ON THE PERFORMANCE OF AN APPLICATION LAYER MULTICAST PROTOCOL

Xiaobing Hou, Central Connecticut State University; Shuju Wu, Central Connecticut State University

Abstract

In this study, the performance of an application layer multicast protocol, namely Adaptive Overlay Multicast (AOM) protocol was evaluated. In this paper, the authors introduce the concepts of fan-outs and foster limits in building application layer multicast trees that exemplify the performance and adaptability characteristics of AOM and network dynamics with extensive simulation results.

Introduction

Overlay multicast (also called application layer multicast) was proposed to provide multicast service at the application layer using P2P connections, thereby removing the dependence on multicast support of the underlying networks [1-4]. Operations such as membership management and routing are implemented at the application layer, and data distribution is over a multicast tree that consists exclusively of end hosts and unicast connections. Both IP and overlay multicast use a tree to achieve distribution efficiency, which results in performance clusters when tree links or nodes are under fault. A performance cluster includes those members that suffer the performance degradation caused by the same fault. The larger the cluster size, the more the group communication is jeopardized. Therefore, tree construction and fault adaptation are important.

The Internet is dynamic and unpredictable in nature. Dynamic events such as group membership changes, node failures, link failures or network congestion can cause the quality of an overlay multicast tree to degrade over time. Any such events are considered faults. A fault caused by dynamic group membership or node failure is easier to detect than the others and the effect on application performance is temporal. However, faults caused by network congestion in the Internet could last much longer. They also cause end-to-end performance degradation without a total loss of connection and, thus, cannot be detected by simple node failure detection mechanisms. Experiments on the MBone [5] have shown that even for a small multicast group of 11 members, each member experiences a very long consecutive loss of up to a few minutes, a situation that occurs in almost every trace. Link loss rates in an MBone group of eight members are measured in one-hour intervals and have been shown to vary between 2% and 35% [6]. On a specific link, loss rates higher than 15% occur frequently and often last about 10 minutes. Also, from the results of other Internet measurements [7], [8], it is not unusual to find long-lasting high-loss periods between Internet nodes, although the average loss rate over a day could be low. When such faults happen in a multicast tree and are close to the multicast source, the size of the performance cluster will be large, which adversely affects most of the group members.

Multicast tree-building algorithms employed by different overlay multicast protocols exhibit different scalability and adaptability characteristics under network dynamics during the multicast session time. A protocol may build a well-formed initial overlay multicast tree under stable network conditions, but may not be able to sustain the application performance in the presence of underlying network perturbations. Multicast protocol performance has been addressed in the context of traditionally reliable IP multicast [9]. Most of the previous overlay multicast protocols focused on the construction of overlay multicast trees. Therefore, the adaptation performance to network dynamics is either passive and limited [3], [4], [10] or not scalable [1], [2]. The adaptation is passive and limited because, although a member periodically looks for a new parent in the tree, it does not use end-to-end performance as a guide and thus may not help end-to-end application performance.

Previously, an Adaptive Overlay Multicast (AOM) that employed both end-to-end and local metrics to build the overlay multicast tree was proposed [11], [12]. Here, however, the concepts of fan-outs and foster limit in AOM are introduced and their effects on the quality of the application layer multicast tree are evaluated. Also presented here are AOM and its fault adaptation algorithm, a simulation study on AOM with different fan-outs, and a comparison of tree quality with a well-know application layer multicast tree protocol.

Related Work

The previous overlay multicast studies focused on self-organizing the group members into a delivery tree and classifying them into centralized, distributed direct-tree and distributed mesh-first approaches. ALMI [13] takes a centralized approach where a central controller builds the overlay and disseminates the tree information to the group members. The NARADA [2] and Gossamer [1] protocols build a mesh first and run a DVMRP-like routing protocol to build the tree. Other protocols like NICE [10] and YOID [4] build the tree directly; i.e., the tree is extended when a new joining member connects to an existing member. All these protocols use Round Trip Time (RTT) as the building metric. HostCast [14] utilizes the shortest end-to-end delay in path finding, but no effort is given to match the overlay multicast tree to the optimized IP multicast tree. None of the above protocols has investigated the loss adaptation issue.

The Host Multicast Tree Protocol (HMTP) [3] is a typical direct-tree protocol using RTT as the only metric. A new member moves as far as possible from the ROOT only if it finds a potential parent closer than the current one, and its RTT to the current parent is longer than the potential parent's RTT to its current parent. Periodically, a member randomly selects another member in its path to the ROOT (or ROOT path) and explores the branch under that member for a new parent. The periodic level-by-level exploration and probing among members accounts for most of the overhead in HMTP.

Recent tree-building approaches include closest-first-searching (CFS) [15], adjacency matrix [16] and minimum diameter spanning tree [17]. Zhang et al. use an approach similar to HMTP, except that a member tries to remember different branches in the tree building process [15]. The objective is to extend the searching range of a node position so that nearby nodes have a better chance of staying together in the tree. The algorithm itself does not consider tree maintenance or issues of adaptability. Mourad and Ahmed rely on an adjacency matrix to build the multicast tree, where matrix information is provided by the underlying P2P architecture [16]. Their application, then, must be tied into a P2P network. Moreno-Vozmediano takes a centralized approach where the multicast source node collects the probing results from every grid node and calculates the minimum spanning tree for multicast file distribution [17]. This approach is expected to be adaptive to network dynamics if continuous probing is applied; however, scalability is the main shortcoming of this centralized approach.

