A Case for Mesh-Tree-Interaction in End System Multicasting*

Anirban Chakrabarti and Govindarasu Manimaran

Dept. of Elecrical and Computer Engineering, Iowa State University
{anirban,gmani}@iastate.edu

Abstract. End System Multicasting (ESM) is fast becoming a feasible alternative to IP multicasting. ESM approaches can be broadly classified into two main categories: (i) Tree first approaches, where an overlay tree is constructed on the physical network, (ii) Mesh first approaches, where a mesh is constructed on the physical network and then a tree is created on the constructed mesh. In this paper, we propose a generic Mesh Tree Interaction (MTI) mechanism, which combines the management efficiency of the mesh first approaches and the performance benefits of the tree first approaches. To achieve this, MTI uses the concept of mesh and enables interactions between the mesh and the underlying multicast tree. Our simulation studies show that MTI results in significant improvement in the quality (average delay metric) of the multicast tree.

1 Introduction

Multicasting has been the most popular mechanism for supporting group communication. In a multicast session, the sender transmits only one copy of each message that is replicated at appropriate routers inside the network and delivered to multiple recipients. For this reason, multicasting typically requires less total bandwidth than separately unicasting messages to each receiver. In order to determine whether to implement multicasting at IP or application layer, implementation complexity vs. performance trade-offs need to be considered. Several prototypes have been developed and IP multicasting has been added as a feature in many commercial routers. In spite of the advancements in the field of IP multicasting, IP multicasting suffers from scalability problem, as each router needs to store group specific information. Also, implementing higher level features like congestion control, flow control, reliability and security have been shown to be more difficult in IP multicasting, than in the unicasting case.

As an alternative to IP multicasting, researchers have proposed the End System Multicasting (ESM) approach [4,5,6,7], wherein the complex multicasting features like replication, group membership management and multicast routing are implemented at the application layer, assuming only the end-systems or hosts are responsible for multicasting. As all the complexities are handled at the hosts rather than at the routers, it offers some distinct advantages over its IP counterpart. (i) ESM is easier to implement, as there is no complexity required at the routers, (ii) Complex functionalities like congestion control, reliable data transfer are handled separately at the unicast level, and

* This research was supported in part by the NSF under grant ANI-0240433

N. Mitrou et al. (Eds.): NETWORKING 2004, LNCS 3042, pp. 186–199, 2004.

therefore manageable, (iii) Adding security features to multicasting is easier as routers are not involved.

In spite of these advantages, ESM has some issues which need future research attention. (i) The quality of the multicast tree produced using ESM is worse than that produced using IP multicasting, (ii) Since each node in the ESM tree is a host, therefore the nodes have limited capability in terms of bandwidth and processor capabilities, (iii) The multicast sessions are unreliable as they depend on the hosts for data transmission.

Multicast trees in ESM can be constructed using two approaches: (a) Tree-first approaches and (b) Mesh-first approaches. In Tree-First approach, members directly select their upstream neighbors from among the known members [3]. In Mesh-first type of approach, a mesh is constructed on the physical network. Narada [4] and NICE [5] are examples of this approach.

2 Problem Statement and Motivation

In this section we formally define the ESM tree management problem, and then provide motivation for the approach taken in this paper to solve the problem.

Given an undirected network $N = (V, E)$, where V is the set of vertices or nodes, and E is the set of edges or links. Let e_{ij} be an edge between nodes i and j, such that $e_{ij} \in E \, \forall i, j \in V$. $D_{i,j}$ be the delay associated with the edge e_{ij}. Let S be the set of all shortest paths in N, and s_{ij} is the shortest path between nodes i and j, such that $s_{ij} \in S$ $\forall i, j \in V$. Let M be the set of members in a multicast session, such that $M \subseteq V$. Let F_i be the fanout constraint of each member i. The problem is to construct a multicast tree $T = (V_M, S_M)$, $S_M \subseteq S$ spanning all members so that the average delay to all members is minimized such that $d_i \leq F_i \ \forall \, i \in M$, where d_i is the degree at node i.

We call the above problem as ESM tree management problem. In this paper, whenever we refer to quality of a multicast tree we use average delay as the metric. The ESM tree management problem can be tackled using two methods. The first method creates a degree constrained spanning tree on a fully connected virtual graph. As shown in [8,9], the problem is NP-Complete. Tree-first techniques use this approach. The second approach, is through construction of a degree-constrained K-spanner on the fully connected virtual graph. A degree-constrained K-spanner is a subset of the fully connected virtual graph such that, each node satisfies the degree constraint and the shortest path between any two node in the K-spanner is not more than K times the shortest path in the fully connected virtual graph. As shown in [10], this problem is also NP-Complete. In this approach, after the construction of the K-spanner, a spanning tree is constructed on the K-spanner. Mesh-first techniques use this approach. In this paper, we propose a technique called Mesh Tree Interaction (MTI), which combines the management ability of the mesh-first approaches, and the performance benefits of the tree-first approaches.

