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

Eli Brosh
Department of EE - Systems
Tel-Aviv University
Ramat-Aviv, Israel 69978
Email: elibrosh@eng.tau.ac.il

Yuval Shavitt
Department of EE - Systems
Tel-Aviv University
Ramat-Aviv, Israel 69978
Email: shavitt@eng.tau.ac.il

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 constrain 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 measures 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 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. 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 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 distribu-

0-7803-8356-7/04/$20.00 (C) 2004 IEEE

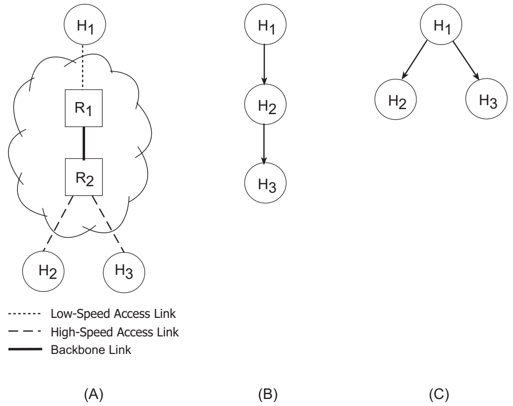

Fig. 1. Comparison between application-layer multicast and network-layer multicast in a simple heterogeneous overlay network

tions 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) 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.* [8] 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]. Note that the suggested solution works both for fully connected topologies, and for structured topologies, used in some p2p overlay networks [9]. 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], [10], [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

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.* [8]. 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, $p(v)$. The original model of Cidon *et al.* [8] 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 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 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.* [11, Figs. 2 and 3] show that about 30-50% of the triangles in the Internet do not obey the triangle inequality.

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 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 non-participating 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 induced by the host set $M \subseteq V$.

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 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 last message. Therefore, we restrict this study to non-lazy multicast schemes, in which a host that has already received a message does not delay message distribution by becoming idle (this term was introduced in [12]). Such schemes correspond to an ordered directed tree T, rooted at s and spanning M. 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 corresponding multicast scheme, such that 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 the message. The reception delay of s is by definition 0. The cost of a multicast tree T is defined as the earliest time at which all the hosts have been notified, i.e., this cost equals to $\max_{v \in M} t_T(v)$.

Given a multicast tree we can easily calculate the optimal ordering using a recursive computation, working bottom-up (the full description of the recursion can be found in [13]). 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-hard using a simple reduction from the telephone broadcast (TB) problem. In the *Telephone model* (see [14]) communication is synchronous, i.e., 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 [15, ND49] for arbitrary graphs.

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, $G_{TB} = (V, E_{TB})$ 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 $c(u, v) = 0 \ \forall (u, v) \in E_{TB}$ and $c(u, v) = n \ \forall (u, v) \notin E_{TB}$. 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 (for $K < n$). The claim follows by noting that for $K \geq n$ the TB problem is trivially solved since it is equivalent to deciding if G is connected. ■

IV. Multicast Algorithms

In the next section we present our solutions for obtaining good multicast trees. We begin with a short review of previous results. 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 [16] for a comprehensive survey), while others have suggested approximation algorithms for optimal broadcasting in general graphs ([17], [18], [12]).

Cidon *et al.* [8] 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 model assumes sequential processing of messages, such that the time needed for a node to handle the transmission of i messages is iP. In addition, they proposed an optimal tree-based broadcast algorithm (see [8]) for complete graphs, and showed that such trees achieve a broadcast delay which is logarithmic in the size of V. Since the overlay model is a generalization of this model, any non-lazy multicast scheme (such as our heuristic algorithm) would provide a logarithmic delay for the case of a homogenous cost overlay network. Raz and Shavitt [19] 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 *heterogenous postal model* [12] is a related model which incorporates (non-uniform) communications and switching delays, i.e., it captures networks which may have different link delays and different switching times between messages. Optimal algorithm for broadcasting in a complete and homogenous cost postal network can be found at [14]. An approximation algorithm for optimal multicasting is given in [12]. This algorithm has a $\log k$ approximation ratio where k is the size of the multicast group. We further discuss this model and the corresponding approximation algorithm in Section IV-A.