Chu et al. studied the overlay multicast protocol in dynamic network environments [18]. Their experiments were carried out on a mesh-first protocol, NARADA, with the results showing that it is important to adapt delay and bandwidth for conferencing applications. In this current study it was felt that it is also necessary to study the dynamic adaptation in direct-tree protocols, first because direct-tree protocols do not have an explicit multicast routing protocol—as in the case of NARADA and Gossamer—to distribute helpful information for the adaptation and, second, because one of the objectives of direct-tree protocols is scalability. A transient study can help analyze whether a protocol is scalable by adapting it to network faults efficiently and on time.

A simple, best-effort approach for improving the data delivery ratio under dynamic network conditions was recently studied in Probabilistic Resilient Multicast (PRM) [19]. The idea is that in addition to forwarding the normal data along the multicast tree, each member randomly chooses a constant number of other members and forwards the new data to each of them with a certain probability. Random forwarding sends duplicate packets to the members that are fault-free, while providing passive loss recovery at the faulty locations. PRM is not a multicast tree protocol but is a best-effort approach for improving the data delivery ratio in an overlay multicast.

An Adaptive Overlay Multicast Approach

In the following sections, fan-outs, foster limit and tree adaptation are reviewed as they relate to AOM. Additional details can be found in a study by Wu et al. [12].

AOM Tree Protocol

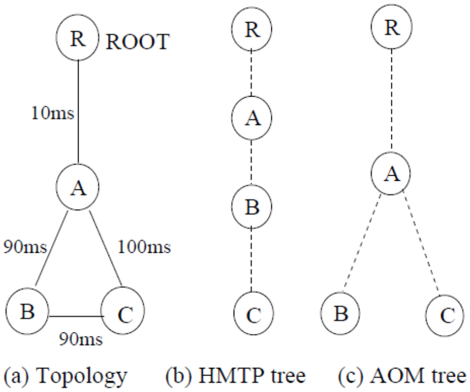
For scalability, the AOM tree protocol takes the direct-tree approach. The tree protocol fulfills the following tasks: tree formation, tree improvement, membership management, loop avoidance, detection and resolution. Most of the previous tree construction protocols only use Round Trip Time (RTT) to local neighbors (referred as local RTT) to connect the members. AOM, on the other hand, uses both End-to-End Delay (EED) to the ROOT and RTTs between the members to determine how to construct the tree. In Equation (1), a member’s (*i*) EED to the ROOT is defined as the sum of its parent's EED and half of the RTT between the member and its parent.

$$EED_{i,ROOT} = EED_{i'sparent,ROOT} + 0.5 \cdot RTT_{i,i'sparent} \quad (1)$$

A member measures its RTT to another member by periodically sending PROBE messages. The measurements are smoothed with exponential averaging. To calculate the EED, a parent puts its current EED in the PROBEREPLY message (the ROOT's EED is 0) and a child updates its own EED, as defined in Equation (2).

$$EED_{new} = EED_{parent} + 0.5 \cdot RTT_{new} \quad (2)$$

A member joins the group by sending a JOIN message to the ROOT. If the ROOT can accommodate the new member, it sends an ACCEPT message with the information of its current children. The new member then starts looking for the most suitable parent. In an AOM tree protocol, the more suitable parent for a member is the one that is closer to the member than the current parent (i.e., smaller RTT), is closer to the ROOT than the member (i.e., smaller EED), and through which the member's new EED is not penalized too much.



(a) Topology (b) HMTP tree (c) AOM tree

Figure 1. Example of Multicast Trees

The reason that AOM uses both EED and RTT as metrics is that EED reflects the *vertical* distance of a member to the ROOT, while RTT reflects the *local* distance between a member and its neighbor; both of them need to be considered to provide the best match between the tree and the underlying network topology. Figure 1a shows a simple network topology; Figure 1b shows a tree built by HMTP [3] that uses only RTT as the metric; Figure 1c shows a tree built by AOM, assuming the joining sequence of A, B and C.

To limit the joining overhead, a member looks for a new parent from a potential set of parents. Assuming that the ROOT is at level 0, a member is at level i if its parent is at level $i-1$. In such a case, they are said to be 1 overlay hop away from each other. The potential set of a level i member is { $member_{l,h}$ }, where $l=i$ and $h<=4$; l is the level and h is the number of overlay hops from this member. The potential set is obtained by probing the ancestors. If a level i member finds a new parent in the potential set, its level becomes $i+1$. Parent searching continues until no new parent can be found. A nice property of such a potential set ($h<=4$) is that when a member initially joins the group and is at level 1 or level 2, it has the opportunity to explore its position in all of the tree branches; when it moves further down the tree, however, the searching is limited to sub-branches.

Due to independent joining sequences and dynamic membership, it is necessary for the members to periodically re-evaluate their positions and continue to optimize the tree structure after joining the group. Since topologically close members are likely to stay close in the overlay by using both EED and RTT metrics, tree improvement is carried locally, i.e., a member only contacts its ancestors for improvement to reduce overhead. The ancestor set of a level i member is { $member_{l,h}$ }, where $l \in \{ i, i-1, i-2 \}$ and $h \in \{1,2,3\}$.

