

PMTA: Potential-based Multicast Tree Algorithm with Connectivity Restricted Hosts

Xiaohui SHI¹, Yang CHEN¹, Guohan LU¹, Beixing DENG¹, Xing LI¹, Zhijia CHEN²

¹Department of Electronic Engineering, Tsinghua University, Beijing 100084, P. R. China

²Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P. R. China
sxh@mails.tsinghua.edu.cn

Abstract — A large number of overlay protocols have been developed, almost all of which assume each host has two-way communication capability. However, this does not hold as the deployment of firewalls and Network Address Translators (NAT) is widespread in the current Internet, which is a challenge to the design and implementation of overlay models and protocols. In this paper, we present Potential-based Multicast Tree Algorithm (PMTA) to enhance the multicast tree construction in presence of connectivity restricted hosts. We evaluate PMTA and previous multicast tree protocols based on real Internet end-to-end delay datasets. According to evaluation results, PMTA outperforms those protocols in terms of all metrics. PMTA reduces *ARDP* by 26%, and it also results in 23%-54% reduction in average overlay latencies. As the results suggest, PMTA can build efficient and effective multicast tree and is suitable for Internet multicast applications in the presence of connectivity restricted hosts.

I. INTRODUCTION

Nowadays, network overlays have become an increasing mainstream in distributed network services and applications, such as overlay network multicast [1-4, 11, 16, 19], peer-to-peer file sharing [7-10] and network security [18]. There are lots of literature on the design, evaluation and implementation of network overlay protocols [1-4, 7-10].

However, all of these protocols assume that every end host has two-way communication capability. This is not true as firewalls and Network Address Translators are widely deployed in the current Internet. The end hosts guarded by those devices, which are called guarded hosts¹, usually cannot accept incoming connections from outside. Measurements in [5] indicate there are as many as 36% guarded hosts in eDonkey and Gnutella. Authors in [16] reported even higher percentages in their Internet overlay multicast experiments. [16] also shows that existing protocols suffer up to 37% degradation in the average latency in such an environment.

The presence of guarded hosts complicates overlay construction. First, not all connections among nodes are valid because of guarded hosts. This issue of asymmetric links requires existing overlay protocols to modify or redesign their algorithms. Note that accommodating guarded hosts in some structured overlays is very difficult to accomplish. Secondly, simple modification of existing overlay protocols may not make effective use of normal hosts to achieve excellent performances, likely producing suboptimal overlays with bad overlay latencies or some highly loaded hosts.

¹ In this paper, a host is guarded if it is unable to accept incoming connections from other hosts. Accordingly, a host which permits incoming connections as well as outgoing connections is a normal host.

In this paper, we propose a multicast tree construction protocol called Potential-based Multicast Tree Algorithm (PMTA). The first key concept of PMTA is potential, which is introduced to build multicast tree in the presence of guarded hosts. Potentials can help hosts join a multicast tree intelligently and efficiently so that the multicast tree is more balanced. The second key improvement in PMTA is a solution to overcome joining failures, which enables both kinds of hosts to join the overlay without rejection.

We evaluate performance of PMTA using several related metrics. According to experimental results, we conclude that PMTA performs better than existing protocols and therefore is effective and efficient in the presence of a large number of guarded hosts. Also PMTA is more practical and adaptive to real Internet applications in terms of several performance enhancements.

The rest of this paper is organized as follows. We first discuss some related works in Section II. We describe Potential-based Multicast Tree Algorithm (PMTA) in Section III. Then we will present performance evaluation in Section IV. Finally, we conclude in Section V.

II. RELATED WORKS

The representative multicast tree algorithm, which we study in this paper, is a latency-based greedy algorithm of multicast tree construction. We use Original Greedy Algorithm (**OGA**) to denote this algorithm. The multicast tree construction of OGA is fully distributed and efficient. The tree of OGA has a single root, which by definition is a member with no parent in the tree. All hosts follow the same steps to find the right positions where they are closest to the root in terms of latencies. When one host wants to join in the tree, it should find the members which can accept new hosts to join as child. Since there may be several members which can accept the new host as a child, the joining host should measure its latency to each of these members and join as a child of the member which can minimize the total overlay distance from this joining host to the root of the multicast tree.

There are efforts on designing routing algorithm using potentials [15]. The authors propose a routing paradigm called PBTA that utilizes the steepest gradient search method to route data packets. The main idea of PBTA is to define a potential field on the network and route packets in the direction of steepest positive gradient in the potential field. The concept of potential is not applied in overlay construction till now. We will use potentials in our multicast tree construction algorithm.