We proceed now to describe our solutions for constructing application layer multicast trees. Since MDM is NP-complete, we seek to devise approximations and heuristics. We begin with developing a logarithmic approximation algorithm based on a modified version of the postal approximation algorithm. The resulting algorithm solves an undirected variant of the multicast problem, thus 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. Furthermore, the approximation algorithm requires high (polynomial) running time due to its need to solve linear programs. Therefore, we devise an alternative cost-effective heuristic algorithm that supports directed overlay networks, and evaluate its performance.

We also discuss shared tree extensions of these algorithms. In the shared tree approach [20] 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 always achieve lower delay, but at the expense of the management and resource usage overhead.

A. Approximation outline

We base our approximation on the algorithm of Bar-Noy *et al.* [12] which handles multicasting in heterogeneous postal models. The postal model represents the communication network using an undirected complete graph $G = (V, E)$. Each node v is associated with a switching time parameter s_v. A sender node v is considered busy (i.e., 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 parameter λ_{uv} which denotes the delay of delivering a message from u to v.

Although both models (postal and overlay) incorporate similar measures for characterizing heterogenous networks, the distribution timings in the postal model differs from the overlay model in the following aspects. (1) In the postal model the communication latency λ_{uv} incorporates the sending time of u, while in the overlay model this sending time is incorporated in the processing delay of v. Thus, the cost (i.e., delay) of delivering the ith message from u to v is the sum of the cost of (u, v) and $i - 1$ times the cost of u in the case of the postal model, and the sum of the cost of (u, v) and i times the cost of u in the case of the overlay model. (2) The postal model assumes that $s_u < \lambda_{uv}, \forall (u, v) \in E$.

Thus, we need to modify the postal approximation algorithm before applying it to the overlay model. We derive our approximation algorithm, called Approx-MDM, using the following steps: (1) we define the *generalized heterogeneous postal (GHP)* model, which eliminates the restriction on the values of the communication and switching (sending) measures. (2) we extend the postal approximation and derive the generalized postal approximation algorithm which handles multicasting in GHP models (3) we construct a cost preserving transformation from the overlay model to the GHP model and apply the generalized postal approximation to compute the multicast tree.

This process increases the original approximation by an additive factor of $O(\log |M|)(p_{max} - p_{min})$ where p_{max} and p_{min} are the maximum and minimum processing delays of the hosts in M, respectively.

Next, we introduce the required definitions.

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., $\lambda_{uv} > 0, \forall (u, v) \in E$.

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

Before proceeding, we provide an outline of the postal approximation algorithm. The interested reader is directed to [12] for the full details.

B. The postal approximation algorithm

In the context of the postal model the multicast problem 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 a minimum time (i.e., minimum delay) scheme that distributes a message from r to the terminal set U where all the nodes in V may participate in the distribution. Observe that we preserve the notations of [12] which denote the multicast group by U.

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 maximum distance (in T) from r to any of the nodes in U. 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 at most linear in the optimal multicast time from r to U_{i-1}.

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

1) Solve a linear program, variant of a multicommodity flow. The resulting set of fractional paths is rounded [12, Theorem 4] producing a set of $|U_i|$ integral paths, one for each terminal.
2) Transform the set of paths into a set of spider graphs - graphs in which at most one node has degree more than two. 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 spider terminal is able to distribute a message to all its nodes in the required linear time.

In [12] it is shown that the resulting tree has a $O(\log |U|)$ multiplicative approximation factor.

C. The generalized postal approximation algorithm

We now describe the *GHP rounding* matrix which enables the support of networks with $\gamma \geq 1$, i.e., GHP models. We preserve the notations of [12]: $P_1, P_2, \ldots$ 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 $\gamma' = max\{1, \gamma\}$. In the generalized postal approximation, the following rounding matrix (termed GHP rounding matrix) is used for rounding the fractional solution of step (1) in the core computation.

