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

Eli Brosh and Yuval Shavitt

***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 constrains. Many previous solutions limited the maximal 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 constraint the load on the hosts.**

In this model the problem is shown to be NP-hard. Therefore, we present a logarithmic approximation algorithm and an alternative heuristic solution. Our heuristic algorithm is shown by simulations to be scalable for large group sizes, and produces results that are very close to optimal.

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 metrics such as delay or throughput. The systems which aim at reducing the overall delay [2], [3], [4], [5], [6], construct a minimum height (or minimum diameter) tree with constrained degrees. The degree constrains 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 metric 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. 1A, 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 minimal 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. 1B 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. 1C), 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

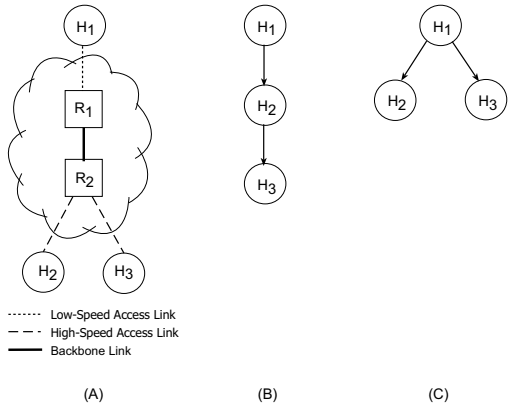

Fig. 1. Comparison between application-layer multicast and network-layer multicast in a simple heterogeneous overlay network

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.

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 [7] that measured the actual end-host heterogeneity of popular peer-2-peer (p2p) overly systems showed that the bandwidth and latency parameters can vary several orders of magnitude across different hosts in the system.

In this paper, we present an application-centric overlay network model which captures the realistic costs involved with application-layer multicast. The model uses a *single delay metric* to characterize multicast performance using the following 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 model serves as a theoretical framework which enables formal comparison of the performance of different multicast algorithms.

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 (we conjecture it is optimal) and thus can scale to large multicast groups, which is a known problem [2]. Note that the suggested solution works both for fully connected topologies, and for structured topologies, used in some p2p overlay networks [8]. Therefore, we address the issue of multicasting in partially connected networks and provide performance bounds for tree and grid graphs.

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], [9], [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 multi-player 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 Section IV we develop approximation and heuristic algorithms for solving this problem. Section V deals with performance analysis of the heuristic algorithm for several overlay topologies. An experimental evaluation of our solutions is presented at Section VI. Finally, Section VII concludes the paper.

II. Overlay Communication Network Model

In this section we define the overlay communication model and the corresponding delay measures which characterize the performance of application-layer multicast solutions.

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.

The overlay network is modelled by a directed complete graph $G = (V, E)$, where V is a set of vertices representing hosts, and E is the set of edges representing the 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, $p(v)$.

The direct communication between hosts is characterized as follows. Assume that at time t, host u initiates processing of a message targeted to 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 minimal time interval between consecutive transmissions.

Although current operating systems and their communication services have mechanisms which allow applications to perform simultaneous (or near simultaneous) message transmissions, the simultaneous effect is overridden by the inherent serialization involved with message transmission through a physical access link. This type of serialization is typically performed at the hardware level by the access equipment. Furthermore, the sequential distribution prohibits the usage of unrealistic application design schemes which relies on simultaneous 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.* [10, Figs. 2 and 3] show that about 30-50% of the triangles in the Internet do not obey the triangle inequality.

The *communication latency* metric is used to represent the end-to-end delay of direct and indirect communications between two hosts in the overlay structure. Given a pair of hosts v_1 and v_k which are connected by a path $p_{v_1,v_k} =< v_1, \ldots, v_k >$ of length k, the communication latency from v_1 to v_k, denoted by $l(p_{v_1,v_k})$, is the sum of communication delays of the overlay links in the path and the processing delays of the traversed hosts, assuming each traversed host distributes the message on its first processing round. Therefore $l(p_{v_1,v_k}) = \sum_{i=1}^{k-1} p(v_i) + c(v_i, v_{i+1})$, where $v_i,\ 1 \leq i \leq k$ denotes the ith host on the path p_{v_1,v_k}. One may view the latency metric as a measure of minimal distribution delay along an overlay path.

III. The 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 formulate the optimal multicast tree problem, also denoted as *minimal 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 non-negative real processing delay $p(v)$ for each vertex $v \in V$, and a non-negative real communication cost $c(u, v)$ for each edge $(u, v) \in E$, find a multicast scheme that minimizes the delay required to disseminate a message from the source host s to all the other hosts in M assuming that only the group members in M may participate in the distribution.

Our goal is to devise a multicast scheme which minimizes the distribution delay, i.e., minimizes the time till all the hosts have received the message. Therefore, we consider only "non-lazy" multicast schemes [11], in which a host which has already received a message does not delay message distribution by becoming idle.

