Approximation and Heuristic Algorithms for Minimum-Delay Application-Layer Multicast Trees

Eli Brosh, Asaf Levin, and Yuval Shavitt, *Senior Member, IEEE*

***Abstract*—In this paper we investigate the problem of finding minimum-delay application-layer multicast trees, such as the trees constructed in overlay networks. It is accepted that shortest path trees are not a good solution for the problem since such trees can have nodes with very large degree, termed high-load nodes. The load on these nodes makes them a bottleneck in the distribution tree, due to computation load and access link bandwidth constraints. Many previous solutions limited the maximum degree of the nodes by introducing arbitrary constraints. In this work, we show how to directly map the node load to the delay penalty at the application host, and create a new model that captures the trade offs between the desire to select shortest path trees and the need to constrain the load on the hosts. In this model the problem is shown to be NP-hard. We therefore present an approximation algorithm and an alternative heuristic algorithm. Our heuristic algorithm is shown by simulations to be scalable for large group sizes, and produces results that are very close to optimal.**

***Index Terms*—Approximation algorithms, overlay networks, peer-to-peer communications.**

I. Introduction

MULTICAST is a key component in the design of group communication applications which require efficient data delivery to multiple destinations. However, IP multicast which implements multicast functionality at the network layer is still not widely deployed in current IP networks. To alleviate this problem, several recent proposals [1] have advocated an alternative approach, termed *application-layer multicast* or *end-host multicast*, which implements multicast functionality at the application layer using unicast network-level services only, forming an overlay network between end-hosts.

The goal of application-layer multicast [2] is to construct and maintain efficient distribution trees between the multicast session participants, minimizing the performance penalty involved with application-layer processing. Many proposals attempt to optimize the cost of the multicast delivery tree using application-level performance measures such as delay or throughput. The systems, which aim at reducing the overall delay [2]–[7], construct a minimum-height (or minimum-diameter) tree with constrained degrees. The degree constraints are used to control the network resource usage, i.e., available bandwidth or stress on the physical links. However, this solution stipulates the usage of a dual-cost optimization objective which mixes network-level and application-level costs to characterize applications performance.

In this paper we advocate an application-centric approach which quantifies system performance using application-level costs only. We claim that the conventional overlay network model and its corresponding delay measure are designed to characterize multicast systems which assume network-level data distribution capabilities. Unfortunately, message processing by end-hosts involves an additional delay penalty which is not captured by such models and is related to application-layer implementations of packet duplication and routing. In particular, the shift of multicast functionality to the upper level influences the simultaneous distribution capabilities of end-hosts, implying a communication model with sequential message distribution. This constraint stems from the fundamental change in the characteristics of the routing infrastructure assumed by the overlay network, attributed to the difference between message distribution speeds of routing nodes (i.e., end-hosts) in overlay networks and packet distribution speeds of routers in conventional physical networks.

For example, consider the simple network of Fig. 1(A), composed of three hosts H_1, H_2, and H_3 and two routers R_1 and R_2 connected using a high-speed backbone, where host H_1 uses a low-bandwidth access link for network connectivity, e.g., modem access, and H_2, H_3 use high-bandwidth LAN access connectivity. Assume that the goal of the overlay system is to devise a multicast tree that provides minimum distribution delay from H_1 to H_2 and H_3. Clearly, a multicast system must be careful to avoid delegating large degree to the low-bandwidth host H_1 in order to eliminate unnecessary bottleneck due to its low-speed data distribution capabilities. Fig. 1(B) depicts the corresponding optimal multicast tree. Now, consider the conventional routing algorithm used by many application-layer multicast architectures that optimize tree delay, namely the shortest path tree algorithm. In this case the shortest path multicast tree reduces to a star topology [Fig. 1(C)], which ignores the performance penalty at the star center. Hence, serialized message distribution which is irrelevant to IP multicast schemes must be accounted for in the evaluation of overlay multicast architectures. Surprisingly, however, many application-layer architectures which optimize tree delay have neglected these implications on the overall performance of group communication applications.

Manuscript received January 20, 2005; revised January 18, 2006; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Byers. The work of E. Brosh was done while the author was with the School of Electrical Engineering, Tel-Aviv University. This work was supported in part by the Israel Science Foundation Center of Excellence Program under Grant 8008/03 and in part by a grant from the EU 6th FP, IST Priority, Proactive Initiative "Complex Systems Research," as part of the EVERGROW integrated project.

E. Brosh is with the Computer Science Department, Columbia University, New York, NY 10027 USA (e-mail: elibrosh@cs.columbia.edu).

A. Levin is with the Department of Statistics, Hebrew University of Jerusalem, Jerusalem 91905, Israel (e-mail: levinas@mscc.huji.ac.il).

Y. Shavitt is with the School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel (e-mail: shavitt@eng.tau.ac.il).

Digital Object Identifier 10.1109/TNET.2007.892840

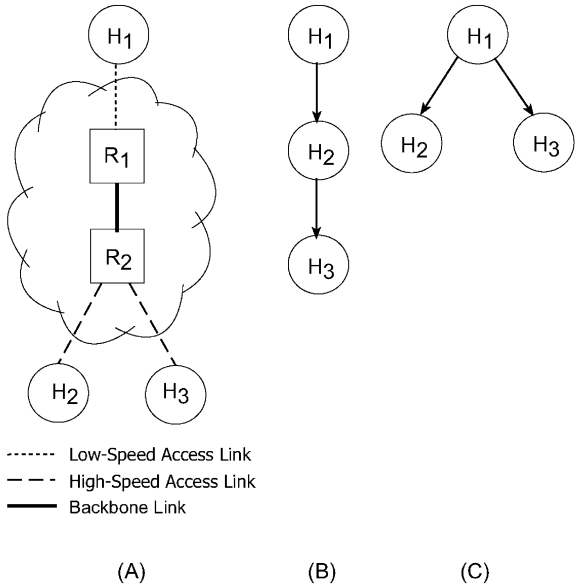

Fig. 1. Comparison between application-layer multicast and network-layer multicast in a simple heterogeneous overlay network.

Another factor which constrains parallel message distributions in overlay networks is the processing capacity of end-host machines. For instance, consider a server which implements router like functionality at the application layer, and therefore may not have enough CPU power to handle message processing at the full speed of its network interfaces. Hence, the effective message distribution rate of an end-host is shaped by two factors, the bandwidth of the access link connecting the host or its local area network to the physical network, and the processing power and the computational load on the host machine. A recent study [8] that measured the actual end-host heterogeneity of popular peer-2-peer (p2p) overlay systems showed that the bandwidth and latency parameters can vary by several orders of magnitude across different hosts in the system.

In this paper, we present an overlay network model which captures the realistic costs involved with application-layer multicast. The model is a mathematical generalization of a communication model developed by Cidon *et al.* [9] for high-speed networks, and similarly it incorporates two separate delay measures. The processing delay measure, which is a reciprocal of the effective message distribution speed of an end-host application, and the communication delay measure, which represents the delay of traversing an overlay link. This framework enables us to characterize the performance of multicast trees using a *single cost*, the overall delay of message distribution.

We use the proposed framework to develop heuristic and approximation algorithms for the basic problem of optimal multicast tree construction. Both the heuristic and the approximation generate minimum-delay trees that intrinsically balance short latency with small degree, and thus avoid an external trial-and-error type of balancing between the two, i.e., we do not impose a maximum degree on our trees. Our heuristic algorithm constructs such trees efficiently and thus can scale to large multicast groups, which is a known problem [2]. Our algorithms supports fully connected topologies as well as structured topologies, used in some p2p overlay networks [10]. We thus address the issue of optimal multicast in partially connected networks and provide performance bounds for tree and grid graph structures.

The presented algorithmic solutions can be effectively used to implement centralized overlay systems, such as p2p- and server-based systems. The heuristic algorithm is particularly useful in the context of two-tier server-based architectures [5], [11], [3] which construct a virtual tree among the servers to provide an efficient content and data delivery services to end-users. Each end-user registers to a server in order to receive multicast services, and the server handles the dissemination of the aggregated traffic. Such semi-static architectures employ reliable servers to provide high-availability service, stipulating a simple implementation with low computational overhead due to minor topology changes. Furthermore, a centralized approach is capable of providing quick and efficient session management services by sharing the computational load among several overlay servers [4].