As mentioned earlier, both Tree-first and Mesh-first approaches are based on NP-Complete problems. Therefore, both the approaches use approximations to construct spanning tree and K-spanner respectively. Independent mesh and tree, though result is simplicity in mesh management, result in creation of low-quality tree as mesh construction is done without taking the actual tree construction into account. Therefore, mesh construction may result in creation of mesh links which do not contribute to the improve-

ment of the quality of the multicast tree. It is to be noted that mesh provides redundancy, however it is the multicast tree which is used for actual data dissemination. Therefore, quality of multicast tree is absolutely critical for group communication. In this paper, we refer to average delay as the 'quality' of the multicast tree.

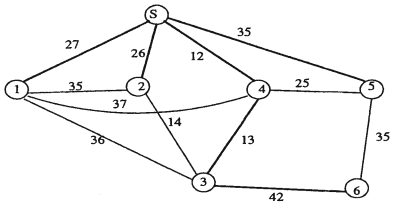

Fig. 1. An Example Mesh

In Figure 1, an example mesh is shown which is a subset of a fully connected virtual graph. Each link in the figure is the shortest path between the nodes. The number shown with each link indicates the delay of the shortest path between the two nodes. The links which are part of the multicast tree are indicated in the figure using darker lines. Let S be the source of the multicast communication; shortest path from node 6 to node 4 has a delay of 12. However, link $(6-4)$ is not part of the mesh, as shown in the figure. Let us assume the fanout limit for each node in this example is 4. Therefore, node 6, if connected to node 4 can provide a better delay path for itself. However, node 4 has reached its fanout limit (in this case 4). It is to be noted that in mesh-first approaches, each node independently tries to satisfy the fanout constraint and find the best neighbor at the mesh level. Since the Mesh-first protocols have no way to identify that link $(6-4)$ if added to the mesh, it will result in a better tree, will not add link $6-4$ to the mesh. This example shows that there is a need for interaction among the constructed mesh and tree so that "unimportant" mesh links can be removed to eventually produce better quality tree. Referring back to the above example, if node 4 had somehow realized, during link addition itself, that links $(1-4)$and $(4-5)$ are "unimportant" links, or links which will not result in a better quality tree, then one of these links could be removed in this case and the link $(6-4)$ can be accommodated such that the quality of the overall multicast tree is improved. In other words, a continuous interaction between mesh and tree is needed to construct a better quality mesh, which eventually leads to the construction of better quality tree. In this paper we propose a mesh-tree interaction approach which achieves the above. MTI identifies whether a link is important (part of the tree) or not (part of the non-tree mesh), and takes action based on the information. MTI technique achieves the following objectives: (a) MTI achieves a better "quality" tree than other Mesh-first protocols. (b) MTI is easily deployable, as group management is still controlled at the mesh level, instead of tree level in case of Tree-first protocols. (c) MTI can be used in isolation, as well as in conjunction with any of the existing Mesh-first protocols like Narada and NICE.

The rest of the paper is organized as follows: In Section 3, an overview of the MTI approach is provided with important definitions to be used for the rest of the paper. In

Section 4, the different steps of MTI are described in detail. A Restricted MTI (R-MTI) approach is outlined in Section 5. Finally, in Sections 6 and 7, simulation results and some concluding statements are provided respectively.

3 Mesh-Tree Interaction (MTI) Overview

Mesh-tree Interaction (MTI) is a mechanism to create "good" quality multicast tree through the improvement of the mesh in an iterative manner. While all the mesh-first protocols create the tree from the mesh, MTI improves the quality of the mesh based on the constructed tree, which in turn improves the quality of the tree. On an abstract level, the main difference between a standard mesh-first approach and MTI lies in the inherent understanding of the nature of the multicast tree, which is used to construct a better mesh. The basic difference is illustrated in Figure 2(a). While in Mesh-first approaches quality of the tree depends enormously on the quality of the underlying mesh, MTI uses an iterative process as tree structure influences the mesh, which in turn influences the tree structure.

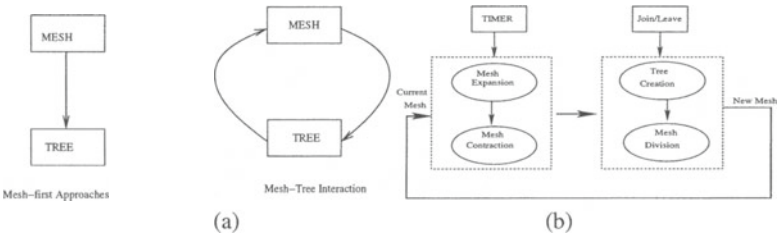

Fig. 2. (a)Mesh-first Approaches vs. MTI, (b) Different steps of MTI

To implement MTI, the mesh is divided into Primary Mesh (PM) and Secondary Mesh (SM), where PM contains all the links that are part of the multicast tree and SM contains all the links that are not part of the multicast tree. The second difference between Mesh-first approaches and MTI is the selection of the "best" neighbors for mesh optimization. Goodness of neighbors are identified by a parameter called the Upstream Correlation Factor (ρ), which determines how "good" the upstream neighbor of a node is. Mesh-tree Interaction has three main steps which differentiate the approach from the traditional mesh-first and tree-first approaches:

Mesh Division: This is the first step where the mesh is divided into Primary Mesh and Secondary Mesh. The intuition behind this is to differentiate between tree links and non-tree links within a mesh. Mesh division also helps MTI to keep track of the important links which constitute the Primary Mesh, and unimportant links which constitutes the Secondary Mesh. This prioritization helps MTI to accommodate links which eventually results in the construction of a better multicast tree.