Without loss of generality we assume that $M \equiv V$, such that the multicast problem reduces to the problem of finding an ordered directed tree T, rooted at s and spanning V. In this tree, the outgoing edges of a non-leaf node u are ordered according to the message distribution order of host u in the multicast scheme, where the ith outgoing edge corresponds to the ith transmission.

The *reception delay* of host $v \in V$, denoted by $t_T(v)$, is the time at which v receives a message from the source host, s. The reception delay of s is defined to be 0. The cost of a multicast tree T is defined as the overall delay of the multicast scheme. This cost equals $\max_{v \in V} t_T(v)$, i.e., the earliest time at which all the hosts have been notified. Given a multicast tree we can easily calculate the optimal ordering using a recursive computation, working bottom-up. Therefore, 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-complete using a simple reduction from the telephone broadcast (TB) problem. In the *Telephone model* (see [12]) communication is synchronous, i.e., each node can either sent or

receive a single message per communication round. The TB problem seeks an optimal broadcast scheme which distributes a message from a source node $r \in V$ to all the nodes in V in a minimal number of rounds. The TB problem is known to be NP-Hard [13, ND49] for arbitrary graphs.

Theorem 1: The decision version of the MDM problem, finding a multicast tree with a delay bound K, is NP-complete.

Proof: The proof follows by applying a reduction from TB which constructs an overlay configuration with unit processing costs and zero communication costs for all the edges in the input graph. The cost of the remaining edges is set to n. ■

IV. Multicast Algorithms

Broadcast and multicast are important communication primitives which have many applications in distributed and parallel systems. The problem of designing efficient broadcast and multicast algorithms which assume sequential message distribution, have been extensively studied in the context of several communication models. One model which was widely investigated is the 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 [14] for a comprehensive survey), while others have suggested approximation algorithms for optimal broadcasting in general graphs ([15], [16], [17]).

The *postal model*, introduced by Bar-Noy *et al.* [12], is a similar homogenous model which captures network communication costs by incorporating a latency parameter λ. It assumes a fully connected communication model in which each node can either transmit or receive a single message per time unit. Another related model which incorporates the processing and communication delay measures is presented by Cidon *et al.* [18] in the context of high-speed communication networks. Raz and Shavitt [19] proposed a similar model for active networks which supports IP-like routing. Optimal broadcast schemes for complete homogenous cost networks can be found at [12], [18].

The *heterogeneous postal model* [20] extends the postal model by assuming non uniform communication costs. In addition the model incorporates a switching time measure which represents the minimal gap between message transmissions. The model represents the communication network using an undirected graph $G = (V, E)$, a switching time function which associates a sending time s_v with each node $v \in V$, and a communication latency function which associates a length λ_{uv} with each pair of nodes $(u, v) \in E$. The communication delay λ_{uv} takes into account the sending time at u and the receiving time at v, and therefore the model assumes that $s_u < \lambda_{uv}, \forall (u, v) \in E$. A $\log k$ approximation algorithm is given in [20] for the problem of optimal multicast where k is the size of the multicast group.

Since the problem of finding the optimal multicast tree is NP-complete, we seek to devise approximations and heuristics. We begin with developing approximation algorithm based on a modified version of the postal approximation algorithm. This algorithm requires undirected overlay graph inputs, implying that its domain is limited to overlay networks with symmetric links. This restriction is in many cases unrealistic due to the widespread deployment of asymmetric access links, such as ADSL and cable-modem connections. The approximation algorithm also requires a high (polynomial) running time. Therefore, we devise an alternative cost-effective heuristic algorithm that supports directed overlay networks, and evaluate its performance. Finally, we analyze homogenous overlay networks and show that 'non-lazy' trees achieve logarithmic multicast delay.

We also discuss shared tree extensions of these algorithms. In the shared tree approach [21] a single tree is constructed for the purpose of multi-source multicast Our analysis show that the presented algorithms can be easily modified to support shared trees without major impact on the performance. Of course, using multiple single source multicast trees will always achieve lower delay, but at the expense of the management and resource usage overhead.

A. Approximation outline

We base the overlay approximation on the postal approximation scheme of Bar-Noy *et al.* [20] originally designed for the heterogeneous postal model. Although both models have common properties, the postal model differs from the overlay model in the following aspects. (1) In the postal model the communication latency of a link incorporates the sending time, while in the overlay model the sending time is incorporated in the processing delay of the sender host. (2) The postal model assumes that $s_u < \lambda_{uv}, \forall (u, v) \in E$.

Thus, we need to adapt the postal approximation algorithm before applying it to the overlay model. We do this in three phases. First, we define the *generalized heterogeneous postal (GHP)* model, which excludes the restriction on the values of the communication and switching measures. Second, we adapt the original postal approximation algorithm to support the GHP model. Finally, we construct a cost preserving GHP configuration and apply the GHP approximation to compute the multicast tree. This process results in an approximation algorithm, Approx-MDM, which increases the original approximation by an additive factor.