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

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 rounding theorem [12, 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 generalized postal approximation algorithm. The generalized postal approximation algorithm is a postal approximation algorithm which employs the GHP rounding matrix instead of the original one.

The correctness of the modified algorithm follows from the fact that the algorithm structure and its underlying 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.

Due to the GHP rounding, step (2) of the core computation gives a set of spiders with the following properties. The diameter of each spider is at most $4 \cdot \gamma' \cdot (\Delta_{T^*} + L_{T^*})$ and the generalized degree of each 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 $0.5 \cdot (\Delta_{T^*} + L_{T^*})$ (as shown in [12, 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.

D. The MDM approximation algorithm

The following is a polynomial algorithm that approximates the minimum multicast delay. The algorithm accepts as an input an overlay network configuration, (G, c, p), which consists of an undirected graph $G = (M, E_M)$ with associated processing and communication cost functions, p and c, respectively, and a source host $\tilde{s} \in M$. Given an overlay network graph (V, E), the input graph, G, is the subgraph induced by the multicast group set $M \subseteq V$ (see Section III).

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

1. Construct a GHP configuration instance $I_{GHP} = (G, s, \lambda)$, from the graph G, switching time function $s_v = p(v), \forall v \in M$ and communication latency function $\lambda_{u,v} = c(u,v) + (p(u) + p(v))/2, \forall (u,v) \in E_M$.

2. Invoke the generalized postal approximation to compute a multicast tree using I_{GHP}, source host $\tilde{s}$, and multicast group $U = M$.

3. Return the computed multicast tree

Let OPT be the minimum multicast delay from $\tilde{s}$ to M, and let n be the size of M. Let p_{max} and p_{min} be the maximum and minimum processing delays of the hosts in M, respectively.

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 M and a host $v \in M$, let $t^{GHP}_{T}(v)$ 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.

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

In order to derive this equation, we used the claim that there is a delay gap of a single processing round (per each traversed host) between the message delivery delays in the postal and overlay models (see Section IV-A).

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 $u \in M$ be a node with the maximum reception delay in T^*_{GHP}. Therefore,

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

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

The constructed I_{GHP} instance satisfies $\gamma < 2$, since $\frac{p(v)}{0.5\cdot(p(v)+p(w))+c(v,w)} < 2$, $\forall(v,w) \in E_M$, 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 delays 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., $p(v) = p, \forall v \in M$. 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 rooted at s can be easily adapted to support multicasting from multiple sources. To enable the multicast from a host $v \in M, v \neq s$, we reverse the direction of the edges on the path from s to v. This modification results in a multicast scheme with a delay which at most $p(v) - p(s) + 2 \cdot C$, where C denotes the cost of T. Therefore, the undirected version of T can be used as a shared tree, such that the multicast delay of any host $v \neq s$ 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.

E. 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.

```
Algorithm LRF($G, p, c, s$)

$t[s] = 0$, set $s$ as the root of a tree $T$
for each $v \in M - \{s\}$
    do $t[v] \leftarrow \infty$
for each $(u, v) \in E_M$
    do $w_{u,v} = c(u, v) + p(u)$
for each $(u, v) \notin E_M$
    do if $v = u$ then $w_{u,v} = 0$ else $w_{u,v} = \infty$
$D, \Pi \leftarrow$ All-Pairs-Shortest-Path($G, W$)
while $M - V[T] \neq \emptyset$
    for each host $u \in M - V[T]$ do
        $m[u] \leftarrow \arg\min_{v:v \in V[T]}\{t[v] + d_{v,u}\}$
    $v \leftarrow \arg\max_{u:u \in M - V[T]}\{t[m[u]] + d_{m[u],u}\}$
    $w \leftarrow v$
    while $w \neq m[v]$ do
        $t[w] \leftarrow t[m[v]] + p(w) + d_{m[v],w}$
        add $w$ to $T$ as a child of $\pi_{m[v],w}$
        $w \leftarrow \pi_{m[v],w}$
    $t[m[v]] \leftarrow t[m[v]] + p(m[v])$, $t[v] \leftarrow t[v] - p(v)$
return $T$
```