Mesh Expansion & Contraction: In mesh expansion, links are added to the secondary mesh which eventually leads to the improvement of the quality of the mesh. Mesh

contraction takes place when certain mesh links are deleted to make way for mesh expansion. The interaction between the different steps is illustrated in Figure 2(b).

Tree Creation: Tree is created using shortest path on the total mesh. Tree creation step leads to the mesh division. Tree creation step is not much different from the tree creation protocol described in [4].

Property 1: $-1 \leq \rho \leq 1$

Property 2: If $\rho^s_{ij} = \delta$, then $\rho^s_{ji} = -\delta$.

Property 3: $\rho^s_{is} = 1$.

Property 4: Let i, j and k are three nodes and source is s, $\rho^s_{ij} > 0$ and $\rho^s_{jk} > 0$, then $\rho_{ik} \geq \frac{\rho^s_{ij} \cdot \Delta_{ij} + \rho^s_{jk} \cdot \Delta_{jk}}{\Delta_{ij} + \Delta_{jk}}$.

Property 5: If i_j is the best upstream neighbor of i_{j-1} $\forall j = 1, 2...n$, and $\rho^s_{i_{j-1}, i_j} > 0$, then $i_1, i_2 \ldots i_n$ cannot form a loop.

Fig. 3. Properties of ρ

3.1 Upstream Correlation Factor (ρ)

To understand Upstream Correlation Factor (ρ), we define the following terms:
Shortest Path Delay (Δ_{ij}): Δ_{ij} determines the delay of the shortest path between nodes i and j.
Upstream Neighbor (η^s_i): η^s_i indicates the upstream neighbor of node i with respect to source s.
Upstream Correlation Factor (ρ^s_{ij}): ρ^s_{ij} determines the quality of the upstream neighbor j of node i, where s is the source of the multicast tree. Mathematically,

$$\rho^s_{ij} = \frac{\Delta_{is} - \Delta_{js}}{\Delta_{ij}} \tag{1}$$

Properties of ρ are shown in Figure 3.

ρ or the Upstream Correlation Factor is an interesting and important metric to determines the quality of the upstream neighbor. $\rho^s_{ij} = 1$ indicates that the shortest path of i goes through j, and $\rho^s_{ij} = -1$ indicates that the shortest path of j goes through i. Higher the value of ρ, lower is the delay of the path through the upstream node. The reason ρ is such an important metric in ESM context is that it determines the quality of the upstream neighbor without actually knowing anything about the actual path. Therefore, the metric can be measured by sending ICMP packets to different nodes and measuring the delay experienced by the packets. Therefore, the metric does not violate the basic premises of the ESM architectures.

Let us illustrate the usefulness of ρ with the help of an example. Let node A and node B have shortest delay path to these source as 10 and 8 respectively. The current delay offered by the two nodes are 10 and 12 respectively. This is a possible scenario, as the current delay path depends on the quality of the underlying mesh. Now, a node C

has shortest delay path to nodes A and B as 2 and 3 respectively. Therefore, the current delay offered to node C, if node A is C's upstream neighbor is 12, while that offered if node B is the upstream neighbor is 15. However, node B has the capability of offering a delay path of 11 to node C. Therefore, it is a "better" upstream neighbor. ρ value reflects this as $\rho_{CA} = 0.5$ and $\rho_{CB} = 1$. From Property 1, ρ_{CB} is the maximum ρ value possible for any neighbor of C. Therefore, B is the best upstream neighbor.

4 Different Steps of MTI

As mentioned earlier, MTI consists of three steps: (a) Mesh Division, (b) Mesh Expansion, (c) Mesh Contraction and (d) Tree Construction. We describe the steps in detail, in the following subsections. In this paper, we only discuss about the first three steps as MTI is flexible, and any tree construction algorithm can be employed.

Mesh Division: In this step, the mesh is divided into Primary Mesh (PM) and Secondary Mesh (SM). PM consists of the links which are part of the multicast tree, and SM consists of all the links which are not part of the multicast tree. Each mesh consists of a two lists. SM consists of two lists SM^+ and SM^-, while PM consists of PM^+ and PM^-. List SM^+ consists of all links having positive ρ value among the SM links, sorted in descending order such that the head of the list contains the link having the maximum ρ value. On the other hand, SM^- consists of all links having negative ρ values among the SM links, arranged in ascending order such that the head of the list contains the link having minimum ρ value. The head and tail of SM^+ are called SM Maximum+ (Γ^+_{SM}), SM Minimum+ (γ^+_{SM}) respectively. The head and tail of SM^- are called SM Minimum- (γ^-_{SM})and SM Maximum- (Γ^-_{SM}) respectively. It is to be noted that, the Minimum and Maximum of SM^+ and SM^- are reversed. The reason behind this is that, the "importance" of a link is higher if the ρ value is lower, if the link has negative ρ value. The links in PM are also arranged in PM^+ and PM^- in a similar way. Let us illustrate the mesh division concept based on the example mesh shown in the Figure 1. For Node 4. PM^+ has only one link having ρ value of 1.0. PM^- also has only one link $(4-3)$, having ρ value of -0.85. SM^+ is empty. SM^- has two links $(4-1)$ and $(4-5)$ having ρ values of -0.41 and -0.52 respectively. $\Gamma^+_{PM} = \gamma^+_{PM}$, and $\Gamma^-_{PM} = \gamma^-_{PM}$, as there is only one link each in PM^+ and PM^-. Link $(4-1)$ is γ^-_{SM} and link $(4-5)$ is Γ^-_{SM}.