Definition 2: The GHP model is a heterogeneous postal model which excludes the restriction on the network costs, such that the edge length parameter in the GHP model is finite and positive, i.e., $\lambda_{uv} > 0, \forall (u,v) \in E$.

The GHP model provides a framework that includes nodes with switching time which is larger than the communication latency to the neighbors. The following measure captures the proportion between switching and communication times.

Definition 3: Given a GHP model with graph $G = (V, E)$, switching time function s, and a communication latency function λ, define $\gamma = \max_{(v,w)\in E}\{\frac{s_v}{\lambda_{vw}}\}$ as the switching to communication ratio of the graph G.

B. The GHP approximation algorithm

Before proceeding to the GHP approximation we provide an outline of the postal approximation algorithm. The interested reader is directed to [20] for the full details.

The problem of multicasting in the postal model is defined as follows. Given a configuration of an undirected graph with associated communication and switching cost functions $(G = (V, E), s, \lambda)$, a set of terminals $U \subseteq V$, and a source node $r \in U$, find the minimal time scheme that distributes a message from r to the terminal set U, where all the nodes in V may participate in the distribution.

The postal approximation algorithm. The basic idea of the algorithm is to find a multicast tree T which minimizes the quantity $\Delta_T + L_T$, where Δ_T denotes the maximum generalized degree (the generalized degree of a node is its actual degree multiplied by the corresponding switching time) of T, and L_T denotes the weighted diameter of T. The algorithm computes a multicast tree, which approximates the cost of the optimal tree T^*, iteratively using l rounds. Let U_i denote the terminal set in the ith round. The algorithm starts with the initial set $U_0 = U$ and terminates when $U_\ell = \{r\}$. In the ith round the algorithm uses the $core$ procedure to compute the following, for any $i \leq l$:

1) a core subset $U_i \subseteq U_{i-1}$ of size at most $\frac{3}{4} \cdot |U_{i-1}|$ where $r \in U_i$
2) a multicast scheme from U_i to U_{i-1}, such that the obtained multicast time is linear in the optimal multicast time from r to U_{i-1}.

The computation of $core(U')$ involves the following steps:

1) Solve a linear program, variant of a multicommodity flow. The resulting set of fractional paths is rounded [20, Theorem 4] producing a set of $|U'|$ integral paths, one for each terminal.
2) Transform the set of paths into a set of spider graphs (see Section V-A). Select an arbitrary terminal from each spider together with nodes which are not spanned by any spider to be included within the resulting core. This selection insures that the chosen terminal is able to distribute a message to all its spider nodes in the required linear time.

In [20] it is shown that the resulting tree has a $O(\log |U|)$ multiplicative approximation factor. This approximation algorithm cannot be applied to overlay networks due to the inherent cost restriction which determines the coefficients of the rounding matrix.

We now describe the *GHP rounding* mechanism that extends the postal approximation domain to support networks with $\gamma \geq 1$, i.e., GHP models. We preserve the notations of [20], $P_1, P_2, \ldots$ denotes the length bounded fractional flow paths, and $V(P_i)$ and $E(P_i)$ denotes the set of nodes and edges in a path P_i, respectively; $f(P_i)$ denotes the amount of flow pushed on path P_i, and $\mathcal{P}^j$ denotes the set of all paths that carry flow of the jth commodity. To simplify the presentation of the results we define $\gamma' = max\{1, \gamma\}$. The following matrix is used for the rounding of the fractional solution:

$$\begin{array}{lrcl} \text{for each } v & s_v \cdot \sum_{i:\, v\in V(P_i)} f(P_i) & \leq & 6\Delta_T \\ \text{for all } j & -4L_T \cdot \gamma' \cdot \sum_{i:\, P_i\in \mathcal{P}^j} f(P_i) & = & -4L_T \cdot \gamma' \end{array}$$

The sum of positive entries in the ith column is:

$$\sum_{v\in V(P_i)} s_v \leq \sum_{(v,w)\in E(p_i)} \lambda_{vw} \cdot \gamma' + s_{t_j} \leq 4L_T \cdot \gamma'.$$

where the second part of the equation follows from the definition of γ. The sum of the negative entries at each column is at most $-4L_T \cdot \gamma'$. By invoking the postal rounding [20, Theorem 4] we get a set of integral paths such that their congestion, i.e., the generalized degree of the graph spanned by a set of paths, is at most $6\Delta_{T^*} + 4L_{T^*} \cdot \gamma'$ and the length of each path is at most $4L_{T^*} \cdot \gamma'$.

The GHP approximation algorithm. The GHP approximation algorithm is a postal approximation algorithm which employees a GHP rounding mechanism instead of the original rounding.

The correctness of the modified algorithm follows from the fact the algorithm structure and its underling theorems and lemmas are not related to the specific switching and communication cost values, except of the constrained selection of the rounding coefficient which we handle appropriately. Therefore it remains to show the approximation ratio.

