Performance Analysis for Overlay Multicast on Tree and M-D Mesh Topologies (II)

Wanqing Tu, Xing Jin, and Cormac J. Sreenan
Department of Computer Science
University College Cork, Cork, Ireland
{wt1,cjs}@cs.ucc.ie

***Abstract*— In previous work of this topic, we have analyzed the worst-case performance for tree-based and mesh-based multicast along the link stress, the number of overlay hops, and the number of shortest paths. The average performance and the difference between the worst-case and the average performance for these metrics are studied now. We present theoretical results on average performance and performance difference. We also program NICE tree and CAN-based multicast in NS2 to evaluate our theoretical prediction and compare the performance of tree-based and mesh-based multicast. Simulation results prove our theoretical analysis. Tree-based multicast suits to real-time and continuous single-source streaming media applications, while mesh-based multicast holds the promise for reliable and multi-source non-real-time and text applications.**

I. Introduction

Overlay multicast is motivated and populated by inter-domain multicast applications. It releases multicast support of underlay network through replicating and forwarding packets at each end host. Such packet process scheme constructs an *overlay topology* that is on top of but different from the underlying physical topology.

An overlay topology is composed of a set of virtual overlay paths. Each overlay path covers several underlying physical links and connects two different end hosts directly in the application layer. The overlay paths usually form two topologies: tree and m-D mesh. A tree is such a topology that the group member (i.e., the end host) may connect to a trunk link with a central transmission facility. We use the terms group member and end host interchangeably in this paper. A m-D mesh is generated by partitioning a m-D Cartesian space among all end hosts such that every end host "owns" its individual, distinct zone within the overall space. For brevity, we call overlay multicast built on top of tree and m-D mesh as tree-based and mesh-based multicast respectively later in this paper.

Tree and m-D mesh are two distinctly different topologies that generate different performances for overlay multicast. In [1], we analyzed tree-based and mesh-based multicast in theory and presented a set of theorems on the worst-case performance along the metrics of *the link stress*, *the number of overlay hops*, and *the number of shortest paths*. In this paper, we are going to continue working on the performance difference of tree-based and mesh-based multicast. Our further research will study the average performance of overlay multicast along *the link stress* and *the number of overlay hops*. Average performance represents the statistical performance of a whole multicast group. Moreover, we will analyze the difference of the worst-cast and the average performance along the above metrics for tree-based and mesh-based multicast respectively. The difference reflects the performance variance among group end hosts. Our results are achieved based on employing the architectures of NICE [10] and CAN-based multicast [8] as the topology models of tree and m-D mesh respectively. For a group of n members, we can derive

- *The average link stress* of tree-based multicast is lower bounded by $\frac{k+k(n-j_1)}{n}$, where k (normally set as 3 in NICE simulations) is a constant parameter of NICE tree, and and $j_1 \in [0, k-1]$; *the average link stress* of mesh-based multicast is upper bounded by $2m + \frac{1}{n}$;
- *The link stress difference* of tree-based multicast is $klog_k^n$; *the*

link stress difference of mesh-based multicast is 1;

- *The average number of overlay hops* of tree-based multicast is lower bounded by $\frac{H^2}{4n}(s-1)^H$, where s is the cluster size and H is the bound of NICE tree; *the average number of overlay hops* of mesh-based multicast is $\frac{mn(n^{\frac{1}{m}}-1)}{2}$;
- *The overlay hop difference* of tree-based multicast is $1+\frac{7H}{4}$; *the overlay hop difference* of mesh-based multicast is $\frac{m(n^{\frac{1}{m}}-2)+1}{2}$.

We then use simulation to observe and prove the metrics analyzed. Simulation results match our theoretic analysis. Compared to the results in [1], we know tree-based and mesh-based multicast further. Tree-based multicast is good for real-time and continuous streaming media multicast communication with single source and large receivers. Mesh-based multicast suits to multi-source non-real-time and text traffic multicast communications due to its load-balanced and multi-path distribution.

The paper is organized as follows. Section II introduces analysis models. Section III presents theorems on the average link stress, and the link stress difference. Section IV presents theorems on the average number of overlay hops, and the overlay hop difference. Section V evaluates the average performance and the performance difference through simulations. Section VI concludes the paper.