In addition, some previous researches focus on the issue of guarded hosts in several aspects. The authors in [12] propose an overlay optimization technique called e^* . The key idea of e^* is its cluster-center election protocol and shortcut-selection heuristics. Actually, e^* does not consider the issue of multicast tree construction. In this paper we mainly focus on multicast tree construction with many guarded hosts existing.

Another research [6] gives two solutions to guarded hosts, Basic Contributor and Enhanced Contributor. The idea in this paper is to utilize bandwidth resources of guarded hosts. However, some of methods given by the authors, such as using UDP-based protocols, cannot be operated in real Internet applications. Also this work does not consider overlay optimization seriously.

Internet will not be able to provide universal connectivity (both outgoing connections and incoming connections for each host) in the foreseeable future. Even with IPv6, firewalls are likely to be present, and NAT may continue to exist because of its functionality in recent years. We study the problem of guarded hosts on application level with no changes to underlying network. We think that our work can be used in such environment and benefit other applications in future.

III. MULTICAST TREE CONSTRUCTION IN THE PRESENCE OF GUARDED HOSTS

We first present a new multicast tree construction algorithm called Potential-based Multicast Tree Algorithm (PMTA) in Section III(A). Then in Section III(B) we propose the scheme to solve the joining failures when the multicast tree is saturated.

A. Potential-based Multicast Tree Algorithm (PMTA)

Without the presence of guarded hosts, all newly joined nodes can provide available downlinks for other nodes to join. The multicast tree built by OGA is optimized to minimize the total delay for the current tree. However, according to OGA, a guarded host can join the tree on any level. When a guarded host joins to the tree on a lower² level, it prevents others nodes from joining to its branch as it provides no more downlinks. This may cause the tree to be highly unbalanced. Certain nodes may have large latencies and be far from the root, and therefore reduce the effectiveness of OGA algorithm.

To overcome the above problem caused by guarded hosts, we force the guarded hosts join the tree on the higher level in order to construct a more balanced tree than OGA. Here, we introduce the notion of potential for tree construction. The potential for a node A is defined as,

$$V(A) = -(Level\ of\ A) = -(Hops\ from\ A\ to\ R) \quad (1)$$

where R represents the root of the tree. The average of all nodes' potentials of a tree is an indicator of the balance of the tree. When the number of nodes is the same for two trees, the average of potentials for balanced tree is higher.

² In this paper, we define that low level is close to the root of the tree, while high level is far from the root.

Fig.1 shows the node joining procedure of PMTA. Host A first contacts the membership server, whose main purpose is to bootstrap new hosts to the tree overlay. A obtains some necessary information about the current overlay and tries to join. Then A executes different joining process according to the current tree and A itself.

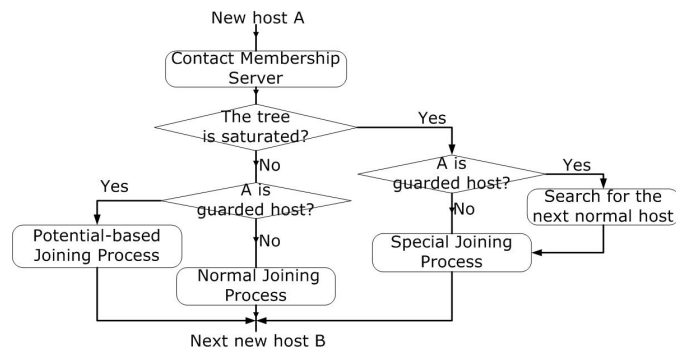


Fig.1. Node Joining Procedure of Potential-based Multicast Tree Algorithm

Potential-based Joining Process: When the current tree is not saturated³ and A is a guarded host, A executes this process. In the guidance of potential, A will get a position with low potential in the tree as possible as it can. If there are still several positions available, A will use OGA to choose one among those positions

Normal Joining Process: When the current tree is not saturated and A is a normal node, A executes this process. It is actually OGA. A does not use potential to search its position because guarded hosts have already taken positions with low potentials so that positions with high potentials are usually left for normal hosts. As a result, even if a normal host does not use potential, it can find a position with high potential with high possibility.

Special Joining Process: When the current tree is saturated, this process is executed. We will explain this process in Section III(B).

B. Avoiding Joining Failures

Another problem of OGA is joining failures. When the multicast tree becomes saturated, no new hosts (guarded and normal) can join the tree any more, and joining failures occur.