Mesh Expansion and Contraction: Mesh expansion forms the second step of MTI which may or may not lead to mesh contraction. Under mesh expansion each node proactively tries to expand the mesh by adding a neighbor to its mesh, which is better than at least one of its current neighbors. The "goodness" of a neighbor is measured by the ρ value mentioned earlier. Higher is the ρ value of the neighbor, better is the neighbor. The main principle behind mesh expansion is that each node (say i) attempts to find its Best Upstream Neighbor ($\mu+$). To identify $\mu+$ each node searches for the neighbor having the maximum ρ. Each searching node (i) searches for a set of candidate neighbors and calculates the ρ value for each of them. The candidate neighbor having the highest ρ value is selected as the best candidate neighbor (say n). If both i and n

have enough resources (fanout is less than the limit), the link is accommodated. In this case, mesh contraction is not called. Otherwise, mesh contraction is called.

In case of mesh contraction, some candidate links are found connected to both i (mentioned as $ReplaceLink_i$) and n (mentioned as $ReplaceLink_n$). If the resource constraint at node i is violated (the fanout at node i exceeds the limit) because of the addition of the link $(i - n)$, $ReplaceLink_i$ is deleted from the mesh to accommodate for link $(i - n)$. Similarly, if the resource constraint at node n is violated because of the addition of the link $(i - n)$, $ReplaceLink_n$ is deleted from the mesh. This step is part of the mesh contraction, as mentioned before. To search for $ReplaceLink_i$, firstly SM^+ of i is searched. The reason behind this is that, all links which are present in SM^+ are not part of the tree. If a link is found (say j) which has lower ρ than ρ_{in} (ρ value of link $i - n$), then j is identified as the $ReplaceLink_i$. If SM^+ is empty, γ^-_{SM} is identified as the $ReplaceLink_i$. If both SM^+ and SM^- are empty, then $ReplaceLink_i$ is identified from PM^+ if ρ value of the link is PM^+ is less than ρ_{in}. To identify $ReplaceLink_n$, if n is not the source same sequence is followed, only this time first SM^- is searched, then SM^+ and then PM^- as ρ changes sign in n (Property 2). However, if n is a source then $ReplaceLink_n$ is identified as a link (say j) in the PM^+ of n, if $\Delta_j > \Delta_{in}$. The reason for using Δ instead of ρ is because $\rho_{in} = 1$, in this case as n is the source (Property 3). The pseudo-code of the mesh is described in the Appendix.

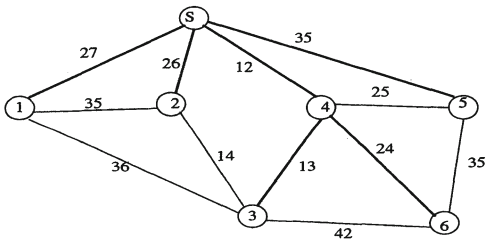

Fig. 4. The New mesh

To illustrate the above algorithm with the help of an example, we refer back to the Figure 1. Let us assume that $\Delta_{6S} = 36$ and $\Delta_{64} = 27$. This means that the shortest distance from node 6 to the source is 36, and the shortest delay from node 6 to node 4 is 27. From the figure, the shortest distance from source to node 4 is 12 *i.e.* $\Delta_{4S} = 12$. Therefore, $\rho^S_{64} = 0.9$ and $\rho^S_{46} = -0.9$. $\gamma^-_{SM} = (4 - 1)$, and $\rho^S_{41} = -0.41$. Node 6 does not violate its fanout constraint by accommodating the link. However, node 4 does. Therefore, mesh contraction algorithm needs to be called at node 4. There exists at least one link in the SM^- of node 4 having lesser importance than link $(4 - 6)$. Therefore, γ^+_{SM} *i.e.* link $(4 - 1)$ is removed from the mesh to accommodate link $(4 - 6)$. Hence, link $(4 - 1)$ is removed from the SM and link $(4 - 6)$ is added to the SM^- of node 4. After tree creation, this link will be added to the tree. After the expansion/contraction of the mesh, the mesh looks like Figure 4. After the mesh addition and tree creation, the average delay improves from 32.17, in the first case to 27.33.

5 Restricted MTI (R-MTI)

Comparing between MTI and any other flat mesh based ESM protocol (Narada for example), the following points need to be considered:

Message Complexity: Message complexity of MTI and any mesh-first protocol is comparable. In both the cases each node need to send $O(n)$ messages, where n is the number of nodes in the mesh.
Message size: In case of standard mesh-first approaches (like Narada), probe messages calculate the distance of the probing node to the potential neighbors. In case of MTI, the probe message should also include the distance of the potential neighbor to the source in addition to the distance information between the two nodes. This requires 4 bytes of extra information in the probing message.
Computational Complexity: In case of ESM, since all multicasting activities are handled at the end systems, therefore computational complexity assumes important proportions. The computational complexity in case of any flat mesh-first protocol is $O(n)$, where n is the number of nodes in the mesh. MTI increases the computational complexity to $O(flogf + fn)$, where f is the fanout limit of the nodes in the mesh.