Fig. 2. Greedy tree construction for the MDM problem

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 minimal time at which the host is free to initiate processing of a new message. The ready time is set to infinity to indicate a 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 $u \in M - V[T]$ 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 non notified host and a notified host, a preprocessing phase of computing all pairs shortest path using the Floyd-Warshall algorithm [21] is implemented. Given a pair of hosts v_1 and v_k connected by a path $< v_1, \ldots, v_k >$ 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 minimal 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 [21]). 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 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 $\delta \rightarrow 0$. For the simplicity of presentation Fig. 3A omits the edges with cost n. Assume that the source host is v_0 and 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. 3B). 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 $k + m\delta$ time units, where m is a 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 more than two), and the lengths of these paths form the following non increasing 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 such a tree when $n = \frac{k \cdot (k+1)}{2}$. 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 $(1+\delta) \cdot k$. 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.

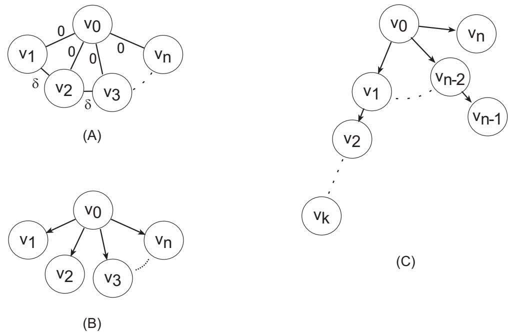

Fig. 3. Example that provides $\sqrt{n}$ approximation ratio for the heuristic tree. (A) The input graph (B) The heuristic tree. (C) An improved tree.

V. TOPOLOGIES

In this section we analyze the performance of broadcasting 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 [9] 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 broadcast algorithms. In the next section, we provide performance bounds 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 4: Any (non-lazy) broadcast scheme provides a factor d approximation for the minimal 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 IV-E, i.e., the cost of a path represents the minimal distribution delay along it. In any (non-lazy) 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 $< v_1, \ldots, v_k >$ 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 $(d-1) \cdot OPT$, and the lemma follows. ■

This result indicates that message distribution along 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 [22]), produces a delay which is up to a multiplicative constant factor higher than the optimal result.

The LRF heuristic achieves an optimal solution for a special class of tree graphs termed spiders, in which at most one node has degree larger than two. The proof can be found in [13].

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-dimensional grid graphs have been previously investigated by Farley *et al.* [23]. 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 (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 5: 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 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 [23]), 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 $i_s = \frac{m+1}{2}$ or $j_s = \frac{n+1}{2}$) 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 $OPT+2$. ■

This results indicates that any SPT based broadcast (e.g., flooding with sense of direction) provides near-optimal result. In addition, in [13] it is shown that the LRF heuristic builds an SPT which provides an optimal solution.

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 [24] 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 the our heuristic with the following schemes.

- Approx-MDM multicast algorithm. The approximation algorithm (see Section IV-A) needs to solve multiple linear programs, and therefore it requires high polynomial order running time. 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.
- 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 [21], where the edge weights are defined using the formulation of section IV-E.
- 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 IV-E) from the source to any other host in the graph. The selected value is a non-tight lower bound on the performance of any multicast scheme. In the graphs shown, this delay bound is labelled as Max. Latency.

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 LRF, 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. In the graphs shown, the LRF results are labelled as Heuristic-MDM.

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. The latter case captures internet-like scenarios. The results for this important case indicate that the heuristic tree outperforms both the approximation tree and the SPT. This performance gap between the heuristic and the approximation, 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.