We first consider the maximum percentage of guarded hosts (P_{max}) a tree could accept. Let N be the number of normal hosts, M be the guarded hosts and D is the maximum outgoing degree of a node. Then N normal hosts can provide at most $N \times D$ outgoing connections. Every node except for the root needs one outgoing degree of other nodes as its incoming connection. Therefore, all nodes need $N+M-1$ connections to build a tree. When the tree is saturated, the number of outgoing connections equals to the number of incoming connections as

$$N \times D = N + M - 1 \quad (2)$$

Hence,

³ A tree is saturated when it cannot accept new hosts to join.

$$P_{\max} = \frac{M}{M+N} = \frac{(D-1) \times N + 1}{D \times N + 1} \xrightarrow{N \rightarrow \infty} \frac{D-1}{D} \quad (3)$$

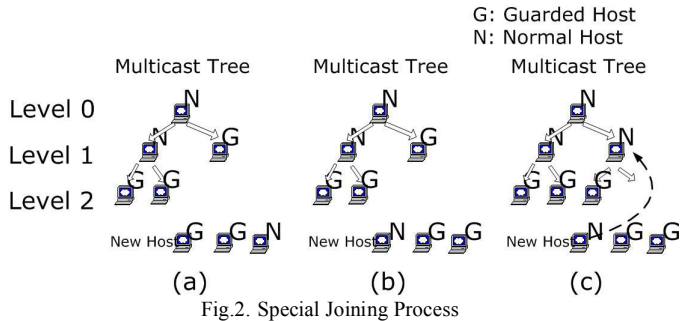
As N grows larger, P_{\max} approaches 50% for $D = 2$. The tree can accommodate 50% guarded hosts at most. When $N \gg 100$, the P_{\max} is already very close to the limit.

In above equations, the tree structure does not affect the P_{\max} . Thus, when a tree becomes saturated, a guarded host cannot join the tree no matter how we change the tree structure. To avoid joining failures, here comes our solution:

Normal host: When a normal host tries to join the saturated tree, it replaces one of the guarded hosts and let that host be its child node. Before it joins, it first views the tree as if there were no guarded hosts and then uses OGA to choose the joining position, deciding which guarded host to replace. To note, this process just affects the guarded host to be replaced.

Guarded host: When a guarded host tries to join, we let it wait until a new normal host joins the tree.

Fig.2 gives an example of this case. Note that maximum node degree is two in this example. In Fig.1(a), the tree is saturated. The next three joining hosts are two guarded hosts, followed by one normal host. In Fig.2(b,c), the guarded hosts will wait until the normal host joins. Then one waiting guarded hosts can join the tree.



IV. PERFORMANCE EVALUATION

A. Metrics

Besides potential, we also evaluate performances of multicast tree algorithms with the following metrics.

1) Relative Delay Penalty (RDP)

Relative Delay Penalty (RDP) [1][11] is defined as the ratio of the overlay latency $D_{i,j}$ between hosts i and j to their unicast latency $D_{i,j}$. In this paper, the latencies in the underlying physical topology are called unicast latencies, while the latencies on an overlay are called overlay latencies. We assume that all unicast paths are routed through the shortest paths so that the *RDP* between any nodes is never less than 1.

Average Relative Delay Penalty (ARDP) [1] is the average *RDP* among all node pairs in the overlay. Especially, in the multicast tree, we study the latencies of all nodes to the multicast tree root. So we define *ARDP* as the average *RDP* of all nodes to the root, which is presented in formula (4).

$$ARDP = \frac{1}{N} \sum_{i=1}^N \frac{D'_{i,root}}{D_{i,root}} \quad (4)$$

Low *ARDP* is an indication that most overlay latencies are close to the respective unicast latencies, therefore all nodes are close to the multicast root. A multicast tree with low *ARDP* is desired in real Internet multicast applications.

2) Rejection Ratio (RR)

Because of guarded hosts, the multicast overlay may get saturated when new hosts cannot join in the overlay. To capture these issues, we use **Rejection Ratio (RR)** [6], which is defined as:

$$RR = \frac{N_{rejection}}{N_{all}} \quad (5)$$

where $N_{rejection}$ is the number of nodes that cannot join in the tree and N_{all} is the number of all nodes. Note that *RR* of an optimal tree should equal to 0.