The transformation performed on the rounded paths, step (2) in the core procedure, yields a set of spiders

which preserve the topological properties of the original algorithm, such that the diameter of each spider is at most $4 \cdot \gamma' \cdot (\Delta_{T^*} + L_{T^*})$ and the generalized degree (of the center) of a spider is at most $6 \cdot \gamma' \cdot (\Delta_{T^*} + L_{T^*})$. Since the algorithm invokes $O(\log |U|)$ iterations of the $core$ procedure and the cost of the optimal tree T^* is at least $0.5 \cdot (\Delta_{T^*} + L_{T^*})$ [20, Lemma 1], we have that the multicast time from the root r to a set of terminals U is at most $O(\log |U| \cdot max\{1, \gamma\})$ times the optimal multicast time.

C. The MDM approximation algorithm

The following polynomial algorithm provides an approximation solution for the MDM problem. The algorithm accepts as an input an overlay network configuration (G, c, p) which consists of an undirected graph $G = (V, E)$ with associated processing and communication cost functions, p and c, respectfully, and a source host $\tilde{s} \in V$.

Algorithm Approx-MDM($\tilde{s}, G, p, c$)

1. Construct a GHP configuration instance $I_{GHP} = (G, s, \lambda)$, from the graph G, switching time function $s_v = p(v), \forall v \in V$ and communication latency function $\lambda_{u,v} = c(u, v) + (p(u) + p(v))/2, \forall (u, v) \in E$.
2. Invoke the GHP approximation to compute a multicast tree using I_{GHP}, source host $\tilde{s}$, and multicast group $U = V$.
3. Return the computed multicast tree

Let OPT be the minimal multicast delay from $\tilde{s}$ to V, and let n be the size of V. Let $p_{max} = \max_{v \in V} p(v)$ and $p_{min} = \min_{v \in V} p(v)$ be the maximal and minimal processing costs in the overlay network.

Theorem 2: The multicast delay of the Approx-MDM algorithm is at most $(OPT + (p_{max} - p_{min})) \cdot O(\log n)$

Proof: Given a multicast tree T which spans V and a host $v \in V$, let $t_{_T}^{GHP}(v)$ be the reception delay of v assuming GHP model timings. By substituting the computed costs of I_{GHP} with the corresponding overlay input costs we get the following relationship between the reception delay costs.

$$t_{_T}^{GHP}(v) = \frac{p(s) - p(v)}{2} + t_{_T}(v) \qquad (1)$$

Consider the following quantities computed assuming GHP model timings. Let OPT_{GHP} be the multicast delay of an optimal tree T^*_{GHP} for the I_{GHP} configuration. Let $u \in V$ be the node with the maximal reception delay in T^*_{GHP}. Therefore we have that

$$OPT_{GHP} \leq OPT + \frac{p(u) - p(s)}{2} \leq OPT + \frac{p_{max} - p_{min}}{2} \qquad (2)$$

where the first inequality follows from Eq. (1).

The constructed I_{GHP} instance satisfies $\gamma < 2$, since $\frac{p(v)}{0.5 \cdot (p(v) + p(w)) + c(v,w)} < 2, \ \forall (v, w) \in E$, and therefore the multicast delay of the resulting tree is at most $OPT_{GHP} \cdot O(\log n)$. Substituting OPT_{GHP} according to equation (2) gives the requested upper bound. ■

When the processing costs are all equal, it improves our approximation for the MDM problem to $O(\log n)$. We do not restrict the communication costs to be homogeneous. The following theorem handles this case.

Theorem 3: Consider an overlay model with homogenous processing costs, i.e., $p(v) = p, \forall v \in V$. The multicast delay of Approx-MDM algorithm for this case is at most $OPT \cdot O(\log n)$.

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

Given a network with symmetric communication costs, a multicast tree $T = (V, E)$ rooted at s can be easily adapted to support multicasting from multiple sources. To perform the multicast from a host $v \in V, v \neq s$, we reverse the direction of the edges on the path from s to v. This modification results in a multicast scheme that requires at most $p(v) - p(s) + 2 \cdot C$ time, where C denotes the cost of T. Therefore, the undirected version of the Approx-MDM multicast tree can be used as a shared tree such that the multicast delay of any host $v \in V, v \neq s$ is at most $2 \cdot (OPT_s + (p_{max} - p_{min})) \cdot O(\log n)$, where OPT_s denotes the optimal multicast delay from s.

D. Heuristic algorithm

We introduce a heuristic tree construction algorithm for the directed version of the MDM problem with n hosts. The proposed algorithm computes the multicast tree incrementally using a greedy approach; for each host not yet included in the tree, a mate host which minimizes its potential reception delay is computed, and the host with maximal delay is chosen to extended the tree along with the hosts on the path to its mate. Fig. 2 shows the steps of the algorithm.