The main applicability of our algorithms is in the context of delay-sensitive multicast applications, which require tight bounds on the end-to-end delays due to jitter and timing constrains. Applications which belong to this category include audio conferencing, real-time media streaming, content distribution services, and multiplayer distributed games.

The rest of this paper is organized as follows. The next section formulates the overlay communication model. In Section III we discuss the problem of optimal multicast tree construction and show that this problem is NP-complete. In Sections V and VI we develop approximation and heuristic algorithms for solving this problem. Section VII deals with performance analysis of the heuristic algorithm for several overlay topologies. An experimental evaluation of our solutions is presented at Section VIII. Finally, Section IX concludes the paper.

II. Overlay Communication Network Model

An *overlay network* is a fully connected virtual network formed by hosts which communicate with each other using a physical network, such as the Internet. The overlay network utilizes the regular unicast services of the physical network to provide communication among hosts, and do not require any special support at the network level. The delay experienced by a message that travels between hosts is composed of two elements: (a) *Communication delay*—which represents the delay of traversing the unicast path between the hosts. This component includes the accumulated propagation and queuing delays of the physical links on the unicast path, and the message reception overhead at the receiver host. (b) *Processing delay*—which represents the delay of processing a message at the sender host. This element includes the overhead of preparing a message for transmission and the transmission delay through the physical access link.

Although current implementations of operating systems enable applications to perform concurrent message transmissions, the concurrent transmissions would be serialized when passing through a physical access link. Typically, this serialization is performed at the hardware level by the access equipment. Thus, sequential distribution of messages should be accounted for in order to avoid unrealistic application design schemes which rely on simultaneous message dissemination capabilities.

We define an overlay network model based on a generalization of a communication model developed by Cidon *et al.* [9]. The overlay network is modeled by a directed complete graph $G = (V, E)$, where V is a set of vertices representing hosts, and E is a set of edges representing unicast paths. We use the terms "host" and "link" to refer to the vertices and edges in the overlay graph. Each overlay edge $(u, v) \in E$ is associated with a communication delay cost $c(u, v)$, and each host $v \in V$ is associated with a bounded and finite processing delay cost $p(v)$. Note that the original model of Cidon *et al.* [9] assumes homogenous processing and communication costs, i.e., $p(v) = P, \forall v \in V$, and $c(u, v) = C, \forall (u, v) \in E$.

The direct communication between hosts can be characterized as follows. Assume that at time t host u initiates processing of a message designated for host v. Then, host u is busy processing this message during the time interval $[t, t + p(u)]$ and the message arrives at host v at time $t + p(u) + c(u, v)$. Therefore, the processing delay measure represents the shortest time interval between consecutive message transmissions.

It is important to note that in our model, the delay costs between pairs of hosts do not necessarily satisfy the triangle inequality. This is a known phenomena in the Internet, stemming in part from policy routing. For example, Jamin *et al.* [12, Figs. 2 and 3] show that about 30–50% of the triangles in the Internet do not obey the triangle inequality.

III. Optimal Multicast Tree Problem

In this section we state our design objective formally and show that the optimal multicast tree problem is NP-complete.

We use the term multicast scheme to refer to the task of distributing a message from a source host to a subset of hosts M in the overlay network. Since one cannot relay on the cooperation of nonparticipating hosts (i.e., hosts which do not belong to the multicast group M), we assume that only the hosts in M are allowed to participate in the distribution. Thus, a multicast scheme in the graph $G = (V, E)$ can be viewed as a broadcast scheme, i.e., the task of distributing a message to the entire network, in the subgraph (M, E_M) induced by the host set $M \subseteq V$.

We formulate the optimal multicast tree problem, also denoted as *minimum-delay multicast (MDM)* problem, as follows.

Definition 1: **The optimal multicast tree problem (MDM):** Given a directed complete graph $G = (V, E)$, a multicast group $M \subseteq V$, a source host $s \in M$, a nonnegative real processing delay $p(v)$ for each vertex $v \in V$, and a nonnegative real communication cost $c(u, v)$ for each edge $(u, v) \in E$, find a multicast scheme which minimizes the delay by which all the hosts in M receive a message from s.

Our objective is to devise a multicast scheme which minimizes the arrival time of the message to the last host. We therefore restrict this study to nonlazy multicast schemes (this term was introduced in [13]), where a host that has already received a message does not delay message distribution by becoming idle. Such schemes correspond to an ordered directed tree T, rooted at s, and spanning M. In this tree the outgoing edges of a nonleaf node u are ordered according to the message distribution order of host u in the corresponding multicast scheme. That is, the ith outgoing edge corresponds to the ith transmission. The *reception delay* of host v, denoted by $t_T(v)$, is defined as the time at which v receives a message from s. By definition, the reception delay of s is zero. The cost of a multicast tree T is defined as the earliest time at which all the hosts have been notified, i.e., $\max_{v \in M} t_T(v)$. Note that it is also possible to look for a tree that minimizes the average message arrival time. However, the average may not be an appropriate metric because improving the delay of nodes with small reception time has no real impact on the application performance while actually decreasing the tree cost.

Given a multicast tree we can easily calculate the optimal ordering using a recursive computation working bottom-up. This recursion is presented in the Appendix. Thus, in the rest of the paper we neglect the ordering and concentrate on finding the optimal tree.

We show that the MDM problem is NP-hard using a simple reduction from the telephone broadcast (TB) problem. In the *Telephone model* communication is synchronous and each node can either send or receive a single message per communication round. The TB problem seeks an optimal broadcast scheme which distributes a message from a root node, r, to all the nodes in a graph in a minimum number of rounds. The TB problem is known to be NP-hard for arbitrary graphs [14].

Theorem 1: The MDM problem is NP-hard.

Proof: We will show that given a delay bound K, deciding if there is a multicast scheme with a distribution delay of at most K is NP-complete. The proof is by a reduction from TB. Given an instance to TB, $G_{\mathrm{TB}} = (V, E_{\mathrm{TB}})$ and $r \in V$, we construct an instance to MDM as follows: a complete overlay network $G = (V, E)$ with unit processing costs $p(v) = 1 \ \forall v \in V$, and communication delay defined as $c(u, v) = 0 \ \forall (u, v) \in E_{\mathrm{TB}}$ and $c(u, v) = K + 1 \ \forall (u, v) \notin E_{\mathrm{TB}}$. We let $s = r$ and $M = V$. In the resulting MDM instance, there is a multicast scheme with a distribution delay at most K if and only if there is a telephone broadcast with at most K rounds. ■

IV. Multicast Algorithms

A. Related Work

Broadcast and multicast are important communication primitives for distributed computing. They have been extensively studied in the context of several communication models which consider sequential message distribution. One model which was widely investigated is the basic telephone model, described in the previous section. Some telephone model studies have focused on the problem of designing optimal broadcast schemes for specific classes of graphs (see [15] for a comprehensive survey), while others have suggested approximation algorithms for optimal broadcasting in general graphs [13], [16]–[19].

Cidon *et al.* [9] presented a communication model for high-speed networks which captures communication costs using two parameters—transmission delay and computation delay. In this model the network is represented by a graph $G = (V, E)$. Each node is associated with a processing delay cost P and each edge is associated with a communication delay cost C. The time needed for a node to handle the transmission of i messages is $i \cdot P$. For complete communication graphs the authors presented an optimal tree-based broadcast algorithm (see [9]) and showed that the broadcast delay using such trees is logarithmic in the

size of V. This implies that any nonlazy broadcast (such as our proposed heuristic algorithm scheme) would lead to a broadcast tree with a logarithmic delay. Raz and Shavitt [20] presented a general version of this model which supports IP-like routing and considered efficient multicasting algorithm (based on balanced trees) for line topologies.

The *postal model* [13] is a related model which incorporates nonuniform communications and switching delays and thus captures networks which may have different link delays and different switching times between messages. Optimal algorithm for broadcasting in complete homogenous-cost postal networks is given in [21]. Several approximation algorithms with polylogarithmic ratios were suggested for minimum time broadcasting in general graphs [13], [18], [19]. For the *undirected* broadcast problem the best known approximation ratio is $\log k/(\log\log k)$ [18], where k is the size of the broadcast group. For the *directed* broadcast problem the upper bound on the approximation ratio is higher and currently stands on $\log k$ [19].