II. Analysis Models

A. Topology Models

NICE and CAN-based multicast are two popular overlay multicast protocols and most of current multicast protocols are developed based on them. Hence, similar as the topology models in [1], we employ NICE and CAN-based multicast as the models of tree-based and mesh-based multicast respectively. Due to the paper length limitation, we are not going to introduce these two models here. But, we give examples for them in Fig. 1. The arrow lines in Fig. 1 (a) show the data paths of NICE tree. The arrow lines in Fig. 1 (b) illustrate the forward flooding multicast routing in a 2-D mesh. For detailed models, please refer to [1].

B. Metric Models

We study the average performance and the performance difference along *the link stress* and *the number of overlay hops* in this paper.

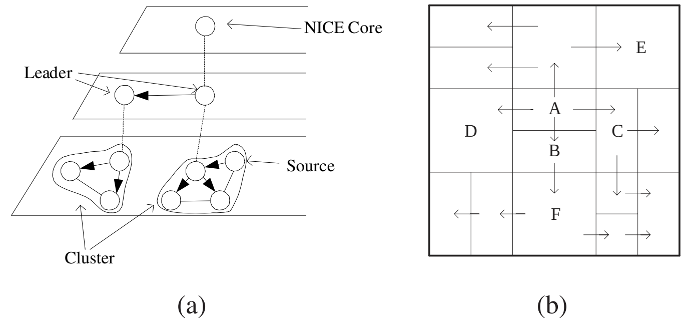

Fig. 1. The example architectures. (a) for NICE tree. (b) for CAN-based multicast in a 2-D mesh.

We now introduce our way to define the average performance and the performance difference.

Average performance is defined as the average value of the summation of the measured metric when considering all of end hosts in a multicast system. We use the following equation to calculate the average performance for a metric M.

$$\bar{M} = \frac{\sum_{i=1}^{n} M_i}{n}, \tag{1}$$

where M_i is the metric achieved by the ith member, and n is the group size. $\bar{M}$ statistically represents the performance of the multicast group along the metric M. And, in general, an acceptable $\bar{M}$ shows that most of group end hosts can receive acceptable performance of the metric M.

Performance difference of the metric M is defined as the difference between the worst-case and the average performance of M. That is,

$$\Delta M = \ddot{M} - \bar{M}, \tag{2}$$

where $\ddot{M} = \max\{M_i | 1 \leq i \leq n\}$ is the worst-case performance of the metric M in the group system. ΔM is defined to reflect the performance variation at different group members. That is, by the calculated ΔM, we can know the performance balance of a measured system. We now analyze the average performance and the performance difference.

III. Link Stress

In [9], *link stress* is defined as the number of identical packet copies passing through the same underlying links. Overlay multicast suffers from link stress (larger than 1) which excessively consumes network capacities. We study the average link stress and the link stress difference for tree-based and mesh-based multicast in this section.

Theorem 1 *For a group of n members, the average link stress of NICE tree is*

$$\bar{LS_{tree}} \geq \frac{k + k(n - j_1)}{n}. \quad (3)$$

Proof. According to NICE construction, the highest tree is generated when each cluster only has k members. For simplicity, we use H to denote the height of the highest NICE tree. Theorem 1 in [1] proves that H is bounded by $\lceil \log_k^{[k+(n-j_1)(k-1)]} \rceil$. We now prove the theorem.

The link stress of the node in the highest layer is sH, where s is the cluster size. There is 1 node in the highest layer. Hence, the total link stress of node in the highest layer is sH.

The link stress of each node in the second highest layer is $s(H-1)$. There are $(s-1)$ nodes whose highest layers are the $(H-1)$th layer. Hence, the total link stress of nodes in the second highest layer is $s(s-1)(H-1)$.

It can be deferred that the link stress of each node in the ith highest layer is $s(H-i+1)$. There are $s^{(i-2)}(s-1)$ nodes whose highest layers are the $(H-i+1)$th layer. Hence, the total link stress of nodes in the $(H-i+1)$th highest layer is $s^{(i-1)}(s-1)(H-i+1)$.

