(s) as the immediate source, anyway.

source this distance is from, as to each node its immediate source its parent.

The goal of the second phase, Pruning, is to form a tree that cuts down as many unnecessary transmissions as possible.

4.3 SWIM Phase II: Pruning

In the beginning of this phase, information about the *Multicast Nodes Seen in Higher Levels* are sent to all nodes, starting from the leaves (nodes furthest from the source), all the way to the source.

In the pruning phase, the goal is for each node to tell its children to keep a selection out of all outgoing links, and eliminate the rest. This is related to the problem of choosing a *set cover* in the following way: Suppose node a has three children, b, c, and d. At the end of Phase I, a knows that b can reach $M_b \subset M$ through its own downstream nodes, c can reach $M_c \subset M$, and d can reach $M_d \subset M$. Suppose that $M_b \cup M_c = M_a = M_b \cup M_c \cup M_d$. That is, between themselves, b and c cover all possible multicast nodes that a is supposed to reach. In this case, a could simply tell d it no longer needs to relay the multicast packets coming from a, thus the link from a to d is taken out of the routing graph, *i.e., pruned*.

Then, the minimum number of child nodes that node a needs to keep in the routing tree, while reaching all elements of M_a is the solution of a minimum set cover problem. It is well known that minimum set cover is an NP-complete problem. Fortunately, there is a well-known greedy set cover algorithm (see, for example, [25],) that is at most $\log(d)$ away from optimal, where d is the size of the largest set.

In summary, here is how the routing tree is obtained: Starting at leaves, every client lets its parent node know the subset of the multicast group that it can reach through its children. Once this process has progressed to the source, the source now has information on which multicast group members it can reach via which child. The source runs the first greedy set cover and assigns its children, c_i, multicast group subsets M_{c_i} to cover. For all i, child i then runs a greedy set cover among its own children to cover M_{c_i}. This progresses through the graph until no node which has an assignment to cover remains; at that point, all multicast nodes have been covered, and the pruned graph is a tree.

5 COMPLEXITY OF SWIM

5.1 Computational Complexity of SWIM

The computational complexity is mainly dictated by the computational complexity of the Greedy Set Cover, which is $O(n^2)$ in terms of number of sets, n [25]. The number of children of a node cannot exceed the total number of nodes: N. Accounting for all nodes, we reach the conclusion that computational complexity scales as fast as $O(N^3)$.

This bound may seem too loose, and indeed extensive simulations have confirmed that average computational complexity is $O(N^2)$. The simulation was done by counting the average number of computational processing with respect to the number of nodes.

5.2 Messaging Overhead in SWIM

The length complexity of the messages is analysed. The worst case message length is obtained when the nodes form a line. Each node is addressed by an address size of $log_2 N$. Each node sends a request packet to reach $O(N)$ nodes, meaning an information of $O(N log_2 N)$. Repeating this for each node on the line means a complexity upper bounded by $O(N^2 log_2 N)$.

6 THE PERFORMANCE OF SWIM

6.1 Depth Optimality

By construction, SWIM forms a multicast tree on which the depth of any destination node is minimal. Hence, SWIM is optimal in terms of minimizing the hopcount of any node from the source and maximum depth (that is, tree height.) This is made precise in the following proposition.

Lemma 1: On the wireless multicast tree computed by SWIM with respect to a given source, the depth of every node is minimal.

Proof. Consider a graph with a multicast group M and source s, and let T be an arbitrary wireless multicast tree rooted at s covering M. Let the depth of multicast node i on T be d_i. This means, there is a neighbor of node i that is within distance $d_i - 1$ of the source. Node i does not enter phase II of the SWIM algorithm before it has heard the h_j announcement of all of its neighbors $\{j\}$, upon which it sets its h_i field to one larger than the smallest of the h_js. Therefore, at the start of Phase II, $h_i \leq d_i$. As phase II operates in levels, and node i will be on level h_i, hence it will have depth exactly h_i, which is smaller than or equal to d_i. ■

It follows by a similar argument that in the multiple source case SWIM minimizes the hop distance of each destination from the nearest source.

6.2 NFN and Tree Depth Simulations

As a way of judging SWIM, we have implemented the M-AODV algorithm, as well as the solution proposed in [4]. Comparisons have been made with respect to average NFN, Maximum and Average Tree Depth. The three algorithms have been run on the same random topologies, generated according to the simulation settings described in [4]:

- Nodes are placed independently according to the uniform distribution in a unit square region of side length 1.
- Transmission range (the maximum distance between two nodes, such that they are connected) is 0.286[2].

All three algorithms were run on the same 10000 independent randomly created topologies:

1) Broadcast Simulation: N(number of nodes) is changed from 20 to 70.
2) Multicast Simulation: $N = 70$ and m(number of multicast clients) varies between 1 to 69.
3) (SWIM Only)Multi Source Simulation: N is increased from 20 to 70 under a broadcast scenario.