The algorithm maintains a ready time attribute $t[v]$ for each host $v \in V$ which records the minimal time at which the host is free to initiate processing of a new message. The ready time is set to infinity to indicate non notified 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 $u \in V - V[T]$ its mate host $m[u] \in V[T]$ by selecting a path which minimizes the potential ready time 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 their new potential processing times, 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.

Algorithm Heuristic-MDM(s, G, p, c)

1. $t[s] = 0$, set s as the root of a tree T
2. for each $v \in V - \{s\}$
3. do $t[v] \leftarrow \infty$
4. for each $(u, v) \in E$
5. do $w_{u,v} = c(u, v) + p(u)$
6. for each $(u, v) \notin E$
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 $V - V[T] \neq \emptyset$
10. for each host $u \in V - V[T]$ do
11. $m[u] \leftarrow \arg\min_{v:v \in V[T]}\{t[v] + d_{v,u}\}$
12. $v \leftarrow \arg\max_{u:u \in V-V[T]}\{t[m[u]] + d_{m[u],u}\}$
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

To be able to calculate the connection cost between a non notified host and a notified host, a preprocessing phase of computing all pairs shortest path using the Floyd-Warshall algorithm [22] is implemented. A shortest path from host u to host v is defined as any path with the minimal communication latency from u to v. The edge weights of the shortest path computation correspond to the communication latency of the overlay links, such that the input to the Floyd-Warshall computation is an $n \times n$ weight matrix $W = (w_{v_i,v_j})$ defined as:

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

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 [22]). 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.

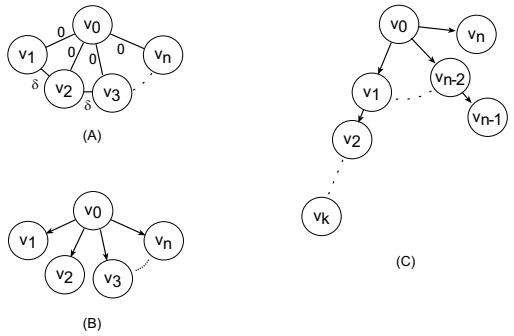

Fig. 3. Example that provides $\sqrt{n}$ approximation ratio for the heuristic tree. (A) The input graph (B) The heuristic tree. (C) An optimal tree.

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 mate host, and $O(n)$ time to update the host paths and extend the tree. The total time per iteration is therefore $O(n^2)$, and the total running time of the heuristic algorithm is $\Theta(n^3)$. We conjecture this time complexity cannot be improved since any algorithm should at least calculate the all pair shortest path.

We show using an example (see Fig. 3A) 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

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

where $\delta \to 0$. For the simplicity of presentation Fig. 3A omits the edges with cost n. Assume that the source host is v_0. Therefore, the heuristic 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. 3B). On the other hand, consider an optimal scheme in which v_0 distributes the message to an ordered set of k paths, $p_1, \ldots, p_k$, such that the number of edges in path p_i, denoted by $|p_i|$, forms the following 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. 3C depicts an optimal tree when $n = \frac{k \cdot (k+1)}{2}$. Since the cost of the optimal tree is at most $(1+\delta) \cdot k$ and $k = O(\sqrt{n})$ we get $\Omega(\sqrt{n})$ approximation ratio for the multicast delay. We conjecture that this example represents the worst case, namely that our heuristic algorithm is an $\sqrt{n}$-approximation.

E. The homogenous case

Consider a fully connected overlay network with homogeneous processing and delay costs, i.e., $p(v) = p, \forall v \in V$, $c(u,v) = c, \forall (u,v) \in E$. We denote this model as the *homogenous overlay network*. Observe that the postal model can be reduced to the homogenous overlay model by selecting $p = 1, c = \lambda - 1$.

In the homogenous overlay network, a non-lazy scheme directs each notified host to distribute the multicast message to a new host every processing interval p. Due to symmetry, any non lazy multicast algorithm which avoids sending duplicate messages to the same destination host will result in an optimal solution. In particular, an optimal solution can be obtained by using the non-lazy centralized Heuristic-MDM algorithm described in section IV-D. It remains to show the convergence rate of message distribution. Using the analysis of [18, Eq. (3)] we derive the following

Theorem 4: In the homogenous model, the maximal number of hosts that can be reached during the time period $(0,t]$ is given by

$$N(t) = \begin{cases} 1 & \text{if } 0 \le t < p+c, \\ N(t-p) + N(t-p-c) & \text{if } t \ge p+c. \end{cases}$$

It is easy to derive upper and lower bounds for $N(t)$, and get that: $2^{\lfloor \frac{t}{p+c} \rfloor} \le N(t) \le 2^{\lfloor \frac{t}{p} \rfloor}$, for any real numbers $t, p, c \ge 0$. Therefore, the optimal algorithm has logarithmic multicast delay in homogeneous overlay networks.