According to the above analysis, the total link stress of all links in the multicast system is

$$\begin{aligned}(TLS)_{tree} &= sH + s(s-1)(H-1) + s^2(s-1)(H-2) + ... + \\ &\quad s^{(H-1)}(s-1)(H-H+1) \\ &= sH + (s-1)H[s + s^2 + ... + s^{(H-1)}] - (s-1)[s + 2s^2 + ... \\ &\quad + (H-1)s^{(H-1)}] \\ &= sH + (s-1)[H\frac{s^H - s}{s-1} + \frac{(s + ... + s^{(H-1)}) - (H-1)s^H}{s-1}] \\ &= \frac{s^{H+1} - s}{s-1} \end{aligned} \quad (4)$$

From (4), obviously, we have $(TLS)_{tree} \geq s^H - 1$ and $(TLS)_{tree} \sim O(kn)$ when n is large enough. It can be inferred that

$$\bar{LS_{tree}} = \frac{s^{H+1} - s}{n(s-1)} \geq \frac{k^{H+1} - k}{n(k-1)}.$$

Since $H = \lceil \log_k^{[k+(n-j_1)(k-1)]} \rceil$, the above equation shows that

$$\bar{LS_{tree}} \geq \frac{k + k(n - j_1)}{n}. \quad (5)$$

Therefore, when n is large enough, we have $\bar{LS_{tree}} \sim O(k)$. Q. E. D.

Corollary 1 *For a group of n members, the link stress difference of NICE tree is*

$$\Delta LS_{tree} \leq klog_k^n. \quad (6)$$

Proof. According to Theorem 1 in [1], the worst-case link stress of a NICE tree with n members is $\ddot{LS_{tree}} = k + klog_k^n$. And, Theorem 1 in this paper has proven that the average link stress of a NICE tree with n members is $\bar{LS_{tree}} \geq \frac{k+k(n-j_1)}{n}$. Therefore, we have

$$\Delta LS_{tree} = \ddot{LS_{tree}} - \bar{LS_{tree}} \leq k[log_k^n + \frac{j_1 - 1}{n}]. \quad (7)$$

When n is large enough, we have $\ddot{LS_{tree}} \leq klog_k^n$. Q. E. D.

We now analyze CAN-based mesh multicast.

Theorem 2 *For a multicast group with n members, the average link stress of m-D CAN mesh is upper bounded by $2m + \frac{1}{n}$.*

Proof. CAN mesh is generated by partitioning and merging zones to deal with nodes' join and departure. CAN utilizes a uniform way to partition and merge mesh zones. Such partition and re-merging are done by assuming along a certain ordering of dimensions. Furthermore, the largest zone is selected to be partitioned when a new member joins, and the smallest zone is selected to be re-merged when a member wants to leave. Therefore, there is no large difference in the split of mesh zones and most of mesh zones may have the same number of neighbors.

Based on the above analysis, in general, each zone has $2m$ neighbors except for one zone who has the most number of neighbors $2m + 1$. And, because CAN-based flooding forwards packets to neighbors who haven't received the packets before, for the average link stress of n members in m-D mesh, we have

$$\bar{LS_{mesh}} = \frac{(n-1) \times 2m + (2m+1)}{n} = \frac{2mn + 1}{n} = 2m + \frac{1}{n}.$$

Q.E.D.

Corollary 2 *For a group of n members, the link stress difference of CAN-based multicast is*

$$\Delta LS_{mesh} = 1 - \frac{1}{n}. \quad (8)$$

Proof. According to Theorem 2 in [1], the worst link stress of CAN-based multicast is $\ddot{LS_{mesh}} = 2m + 1$. And, we have proven that $L\bar{S}_{mesh} = 2m + \frac{1}{n}$ in the above theorem. Therefore, we have

$$\Delta LS_{mesh} = L\ddot{S}_{mesh} - L\bar{S}_{mesh} = 1 - \frac{1}{n}.$$

Q.E.D.

Remark 1: Based on the analysis in Corollaries 1 and 2, when n is large enough, we have $\Delta LS_{tree} = klog_k^n >> 1 = \Delta LS_{mesh}$. That is, the links in tree topology suffer from larger difference in link stress than the links in mesh topology. It shows that mesh routing disperses traffic load more evenly than tree routing.