B. Comparison of Postal and Overlay Models

Although both models (the postal and overlay) incorporate similar measures for characterizing heterogenous networks, to the best of our knowledge postal-based approximation algorithms cannot be directly used to solve the MDM problem. This results from a timing difference between the two models. Before moving on to describe this discrepancy let us first give a formal description of the postal model.

Similarly to the overlay model the postal model represents the communication network using a complete graph $G = (V, E)$. Each node v is associated with a delay parameter s_v such that v is considered busy (engaged only with the current transmission) in the first s_v time units following the previous transmission time. Each link $(u, v) \in E$ is associated with a delay parameter λ_{uv} which represents the delay of delivering a message from u to v.

The difference between the models can be categorized as follows.

- In the postal model the communication latency λ_{uv} incorporates the sending time of node u, whereas the overlay model incorporates the sending time in the processing delay of v. Thus, the delay of delivering the ith message from u to v is $c(u, v) + (i - 1) \cdot s_u$ for the postal model and $c(u, v) + i \cdot s_u$ for the overlay model (assuming all costs are represented by postal notations $\forall u, v : p(u) = s_u, c(u, v) = \lambda_{u,v}$).
- The postal model enforces node delay constraints by assuming that $s_u < \lambda_{uv}, \forall (u, v) \in E$.

To approximate the directed MDM problem one can use an existing postal-based approximation algorithm. Consider an overlay configuration (G, p, c), where $G = (M, E_M)$ is a directed graph and c and p are associated communication and processing delay functions, respectively. We construct a cost-preserving postal configuration (G, s, λ), such that $\forall u, v \in M : \lambda_{u,v} = p_u + c(u, v), s_u = p(u)$. The key property of this construction is that it guaranties timing equivalency between the models. Thus, using the transformation on the edge costs, any postal-based α-approximation[1] algorithm for the directed broadcast problem would lead to α-approximation algorithm for the directed MDM problem.

The latter approach can be used to solve the undirected MDM problem as well (by replacing each undirected edge with two antiparallel directed edges with equal length). However, as noted in Section IV-A the current best known approximation ratio for the undirected broadcast problem is lower than the one for the directed problem. This motivated us to develop a postal-based approximation for the undirected MDM problem. In the following section we present a more sophisticated transformation technique, which given a postal-based α-approximation algorithm for the undirected broadcast problem constructs an $O(\alpha)$-approximation algorithm for the undirected MDM problem.

In Section VI we use an alternative heuristic approach to develop a greedy algorithm for the MDM problem which admits a simple implementation. We also show that the suggested approximation and heuristic algorithms can be easily extended to generate shared trees without suffering from significant performance degradation. The shared-tree approach implies that a single tree is constructed for the purpose of multisource multicast (see [22]). Of course, using multiple single source multicast trees always achieve lower delay, but at the expense of the management and resource usage overhead.

V. Approximation Algorithm

Given a postal-based α-approximation algorithm for the undirected broadcast problem, our goal is to design an $O(\alpha)$-approximation algorithm for the undirected MDM problem. We consider a two step design. First, we develop a basic approximation algorithm which guarantees the desired approximation ratio up to an additive factor. In the second step we use the basic algorithm as a building block to derive an improved approximation algorithm, and get the desired $O(\alpha)$-approximation ratio.

A. Basic MDM Approximation Algorithm

We now describe a polynomial-time algorithm for approximating the minimum multicast delay in an overlay network. The input is composed of an overlay network configuration (G, c, p) and a source host $r \in M$, where $G = (M, E_M)$ is an undirected graph and p and c are the associated processing and communication cost functions, respectively. Recall from Section III that (M, E_M) is the subgraph induced by the multicast group M. The basic approximation algorithm is given below.

Algorithm Approx-MDM (G, p, c, r)

1. Construct a Postal configuration instance $I_P = (G, s, \lambda)$ that consists of the graph G, switching time function $s_v = p(v), \forall v \in M$, and communication latency function $\lambda_{u,v} = c(u, v) + (p(u) + p(v))/2, \forall (u, v) \in E_M$.
2. Use a postal-based approximation algorithm to compute a multicast tree in I_P rooted at r.
3. Return the computed multicast tree.

Let OPT be the minimum multicast delay in (G, c, p) for source host r and multicast group M. Let $p_{\max}$ and $p_{\min}$ be

[1]An α-approximation algorithm is an algorithm that guarantee to achieve a result which is at most α times worse than the optimal.

the maximum and minimum processing delays of the hosts in M, respectively.

Theorem 2: The multicast delay of the solution that Approx-MDM algorithm returns, is at most $(\text{OPT} + (p_{\max} - p_{\min})) \cdot \alpha$.

Proof: Consider an arbitrary multicast tree T which spans M. Let $t_T^P(v)$ denote the reception delay of a node $v \in M$ assuming postal model delays defined by the configuration I_P. By substituting the computed costs of I_P with the corresponding overlay input costs we get the following relationship between the reception delay costs:

$$t_T^P(v) = \frac{p(v) - p(s)}{2} + t_T(v). \tag{1}$$

The latter equality follows from the delay gap between the models which consists of a single processing round per each traversed host (see Section IV-B).

Consider the following quantities computed assuming postal model delays defined by I_P. Let OPT_P be the multicast delay of an optimal tree T_P^* (in the postal model), and let $u \in M$ be a node with a maximum reception delay in T_P^*. Therefore

$$\text{OPT}_P \leq \text{OPT} + \frac{p(u) - p(s)}{2} \leq \text{OPT} + \frac{p_{\max} - p_{\min}}{2} \tag{2}$$

where the first inequality follows from (1). Substituting OPT_P according to inequality (2), and using the fact that the postal delay of the resulting solution is at most αOPT_P (since we use an α-approximation algorithm), gives the requested upper bound. ∎

When the processing delays are all equal the approximation ratio reduces to α. Note that we do not restrict the communication costs to be homogeneous.

Theorem 3: Consider an overlay model with homogenous processing costs, i.e., $p(v) = p, \forall v \in M$. In this configuration the multicast delay of Approx-MDM is at most $\text{OPT} \cdot \alpha$.

Theorem 3 can be obtained by substituting $p_{\max} = p_{\min} = p$ in Theorem 2.

Moreover, when the processing delays are all at most OPT, our approximation algorithm for the MDM problem is an $O(\alpha)$-approximation. This fact follows by Theorem 2.

Theorem 4: Consider an overlay model with $p(v) \leq \text{OPT}, \forall v \in M$. In this configuration the multicast delay of Approx-MDM is at most $(3/2)\alpha \cdot OPT$.

Recall from Section IV-B that the postal model enforces node delay constraints $s_u < \lambda_{uv}, \forall (u, v) \in E$. Observe that we can premultiply λ_{uv} in Approx-MDM by 2 for all u, v so the resulting I_P instance satisfies this condition of the postal model. In this case we lose a factor of 2 in the approximation ratio. Thus, in the rest of the paper we can ignore these constraints and assume general mode delays.

Given an undirected network configuration we can modify the (rooted tree) solution to support multisource multicast. Let T be such a multicast tree rooted at s. To multicast from an arbitrary host $v \in M$ we simply reverse the direction of the edges on the path from s to v, i.e., the undirected version of T is the shared tree. The multicast delay of an arbitrary host $v \neq s$ is at most $2 \cdot (OPT_s + (p_{\max} - p_{\min})) \cdot \alpha)$ where OPT_s denotes the optimal multicast delay from s.

B. Improved Approximation Algorithm for the MDM Problem

We now show how to improve our approximation algorithm, and get an $O(\alpha)$-approximation. We first assume that we know OPT in advance. Our improvement is based on the following insight.

Observation 5: Let T^* be an optimal solution to the MDM problem, then each vertex v such that $p(v) > \text{OPT}$, is a leaf in T^*.

Proof: A vertex v with $p(v) > \text{OPT}$ does not transmit to other vertices during the reception delay of T^*. ∎

We identify the subset of vertices $S \subseteq M$ such that for every $v \in S$ $p_v \leq \text{OPT}$. T^* induces a connected subtree over S.

Observation 6: The optimal solution delay of the instance of MDM defined by the induced subgraph of G over S, with the restriction of p and c on S has a reception delay of at most OPT.