V. Topologies

In this section we analyze the performance of the heuristic tree for the special case of partially connected overlay networks. Partial connectivity, which assumes arbitrary or structured graphs, is an important model which arises in several contexts.

Partial connectivity is implement 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 therefore enforce connectivity constrains to achieve efficient resource utilization. Structured p2p systems [8] 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 rise 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.

Partial topologies are also relevant to the case of active networks [19], which have similar properties to those of overlay networks. It is possible to view the overlay network as an application level implementation of the active network model, where the active network uses programmable routers to add new functionality and services to the network. For example, Raz and Shavitt [19] 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.

Therefore, in order to support networks with partial connectivity an extended overlay model is assumed; in this model 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.

For general graph topologies our analysis focuses on the performance of the broadcasting communication primitive in which a source host disseminates a message to the rest of the hosts in the graph. In the next section, we analyze the broadcast performance of the heuristic tree for several common undirected graph topologies.

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 non-leaf hosts.

Lemma 5: Any non-lazy broadcast scheme provides a factor d approximation for the minimal broadcast delay for a tree graph $T = (V,E)$ with a maximal degree of d.

Proof: Denote by s the source host. In any non lazy scheme, the time that the last notified leaf, denoted by v^*, receives a message is at most $l(p_{s,v^*})$, i.e., the latency of the unique path from s to v^* in T, plus the additional processing delay imposed by each host on the path p_{s,v^*}. Since the degree of the tree is bounded by d, this delay is at most $(d-1) \cdot \sum_{i=0}^{k-1} p(v_i)$ where v_i, denotes the ith host on this path such that $v_0 = s$,$v_k = v^*$. It is easy to see that this additional delay is at most $(d-1) \cdot l(p_{s,v^*})$, and the lemma follows. ■

This result indicates that a distribution along a degree-constrained multicast tree at an arbitrary order, such as delivery schemes used by overlay multicast systems which ignore sequential distribution of messages, produces a delay which is up to a multiplicative constant factor higher than the optimal result.

The heuristic algorithm achieves optimal solution for a special class of tree graphs termed spiders, in which at most one node has degree larger than two. The proof is omitted due to space limitations.

B. Grids

This section investigates broadcasting in the context of homogenous 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 homogenous cost model where $p(v) = 1, \forall v \in V, \quad c(u, v) = 0, \forall (u, v) \in E$. 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-dimensional grid graphs have been previously investigated by Farley and Hedetniemi [24]. They have shown that:

Given a grid graph $G_{m,n}$ with a node v at position (i, j). Then

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

where $b(v)$ denotes the optimal broadcast time 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, i.e., $||u - v|| = |i_u - i_v| + |j_u - j_v|$.

The following Theorem shows that the heuristic tree provides an optimal solution for broadcasting in grid graphs. The proof for this case assumes that the heuristic algorithm uses a tie-breaking strategy to handle multiple path choices when connecting a new non-notified host to the constructed tree. The strategy selects a path which satisfies the following conditions. (a) the path has the minimal latency among all the paths leading to the constructed tree (b) the path uses the minimal number of direction changes in the grid topology. This strategy greatly simplifies the analysis, since it implies that the algorithm uses one-turn paths, i.e., paths with only one direction change, or zero-turn paths, i.e., horizontal or vertical paths.

Lemma 6: The Heuristic-MDM algorithm provides an optimal solution for a homogenous grid graph $G_{m,n} = (V, E)$.

Proof: Let T denote the computed heuristic tree, rooted at s. Since the heuristic algorithm uses max-min criteria for the selection newly of notified hosts, it follows that T is an SPT. The proof is omitted due to space limitations. This implies the following degree delegation in T. If s is a corner host than its degree is 2 and rest of the nodes have maximal out-degree of 2. If s is a side or interior host, than the out-degree of the interior nodes which share a common coordinate with s is 3 and the maximal out-degree of the rest of the nodes is 2. The degree of s is 3 when s is a side host, and 4 when it is an interior host. Define $S_3 = \{v : deg(v) = 3, v \neq s\}$ as the set of hosts with out-degree 3, 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 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 we get that the optimal broadcast time from the root of a T_2 tree with height d requires d time units. 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 time from a corner host requires D units of time, 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 time 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 time of T.

Let T_3 denote a subtree in T_c rooted at a child of s. Consider the case of trinary T_3 trees. 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 $h(T_{v_3}) = h(T_{v_2}) > h(T_{v_1})$. The operation of the heuristic algorithm insures that a subtree T_3 with height d wont contain a host $v \in S$ which has two subtrees in which the maximal distance from the leaves to the root is $d-1$. Denote this assumption as the heuristic path selection restriction (the proof for this claim is omitted due to space limitations). By using a bottom-up recursive computation of the broadcast time we have that the broadcast time from the root of a T_3 with height d is d when there is a single leaf at depth d, and $d+1$ when there exists two leaves at depth d.