A single member leaving will cause the tree to become partitioned. Therefore, before a member leaves the group, it notifies its parent and children. Each child then chooses the closest ancestor (minimum RTT) or ROOT as new parent. Partitions caused by an unexpected member or link fault are detected either by the fault adaptation algorithm or by continuous loss of the PROBEREPLY messages. The simplest way to resolve the loop is to let each member attach its ROOT path information in the PROBE and PROBEREPLY messages. A member detects the loop by finding itself in the middle of its ROOT path and breaks the loop by re-joining the ROOT.

Performance Monitoring and Fault Detection

The previous direct-tree protocols, including HMTP, do not actively monitor end-to-end performance metrics. Therefore, they adapt only to local delay conditions as dictated by RTT increases. In AOM, a member monitors the performance of not only its current ROOT path, but also the paths through its ancestors (backup paths). Therefore, when a fault happens on the ROOT path, the member can select a backup path with better performance for its performance cluster. Currently, end-to-end delay and end-to-end loss rates are used as performance metrics for AOM.

A member monitors the EED on its ROOT path by periodically probing its parent. The EEDs on the backup paths are measured in the same way but less frequently because no other important information is exchanged on these paths. To prevent problems of instability, periodic measurements are smoothed with exponential averaging.

Since the ROOT path is used for data distribution, its loss rate, $l_{m,ROOT}$, can be measured by the application data. Loss rate on a backup path $l_{m,a,ROOT}$ is calculated as

$$l_{m,a,ROOT} = 1-(1-l_{a,ROOT})*(1-l_{m,a})$$

where $l_{a,ROOT}$ is the loss rate on the ancestor a 's ROOT path and $l_{m,a}$ is the loss rate on the overlay link between the member and the ancestor, a . Since there is no application data on this link, the member asks the ancestor to periodically send a test packet.

Loss measurement of $l_{m,a}$ is a variation of the Average Loss Interval (ALI) method [20]. ALI is a better loss rate estimator than the Dynamic History Window (DHW) used by RON (resilient overlay networks) [21] and the Exponentially Weighted Moving Average (EWMA). ALI properly considers the effects of both recent and earlier loss events. Like ALI, AOM uses the weighted average loss over a few measurement intervals; but unlike in ALI, where the intervals are decided by every single loss event, the intervals in AOM are of equal lengths. This is because the two methods serve different purposes: ALI works for TCP-friendly congestion control and expects the sender to respond to every loss event; the purpose of this current study, however, was to detect the loss rate over a period of time. For details on the algorithm, please refer to the study by Wu [12].

Fault Adaptation

Without an adaptation algorithm, the only chance for protocols such as HMTP to recover from an EED fault is periodic improvement, where RTT is used to look for a closer member. Therefore, a member can bypass the EED fault if it finds a closer member not suffering the fault. Since a closer member is not necessarily an EED fault-free member, the result is random. In addition, a faulty link may affect many members' EEDs but not local RTTs, resulting in no switching efforts at all. In AOM, once the EED fault is detected, a member actively probes the ancestors for the most up-to-date EEDs and loss rates, and starts the fault adaptation algorithm, as summarized in Table 1.

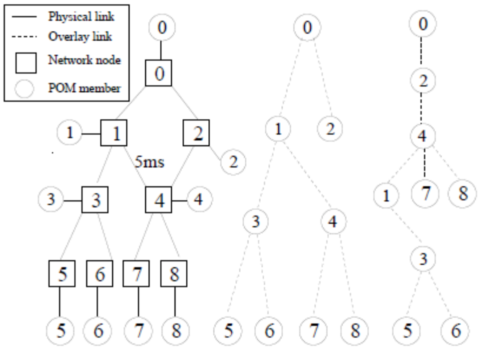
It is worth pointing out the difference between tree improvement and fault adaptation. Both of them involve looking for new parents. However, the tree improvement process creates a more efficient tree, while the fault adaptation process satisfies the performance requirement.

Benefits of End-to-End Performance Monitoring

In this section, the benefits of using EED over RTT alone to adapt to network faults are presented. For better clarification, both cases are simulated in AOM. However, the case of using a local metric will apply to other protocols like HMTP. Figure 2 shows a 9-node network topology, the overlay multicast tree before the fault happens and the new overlay multicast tree when end-to-end delay is used as a fault adaptation metric in AOM. Every physical link has a delay of 10ms with the exception of link 14 which has a delay of 5ms. This is to ensure that, initially, member 4 selects member 1 as its parent in the tree. At a time of 50 seconds into the simulation, the delay of link 01 increases to 1.2 seconds, causing members 1, 3, 4, 5, 6, 7 and 8 to suffer large end-to-end delays. The total simulation time is 180 seconds.