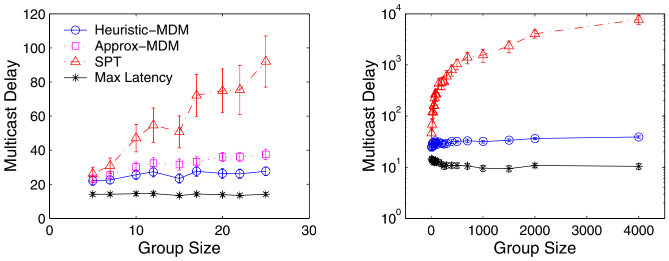

Fig. 4. The multicast delay for a clique topology with random network costs from $[1, 10]$

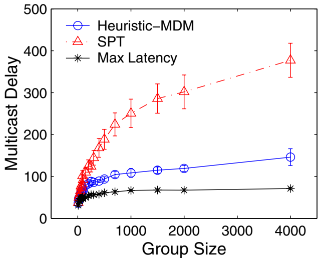

Fig. 7. The multicast delay for a power-law topology with random network costs from $[1, 10]$

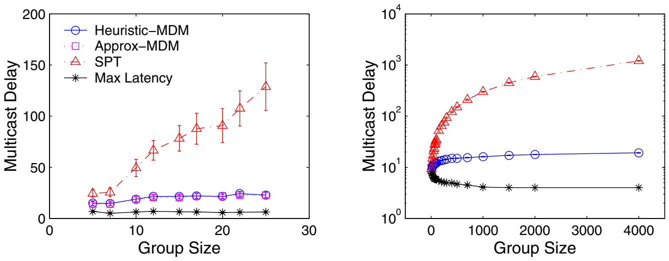

Fig. 5. The multicast delay for a clique topology with random processing costs from $[1, 10]$ and unit communication costs

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 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 [8]) 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.

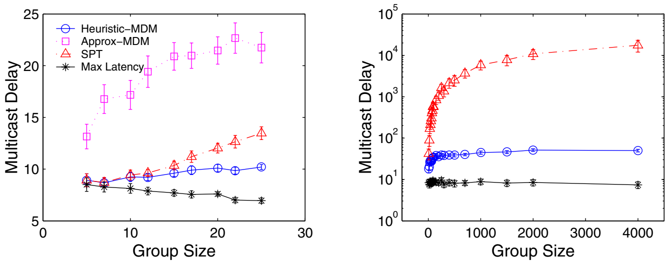

Fig. 6. The multicast delay for a clique topology with random communication costs from $[1, 10]$ and unit processing costs

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 of the algorithm. 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.

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 provides worst-case performance and produces non-scalable results.

VII. Concluding Remarks

In this work we looked at building efficient application layer multicast trees. We have presented two solutions to the MDM problem, a logarithmical proven approximation and a heuristic. 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.

Our approximation has an additive factor that depends on the value of the processing delays. We are now working on a better approximation which depends only on the size of the multicast group.

Acknowledgment

We thank Israel Cidon, Avishai Wool, and the anonymous reviewers 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 Network magazine, Special issue on "Multicasting: An Enabling Technology"*, Jan. / Feb. 2003.

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

[3] D. Pendarakis, S. Shi, D. Verma, and M. 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'02*, New-York, NY, USA, June 2002.

[5] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *IEEE INFOCOM'03*, San Francisco, CA, USA, Apr. 2003.

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

[7] S. Saroiu, P. K. Gummadi, and S. 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] I. Cidon, I. Gopal, and S. Kutten, "New models and algorithms for future networks," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 769 – 780, May 1995.

[9] 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'03*, San Francisco, CA, USA, Apr. 2003.

[10] S. Y. Shi, J. Turner, and M. Waldvogel, "Dimensioning server access bandwidth and multicast routing in overlay networks," in *NOSSDAV 2001*, June 2001, pp. 83–92.

[11] 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*, vol. 9, no. 5, pp. 525–540, Oct. 2001.

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

[13] E. Brosh, "Approximation and heuristic algorithms for minimum delay application-layer multicast trees," Master's thesis, Departement of EE-Systems, Tel-Aviv University, Ramat-Aviv, Israel, 2003.

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

[15] M. Garey and D. Johnson, *Computers and Intractability*. San Francisco: Freeman, 1979.

[16] 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.

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

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

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

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

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

[22] P.Francis. (1999) Yoid: extending the multicast inetrnet architecture. [Online]. Available: http://www.aciri.org/yoid/

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

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