Now, we need to check all the combinations of hosts at depth D and $D-1$ in the T_3 trees linked to the source s. First, consider the case when s is a side host linked with three T_3 trees. If s is a middle side host, there are two nodes 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 time from a corner host is at most $D+1$. If these two hosts reside in different subtrees, then the maximal height of the third subtree is $D-2$ and the broadcast time is again at most $D+1$. 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 $D-2$. The broadcast time is at most D, and the lemma follows for this case. The case of broadcasting from an interior source host requires similar analysis. By checking all the possible combinations we get that the broadcast time from an interior node obeys the optimal time, which completes the proof of the lemma. ■

Corollary 7: The broadcast delay of a shortest path tree for homogenous cost grid graph $G_{m,n} = (V, E)$ is at most $OPT + 2$

Proof: If we remove the heuristic path selection restriction, the broadcast time from the root of a T_3 is increased by at most one unit of time, and therefore the total broadcast time can be increased by at most two units of time. ■

VI. A 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 [25] 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. 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 average degree of approximately 4.38.

In our simulations we compare the performance of the Heuristic-MDM algorithm with the following schemes.

- Approx-MDM multicast algorithm. In our simulation environment which includes 1.5Ghz PCs with 512M RAM, the Approx-MDM algorithm was able to effectively solve problems with up to 25 hosts. This limitation is due to the high running time of the algorithm, which is at least $\Theta(n^7)$ [17].
- 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 [22], where the edge weights are defined using the formulation of section IV-D.
- Latency bound. Since the MDM problem is NP-Hard (see Section III) the optimal solution could not be computed. Instead, the maximal value of the minimal communication latencies between the source and the group members is computed. This maximal latency is a lower bound on the performance of any multicast scheme.

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 handles larger group sizes up to 4000 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 analyze the scaling properties of the heuristic and SPT trees.

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 Heuristic-MDM, Approx-MDM, 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. The left range in each pair is the communication cost range, and the right range is the processing 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 3 times smaller than the approximation cost. This performance gap stems from the fact that the approximation scheme constructs trees with logarithmic height. The usage of logarithmic height trees increases the probability of selecting high cost communication delays, and therefore reduces the average efficiency of approximation trees.

As expected SPT provides the worst case performance, providing a cost function which is almost linearly proportional to the tested group size. Observe that the multicast delay is plotted on a logarithmic scale, such

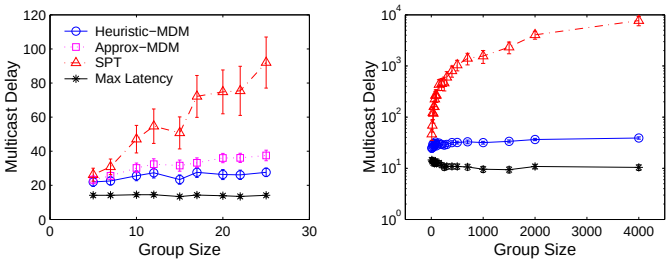

Fig. 4. The multicast delay for a clique topology with random network costs from $[1, 10]$

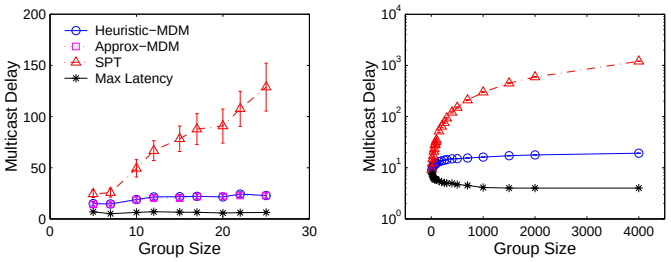

Fig. 5. The multicast delay for a clique topology with random processing costs from $[1, 10]$ and unit communication costs

that the linear performance degradation is shown using a logarithmic curve. The SPT performance is consistent with the tree construction mechanism which makes no attempt to minimize the degree of the resulting tree. The quality of the SPT is determined according to the dominance of the communication costs, such that 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 Section IV-E) of the heuristic.

We used a 4000 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 minimal 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.

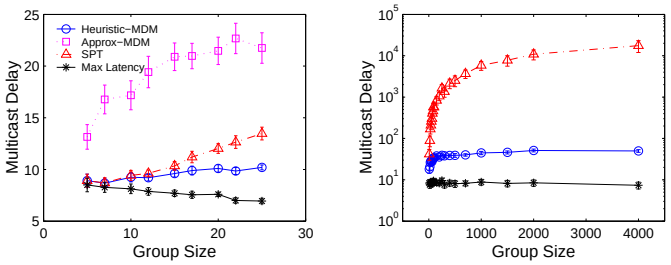

Fig. 6. The multicast delay for a clique topology with random communication costs from $[1, 10]$ and unit processing costs