Proof: The subtree of T^* induced over S is a feasible solution with a reception delay of at most OPT. ∎

We approximate the MDM using the Approx-MDM algorithm where the input graph is the induced subgraph of G over S. This approximate solution has a delay of at most $(3/2)\alpha \cdot \text{OPT}$. This is so by Theorem 4 since $p_v \leq \text{OPT}$ $\forall v \in S$. Now, we are left with the problem that the vertices in S needs to send the message to $M \setminus S$. Note that the vertices in $M \setminus S$ do not participate in broadcasting.

We identify a bipartite graph $BG = (S, M \setminus S, E_{BG})$ (of allowed transmissions) such that its sides are S and $M \setminus S$, and there is an edge (u, v) in BG if and only if $c(u, v) \leq \text{OPT}$. Next, we identify a degree bound for each vertex u, d_u as follows: $d_u = \min\{n - 1, \lfloor(\text{OPT}/p_u)\rfloor\}$ for all $u \in S$ and $d_u = 1$ for all $u \in M \setminus S$.

We find a maximum sized B-matching in BG with degree bounds d. This can be computed using a maximum flow computation in a bipartite graph. If the B-matching does not span all the vertices in $M \setminus S$ then our guess of OPT is too small. Whereas, if we use a value of OPT that is at least as large as the real one, then T^* induces a feasible B-matching that spans all the vertices in $M \setminus S$. In the last case we add the set of edges that belong to the B-matching to our constructed tree, and we obtain a spanning tree that spans M.

Lemma 7: If we know the actual value of OPT, then the above algorithm is an $((3/2) + 2) \cdot \alpha$-approximation.

Proof: Denote by T_{alg} the solution returned by the algorithm. For each $v \in S$ the reception delay of v in T_{alg} is at most $(3/2)\alpha \cdot \text{OPT}$. At time $(3/2)\alpha \cdot \text{OPT}$ the vertices in S have already received the message from the source $\tilde{s}$.

For a vertex $v \in M \setminus S$, by construction the reception delay of v is at most the reception delay of its father u in T_{alg} (where $u \in S$) plus the length of time from the time $(3/2)\alpha \cdot \text{OPT}$ until u starts transmitting to v plus $c(u, v)$. The reception delay of u in T_{alg} is at most $(3/2)\alpha \cdot \text{OPT}$ because at time $(3/2)\alpha \cdot \text{OPT}$ the vertices in S have already received the message. The edge (u, v) belongs to the B-matching, and therefore the number of edges in BG that are adjacent to u is at most d_u. Therefore, since the time $(3/2)\alpha \cdot \text{OPT}$ u transmits to at most d_u vertices, and therefore it starts to transmit to v at time at most $(3/2)\alpha \cdot \text{OPT} + d_u p_u \leq ((3/2) + 1)\alpha \cdot \text{OPT}$. Since $(u, v) \in E_B, c(u, v) \leq \text{OPT}$, and therefore the reception delay of v is at most $((3/2) + 2) \cdot \alpha \cdot \text{OPT}$. ∎

Algorithm LRF(G, p, c, s)

1. $t[s] = 0$, set s as the root of a tree T
2. for each $v \in M - \{s\}$
3. do $t[v] \leftarrow \infty$
4. for each $(u, v) \in E_M$
5. do $w_{u,v} = c(u, v) + p(u)$
6. for each $(u, v) \notin E_M$
7. do if $v = u$ then $w_{u,v} = 0$ else $w_{u,v} = \infty$
8. $D, \Pi \leftarrow$ All-Pairs-Shortest-Path(G, W)
9. while $M - V[T] \neq \emptyset$
10. for each host $u \in M - V[T]$ do
11. $m[u] \leftarrow \arg\min_{v:v \in V[T]}\{t[v] + d_{v,u}\}$
12. $v \leftarrow \arg\max_{u:u \in M - V[T]}\{t[m[u]] + d_{m[u],u}\}$
13. $w \leftarrow v$
14. while $w \neq m[v]$ do
15. $t[w] \leftarrow t[m[v]] + p(w) + d_{m[v],w}$
16. add w to T as a child of $\pi_{m[v],w}$
17. $w \leftarrow \pi_{m[v],w}$
18. $t[m[v]] \leftarrow t[m[v]] + p(m[v]), t[v] \leftarrow t[v] - p(v)$
19. return T

Fig. 2. Greedy tree construction for the MDM problem.

In order to obtain an $O(\alpha)$-approximation algorithm we note that if we know the set S, the edge set E_{BG}, and the set of degree bounds for all the vertices in S that corresponds to the value of OPT, then our algorithm is an $O(\alpha)$-approximation. Note that all together there are $O(n + m + n^2)$ critical values in which these information is changing, where m and n denote the number of edges and nodes in G, respectively. Thus, we define a set Cr of critical values, and $t \in Cr$ if the execution of the algorithm for OPT $= t + \epsilon$ and OPT $= t - \epsilon$ differ for all $\epsilon > 0$. Moreover, Cr can be identified in $O(n^2)$ time. We sort Cr and apply a binary search over it. In each iteration we pick a value $\tilde{\text{OPT}}$ of OPT that is between a pair of adjacent critical values, and we apply our algorithm with this guess. If we get a feasible solution such that its reception delay is at most $((3/2) + 2)\alpha \cdot \tilde{\text{OPT}}$ then OPT $\leq \tilde{\text{OPT}}$, otherwise OPT $> \tilde{\text{OPT}}$. This binary search uses $O(\log n)$ iterations. Each iteration is polynomial, and therefore the resulting algorithm is polynomial.

Picking the best feasible tree that we obtain during the binary search gives an $O(\alpha)$-approximation algorithm. We summarize this by the following theorem.

Theorem 8: There is a polynomial-time $O(\alpha)$-approximation algorithm for MDM.

Note that if the α-approximation algorithm for the postal model is strongly polynomial time algorithm, then our approximation algorithm for the MDM problem is also strongly polynomial.

VI. Heuristic Algorithm

We introduce a heuristic tree construction algorithm, named largest ready time first (LRF), which solves the directed variant of the MDM problem. The proposed algorithm computes the multicast tree incrementally using a greedy approach; for each host not yet included in the tree, the algorithm computes its minimum reception delay, and the host with the maximal delay quantity is selected. The tree is extended with the hosts on a minimum-delay path between the selected host and a notified host. Fig. 2 shows the steps of the algorithm.

The input to this algorithm is the same as the input to the Approx-MDM, except for the source host which is denoted by s. The algorithm maintains a ready time attribute $t[v]$ for each host $v \in M$ which records the minimum time at which the host is free to initiate processing of a new message. The ready time is set to infinity to indicate nonnotified host. The constructed tree is denoted by T and the corresponding set of notified hosts by $V[T]$. In each iteration, the algorithm determines for each host $u \in M - V[T]$ its mate host $m[u] \in V[T]$ by selecting a path which minimizes the ready time attribute of u, setting v to indicate the host with the maximal reception delay. Then, it updates the ready time of the hosts on the path from $m[v]$ to v to reflect the processing time involved with delivering a message to the newly notified host v, and it adds the path hosts to the constructed tree T. The variable w indicates the current updated host. The algorithm terminates when all the hosts are notified.

To be able to calculate the connection cost between a nonnotified host and a notified host, a preprocessing phase of computing all pairs shortest path using the Floyd–Warshall algorithm [23] is implemented. Given a pair of hosts v_1 and v_k connected by a path $\langle v_1, \ldots, v_k \rangle$ of length $k - 1$, the cost of this path is defined as $\sum_{i=1}^{k-1} p(v_i) + c(v_i, v_{i+1})$, where $v_i, 1 \leq i \leq k$ denotes the ith host on this path, i.e., this cost represent the minimum distribution delay (along the specified path) from v_1 to v_k. A shortest path from host u to host v is defined as any path between these hosts with minimum cost. Therefore, the input to the Floyd–Warshall computation is an $n \times n$ weight matrix $W = (w_{v_i,v_j})$ defined as

$$w_{v_i,v_j} = \begin{cases} p(v_i) + c(v_i, v_j), & \text{if } v_i \neq v_j \\ 0, & \text{otherwise.} \end{cases}$$

where n denotes the size of M. The output of the all pairs shortest path computation is composed of two $n \times n$ matrices; all pairs distance matrix $D = (d_{v_i,v_j})$ and predecessor matrix $\Pi = (\pi_{v_i,v_j})$ (see [23]). Observe that the shortest path from the source s to any host v is a lower bound on the cost of the optimal tree.