Table 1. Fault Adaptation Algorithm

```
1. On detection of faults at member $m$ :
   probe $S_a = \{ROOT, ancestors\}$ for RTTs, EEDs and loss rates.
2. Wait for reply, then update performance metrics
   through ancestor $a$ in $S_a$ as:
     $RTT_{new} = \beta \cdot RTT_{new} + (1-\beta) \cdot RTT_{old}$;
     $EED_{m,a,ROOT} = EED_{a,ROOT} + RTT_{new,a}/2$;
3. Add $a$ to potential parent list $pl$ if:
     $EED_{m,a,ROOT} < scale \cdot EED\_LIMIT$ &&
     $l_{m,a,ROOT} < scale \cdot LOSS\_LIMIT$
4. Find closest potential parent:
     $cur\_potential\_parent = min\_rtt(pl)$;
   if $cur\_potential\_parent == NULL$ , adaptation fails, end.
     $pl = delete(pl, cur\_potential\_parent)$;
   join( $cur\_potential\_parent$ ).
5. If not accepted by $cur\_potential\_parent$ , go to step 4.
```



(a) Topology (b) initial tree (c) after fault

Figure 2. A 9-node Network (a), Initial Tree (b), Tree After Adaptation (c)

Figure 3 shows the simulation results of members 1, 4 and 6. It can be seen that member 4 (and, thus, its children 7 and 8) changes its path before member 1 and recovers from the fault by attaching to member 2. Member 1 could not adapt to the fault by itself as the underlying routing algorithm happens to use the faulty link to probe member 2. However, member 4 invites member 1 after it switches its sub-cluster to member 2. From the figure, it can also be seen that member 6 adapts to the fault. However, this is not because member 6 selects a new path but rather because its grandparent, member 1, changes the ROOT path to a better position. Therefore, the sub-cluster (5 and 6) recovers from the fault without extra probing and adaptation overhead. Conversely, Figure 4 shows that if only RTT is used as the performance metric, none of these members (1, 4 or 6) could adapt to the delay fault, in spite of the existence of better paths, and the multicast tree would not change.

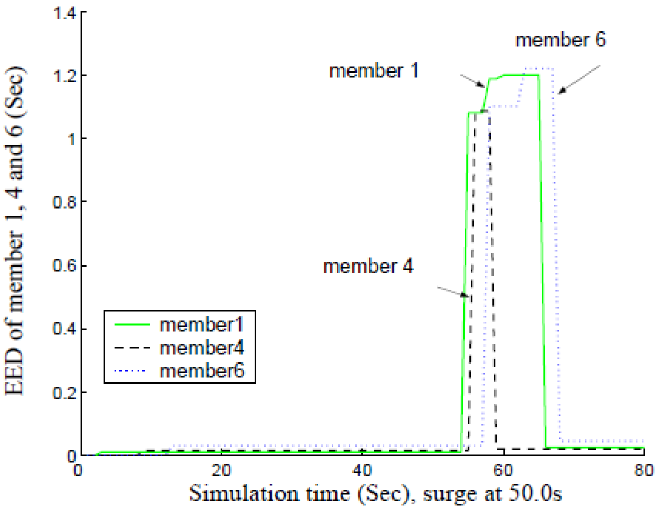


Figure 3. Adaptation of EED using End-to-End Metric

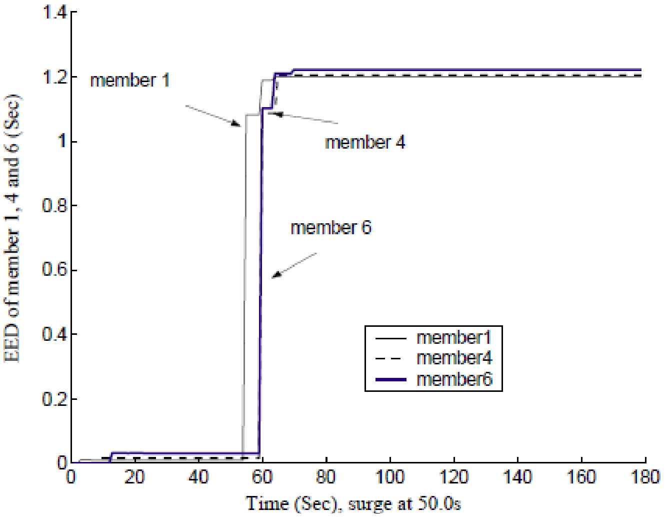


Figure 4. Adaptation of EED using Local RTT

Fan-Outs and Foster Limit

End-hosts over the Internet are heterogeneous; thus, the maximum number of unicast connections that can be set up to forward the application data depends on factors such as bandwidth capacity, traffic load and host processing power, and may vary from time to time. This connection limit is called a fan-out limit.

A member's fan-outs should be those that are best for the tree quality. If a new connection request is simply refused when the fan-out size reaches its limit, the resulting tree may be of inferior quality. To solve this problem, the connections accepted by a member are classified as fan-out connections and foster connections. Fan-out connections are used to forward application data and are restricted by the fan-out limit. Foster connections are used to construct the tree and are restricted by a foster limit. If a connection request cannot be treated as a fan-out connection, it is accepted as a foster connection for a period of time. Since the control packets used for tree construction are of small size and are sent much less frequently than data packets, a member can manage many more foster connections.