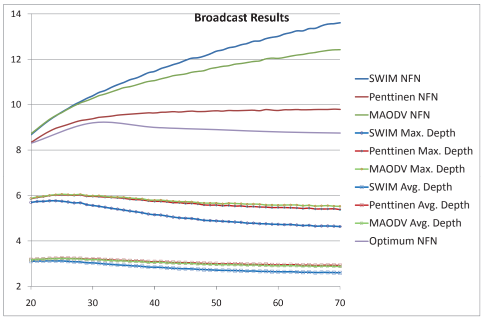

Fig. 4. Broadcast Simulation results. The optimum NFN result is taken from [4].

The results indicate where SWIM stands compared to the solution in [4] in the trade-off between tree depth and NFN. The algorithm proposed in [4], which is the best known solution in terms of NFN from the literature, achieves better average NFN values than SWIM, especially as the network size increases. However, meanwhile the gap between the two algorithms in terms of average and maximum depth widens in SWIM's favor.

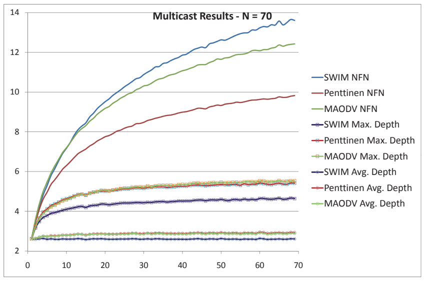

Fig. 5. Multicast Simulation results.

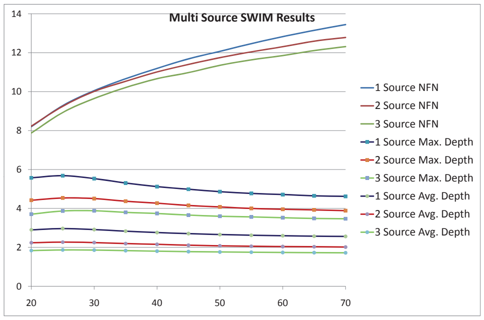

Fig. 6. MultiSource simulation results.

6.3 Average and Maximum Delay Simulations:

Another performance metric of great practical significance is delay. To simulate the algorithms, we have implemented the three algorithms; SWIM, MAODV and [4] in the real-time event simulator ns3 and measured two metrics: Average and Maximum Delay.

The simulation setup uses 802.11b WiFi MAC protocol in AdHoc mode with a 1Mbps bandwidth. The total number of nodes in the network, N, is increased from 20 to 70. For each case, the results are averaged over 100 topologies. The source generates a UDP flow with rate 100kbps. All the three algorithms are run on the same topology for 150 seconds and there is no mobility. The propagation loss model is Log Distance Model. The results are presented in the rest of this section.

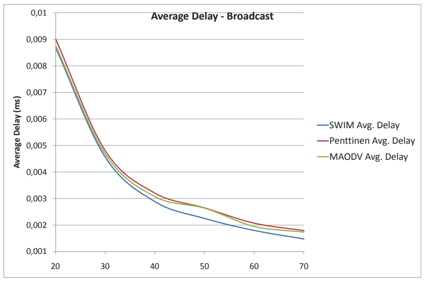

Fig. 7. Broadcast case: Average Delay Comparison

---

[2] This number has been picked because it is the same value that was used in [4], and it corresponds to nodes having transmission radius 100 units located in a square area with 350 unit sides.

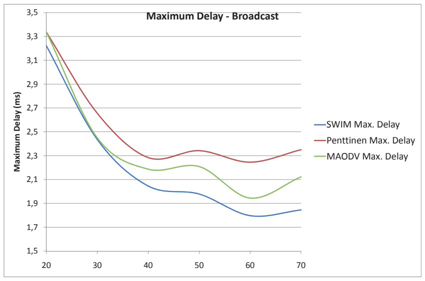

Fig. 8. Broadcast case: Maximum Delay Comparison

SWIM tries to decrease the NFN by maximizing the links per a forwarding node while maintaining depth optimality, and the larger the number of links, the more points of failure. Hence, SWIM will trade off throughput in favor of minimizing delay. However, there is a mechanism to increase throughput in a SWIM implementation: Generating alternative redundant paths. Alternative paths will cause no additional delay, but they will lead to increased energy dissipation.

7 GENERATION OF ALTERNATIVE PATHS

The main idea is to select, at each step, an alternative set cover that complements the original one. The alternative route algorithm starts a second set cover on a non-intersecting set with the main route on each node and distributes the information as in Part II.

8 CONCLUSIONS AND FUTURE DIRECTIONS

We constructed a distributed multicast-tree generation algorithm, SWIM, achieving an average complexity of only $O(N^2)$. SWIM is depth-optimal and obtains a low NFN. The key reason is that SWIM applies a competitive Greedy Set Cover algorithm at each level from source(s) to destinations. The number of messages to be exchanged at startup is minimal. We believe that SWIM offers an implementable solution that will effectively relieve bottlenecks at network gateways. A method is also developed for creating alternative routing trees. The performance with respect to the other existing approaches under mobility is an interesting future work. Other future directions include the weighted-graph case and selecting the number of alternative graphs based on a given throughput goal.