IV. Number of Overlay Hops

The number of overlay hops refers to the number of packet hops in the overlay network between two end hosts. Larger number of overlay hops means more times of end host process. Since end hosts have lower capacities (e.g., CPU speed) to process packets, the small number of overlay hops is expected. In this section, we are interested in the average number of overlay hops and the overlay hop difference of tree-based and mesh-based multicast.

Theorem 3 *For a multicast group with n members, the average number of overlay hops experienced by packets passing through NICE tree is $O\bar{H}_{tree} \geq \frac{H^2}{4n}(k-1)^H$.*

Proof. For simplicity, we consider the average number of overlay hops when packet multicast begins at NICE core.

Suppose NICE cluster size is s, and the height of NICE tree is H that is calculate as $\lceil log_k^{[k+(n-j_1)(k-1)]}\rceil$, where $j_1 \in [1, k)$ is the number of members that haven't joined in any clusters at the lowest layer of NICE tree. We first consider the group member in the highest layer (i.e., NICE core). It can be inferred that there are $(H-1) \times (s-1)$ group members who are NICE core's cluster members. There is 1 overlay hop from NICE core to these cluster members. Hence, the total number of overlay hops experienced by packets to NICE core's cluster members is $(H-1) \times (s-1)$.

We now consider the $(s-1)$ group members in the second highest layer. Each of these group members has $(H-2) \times (s-1)$ cluster members. Therefore, there are totally $(H-2) \times (s-1)^2$ group members belonging to the clusters of the $(s-1)$ group members in the second highest layer. As we known, there are 2 overlay hops from NICE core to these cluster members. Hence, the total number of overlay hops experienced by packets to reach these cluster members is $2 \times (H-2) \times (s-1)^2$.

Similarly, we consider the ith highest layer. There are $(s-1)^{i-1}$ group members in this layer. And, each of these group members has $(H-i) \times (s-1)$ cluster members. Since packets from NICE core experience i overlay hops to reach the cluster members, the total number of overlay hops experienced by packets to reach the ith highest group members' cluster members is $i \times (H-i) \times (s-1)^i$.

Based on the above analysis, it can be inferred that the total number of overlay hops in multicast system, $(TOH)_{tree}$, is

$$(TOH)_{tree} = \sum_{i=1}^{H} i \times (H-i) \times (s-1)^i.$$

Therefore, we have

$$O\bar{H}_{tree} = \frac{(TOH)_{tree}}{n} = \frac{\sum_{i=1}^{H} i \times (H-i) \times (s-1)^i}{n}.$$

Define $g(i) = i * (H-i)$, we have $g'(i) = H - 2i$. It shows, when $i = \frac{H}{2}$, $g(i)$ has the minimum value. Therefore, we have

$$\begin{aligned} O\bar{H}_{tree} &\geq \frac{\sum_{i=1}^{H} \frac{H}{2} \times (H - \frac{H}{2}) \times (s-1)^i}{n} \\ &= \frac{H^2}{4n} \frac{(s-1)^{H+1} - (s-1)}{s-2} \\ &\geq \frac{H^2}{4n}[(s-1)^H - 1]. \end{aligned} \tag{9}$$

When n is large enough, we have that $O\bar{H}_{tree}$ is lower bounded by $\frac{H^2(s-1)^H}{4n}$. It shows that $O\bar{H}_{tree} \geq \frac{H^2(k-1)^H}{4n}$ since $s \in [k, 3k-1]$. Q. E. D.

Corollary 3 *For a multicast group with n members, the overlay hop difference of NICE tree is upper bounded by $1 + \frac{7H}{4}$.*

Proof. We first compare $O\bar{H}_{tree}$ and $\frac{H}{4}$. When n is large enough, we have

$$\begin{aligned} \frac{O\bar{H}_{tree}}{\frac{H}{4}} &= \frac{H}{n}(s-1)^H \geq \frac{H}{n}[k + (n-j_1)(k-1)] \\ &= \frac{Hk}{n} + \frac{(k-1)n}{n} - \frac{(k-1)j_1}{n} \geq 1. \end{aligned}$$

The above expression shows that $O\bar{H}_{tree} \geq \frac{H}{4}$. Based on this, ΔOH_{tree} is calculated as

$$\Delta OH_{tree} = O\ddot{H}_{tree} - O\bar{H}_{tree} \leq 1 + 2H - \frac{H}{4} = 1 + \frac{7H}{4}. \tag{10}$$

Q.E.D.