During the fostering period, several changes could occur. First, the new child may find a more suitable position in the current tree branch and move down. Second, due to periodic tree improvement, an existing child may move or become the child of a new member. Third, if none of the fan-outs or the foster child finds a better position, the tree stays unchanged. For the third case, the foster child or a fan-out will be forced to move away depending on its RTT to the parent. During the fostering period, a foster child can receive the application data from its old parent or from a randomly selected member if it is in the initial joining period. Results in the next section show how foster connections improve tree quality.

Effects of Foster Limit on Protocol Performance

This section presents the effects of foster limit on AOM performance and compares them with HMTP [3], a typical direct-tree protocol. First, tree quality is evaluated in randomly generated 1,000-node transient-stub network topologies. In this part, network conditions are static in that 1) link delays are pre-assigned and do not change during the simulations, and 2) the members do not leave the multicast group during the simulation period. The second part focuses on the adaptability of the two schemes. Faults like delay and loss-rate surges are added to randomly selected links in order to observe how the schemes respond to such events.

Performance Metrics

The following metrics are used to evaluate the tree quality:

- Group EED ratio: the group members' EED ratios are averaged. A member's EED ratio is the ratio of its overlay EED and its EED in the Shortest Path Source Tree (SPST) in an IP multicast. This metric measures the increase in EED in an overlay multicast.
- Average link stress: assuming that $LS(i)$ is the number of duplicate packets on a link i, the average link stress is defined as: $\sum_{i,LS(i)>=1} LS(i)$ / $\sum_{i,LS(i)>=1} 1$ and reflects the load added to a link by an overlay multicast protocol.
- PDF for link stress: Link stress shows the distribution of the stress over the physical links as well as the most stressed link.
- CDF for path length: path length is defined as the number of physical links (hops) in a member's ROOT path. A longer path is not desirable because it often implies a larger EED and certainly more processing overhead.
- Tree cost: This is the total number of physical links used by the tree. Tree cost ratio is defined as the ratio of overlay multicast tree cost to the corresponding SPST tree cost.
- Control overhead: This is the total amount of control traffic used to build and maintain the tree.

Performance Evaluation

The simulation was implemented using Network Simulator-2 [22]. The 1,000-node transient-stub network topology was randomly generated by GT-ITM [23]. For simplicity, links are assigned symmetric random delays. Since both AOM and HMTP use RTT to estimate the delay between two members, symmetric delay does not favor either of them. The simulation results reflect the average of 10 runs with a C.I. of 95%, except for those that describe the transient behavior. Table 2 summarizes the values of the simulation parameters.

Table 2. Simulation Parameters

Parameters	AOM	HMTP
Improvement period	60 seconds	30 seconds
Foster limit	0, 50	0
Join time	uniform(0,1500) seconds	
Simulation time	2500 seconds	
Fan-out limit	10	
Group density	5%, 8%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%	

Figure 5 shows that for a group size ranging from 50 to 900, an AOM member has, on average, a much smaller EED than it would in HMTP. This is due to the fact that the tree algorithm in AOM considers not only the RTTs but also the EEDs to the ROOT. A member in AOM has limited tolerance to increase its EED. This avoids the long paths that could occur in HMTP, as will be shown later. With the exception of when the group size is 50, the group EED ratio of AOM is at least 60% less than that of HMTP. Another observation is that for a group size of 50 to 900, the group EED ratio in AOM remains low and stable, while in HMTP it increases by 60%. This means that the AOM tree matches the underlying network topology better and the AOM scheme is more scalable. It is also shown in Figure 5 that fostering a few members for the purpose of tree construction improves the EED ratio. At a large group size, fostering 50 children in AOM decreases the EED ratio by 15% over no fostering.

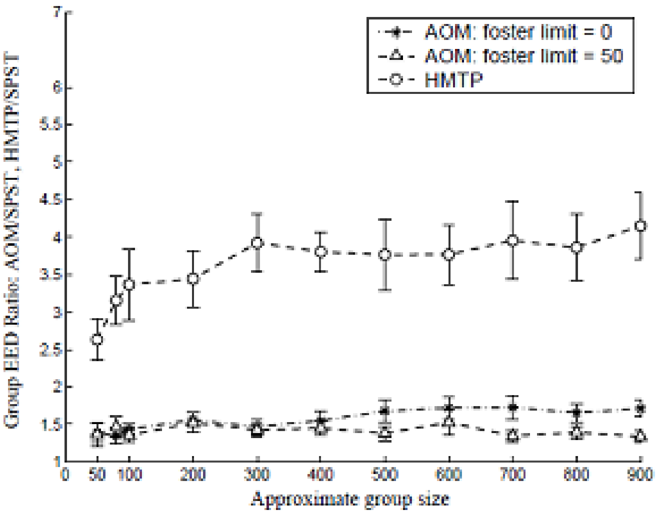


Figure 5. Tree Quality: Group EED Ratio

Unlike in an IP multicast tree where every link has a stress of 1, some links in an overlay multicast tree have duplicate packets. The link stress is affected by the group size and whether the tree matches the underlying network topology. Figure 6 shows the average link stress of HMTP and AOM. In both schemes, average link stress increases with group size. This is because the more members there are, the more likely that some links will be used repeatedly. When the group size is small, HMTP builds lower-stress trees than AOM. However, in large groups, AOM with 50 foster children outperforms both HMTP and AOM with no foster children by up to 17.36%.