This algorithm can be extended to support a shared-tree solution using the following modification. At the initialization phase the longest path in the graph G is computed using the weight matrix W, and the hosts on this path are used as the initial set of notified hosts in T. The shared-tree variant uses this initial selection instead of the original one and proceeds with normal tree construction as in the original algorithm.

The complexity analysis of this algorithm is straightforward. The all pairs shortest path computation requires $\Theta(n^3)$ time. Each iteration requires $O(n)$ time to find a single mate host, and $O(n)$ time to extend the tree. The total time per iteration is therefore $O(n^2)$, and the total running time of the LRF heuristic is $\Theta(n^3)$.

We show using an example [see Fig. 3(A)] a lower bound on the approximation ratio of the heuristic tree. Consider the following complete undirected graph $G = (V, E)$ with $n + 1$ hosts denoted by $v_0, \ldots, v_n$, with processing costs defined as $p(v) = 1, \forall v \in V$, and communication costs $c(v_i, v_j)$ defined as

$$c(v_i, v_j) = \begin{cases} 0, & \text{if } i = 0, \quad j = 1, \ldots n \\ \delta, & \text{if } 1 \leq i \leq n - 1, \quad j = i + 1 \\ n, & \text{otherwise} \end{cases}$$

where $\delta \to 0$. For the simplicity of presentation Fig. 3(A) omits the edges with cost n. Assume that the source host is v_0 and

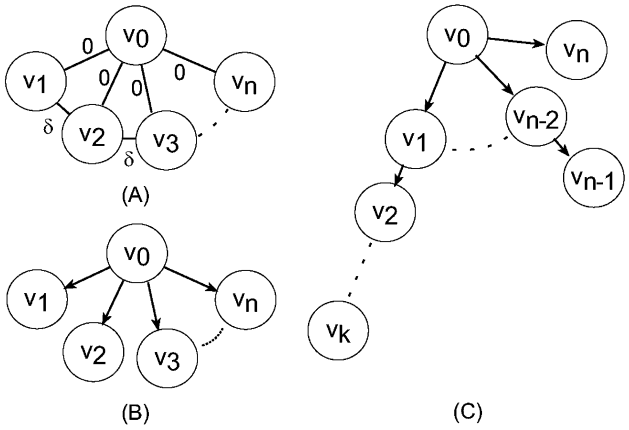

Fig. 3. Example that provides $\sqrt{n}$ approximation ratio for the heuristic tree. (A) The input graph (B) The heuristic tree. (C) An optimal tree for $n = (k \cdot (k+1))/2$.

that $M = V$. Therefore, the LRF scheme would have v_0 distribute the message to the rest of the hosts using n processing rounds, such that the tree cost is n [see Fig. 3(B)]. On the other hand, consider an improved scheme in which v_0 distributes the message to k hosts, and the last host in the graph receives the message in $k + t\delta$ time units, where t is some positive integer. Let $|p_i|$ denote the length (i.e., the number of edges) of path p_i. Such a scheme can be obtained by a tree composed of k paths $p_1, \ldots, p_k$ which share only single host, v_0 (i.e., only v_0 has an out-degree larger than two), and the lengths of these paths form the following nonincreasing sequence: $|p_i| - 1 = |p_{i+1}|$,$1 \leq i \leq k-1$, whereas for a single index j in this set we may have $|p_j| = |p_{j+1}|$. Fig. 3(C) depicts such a tree when $n = (k \cdot (k+1))/2$. Assume that v_0 distributes the message to these paths, i.e., to its k children in the tree, according to a decreasing path length order. Therefore, the cost of the optimal tree is less than $(1+\delta) \cdot k$. Note that in this particular scheme $t = k$. Since the set of paths span all the hosts in V we have that $k = O(\sqrt{n})$, and therefore we get $\Omega(\sqrt{n})$ approximation ratio for the multicast delay. We conjecture that this example represents the worst case, namely that our LRF heuristic algorithm is an $\sqrt{n}$-approximation.

VII. Topologies

In this section we analyze the performance of broadcast in partially connected overlay networks (e.g., structured graph topologies) which are widely used in various contexts.

Partial connectivity is implemented by many data distribution services, such as content distribution networks and multimedia streaming systems, which utilize a dedicated network of leased lines and virtual connections to provide connectivity among application servers. These systems optimize resource usage and thus enforce connectivity constrains to achieve efficient resource utilization. Structured p2p systems [10] are another class of applications which utilize partial connectivity overlays. Despite the fact that many of these systems employ distributed architectures, our centralized application-centric approach can still be used to provide theoretical performance bounds on the multicast delay in such systems.

Partial connectivity may also arise in cases where due to anonymity requirements not all the hosts are aware of each other and thus connectivity is sparse. That is, hosts use local policies to override universal connectivity. For example, consider security policies in the Internet, which limit the connectivity of hosts located behind firewalls and NAT facilities.

Performance analysis of arbitrary topologies is relevant also for active networks [20]. Active networks use programmable routers to add new functionality and services to the network, and thus may be viewed as a network-level implementation of overlay networks. For example, Raz and Shavitt [20] have used a framework that considers the processing and communication delays in active networks to develop and analyze the time complexity of several basic algorithms, including multicasting. Their framework uses the processing delay measure to capture the delay imposed by a software router implementing copy and forward of packets.

To support networks with partial connectivity an extended overlay model is assumed where the communication cost of an overlay link (u, v) is set to infinity, i.e., $c(u, v) = \infty$, to indicate the absence of direct communication from u to v. In the following section we develop performance bounds for the minimum-delay multicast problem in several common graph topologies. The good performance on the examined topologies may indicate that our heuristic can perform well in other configurations.

A. Trees

We consider broadcasting in tree graphs. In these graphs each node has a single path from the root, implying that any broadcast scheme is characterized only by the message distribution order of nonleaf hosts. Recall that such instances can be solved by dynamic programming procedure in polynomial time.

Lemma 9: Any (nonlazy) broadcast scheme provides a factor d approximation for the minimum broadcast delay for a tree graph T with a maximal degree of d.

Proof: Denote by s the source host. We use the path cost notation defined in Section VI, i.e., the cost of a path represents the minimum distribution delay along it. In any (nonlazy) broadcast scheme the delay by which the last notified host, denoted by v, receives a message is composed of two quantities, the cost of the path from s to v, and the sum of the additional processing delays invoked by the hosts on this path (the additional delay of v is assumed to be zero). By definition, the former quantity is no more than OPT, where OPT denotes the optimal broadcast delay. We denote by $\langle v_1, \ldots, v_k \rangle$ the path of length $k-1$ which connects between s and v, such that $v_1 = s$ and $v_k = v$. Due to the bound on the degree of the tree, each node may delay the processing by at most $d-1$ processing rounds, and therefore the sum of the additional processing delays is at most $(d-1) \cdot \sum_{i=0}^{k-1} p(v_i)$, where $v_i, 1 \leq i \leq k$ denotes the ith host on the path from s to v. It is easy to see that this quantity is at most $(d-1) \cdot \mathrm{OPT}$, and the lemma follows. ■

This result implies that multicasting in a degree-bounded tree at an arbitrary order, such as the delivery schemes used by overlay multicast systems which ignore sequential distribution of messages (see for example [24]), produces a delay which is up to a multiplicative constant factor higher than the optimal result.

B. Grids

This section investigates broadcasting in the context of homogeneous rectangular grid graphs. Let $G_{m,n} = (V, E)$ denote an $m \times n$ grid graph. Each host in this graph is uniquely identified by a row and column indexes (i, j), where $1 \leq i \leq m$ and $1 \leq j \leq n$. The broadcast analysis is conducted assuming a homogeneous cost model where $p(v) = 1, \forall v \in V$, $c(u, v) = 0, \forall (u, v) \in E$. This particular selection reduces the model to the well-known telephone model, and enables the usage of known results in grid broadcasting.