For mesh-based multicast, we achieve the following theorem and corollary.

Theorem 4 *For a multicast group with n members that are mapped into a m-D CAN mesh $d_1 \times d_2 \times ... \times d_m$, where $d_i (i \in [1, m])$ is the number of mesh zones along the i-th dimension, if $d_1 = d_2 = ... = d_m$, the average number of overlay hops of CAN-based multicast is* $\frac{mn(n^{\frac{1}{m}}-1)}{2}$.

Proof. We first calculate the total number of overlay hops of all group members.

Suppose the sender is in the zone $(0, 0, ..., 0)$. Any group member in the m-D mesh zone $(x_1, x_2, ..., x_m)$ has the number of overlay hops $(x_1 + x_2 + ... + x_m)$ to the sender. Hence, the total number of overlay hops of all group members in the m-D mesh is

$$(TOH)_{mesh} = \sum_{x_1=0}^{n^{\frac{1}{m}}-1} \sum_{x_2=0}^{n^{\frac{1}{m}}-1} ... \sum_{x_m=0}^{n^{\frac{1}{m}}-1} (x_1 + x_2 + ... + x_m). \quad (11)$$

To calculate (11), we know each coordinate $x_i (i \in [1, m])$ has the same chance to select a number in $[0, n^{\frac{1}{m}} - 1]$ as its value. In another words, in the view of all dimension coordinates of all group members, each number in $[0, n^{\frac{1}{m}} - 1]$ is used as coordinates by the same times as other numbers in this range. If we denote the times that each number is employed as coordinates by all group members as F, then $(TOH)_{mesh}$ can be expressed by

$$(TOH)_{mesh} = F \sum_{i=0}^{n^{\frac{1}{m}}-1} i = F \frac{n^{\frac{1}{m}}(n^{\frac{1}{m}} - 1)}{2}.$$

We now achieve the times that each number in $[0, n^{\frac{1}{m}} - 1]$ appears in the right expression of (11) (i.e., F). According to the knowledge of Probability and Statistics, since each number has equal chance to be used by coordinates in a m-D mesh, we have $F = mn^{\frac{m-1}{m}}$. Then, it is inferred

$$(TOH)_{mesh} = mn^{\frac{m-1}{m}} \frac{n^{\frac{1}{m}}(n^{\frac{1}{m}} - 1)}{2}. \quad (12)$$

Hence, the average number of overlay hops for mesh topology is

$$\bar{OH}_{mesh} = \frac{mn^{\frac{m-1}{m}} \frac{n^{\frac{1}{m}}(n^{\frac{1}{m}}-1)}{2}}{n} = \frac{m(n^{\frac{1}{m}} - 1)}{2}. \quad (13)$$

Q.E.D.

Corollary 4 *For a multicast group with n members that are mapped into a m-D CAN mesh $d_1 \times d_2 \times ... \times d_m$, where $d_i (i \in [1, m])$ is the number of mesh zones along the i-th dimension, if $d_1 = d_2 = ... = d_m$, the overlay hop difference of CAN-based multicast is* $\frac{m(n^{\frac{1}{m}}-1)}{2}$.

Proof. According to Theorem 4 in [1], we have $\ddot{OH}_{mesh} = mn^{\frac{1}{m}} - m$. And, we have proven in the above theorem that $\bar{OH}_{mesh} = \frac{m(n^{\frac{1}{m}}-1)}{2}$. Therefore, ΔOH_{mesh} is calculated by

$$\Delta OH_{mesh} = \ddot{OH}_{mesh} - \bar{OH}_{mesh} = mn^{\frac{1}{m}} - m - \frac{m(n^{\frac{1}{m}} - 1)}{2}$$
$$= \frac{m(n^{\frac{1}{m}} - 1)}{2}.$$

Q.E.D.