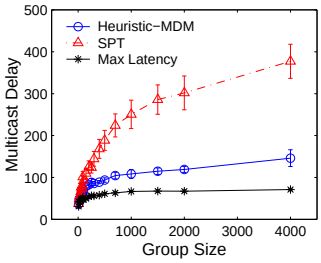

Fig. 7. The multicast delay for a power-law topology with random network costs from $[1, 10]$

Next, we consider the case of partially connected overlay networks derived using the power-law topology generator. In this case, we weren't able to apply the approximation scheme due to the implicit full-connectivity assumption inherent in the heterogeneous postal model. This assumption makes the postal approximation, and consequently the Approx-MDM algorithm, unsuitable for partially connected graphs. This limitation cannot be bypassed since invoking the approximation scheme on arbitrary graphs may result in a partially connected core subset, making the following core computations problematic. Therefore we compare the performance of the heuristic tree with SPT, 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 3 times higher than the non-tight lower bound.

VII. Concluding Remarks

In this work we looked at building efficient application layer multicast trees. We presented a new model that captures the trade offs between the desire to select shortest path trees and the need to constraint the load on the hosts. We defined the minimum delay multicast tree problem, and presented both an approximation and a heuristic for its solution. Our simulation study shows that

the heuristic algorithm provides a cost effective solution for the MDM problem, which is very close to optimal.

REFERENCES

[1] A. El-Sayed, V. Roca, and L. Mathy, "A survey of proposals for an alternative group communication service", *IEEE Network magazine, Special issue on "Multicasting: An Enabling Technology"*, Jan. / Feb. 2003.

[2] Yang-Hua Chu, Sanjay G. Rao, and Hui Zhang, "A case for end system multicast", in *ACM SIGMETRICS 2000*, Santa Clara, CA, USA, June 2000, ACM, pp. 1–12.

[3] Dimitris Pendarakis, Sherlia Shi, Dinesh Verma, and Marcel Waldvogel, "ALMI: An application level multicast inproceedings infrastructure", in *Proceedings of the 3rd USNIX Symposium on Internet Technologies and Systems (USITS '01)*, San Francisco, CA, USA, Mar. 2001, pp. 49–60.

[4] S. Shi and J. Turner, "Routing in overlay multicast networks", in *IEEE Infocom 2002*, June 2002.

[5] Suman Banerjee, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee, and Samir Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications", in *IEEE Infocom 2003*, San Francisco, CA, USA, Apr. 2003.

[6] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy, "Scalable application layer multicast", in *ACM Sigcomm 2002*, Pittsburgh, PA, USA, Aug. 2002.

[7] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, "A measurement study of peer-to-peer file sharing systems", in *Multimedia Computing and Networking 2002 (MMCN'02)*, San Jose, CA, USA, Jan. 2002.

[8] 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 *IEEE Infocom 2003*, San Francisco, CA, USA, Apr. 2003.

[9] Sherlia Y. Shi, Jon Turner, and Marcel Waldvogel, "Dimensioning server access bandwidth and multicast routing in overlay networks", in *NOSSDAV 2001*, June 2001, pp. 83–92.

[10] 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 Transactions on Networking*, Oct. 2001.

[11] Amotz Bar-Noy, Sudipto Guha, Joseph (Seffi) Naor, and Baruch Schieber, "Multicasting in heterogeneous networks", in *STOC'98*, 1998, pp. 448–453.

[12] Amotz Bar-Noy and Shlomo Kipnis, "Designing broadcasting algorithms in the postal model for message passing systems", in *Proc. of SPAA*, 1992, pp. 13–22.

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

[14] 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.

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

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

[17] M. Elkin and G. Kortsarz, "Combinatorial logarithmic approximation algorithm for the directed telephone broadcast problem", in *STOC*, 2002, pp. 438–447.

[18] Israel Cidon, Inder Gopal, and Shay Kutten, "New models and algorithms for future networks", *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 769 – 780, May 1995.

[19] Danny Raz and Yuval Shavitt, "New models and algorithms for programmable networks", *Computer Networks*, vol. 38, no. 3, pp. 311–326, 2002.

[20] Amotz Bar-Noy, Sudipto Guha, Joseph (Seffi) Naor, and Baruch Schieber, "Message multicasting in heterogeneous networks", *SIAM Journal on Computing*, vol. 30, no. 2, pp. 347–358, 2001.

[21] L. Wei and D. Estrin, "The trade-offs of multicast trees and algorithms", in *ICCCN'94*, 1994.

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

[23] P.Francis, "Yoid: extending the multicast inetrnet architecture", *White paper http://www.aciri.org/yoid/*, 1999.

[24] A.M. Farley and S.T. Hedetniemi, "Broadcasting in grid graphs", in *the 9th S-E conf. combinatorics, graph theory, and computing*, 1978, pp. 275–288.

[25] R. Albert and A.L. Barabasi, "Topology of evolving networks: local events and universality", *Physical Review Letters*, vol. 85, no. 24, pp. 5234–5237, 11 Dec. 2000.