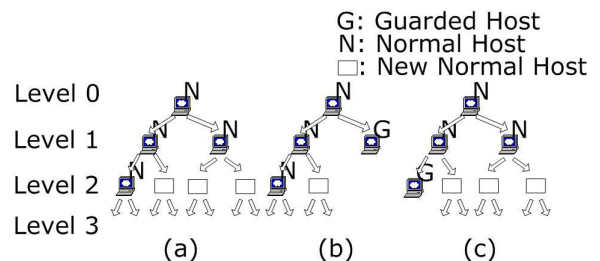
3) Supply Capability (SC)

Supply Capability (SC) is defined as

$$Supply\ Capability\ (SC) = L + 1 - \log_D M_{L+1} \quad (6)$$

where L is the highest level of the multicast tree. M_{L+1} is the maximum number of hosts that the tree is able to accept on Level $L+1$. And D is the maximum node degree of the tree. We define that *SC* is $L+1+\epsilon$ when all leaves of the tree are guarded hosts. When all leaves are normal hosts, Supply Capability is 0. As we can see, a **lower** *SC* describes that the tree could provide more positions on its highest level, which is desired in multicast applications.

For example, we consider Level 3 of the tree in Fig.3. Outgoing degree of all nodes is 2. In Fig.3(a), the tree has 4 leaves on the level 2, so *SC* of this tree is $2+1-\log_2(4*2)=0$. Considering the guarded hosts however makes *SC* more complicated. In particular, *SC* becomes sensitive to the structure of the tree overlay for the same set of hosts. In Fig.3(b), the tree has 2 leaves on the level 2, so *SC* is $2+1-\log_2(2*2)=1$. We can observe that the optimal tree in terms of guarded hosts is one where guarded hosts preferentially choose the positions on the higher levels, just as Fig.3(c). In contrast to Fig.1(b), there are 3 leaves on the level 2, so *SC* is $2+1-\log_2(3*2)=0.415$ for this tree.



B. Evaluation Methodology

We conduct our simulations with two datasets derived from measurements of real Internet. The first dataset is King dataset [17], which includes the round-trip latencies among

1740 Internet DNS servers. The second dataset is PlanetLab dataset [14], which includes the round-trip latencies among 226 hosts on PlanetLab.

Each of our experiments is conducted with a particular node group (from 300 to 1740), a particular node degree (from 2 to 8), and a particular percentage of guarded hosts (from 20% to 50%). Given a group size, we randomly select N nodes from the datasets. We generate M random sequences of the nodes to join. For each sequence, we randomly assign nodes to be guarded hosts with the probability p . We repeat each experiment with M sequences. Finally, we compute the arithmetic mean of the metrics of those M experiments. In our experiments, N is set to between 300 and 1740. M is set to 100. And p is set to the percentage of guarded hosts.

C. Evaluation Results

1) Rejection Ratio (RR)

In this section, we do not study the degree of nodes. We simply fix the same maximum outgoing degree of all nodes to simplify the bandwidth consideration.

Table I gives the *Average Rejection Ratio (ARR)* of OGA. As we can see, *ARR* under 50% guarded hosts is higher than the *ARR* under only 30% guarded hosts. In addition to the increase with percentage of guarded hosts, *ARR* under high degree is less than *ARR* under low degree.

TABLE I
AVERAGE REJECTION RATIO (ARR) of OGA

Percentage of Guarded Hosts \ Outdegree	30%	50%
2	29.93%	90.40%
3	6.83%	22.88%
4	3.99%	8.98%

We can overcome joining failures by using the solution explained in Section III(B), both *RR* of OGA* and *RR* of PMTA are 0.

2) Potential

In order to compare these algorithms effectively, we should investigate them under the same dataset. Because of joining failures, OGA cannot make use of all data in datasets. Thus, we combined OGA with Special Joining Process in Section III(B) to construct OGA*, and use OGA* as a representative multicast tree protocol when comparing with PMTA. We will conduct a detailed evaluation of PMTA and OGA* below.

Fig.4 shows potentials of overlays under various node degrees. And Fig.5 presents potentials of overlays under various percentages of guarded hosts. Note that the variable of y-axis is minus potential. The two figures show that PMTA produces higher potentials than OGA* does. That is to say, the average overlay hop in PMTA-Overlay is less than that of OGA*-Overlay so that nodes in PMTA-Overlay are much closer to the root of the multicast tree. According to potential, PMTA produces more balanced trees than OGA*. Meanwhile, the average potentials in PMTA-Overlay remain almost the

same value without drastic variations when the percentage of guarded hosts increases, which is desired in overlay multicast.