Remark 2: According to Theorem 3, when n is large enough, $\bar{OH}_{tree} \approx \frac{[log_k^n]^2}{4}$. According to Theorem 4, when n is large enough, we have $\bar{OH}_{mesh} = \frac{mn^{\frac{1}{m}}}{2}$. It can be inferred that $\bar{OH}_{tree} < \bar{OH}_{mesh}$. The average number of overlay hops statistically observes the whole multicast system's performance in the number of overlay hops. For the physical meaning, $\bar{OH}_{tree} < \bar{OH}_{mesh}$ shows that tree routing introduces shorter delays than mesh routing. Furthermore, according to corollaries 3 and 4, we have $\Delta OH_{tree} << \Delta OH_{mesh}$ when n is large enough. It indicates that the group members in mesh-based multicast experience larger difference in the number of overlay hops than group members in tree-based multicast. Therefore, under the same network status, members in the mesh-based multicast experience larger delay difference (i.e., delay jitter) than members in the tree-based multicast.

V. Simulation Evaluation

In this section, we use simulations in NS2 [19] to evaluate the analyzed metrics for NICE tree and CAN-based multicast.

Fig. 2 gives the backbone network in the simulations. The backbone network is a combination of two MCI-ISP backbones. In this topology, all nodes are routers. Group members (i.e., end hosts) connect to the backbone network directly or indirectly through some intermediate network components (e.g., hubs). Links in the backbone network have 1000Mbps bandwidth, and links in the local area network have 100Mbps bandwidth. The link propagation delays are 2ms and 1ms for backbone network and local area network respectively. Simulation traffic is 1.5Mbps MPEG-1 video streams. For NICE simulation, we set $k = 3$; and for CAN-based multicast, the program maps group members into a 2-D mesh.

Fig. 3 illustrates the average link stress performance. According to Theorem 1, $\bar{LS}_{tree} \geq \frac{k+k(n-j_1)}{n} = k + \frac{k-j_1}{n}$; and according to Theorem 2, $\bar{LS}_{mesh} = 2m + \frac{1}{n}$. Both $\bar{LS}_{tree}$ and $\bar{LS}_{mesh}$ have

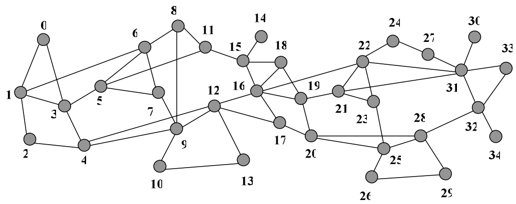

Fig. 2. The experimental backbone network.

stable trend with the change of the number of group members. Further, the average link stress performance has slight decrement with the increasing of n. In this simulation, due to the parameters $k = 3$ and $m = 2$, CAN-based multicast generates a bit larger link stress than NICE tree. In practice, these two multicast protocols may generate very close performance in the average link stress.

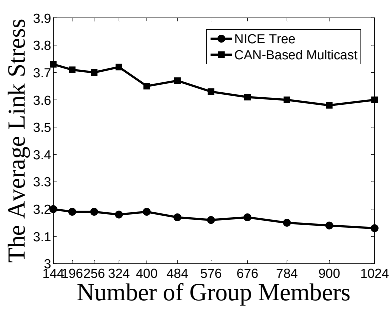

Fig. 3. Average link stress performance of link stress.

Fig. 4 plots the link stress difference curves for NICE tree and CAN-based multicast. As we predicted in the theoretical analysis, NICE tree has larger link stress difference than CAN-based multicast which shows the unbalanced traffic load in NICE multicast systems. As we have analyzed in [1], NICE tree generates a larger worst-case link stress than CAN-based multicast. While for the average link stress, they have very close performance. The smaller average link stress of NICE tree is because that most of NICE members are assigned in the lowest layer in which they need not occupy links to forward packets.