Since the message and computational complexity of MTI increases linearly, then the scalability of the protocol suffers for high number of nodes in the mesh. A message complexity of $O(n)$ has the potential of message explosion, if the number of nodes in the mesh increases. Therefore, there is a need to device means to reduce or restrict the number of messages transmitted. Reduction of message complexity motivates the development of Restricted MTI (R-MTI). In R-MTI, the potential neighbor search is only restricted to the neighbors in either SM^- and PM^-. The ρ value of neighbors of SM^+ and PM^+ are calculated based on Property 4. R-MTI helps to restrict the worst-case message complexity from $O(n)$ under normal MTI, to $O(f^2)$, where f is the fanout limit of the nodes. Under normal case, it will be still less because search will be restricted to only neighbors having negative ρ. Similarly, the computational complexity is reduced from $O(flogf + fn)$ to $O(flogf + f^2)$. Though R-MTI has low computational and message complexity, it produces lower "quality" tree as the search space is restricted. Therefore, R-MTI introduces a trade-off between message and computational complexity with the "quality" of the multicast tree. In the simulation section R-MTI is studied vis-a-vis MTI and Narada to quantify this trade-off.

6 Simulation Studies

In order to evaluate the effectiveness of our MTI model, we conducted extensive simulation studies using ns [11]. In our simulation studies, we compared our MTI model with Narada as well as several Centralized algorithms. The various inputs for the simulation studies were generated as follows: (a) Random network topologies were generated based on a given input parameter "graph density." This parameter determines the average node degree and hence the connectivity of the network. The higher the value, the denser the topology. (b) The selection of receivers for a given multicast session were uniformly distributed from the node set. (c) Members join and leave the multicast group, and the

mesh is reorganized assuming a Distance Vector protocol running on the mesh level. (d) For each point in the graph, an average of 10 simulation runs were conducted.

The *default parameters:* are (i) Total number of nodes = 1000, (ii) 20% of all nodes are end hosts, (iii) Average Node degree = 4, (iv) Average number of members = 100, (v) Average link bandwidth = $15Mbps$, (vi) Average link delay = $12.5ms$ (vii) Member join/leave inter-arrival time = $100ms$, (viii) Average fanout of the nodes = 4.0.

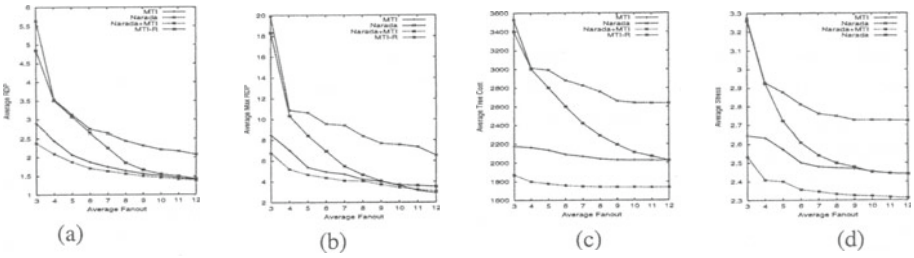

Fig. 5. Variation of (a) ARDP, (b) AMRDP, (c) Average Tree Cost and (d) Average Stress with varying Fanout Limit

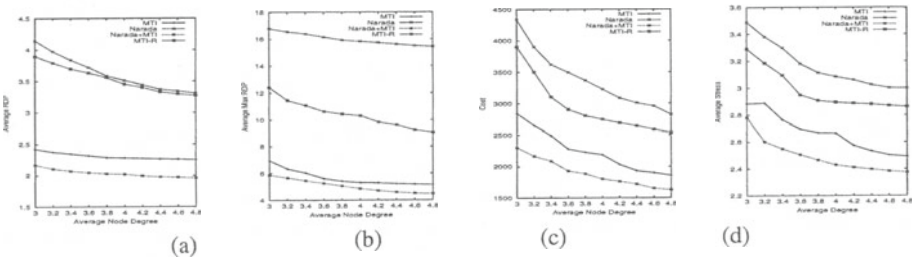

Fig. 6. Variation of (a) ARDP, (b) AMRDP, (c) Average Tree Cost and (d) Average Stress with varying Average Network Density

In order to compare the effectiveness of the various models, we evaluated the models according to the following performance metrics: (a) **Average Relative Delay Penalty (ARDP):** Relative Delay Penalty (RDP) is defined as the ratio of the delay provided by the current multicast tree to that provided by unicast averaged for all the members. (b) **Average Maximum Relative Delay Penalty (AMRDP):** Maximum Relative Delay Penalty (MRDP) is defined as the maximum RDP suffered by a node, among all the nodes currently in session. (c) **Average Stress:** Stress is defined as the average number of unicast flows per tree link. (d) **Average Tree Cost:** The average cost of multicast tree averaged out over time were compared to evaluate the effectiveness of the algorithms.

In order to evaluate the effectiveness of different approaches, we studied the effects of the following parameters: (a) Fanout constraint, (b) Group Dynamics, (c) Network Density and (d) Group Size.