The problem of finding an optimal broadcast scheme in 2-D grid graphs have been previously investigated by Farley *et al.* [25]. They have shown that given a grid graph $G_{m,n}$ with a node v at position (i, j), then

$$b(v) = \begin{cases} D+2, & \text{if } i = j = \frac{m+1}{2} = \frac{n+1}{2} \\ D+1, & \text{if } i = \frac{m+1}{2} \quad \text{or} \quad j = \frac{n+1}{2}, \quad i \neq j \\ D, & \text{otherwise.} \end{cases}$$

where $b(v)$ denotes the optimal broadcast time (i.e., delay) from v, and D denotes the maximal distance from v to a corner node in $G_{m,n}$. The distance between a pair of nodes u and v in positions (i_u, j_u) and (i_v, j_v), respectively, is defined as the number of edges on the shortest path between them.

Next, we provide a new result on broadcasting in grid graphs using shortest path trees. Let OPT denote the cost of an optimal solution.

Theorem 10: The broadcast delay of a shortest path tree for homogenous cost grid graph $G_{m,n} = (V, E)$ is at most OPT$+2$

Proof: Let s denote the source host, and let T denote a directed shortest path tree (SPT) rooted at s. The SPT structure implies the following degree delegation in T. If s is a corner host then its degree is 2 and the rest of the hosts have maximal out-degree of 2. If s is a side host or interior host, then the maximal out-degree of the interior hosts which share a common coordinate with s is 3 and the maximal out-degree of the rest of the hosts is 2. The degree of s is 3 when s is a side host, and 4 when it is an interior host. Let S_3 denote the set of hosts in $V \setminus \{s\}$ such that the out-degree of these hosts in T is 3, i.e., $S_3 = \{v : \deg(v) = 3, v \neq s\}$ where $\deg(v)$ denotes the out-degree of v in T.

Let T_2 be a binary subtree of T rooted at r, such that r is either a child of $v \in S_3$ or a side host which is a child of s. The grid topology implies that a subtree of height d, rooted at an internal node of T_2, has a single leaf at depth d. Therefore, by using a bottom-up recursive computation (see Section III) we get that the optimal broadcast delay from the root of a T_2 tree with height d is d. If s is a corner host then T has two T_2 subtrees linked to it (that is, the root of each subtree is a child of s). Since only one of these trees has a height of $D-1$ while the height of the other is at most $D-2$, the broadcast delay from a corner host is D. This delay achieves the optimal value (devised by Farley *et al.* in [25]), and the lemma follows for this case.

The other cases are analyzed using a compressed version of T. A T_2 tree with height d can be "compressed" to a path with d edges which preserve the broadcast delay of the tree. The compressed version of T, denoted as T_c, is produced by replacing all the T_2 subtrees with their corresponding paths. This compression does not modify the broadcast delay of T.

Let T_3 denote a subtree in T_c rooted at a child of s. By definition, the maximal out-degree of this tree is 3. Next, we consider the case of T_3 trees which include at least a single node with an out-degree of 3. The grid topology implies that a subtree of height d rooted at an internal node of $T_3, v \in S_3$, may have at most two leaves at depth d. Each host $v \in S_3$ has three children in T, v_1, v_2 and v_3, ordered according to the height of the subtrees rooted at these hosts, such that $h(T_{v_1}) \leq h(T_{v_2}) \leq h(T_{v_3})$ where T_{v_i}, $i = 1, 2, 3$ denotes the subtree rooted at v_i, and $h(T_{v_i})$ denotes the height of T_{v_i}. Given a subtree of height d rooted at v with a single leaf at depth d, the grid topology implies that $h(T_{v_3}) > \max\{h(T_{v_2}), h(T_{v_1})\}$. If the subtree has two leaves at depth d, then $h(T_{v_3}) = h(T_{v_2}) > h(T_{v_1})$. By using a bottom-up recursive computation we get that the broadcast delay from the root of a T_3 tree with height d is at most $d+1$ when there is a single leaf at depth d, and at most $d+2$ when there are two leaves at depth d.

If s is a side host, the root of T is linked with three T_3 subtrees. If s is a middle side host (i.e., a host with coordinate (i_s, j_s) such that $i_s = (m+1)/2$ or $j_s = (n+1)/2$) there are two hosts at distance D from s. If these two hosts reside in the same T_3 tree, then the maximal height of the remaining T_3 trees is $D-2$ and we have that the broadcast delay from a corner host is at most $D+2$. If these two hosts reside in different subtrees, then the maximal height of the third subtree is $D-2$ and the broadcast delay is again at most $D+2$. In the case of a non-middle side host, the single host at distance D is located at one of the T_3 trees and the maximal height of the remaining trees is $D-2$, and therefore the broadcast delay is at most $D+2$. Therefore, the lemma follows for this case.

If s is a interior host then T has four T_3 subtrees linked to it. By checking all the possible combinations of tree heights and the location of the hosts at distances D and $D-1$, it can be easily shown that the broadcast delay from an interior host is at most OPT $+ 2$. ■

The latter result implies that any SPT-based broadcast (e.g., flooding with a sense of direction) leads to a nearly optimal result.

VIII. Simulation Study

In this section we analyze the average performance of the proposed algorithms on random networks assuming various group sizes and wide range of network costs.

The simulations assume two undirected network topologies, fully connected and partially connected overlay graphs. The topologies of the physical networks and the partially connected overlays are constructed using a power-law graph generator. This generator is based on the Notre Dame model [26] which constructs undirected graphs with power-law node degree frequency distribution using an input parameter set m_0, m, p, q. This parameter set defines the properties of the resulting graph: m_0 is the initial node set, p is the probability to add m new links, and q is the probability to rewire m links. A common parameter set $m_0 = 3, m = 2, p = 0.1, q = 0$ was used to derive all the topologies. This set results in graphs with an average degree of approximately 4.38. In addition, in all the simulations we have selected the multicast group to include all the hosts in the network.

In our simulations we compare the performance of our heuristic algorithm (labelled as H-MDM in the shown graphs) with the following schemes.

- *Approx-MDM.* The underlying postal-based approximation was selected to be Bar-Noy's logarithmic approximation algorithm [13]. This algorithm needs to solve multiple linear programs and therefore it requires high polynomial order running time. In our simulation environment which includes 1.5-GHz PCs with 512M RAM, the Approx-MDM algorithm was able to effectively solve problems with up to 25 hosts.
- *Shortest Path Tree.* This tree is evaluated to assess the performance penalty involved with SPT routing, a common routing scheme employed by many overlay multicast systems. The SPT is computed using Dijkstra's algorithm [23], where the edge weights are defined using the formulation of Section VI.
- *Degree Bounded Tree.* We compare against a scheme which limits the degree of the tree nodes, MDDBST, developed in [11] for application-level multicast systems. This algorithm is a heuristic for computing minimum-diameter, degree bounded tree (an NP-hard problem) and is structured similarly to Prim's algorithm for minimum spanning tree [23]. Unlike many other application-layer tree construction suggestions this scheme allows the user to enforce a degree bound on each node. In the simulations we set the degree bound of a node that has a processing delay p (where p is uniformly selected from the range $[p_l, p_h]$) to be $p_h - p$, i.e., the degree bound is a linear function of the processing delay.
- *Delay bound.* Since the MDM problem is NP-hard (see Section III) the optimal solution could not be computed. Instead, we select the maximum cost of the shortest path (as defined in Section VI) from the source to any other host in the graph. The selected value is a nontight lower bound on the performance of any multicast scheme. In the graphs shown this delay bound is labelled as M-SPATH.

A. Simulation Results

First we describe the format of the plotted graphs. In all the presented results we apply 40 independent simulation experiments per each data point, plotting the mean value with error bars representing a 95% confidence interval. In the case of fully connected overlay networks, we present the simulation results using two plots, one that covers small group sizes up to 25 members and another which shows larger group sizes up to 400 members. Thus, the performance of the heuristic and approximation trees is compared in the context of small group sizes, while large group sizes are used to demonstrate the scaling properties of the heuristic tree versus the SPT and MDDBST.

Next, we present the results for the case of a fully connected overlay network. Figs. 4–6 plot the costs, i.e., the multicast delays, of the LRF, Approx-MDM, MDDBST, and shortest-path trees as a function of the multicast group size. In each simulation the network costs are randomly selected using a discrete uniform distribution on the intervals $([1,10],[1,10])$, $([1,1],[1,10])$, $([1,10],[1,1])$, respectively.