Fig. 5 illustrates the average number of overlay hops. $\bar{OH}_{mesh}$ increases with the increment of n as we predicted in Theorem 4 $\bar{OH}_{mesh} = \frac{m(n^{\frac{1}{m}}-1)}{2}$. $\bar{OH}_{tree}$ increases very slowly with the increment of n which matches our results in Theorem 3 $\bar{OH}_{tree} = \frac{H^2}{4n}(k-1)^H$. In our simulation, $m = 2$ in CAN-based multicast. Therefore, for the groups with $\{144, 196, 256, 324, 400, 484, 576, 676, 784, 900, 1024\}$

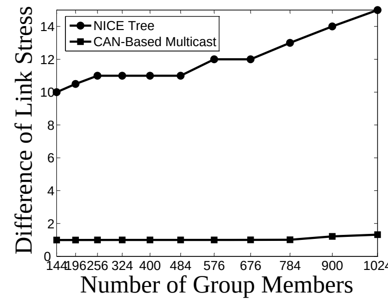

Fig. 4. Link stress difference performance.

members, based on Theorem 4, $\bar{OH}_{mesh}$ should be $\{11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31\}$ respectively. The simulation shows that $\bar{OH}_{mesh}$ are $\{8.3, 10.5, 13, 14.8, 17, 19, 20.5, 22.5, 24, 2628.3\}$ respectively. Simulation results match the theoretical analysis. For tree topology, the simulation set $k = 3$, and constructs $\{2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4\}$ layers for $\{144, 196, 256, 324, 400, 484, 576, 676, 784, 900, 1024\}$ group members. Inputting these parameters into the result of Theorem 3, it can be seen that simulation results prove the theorem. Furthermore, the figure shows that tree topology generates much less average number of overlay hops than mesh topology. It means that packets distributed by tree routing achieve shorter delay performances than packets distributed by mesh routing.

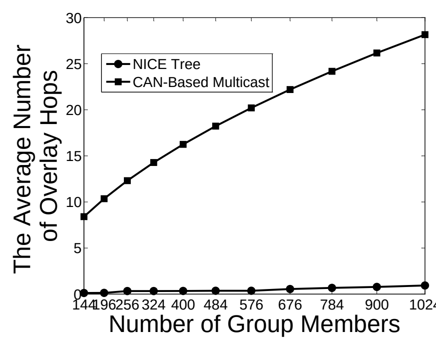

Fig. 5. Performance of the average number of overlay hops.

Fig. 6 plots the curves of the difference in the number of overlay hops for tree-based and mesh-based multicast. If we input simulation parameters into the corresponding equations in Corollaries 3 and 4, it shows that simulation results meet the theoretical results. As we compared in Remark 2, $\Delta OH_{tree} << \Delta OH_{mesh}$ is proved in this figure. The results tell us that tree topology can achieve smaller delay jitter performance than mesh topology.

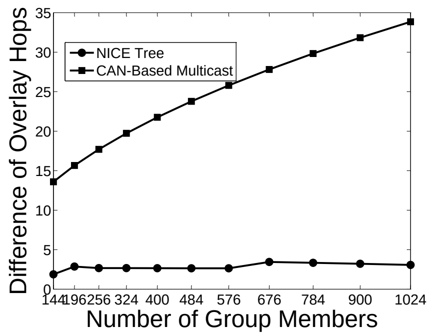

Fig. 6. Performance of the average number of overlay hops.

VI. Conclusion

In this paper, we studied the average performance and the performance difference of the link stress and the number of overlay hops for tree-based multicast and mesh-based multicast. Our theoretical analysis and simulation observation show that mesh-based multicast introduces load balanced transmission, while tree-based multicast is easier to cause bottleneck; mesh-based multicast occupies most of the links in the multicast systems, while tree-based multicast distributes packets through common links on the tree; mesh-based multicast suffers from longer delay transmission, while tree-based multicast is efficient in distributing packets; mesh-based multicast may generate larger delay jitter for packets, while tree-based multicast is good at achieving continuous packet receiving. Based on these properties, we think that tree-based multicast suits to real-time and continuous streaming media transmission, and mesh-based mulitcast may scale to large size groups in multi-source text and non-real-time streaming media applications.

References

[1] W. Tu, Performance Analysis for Overlay Multicast on Tree and M-D Mesh Topologies, in Proc. of IEEE International Conference on Communications 2007 (ICC'07), June 24-28, 2007, Glasgow, Scotland, UK.