Effect of Fanout Limit: In this set of experiments, the fanout limit of each node participating in the ESM session was varied and its effect was studied on different performance metrics for different approaches. The results are shown in Figure 5. In Figure 5(a), the

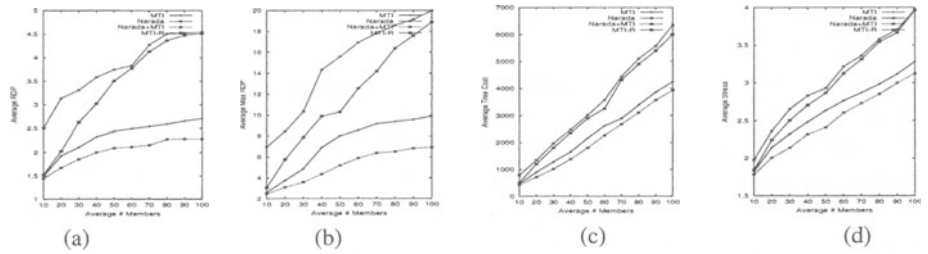

Fig. 7. Variation of (a) ARDP, (b) AMRDP, (c) Average Tree Cost and (d) Average Stress with varying Average Group Size

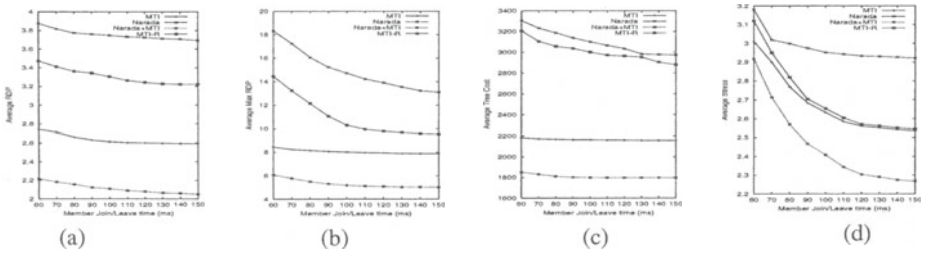

Fig. 8. Variation of (a) ARDP, (b) AMRDP, (c) Average Tree Cost and (d) Average Stress with varying Average Join/Leave Time

average RDP (ARDP) is studied with varying fanout limit. With the increase of fanout limit, each node in the multicast group can accommodate more mesh links, and therefore reduces the average delay of the overall tree. This trend can be observed for all the approaches. Comparing the relative performances of different approaches, the ARDP of the trees constructed using MTI is $50 - 60\%$ of that of the Narada trees. Narada and MTI combination reduces the ARDP value further by $10 - 15\%$, justifying that MTI and Narada can be combined in practice to get a better quality without losing the inherent easy maintenance of the mesh-first approaches. ARDP of the MTI-R approach, is lower (better) than that of Narada and gets closer to MTI with increase in fanout limit. The reason behind this is that, with increase in fanout limit, more links nodes are searched for the identification of better upstream neighbor.

Though ARDP is the primary metric used in MTI, the unique method of selecting best neighbors using ρ value reduces AMRDP, Average Tree Cost and Stress also. This point is justified in Figures 5(b), (c) and (d). The trends exhibited by each of these metrics is similar to that exhibited by the ARDP metric. Narada has the highest Average Tree Cost, MTI and Narada combination has the lowest. MTI and MTI-R lies somewhere in between. Trees produced by the MTI-R approach is similar to the MTI approach with increase in fanout limit.

Effect of Network Density: In this set of experiments the node degree of the physical network is varied and its effect is studied on the four different performance metrics mentioned above. Higher the average node degree, denser is the physical topology. Increasing the density of the nodes in the physical network has significant effect on the trees created on the overlay. Figure 6 illustrates the effect. As the network becomes

denser, the chances of two paths on the overlay going through the same physical links gets substantially reduced. This effect gets reflected the Average Stress metric (shown in Figure 6(d)), and in turn on all the other metrics used for performance analysis. Average Stress decreases with increasing node degree.

Figure 6(a) shows the variation of ARDP with average node degree of the physical network. As the network becomes denser, then there are more options to go from one node to another, as a result the chances of getting a better delay path increases. This phenomenon is reflected in Figure 6(a). Among the different approaches, combination of MTI and Narada has the lowest ARDP value. ARDP value in case of MTI is nearly 50% less than that of Narada. R-MTI is in between Narada and MTI and ARDP of most R-MTI trees are around 10% lower than that of Narada, while $20-30\%$ more than MTI. Similar trends are also witnessed for AMRDP, Average tree cost and Average Stress metrics.

Effect of Group Size: In this set of experiments, the average group size of a multicast session is increased. The results are shown in Figure 7. As the group size increases the size of the overlay increases. Therefore, the average cost of the multicast tree increases, as more members are part of the tree. Delay and stress metrics also show an increase, as more members join the group which may be farther away from the rest of the tree, resulting in increase in these performance metrics.

In Figure 7(a), the variation of ARDP metrics is shown with average group size. At low group size, ($\leq$ 60), R-MTI performs similar to MTI and its ARDP is nearly 50% less than that of Narada. The combination is 10% less than MTI. With increase in group size, the ARDP value increases for all approaches as the distance between nodes increases. The increase of R-MTI is maximum, as the ARDP value is nearly equal to that of Narada for group size $\geq$ 160. Increase of Narada is approximately linear with group size, while that of MTI and the combination is sub-linear. Therefore, ARDP of MTI is approximately 80% less than Narada at higher values of group size ($\geq$ 160). Variation of AMRDP is shown in Figure 7(b), which is similar to RDP.