Figure 7 shows the pdf of the link stresses collected from 10 simulation runs. The group density of the simulations was 85%. The largest link stress in the figure is the largest link stress to appear in the 10 simulation runs. It can be seen that almost 99% of the links in both schemes have a stress number of less than 7; however, there are a few heavily stressed links in each scheme. In a group of about 800 members, the worst link stress to appear in 10 runs was 28 in AOM for both foster limits. HMTP had a smaller worst-link stress of 22.

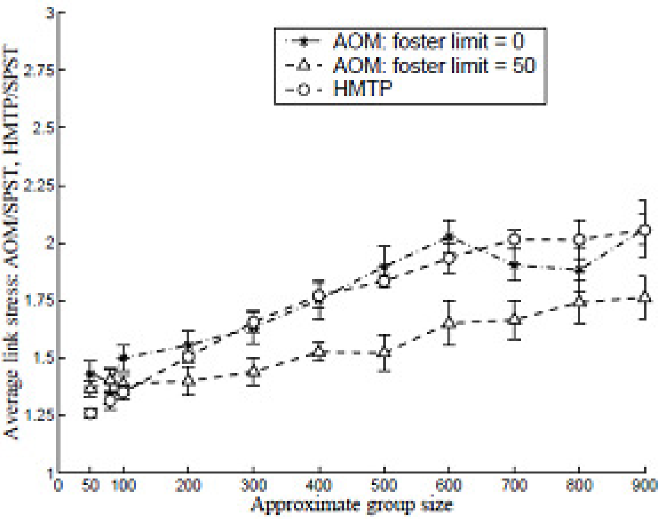


Figure 6. Tree Quality: Average Link Stress

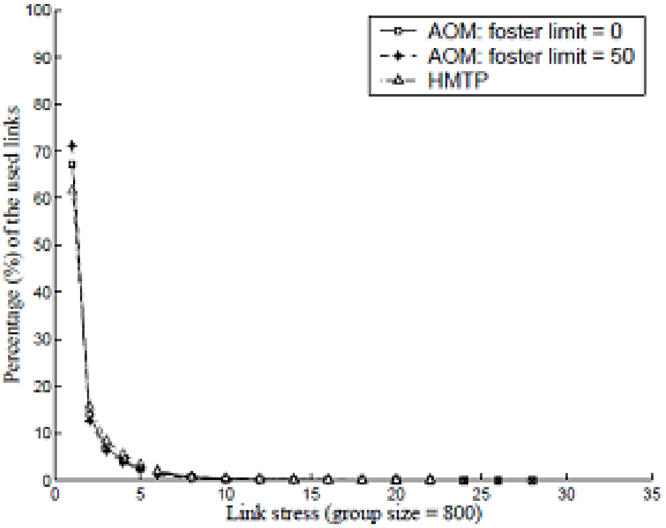


Figure 7. Tree Quality: *pdf* of Link Stress

An ideal overlay multicast tree should provide short ROOT paths and low link stress. However, it is difficult to achieve both objectives at the same time. One extreme is the multiple unicasts tree in which the paths are short but the link stresses are high. Another extreme is that the path is extremely long, but the link stress is low. Often, longer paths result in larger EEDs.

Figure 8 shows the CDF of the ROOT path length in AOM, HMTP and multiple unicasts. Path length is the number of physical links involved in a member's ROOT path. In the figure, the path lengths of all the members in a group were collected from 10 simulation runs with a group density of 85%. As can be seen, the multiple unicasts tree had the shortest path length, which was also the lower bound of the overlay multicast tree. The path length of AOM was moderate because the longest path length was 60 hops when the foster limit was 0. At least 30% of the HMTP members have a ROOT path longer than the longest path in AOM.

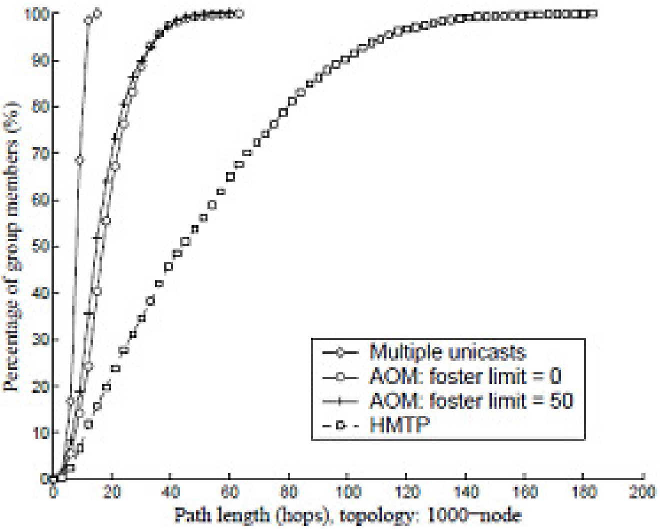


Figure 8. Tree Quality: CDF of Path Length

Tree cost reflects the total resources consumed by the overlay multicast group, such as bandwidth and processing power. Figure 9 compares the average tree cost of AOM and HMTP. For each group size, the result was the average of 10 trees and was normalized by the cost of the corresponding SPST. As can be seen in large groups, fostering children in building multicast trees in AOM saves 20%-32% more network resources than HMTP.