[2] Y. H. Chu, S. Rao, S. Seshan, and H. Zhang. A Case for End System Multicast. Distance Vector Multicast Routing Protocol - DVMRP. RFC 1812, June 1995. In Proc. of ACM SIGMETRICS 2000, page 1-12, June 17-21, 2000, Santa Clara, California, USA.

[3] P. FRANCIS. Yoid: Extending The Internet Multicast Architecture. Available at http://www.aciri.org/yoid/docs/index.html, April 2000.

[4] Y. H. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on The Internet Using An Overlay Multicast Architecture. In Proc. of ACM SIGCOMM 2001, page 55-67, August 27-31, 2001, San Diego, CA.

[5] D. Pendarakis, S. Shi, D. Verma, and M.Waldvogel. ALMI: An Application Level Multicast Infrastructure. In Proc. of The 3rd Usenix Symposium on Internet Technologies and Systems (USITS01), page 49-60, March 2001, San Francisco, California.

[6] Y. Chawathe, S. McCanne, and E. A. Brewer. RMX: Reliable Multicast for Heterogeneous Networks. In Proc. of IEEE INFOCOM 2000, page 795-804, March 26-30, 2000, Tel Aviv, Israel.

[7] S. Shi, and J. Turner. Routing in Overlay Multicast Networks. In Proc. of IEEE INFOCOM 2002, page 1200-1208, June 23- 27, 2002, New York, USA.

[8] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-Level Multicast Using Content-Addressable Networks. In Proc. of Of The 3rd International Workshop on Network Group Communication, page 14-29, November 7-9, 2001, London, UK.

[9] S. Ratnasamy, P. Francis, M. Handley, and R. Karp. A Scalable Content-Addressable Network. In Proc. of ACM SIGCOMM 2001, page 161-172, August 27-31, 2001, San Diego, California, USA.

[10] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In Proc. of ACM SIGCOMM, page 205-217, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.

[11] B. Zhang, S. Jamin, and L. Zhang. Host Multicast: A Framework for Delivering Multicast to End Users. In Proc. of IEEE INFOCOM 2002, page 1366-1375, June 23-27, 2002, New York, USA.

[12] M. Castro, P. Druschel, A. M. Kermarrec, and A. Rowstron. SCRIBE: A Large- Scale and Decentralized Application-Level Multicast Infrastructure. IEEE Journal on Selected Areas in communications (JSAC), vol. 20, no. 8, page 1489-1499, October 2002.

[13] A. Rowstron, and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems. Available at http://research.mecrosoft.com/antr/PAST/, 2001.

[14] J. Jannotti, D. K. Gifford, K. L. Johnson, M. Frans Kaashoek, and J. W. OToole Jr. Overcast: Reliable Multicasting with An Overlay Network. In Proc. of The 4th Usenix Symposium on Operating Systems Design and Implementation, page 194-210, October 22-25, 2000, Paradise Point Resort, San Diego, California, USA.

[15] S. Banerjee, C. Kommareddy, B. B. K. Kar, and S. K huller. Construction of An Efficient Overlay Multicast Infrastructure for Real-Time Applications. In Proc. of IEEE INFOCOM 2003, page 1521-1531, 30 March 30-April 3, 2003, San Francisco, USA.

[16] A. Riabov, and L. Z. Zhen Liu. Overlay Multicast Trees of Minimal Delay. In Proc. of The 24th International Conference on Distributed Computing Systems (ICDCS 2004), page 654-661, March 23-26, 2004, Tokyo, Japan.

[17] W. Tu, and W. Jia. A Scalable and Efficient End Host Multicast for Peer-to-Peer Systems—DSCT. In Proc. of IEEE Global Telecommunication Conference 2004 (Globecom 2004), November 29-December 3, Dallas, Texas USA, IEEE Computer Society Press.

[18] W. Tu, and W. Jia. An End Host Multicast Protocol for Peer-to-Peer Systems. In Proc. of The 30th Annual IEEE Conference on Local Computer Networks (LCN 2005), November 2005, IEEE Computer Society Press.

[19] UC Berkeley, LBL, USC/ISI, and Xerox PARC. Ns Notes and Documentation. October 20, 1999.