In Figure 7(c) and (d), the variation of Average Tree Cost and Stress is shown with average group size. Both the parameters increase approximately linearly with average group size. The relative performance of the approaches match that of ARDP.

Effect of Group Dynamics: In Figure 8 variation of group dynamics is shown on the performance metrics. Group dynamics is measured by the join/leave time. Higher the join/leave time interval, lesser dynamics is the group. All the performance metrics increase with the increase in the group dynamics. The reason for this is that, the multicast tree and the mesh size gets bigger at a faster rate than the optimization when the group dynamics is very high.

MTI is more immune to group dynamics, as the ARDP, Average Cost and AMRDP increase approximately $3-5\%$ for MTI, while $5-7\%$ for Narada.

7 Conclusion

In this paper, we have proposed a mesh-tree interaction (MTI) approach which does not compromise on the inherent simplicity in management of the mesh first approaches, however builds a much better quality multicast tree than the mesh-first approaches. The

main principle behind the MTI approach is that the "quality" of the mesh is improved based on the underlying multicast tree, which in turn improves the quality of the tree itself. Thus, by keeping the mesh structure, the management simplicity of mesh-first approaches is maintained, and the iterative tree-building mechanism improves the quality of the tree dramatically. In this paper, we have carried out extensive simulation studies illustrating the MTI approach. In comparison to other mesh-first approaches like the Narada, MTI improves the ARDP metric by nearly $40 - 50\%$ and cost of the multicast tree improves by $20 - 30\%$ for lower fanout constraints (4-5). MTI can also be applied in conjunction with other mesh management techniques like Narada, which further improves ARDP by $10 - 15\%$. Future work includes: (i) Extending MTI to hierarchical mesh management techniques like NICE. (ii) Interoperability of MTI tree management techniques with that of IP tree management techniques.

References

1. S. Deering, "Multicast Routing in Internetworks and extended lans," in *Proc. SIGGCOM,* pp. 55-64, Aug. 1988.
2. C. Diot, B.N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network,* pp.78-88, Jan./Feb. 2000.
3. P. Francis, "Yoid: Your own Internet Distribution," *www.aciri.org/yoid,* Apr. 2000.
4. Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang, "Enabling Conferencing Applications on the Internet using an Overlay Multicasting Architecture," in *Proc. SIGGCOM,* Aug. 2001.
5. S. Banerjee, B. Bhattacharya, and C. Kommareddy, "Scalable Application Layer Multicast," in *Proc. SIGGCOM,* Aug. 2002.
6. D. Pendarakis, S. Shi, D. Verma, and M. Waldgovel, "ALMI: An Application Level Multicast Infrastructure," in *Proc. USENIX Symp. on Internet Technologies and Systems,* Mar. 2001.
7. M. Castro, P. Druschel, A. -M. Kermarrec, and A. Rowstron, "SCRIBE: A Large-scale and decentralized application-level multicast infrastructure," in *IEEE JSAC,* vol. 20, no. 8, Oct. 2002.
8. N. Deo and S. L. Hakimi, "The shortest Generalized Hamiltonian Tree," in *Proc. Annual Allerton Conference,* pp. 879-888, 1968.
9. G. Zhou and M. Gen, "Application to Degree Constrained Minimum Spanning Tree Problem using Genetic Algorithm," in *Engineering Design and Automation,* vol. 3, no. 2, pp. 157-165, 1997.
10. G. Korsatz and D. Pelleg, "Generating Low-Degree 2-Spanners," in *SIAM Journal on Computing,* vol. 27, no. 5, pp. 1438-1456, Oct. 1998.
11. UCB/LBNL/VINT Network Simulator - ns (version 2), Available at www.isi.edu/nsnam/ns.

Appendix

Property 1: $-1 \leq \rho \leq 1$

Proof: Let us assume that $\rho_{ij}^{s} > 1$. Then, from Equation 1 we get,

$$\Delta_{is} > \Delta_{ij} + \Delta_{js} \tag{2}$$

Equation 2 shows that there is an alternate path through j which is shorter than the shortest path $i - s$. This leads to contradiction, therefore

$$\rho_{ij}^{s} \leq 1 \tag{3}$$

To prove the lower bound, let us assume that $\rho_{ij}^{s} < -1$. Therefore, from Equation 1, we get

$$\Delta_{js} > \Delta_{ij} + \Delta_{is} \tag{4}$$

Since the network is undirected, $\Delta_{ij} = \Delta_{ji}$. Therefore, Equation 4 leads to a contradiction as there exists a shorter path than the shortest from j to s through i. Therefore,

$$\rho_{ij}^{s} \geq -1 \tag{5}$$

Equations 3 and 5 prove the property.

Property 2: If $\rho_{ij}^{s} = \delta$, then $\rho_{ji}^{s} = -\delta$.

Proof: From Equation 1,

$$\rho_{ij}^{s} = \frac{\Delta_{js} - \Delta_{is}}{\Delta_{ji}} = -(\frac{\Delta_{is} - \Delta_{js}}{\Delta_{ij}}) = -\rho_{ij}^{s} \tag{6}$$

Equation 6 proves the Property.