Figure 10 shows the change of the tree cost in a typical run of each scheme. About 800 members join the group in the first 1500 seconds. At the initial phase, the tree cost increases rapidly. After all of the members join the group, the tree cost begins to decline as the improvement algorithm continues to work. It can be seen that AOM with a foster limit of 50 had the smallest tree cost and fastest convergence.

Both AOM and HMTP need control packets to build and maintain the tree. In HMTP, control traffic is used to refresh information between parents and children, measure the RTTs and query for the information used by the tree improvement algorithm; control traffic in AOM is used to exchange information between a member and its ancestors. In the simulations, 40-byte control packets were used.

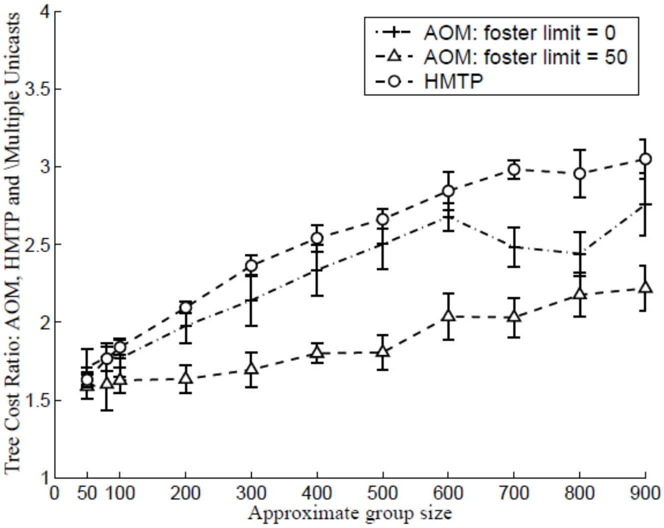


Figure 9. Tree Quality: Cost Ratio

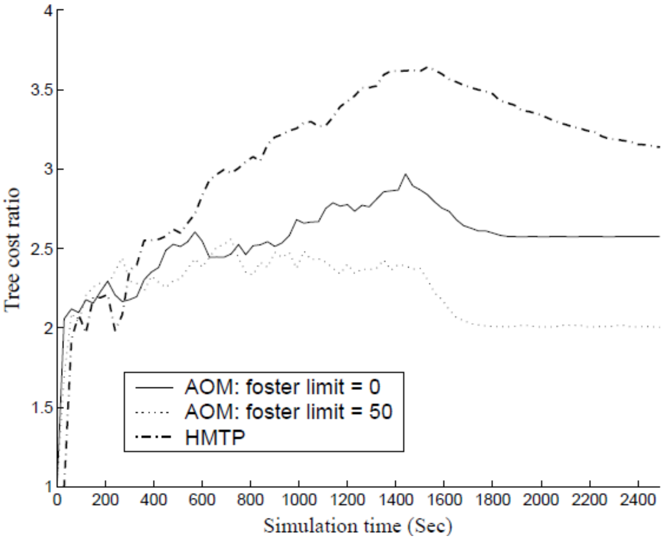


Figure 10. Convergence of Tree Cost Ratio

Figure 11 shows that the control traffic load added by the two schemes increased with the group size. At a group size of 900, the control traffic load reached 30KB/sec in HMTP but was 20% lower in AOM with a foster limit of 50 and 40% lower without fostering. Such control traffic load is not large in the sense that it is distributed across the entire network rather than on a single link. AOM with a foster limit of 50 incurs more control traffic than with no fostering.

Conclusions

End-hosts over the Internet are heterogeneous; therefore, the maximum number of unicast connections that can be set up to forward the application data (called fan-outs) depends on factors such as bandwidth, traffic load and host processing power, and may vary from time to time. This study showed that appropriate fan-out limits and the foster limit (temporary connections used for multicast tree construction rather than application data distribution) can improve overall tree quality and application performance.

Multicast applications have different performance requirements. For example, media streaming applications are sensitive to delay, loss and available bandwidth; content distribution, such as server replication and large software distribution, can be loss intolerant; a delay jitter requirement must be satisfied in voice applications. Therefore, future work with AOM should consider more end-to-end performance metrics such as available bandwidth and the jitter requirement.

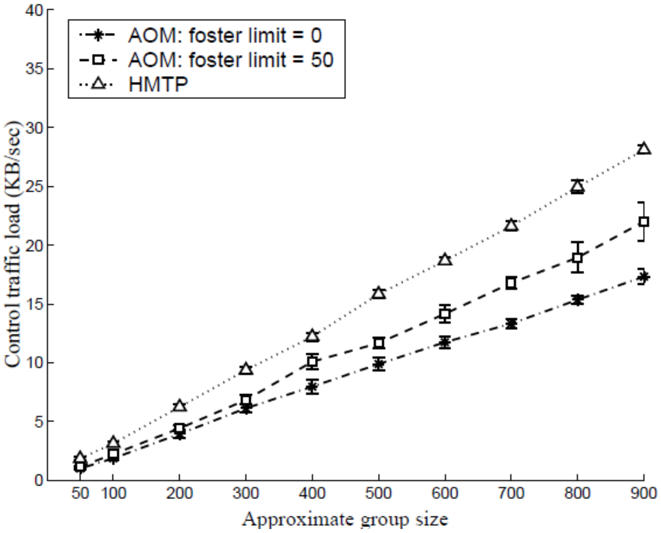


Figure 11. Overhead Traffic Load Added by the Schemes

References

[1] Chawathe, Y., McCanne, S. & Brewer, E. (2000). *An architecture for Internet content distribution as an infrastructure service*. Unpublished report available at http://www.cs.berkeley.edu/yatin/papers.