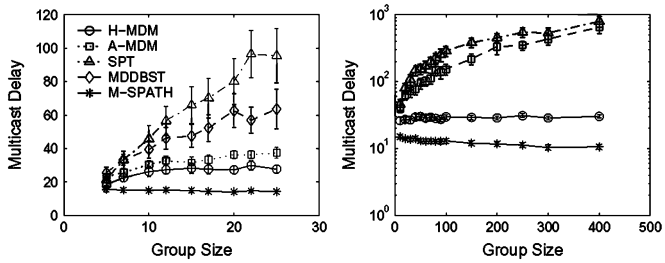

Fig. 4. Multicast delay for a clique topology with random network costs from $[1, 10]$.

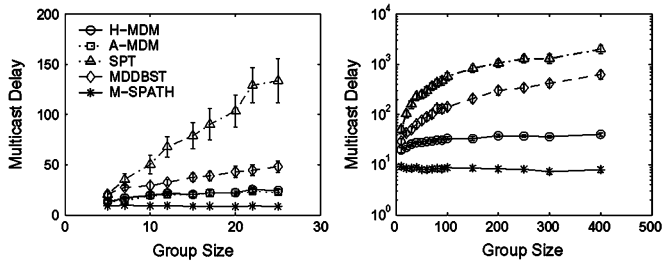

Fig. 5. Multicast delay for a clique topology with random processing costs from $[1, 10]$ and unit communication costs.

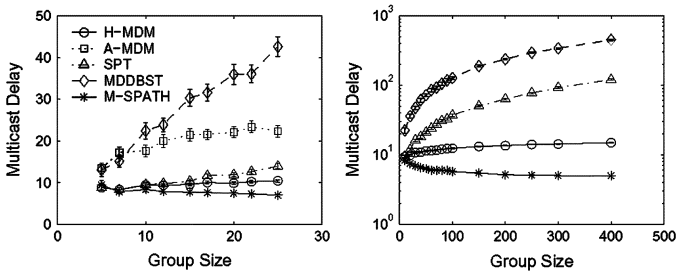

Fig. 6. Multicast delay for a clique topology with random communication costs from $[1, 10]$ and unit processing costs.

The left range in each pair is the communication cost range, and the right range is the processing cost range.

According to Fig. 4, the cost of the heuristic tree is up to 30% smaller than the cost of the approximation tree. Fig. 5 indicates that the trees achieve similar cost when the processing costs dominate the communication costs. Fig. 6 shows that in the alternative case of network with dominating communication costs, the heuristic tree cost can be up to three times smaller than the approximation cost. The latter case captures Internet-like scenarios. We note that the performance gap between the heuristic and the approximation, stems from the fact that the approximation scheme inherently constructs trees with logarithmic height. Such trees are more likely to include high-cost communication delays, which increase their inefficiency compared to heuristic trees.

The plots indicate that the cost of the degree bounded tree is significantly higher then the cost of our heuristic tree, and that the performance difference is highest in Internet-like scenarios. Note that the performance of MDDBST can most likely be improved by trial-and-error exploration of the possible node degree bound space, which grows exponentially with the number of nodes. As expected, SPT which does not attempt to minimize

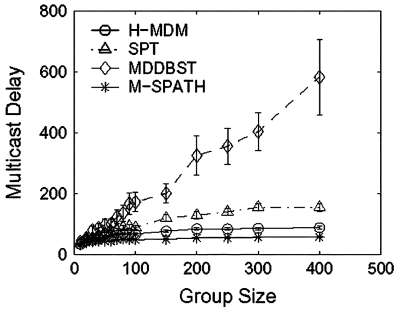

Fig. 7. Multicast delay for a power-law topology with random network costs from $[1, 10]$.

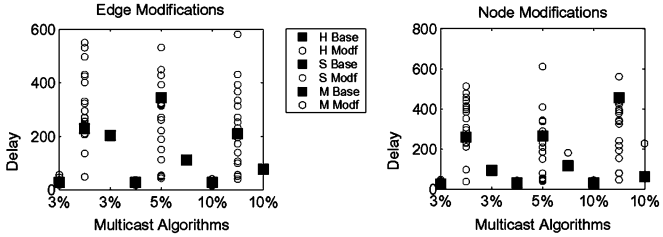

Fig. 8. Sensitivity of LRF (labelled H), SPT (labelled S), and MDDBST (labelled M) to (left plot) 3%, 5%, and 10% edge or (right plot) node modifications.

the tree degrees has low performance. Its cost function is almost linearly proportional to the tested group size (the delay curves exhibit logarithmic-like growth rate since they are shown using a logarithmic scale). The quality of the SPT is determined by the dominance of the communication costs, i.e., the applicability of SPT is limited to small multicast groups in overlay networks with dominating communication costs (Fig. 6).

The previous experiments were repeated using other cost intervals, $[1, 5]$, $[1, 100]$, preserving the methodology of network cost selection. The obtained results were consistent with the previous outcomes. We also simulated near homogeneous costs and verified the logarithmic convergence rate (see [9]) of LRF.

We used a 400-node physical network, based on a power-law graph, to simulate fully connected overlay structures over the Internet. In each simulation the multicast group hosts were attached to a randomly selected uniformly distributed set of edge nodes in the power-law topology. The communication costs were derived according to the minimum hop count, yielding an average overlay link cost of 4.8 hops with a maximal value of 9 hops. The processing costs were randomly selected from the discrete intervals $[1, 5]$, $[1, 10]$, and $[1, 100]$. Unsurprisingly, the obtained results were similar to the previous results which use random cost selection, and therefore the corresponding graphs are omitted.

Next, we consider the case of partially connected overlay networks derived using the power-law topology generator. In this case, we were not able to apply the approximation scheme due to the implicit full-connectivity assumption of the algorithm. Therefore, we compare the performance of the heuristic tree with SPT and MDDBST, using the same network costs as in the fully connected case. The results indicate that the heuristic tree scales well, such that its maximal cost is up to 80% higher than the lower bound, which is not tight. Fig. 7 shows a typical large scale result with processing and communication costs randomly selected from the discrete intervals $([1, 10], [1, 10])$. The large-scale results for a clique topology are similar. For example, see Fig. 4 in which the maximal cost of the heuristic tree is up to three times higher than the nontight lower bound.

The main conclusion drawn from the simulations is that the heuristic algorithm produces results which are very close to the optimal for almost any group size, showing a logarithmic-like growth rate. Furthermore, the average performance of the heuristic algorithm is similar or better than the performance of the approximation algorithm, whereas the SPT and MDDBST provide lower performance and produce nonscalable results.

B. Sensitivity Analysis

We study the sensitivity of our heuristic solution to small changes in the network and compare it to the sensitivity of the the shortest path tree and MDDBST algorithms. For this purpose we generate a 100-node click network with randomly assigned processing and communication costs drawn from a uniform discrete distribution in the interval $[1, 10]$. We then randomly select 3%, 5%, and 10% of either the nodes or edges, reassign their costs with randomly generated discrete values uniformly distributed in an interval $[1, 10]$, and recompute the delay of the examined algorithms. We repeat the last step 20 times.

Fig. 8 shows the distribution of the multicast delays under the examined algorithms as a function of the portion of edges or nodes being modified. The highlighted marker is the baseline, i.e., the cost of the unmodified network.

The graphs indicate that our heuristic algorithm is insensitive to either small edge cost changes or small node cost changes, unlike SPT. It is interesting to note that MDDBST is also very insensitive to small changes. However, the cost of MDDBST trees is significantly higher then the cost of our heuristics trees.

IX. Concluding Remarks

In this work we looked at building efficient application-layer multicast trees. We have presented two solutions to the minimum-delay multicast problem: an approximation algorithm and a heuristic algorithm. It is interesting to see that in practice the heuristic achieves much shorter delays than the approximation for the cases that represents the Internet, i.e., networks with communication delays larger than processing delays; and both are better than the previously advocated shortest path trees.

Appendix
Optimal Recursive Computation

Consider a tree $T = (V, E)$ rooted at s with associated processing and communication costs p and c, respectively. Our goal is to compute the optimal cost and ordering for T. To this end we employ a bottom-up recursive computation approach.