Property 3: $\rho_{is}^{s} = 1$.

Proof: The property can be proved by substituting $\Delta_{ss} = 0$ in Equation 1.

Property 4: Let i, j and k are three nodes and source is s, $\rho_{ij}^{s} > 0$ and $\rho_{jk}^{s} > 0$, then $\rho_{ik} \geq \frac{\rho_{ij}^{s} \times \Delta_{ij} + \rho_{jk}^{s} \times \Delta_{jk}}{\Delta_{ij} + \Delta_{jk}}$.

Proof: From Equation 1,

$$\rho_{ik} = \frac{\Delta_{is} - \Delta_{ks}}{\Delta_{ik}} \geq \frac{\Delta_{is} - \Delta_{ks}}{\Delta_{ij} + \Delta_{jk}} \tag{7}$$

The Property can be proved by substituting the values of Δ_{is} and Δ_{js} from Equation 1 to Equation 7.

Property 5: If $i_j = \eta_{i_{j-1}}^{s} \quad \forall j = 1, 2...n$, and $\rho_{i_{j-1}, i_j}^{s} > 0$, then $i_1, i_2 \ldots i_n$ cannot form a loop.

Proof: Let us assume that i, j and k are nodes such that $j = \eta_i^s$, $k = \eta_j^s$ and $i = \eta_k^s$ *i.e.*, i,j and k form a loop. Also, $\rho_{ij}^{s}, \rho_{jk}^{s}, \rho_{ki}^{s} > 0$. From Equation 1, we get $\Delta_{is} > \Delta_{js}$,$\Delta_{js} > \Delta_{ks}$ and $\Delta_{ks} > \Delta_{is}$. This leads to contradiction, therefore such a loop cannot occur. This argument can be extended to prove the Property $\forall \; k = i_1, i_2 ... i_n$

Mesh Expansion & Contraction

1. If Current Fanout of i is less than the fanout limit of i goto 9. Therefore, if i can accommodate the link it will as long as n can also accommodate the link.
2. If SM^{+} of i is empty goto 4. To accommodate the link, some link of i need to be removed, since SM^{+} is empty, links from SM^{-} are searched.
3. If $\rho_{in} > \rho_{\gamma_{SM}^{+}}$
   a) This means that link $(i - n)$ is "better" than at least one link in, SM^{+}
   b) Set $ReplaceLink_i = \gamma_{SM+}$ *i.e.* γ_{SM+} is chosen as the likely candidate for removal from the mesh.
   c) Goto 9

4. Otherwise Goto 18 *i.e* current link is "good" enough, therefore the link is not added to the mesh.
5. If SM^- of i is empty Goto 7 *i.e.* SM is empty, therefore PM is searched.
6. If SM^- of i is non-empty
   a) set $ReplaceLink_i = \gamma_{SM^-}$ *i.e.* γ_{SM^-} is a likely candidate for removal.
   b) Goto 9 *i.e.* check whether n can accommodate the link.
7. If $\rho_{in} > \rho_{\gamma^+_{PM}}$
   a) Current Link is better than the link in PM^+, therefore PM^+ is the likely candidate for removal.
   b) Set $ReplaceLink_i = \gamma_{PM^+}$ *i.e.* γ_{PM^+} is a likely candidate of removal from the mesh.
   c) Goto 9
8. Otherwise Goto 18, *i.e.* no candidate can be found. Therefore, the link is not "good" enough.
9. If current fanout of n is less than the fanout limit of n Goto 17 *i.e.* the link can be added without removing any current link in n.
10. If n is a source node Goto 17.
11. If SM^- of n is empty Goto 13 *i.e.* n is not a source and SM^+ for candidates.
12. If $\rho_{in} < \rho_{\gamma^-_{SM}}$
    a) Current link is "better" than at least one link in SM^- of n.
    b) set $ReplaceLink_n = \gamma_{SM^-}$ *i.e.* γ_{SM^-} is the likely candidate for removal.
    c) Goto 18
13. If SM^+ of n is empty Goto 15
14. If SM^+ of n is non-empty
    a) set $ReplaceLink_n = \gamma_{SM^+}$ *i.e.* γ_{SM^+} is a likely candidate of removal from the mesh.
    b) Goto 18
15. If $\rho_{in} < \rho_{\gamma^-_{PM}}$
    a) Current link is "better" than at least one link in PM^- of n, therefore γ^-_{PM} is the likely candidate candidate for removal.
    b) set $ReplaceLink_n = \gamma_{PM^-}$ *i.e.* γ_{PM^-} is a likely candidate of removal from the mesh.
    c) Goto 18
16. Otherwise Goto 19
17. If $\Delta_{in} < \Delta_{jn}, j \in PM^+$
    a) set $ReplaceLink_n = j$ *i.e.* j is a likely candidate of removal from the mesh. Here Δ is used as a parameter instead of ρ because, in this case $\rho_{in} = 1$ (Property 3).
    b) Goto 18
18. The current link is added
    a) Link $(i - n)$ is added to the secondary mesh
    b) Links $ReplaceLink_i$ and $ReplaceLink_n$ are removed from the mesh of i and n respectively.
19. Exit