[2] Chu, Y. Rao, S. G. & Zhang, H. (2002). A case for end system multicast. *IEEE Journal on Selected Areas in Communication*, 20(8), 1455-1471.

[3] Zhang, B., Jamin, S. & Zhang, L. (2002). Host mul-

ticast: a framework for delivering multicast to end users. *Proceeding of IEEE INFOCOM*.

[4] Francis, P. (2000). *Yoid: Extending the Multicast Internet Architecture*. White paper available at http://www.aciri.org/yoid/.

[5] Yajnik, M., Kurose, J. & Towsley, D. (1996). Packet loss correlation in the mbone multicast network. *Proceeding of IEEE Global Conference on Communications*.

[6] Cáceres, R., Adams, A., Bu, T., Duffield, N. G., Friedman, T., Horowitz, J. et al. (2000). The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior. *IEEE Communications*, 38(5).

[7] Labovitz, C., Ahuja, A. Bose, A. & Jahanian, F. (2000). Delayed internet routing convergence. *Proceeding Of ACM SIGCOMM*, (pp. 175-187).

[8] NLANR active measurement project. (n.d.). Retrieved February 1, 2006, from http://watt.nlanr.net/ .

[9] Birman, K. P., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M. & Minsky, Y. (1999). Bimodal multicast. *ACM Transactions On Computer Systems*, 17(2), 41-88.

[10] Banerjee, S., Bhattacharjee, B. & Kommareddy, C. (2002). Scalable application layer multicast. *Proceedings of ACM SIGCOMM*.

[11] Wu, S. & Banerjee, S. (2004). Improving the Performance of Overlay Multicast with Dynamic Adaptation. *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV.

[12] Wu, S., Banerjee, S., Hou, X. & Thompson, R. A. (2004). Active Delay and Loss Adaptation in Overlay Multicast. *Proceeding of IEEE International Conference on Communications*.

[13] Pendarakis, D., Shi, S., Verma, D. & Waldvogel, M. (2001). ALMI: an application level multicast infrastructure. *Proceedings of the 3rd USENIX Symposium on internet technologies and systems (USITS)*.

[14] Li, Z. & Mohapatra, P. (2003). HostCast: A New Overlay Multicasting Protocol. *Proceeding of IEEE International Conference on Communications*.

[15] Zhang, X., Li, X., Wang, Z. & Yan, B. (2008). A Delivery Tree Building Approach for Application Layer Multicast. *International Symposium on Computer Science and Computational Technology*, (pp. 630-634).

[16] Mourad, A. & Ahmed, M. (2008). A Scalable Approach for Application Layer Multicast in P2P Networks. *Pervasive Computing and Communications, IEEE International Conference*, (pp. 498-503).

[17] Moreno-Vozmediano, Y. R. (2009). Application Layer Multicast for Efficient Grid File Transfer. *International Journal of Computer science and Applications*, 6(1), 70-84.

[18] Chu, Y., Rao, S. G., Seshan, S. & Zhang, H. (2001). Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture. *ACM SIGCOMM*, San Diago, CA.

[19] Banerjee, S., Lee, S., Bhattacharjee, B. & Srinivasan, A. (2003). Resilient Multicast Using Overlays. *ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, San Diego, CA.

[20] Floyd, S., Handley, M., Padhye, J. & Widmer, J. (2000). Equation-based Congestion Control for Unicast Applications. *Proceeding of ACM SIGCOMM* (pp. 43-54).

[21] Andersen, D., Balakrishnan, H., Kaashoek, F. & Morris, R. (2001). Resilient Overlay Networks. *Proceeding of the 18th ACM SOSP*, Banff, Canada.

[22] The Network Simulator ns-2. (n.d.). Retrieved May 20, 2009, from http://www.isi.edu/nsnam/ns/.

[23] GT-ITM: Georgia Tech Internetwork Topology Models. Retrieved June 18, 2009, from http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html.

Biographies

XIAOBING HOU is currently an Assistant Professor at the Computer Electronics and Graphics Technology Department at Central Connecticut State University. He received the Ph.D. degree in Information Science from the University of Pittsburgh in 2006. Dr. Hou's teaching and research interests are in the area of computer networking and information security. He is a member of IEEE and ACM. Dr. Hou may be reached at xhou@ccsu.edu

SHUJU WU is currently an Associate Professor at the Computer Electronics and Graphics Technology Department at Central Connecticut State University. She received her Ph.D. degree in Information Science from the University of Pittsburgh in 2004. Dr. Wu's teaching and research interests include computer communications and networks, multimedia systems, performance modeling and evaluation, and network applications. Dr. Wu may be reached at swu@ccsu.edu