For each node $v \in V$ we compute the quantity $M(v)$ which represents the optimal cost of the subtree rooted at v, i.e., the minimum time for delivering a message from v to all the nodes

in the subtree. In addition, we compute an auxiliary quantity $m(v)$ which represents the optimal multicast delay from v's parent, u, to the subtree rooted at v assuming zero processing overhead at u

$$m(v) = \begin{cases} M(v) + c(u,v), & v \neq s \\ M(v), & v = s. \end{cases} \tag{3}$$

Consider a nonleaf node v with k children $v_1, \ldots, v_k$. Let $r(v,i)$ be a rank function that returns a child node of v with the ith largest m quantity. The optimal cost of a subtree rooted at v can be expressed by the recursion

$$M(v) = \max_{1 \leq i \leq k} \{m(r(v,i)) + i \cdot p(v)\} \tag{4}$$

and the optimal cost of a subtree rooted at a leaf node is defined to be zero.

The optimal multicast delay for tree T is simply $M(s)$, where the optimal ordering follows the rank function, i.e., node v delivers its i message to $r(v,i)$. The time complexity of this computation is $\Theta(|V| \log |V|)$.

Lemma 11: The recursive computation provides an optimal solution for the MDM problem in a tree graph.

Proof: The proof is by induction on the height of the tree. The basis is trivial. Inductive step: consider a tree of height k and assume that the lemma holds for the subtrees (of height at most $k-1$) linked to the tree root, v. Let l be the number of such subtrees. Assume that v distributes the message to its children in an arbitrary order and let v_{r_i} be the child that received the ith transmission. Due to the induction assumption the cost of the subtree rooted at v is $\max_{1 \leq i \leq l}\{m(v_{r_i}) + i \cdot p(v)\}$. This cost is minimized when v orders its transmission according to the m quantity of the its children, from highest to lowest. ■

ACKNOWLEDGMENT

The authors thank Israel Cidon and Avishai Wool for their helpful comments that improved our presentation in many ways. The first author is deeply grateful to Zvika Lotker for valuable and motivating discussions, and to Yoav Karniely for his help with the simulations.

REFERENCES

[1] A. El-Sayed, V. Roca, and L. Mathy, "A survey of proposals for an alternative group communication service," *IEEE Netw. Mag.*, vol. 17, no. 1, pp. 46–51, Jan./Feb. 2003.

[2] Y.-H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *Proc. ACM SIGMETRICS 2000*, Santa Clara, CA, Jun. 2000, pp. 1–12.

[3] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An application level multicast inproceedings infrastructure," in *Proc. 3rd USNIX Symp. Internet Technologies and Systems (USITS'01)*, San Francisco, CA, Mar. 2001, pp. 49–60.

[4] S. Shi and J. Turner, "Routing in overlay multicast networks," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, vol. 3, pp. 1200–1208.

[5] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003, vol. 2, pp. 1521–1531.

[6] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proc. ACM SIGCOMM 2002*, Pittsburgh, PA, Aug. 2002, pp. 205–217.

[7] N. Malouch, Z. Liu, D. Rubenstein, and S. Sahu, "A graph theoretic approach to bounding delay in proxy-assisted, end-system multicast," in *Proc. 10th IEEE Int. Workshop on Quality of Service (IWQoS)*, Miami, FL, May 2002, pp. 106–115.

[8] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proc. Multimedia Computing and Networking 2002 (MMCN'02)*, San Jose, CA, Jan. 2002, pp. 156–170.

[9] I. Cidon, I. Gopal, and S. Kutten, "New models and algorithms for future networks," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 769–780, May 1995.

[10] M. Castro, M. B. Jones, A.-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman, "An evaluation of scalable application-level multicast built using peer-to-peer overlays," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003, vol. 2, pp. 1510–1520.

[11] S. Y. Shi, J. Turner, and M. Waldvogel, "Dimensioning server access bandwidth and multicast routing in overlay networks," in *Proc. NOSSDAV 2001*, Port Jefferson, NY, Jun. 2001, pp. 83–92.

[12] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: A global internet host distance estimation service," *IEEE/ACM Trans. Netw.*, vol. 9, no. 5, pp. 525–540, Oct. 2001.

[13] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber, "Message multicasting in heterogeneous networks," *SIAM J. Comput.*, vol. 30, no. 2, pp. 347–358, 2001.

[14] M. R. Garey and D. S. Johnson, *Computers and Intractability*. San Francisco, CA: Freeman, 1979.

[15] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks*, vol. 18, no. 4, pp. 319–349, 1988.

[16] R. Ravi, "Rapid rumor ramification: Approximating the minimum broadcasting time," in *Proc. 35th IEEE Symp. Foundations of Computer Science*, 1994, pp. 202–213.

[17] G. Kortsarz and D. Peleg, "Approximation algorithm for minimum time broadcast," *SIAM J. Discrete Math.*, vol. 8, pp. 401–427, 1995.

[18] M. Elkin and G. Kortsarz, "A sublogarithmic approximation algorithm for the undirected telephone broadcast problem: A path out of a jungle," in *Proc. ACM Symp. Discrete Algorithms (SODA)*, Baltimore, MD, 2003, pp. 76–85.

[19] M. Elkin and G. Kortsarz, "Combinatorial logarithmic approximation algorithm for the directed telephone broadcast problem," in *Proc. ACM Symp. Theory of Computing (STOC)*, Montreal, QC, Canada, 2002, pp. 438–447.

[20] D. Raz and Y. Shavitt, "New models and algorithms for programmable networks," *Comput. Netw.*, vol. 38, no. 3, pp. 311–326, 2002.

[21] A. Bar-Noy and S. Kipnis, "Designing broadcasting algorithms in the postal model for message passing systems," in *Proc. Symp. Parallelism in Algorithms and Architectures (SPAA)*, San Diego, CA, Jun. 1992, pp. 13–22.

[22] L. Wei and D. Estrin, "The trade-offs of multicast trees and algorithms," in *Proc. Int. Conf. Computer Communications and Networks (ICCCN'94)*, San Francisco, CA, Sep. 1994, pp. 902–926.

[23] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. New York: McGraw-Hill, 1990.

[24] P. Francis, "Yoid: Extending the multicast inetrnet architecture," White Paper [Online]. Available: http://www.aciri.org/yoid/ 1999

[25] A. M. Farley and S. T. Hedetniemi, "Broadcasting in grid graphs," in *Proc. 9th S-E Conf. Combinatorics, Graph Theory, and Computing*, 1978, pp. 275–288.

[26] R. Albert and A. L. Barabasi, "Topology of evolving networks: Local events and universality," *Phys. Rev. Lett.*, vol. 85, no. 24, pp. 5234–5237, Dec. 11, 2000.

Eli Brosh received the B.Sc. and M.Sc. degrees from Tel-Aviv University, Tel-Aviv, Israel. He is currently working toward the Ph.D. degree at Columbia University, New York, NY.

He has worked in the telecommunications industry as a Systems Engineer and Architect.

Asaf Levin received the B.Sc., M.Sc., and Ph.D. degrees from Tel-Aviv University, Tel-Aviv, Israel.

He is currently a Lecturer in the Department of Statistics, Hebrew University of Jerusalem, Jerusalem, Israel. His research is focused on combinatorial optimization and approximation algorithms.

Yuval Shavitt (S'88–M'97–SM'00) received the B.Sc. degree in computer engineering (*cum laude*), the M.Sc. degree in electrical engineering, and the D.Sc. degree from the Technion—Israel Institute of Technology, Haifa, in 1986, 1992, and 1996, respectively.

From 1986 to 1991, he served in the Israel Defense Forces, first as a System Engineer and the last two years as a Software Engineering Team Leader. After graduation, he spent a year as a Postdoctoral Fellow at the Department of Computer Science, The Johns Hopkins University, Baltimore, MD. Between 1997 and 2001, he was a Member of the technical staff at the Networking Research Laboratory, Bell Laboratories, Lucent Technologies, Holmdel, NJ. Starting October 2000, he has been a Faculty Member in the School of Electrical Engineering, Tel-Aviv University Tel-Aviv, Israel. His recent research focuses on Internet measurement, mapping, and characterization and on QoS in networks. He was an Editor of *Computer Networks* in 2003–2004, and served as a Guest Editor for the *World Wide Web Journal.*

Dr. Shavitt has served as a TPC member for INFOCOM 2000–2003 and 2005, IWQoS 2001 and 2002, ICNP 2001, IWAN 2002–2005, Tridentcom 2005–2006, and more, and on the executive committee of INFOCOM 2000, 2002, and 2003 and was a Guest Editor for the IEEE JOURNAL ON SELECTED TOPICS IN COMMUNICATIONS.