9 ACKNOWLEDGMENTS

This work has been partly supported by the Communications division of ASELSAN.

REFERENCES

[1] B. Awerbuch and Y. Azar, "Competitive multicast routing," *Wireless Networks*, vol. 1, pp. 107–114, 1995.

[2] M. C. Xudong Wang, Edward Knightly and A. Ephremides, "Guest editorial, a special issue on 'wireless mesh networks'," *Elsevier Ad Hoc Networks*, vol. 5, pp. 649–651, aug 2007.

[3] J. Zheng and M. J. Lee, "A resource-efficient and scalable wireless mesh routing protocol," *Elsevier Ad Hoc Networks*, vol. 5, pp. 704–718, aug 2007.

[4] A. Penttinen, "Minimum cost multicast trees in ad hoc networks," in *Proceedings of IEEE ICC 2006*, pp. 3676–3681, 2006.

[5] R.-M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations, Plenum Press*, 1975.

[6] P. Winter, "Steiner problem in networks: A survey," *Networks*, vol. 17 no. 2, p. 129167, 1987.

[7] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Proc. of European Symposium on Algorithms*, p. 179193, 1996.

[8] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resillient overlay networks," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, pp. 131–145, October 2001.

[9] Y. Amir, C. Danilov, R. Musaloiu-E., and N. Rivera, "An inter-domain routing protocol for multi-homed wireless mesh networks," in *IEEE International Symposium on World of Wireless Mobile and Multimedia Networks WoWMoM 2007*, pp. 1–10, 2007.

[10] B. Aoun, R. Boutaba, and Y. Iraqi, "Gateway placement optimization in wmn with qos constraints," *IEEE Journal on Selected Areas in Communications*, 2006.

[11] R. Langar, N. Bouabdallah, and R. Boutaba, "A comprehensive analysis of mobility management in mpls-based wireless access networks," *IEEE/ACM Transactions on Networking*, 2007.

[12] A.Banchs, W.Effelsberg, C.Tschudin, and V.Turau, "Multicasting multimedia streams with active networks," in *23rd IEEE Conference on Local Computer Networks (LCN98)*, (Lowell, MA), October 1998.

[13] H. G. Carlos de Morais Cordeiro and D. P. Agrawal, "Multicast over wireless mobile ad hoc networks: Present and future directions," *IEEE Network*, pp. 52–59, January/February 2003.

[14] W. Jia, L. Cheng, and G. Xu, "Efficient multicast routing algorithms on mesh networks," in *Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing(ICA3PP'02)*, pp. 110–117, 2002.

[15] J. L. Wong, G. Veltri, and M. Potkonjak, *System-Level Power Optimization for Wireless Multimedia Communication*, ch. 5, pp. 69–85. Springer US, 2002.

[16] B. Tavli and W. B. Heinzelman, "Mc-trace: Multicasting through time reservation using adaptive control for energy efficiency," *MILCOM*, vol. 4, pp. 2672–2678, 2005.

[17] B. Tavli and W. B. Heinzelman, "Mh-trace: Multi hop time reservation using adaptive control for energy efficiency," *IEEE J. Select. Areas Commun.*, vol. 22, pp. 942–953, 2004.

[18] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proceedings of IEEE Infocom 2000*, p. 586594, 2000.

[19] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Distributed algorithms for energy-efficient broadcasting in ad hoc networks," in *Proceedings of the IEEE MILCOM'02*, pp. 820–825, Oct 2002.

[20] F. Li and I. Nikolaidis, "On minimum-energy broadcasting in all-wireless networks," in *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, (Tampa, FL), pp. 193–202, November 2001.

[21] M. Min, O. Prokopyev, and P. Pardalos, "Optimal solutions to minimum total energy broadcasting problem in wireless ad hoc networks," *Journal of Combinatorial Optimization*, vol. 11, pp. 59–69(11), February 2006.

[22] O. Egecioglu and T. Gonzalez, "Minimum-energy broadcast in simple graphs with limited node power," in *Proc. IASTED Int. Conf. on Parallel and Distributed Computing and Systems*, (Anaheim, Canada), pp. 334–338, 2001.

[23] M. Cagalj, J.-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, (Atlanta, Georgia, USA), pp. 172–182, Sep 2002.

[24] G. V. Raju, T. B. Reddy, and C. S. R. Murthy, "A near optimal localized heuristic for voice multicasting over ad hoc wireless networks," *ICC*, pp. 1648–1653, 2007.

[25] D. S. Johnson, "Approximation algorithms for combinatorial problems," pp. 38–49, 1974.