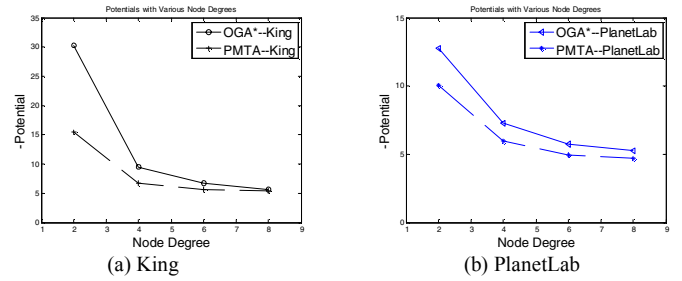


Fig.4. Potentials under Various Node Degrees (50% guarded hosts)

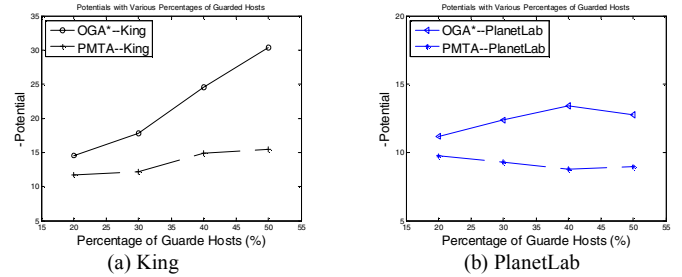


Fig.5. Potentials under Various Percentages of Guarded Hosts (node degree=2)

Fig.6 shows Cumulative Distribution Function (CDF) of potentials in overlay, which indicates PMTA greatly enhance potentials of all nodes in overlay. Note that the variable of x-axis is minus potential. PMTA performs effectively especially in the larger overlays, which can be seen in this figure.

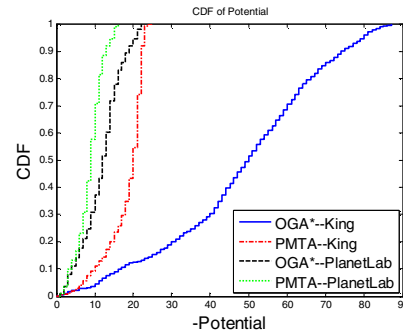


Fig.6. CDF of Potentials in Overlay (degree=2, 50% guarded hosts)

The improvement of potential will affect the performance of the multicast tree in several aspects, including *RDP* and *SC*. We will show these enhancements in the following sections.

3) ARDP and CDF of RDP

Fig.7 shows the *ARDP* of PMTA and OGA* in various sizes of overlays (from 300 nodes to over 1700 nodes). In Fig.7(a), we give a snapshot of two algorithms under various percentages of guarded hosts (from 10% to 50%). We also present a picture of two algorithms under various node degrees in Fig.7(b). It is very clear that PMTA results in much lower *ARDP* than OGA*. With 10% guarded hosts, we can

see only small change in *ARDP*. With 50% guarded hosts, PMTA results in around 26% reduction in *ARDP*.

Node degree is an important parameter for multicast tree construction. Node degree indicates the workload of nodes in overlays. Nodes with higher degree contribute more resources in multicast trees. To simplify this problem, we assume maximum outgoing degree in each experiment. Fig.8 shows the comparison of *ARDP* between PMTA and OGA* on overlays with 50% guarded nodes. PMTA produces lower *ARDP* than OGA*, which means that nodes in PMTA-Overlay are closer to the root than nodes in OGA*-Overlay. Especially, with low outgoing degrees, PMTA performs much better than OGA* in the environments where many hosts on Internet cannot provide much outgoing bandwidth.

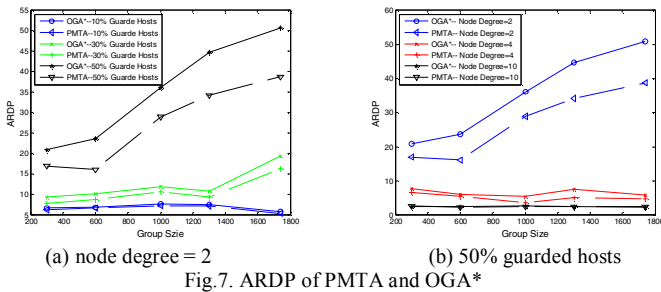


Fig.7. ARDP of PMTA and OGA*

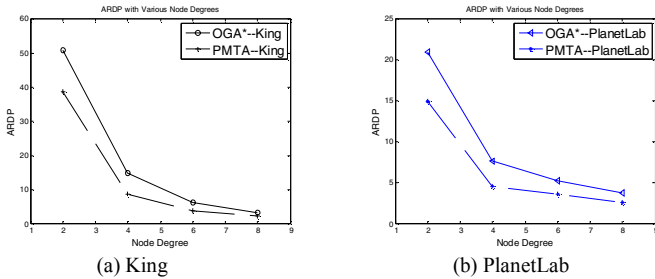


Fig.8. ARDP under Various Node Degrees (50% guarded hosts)

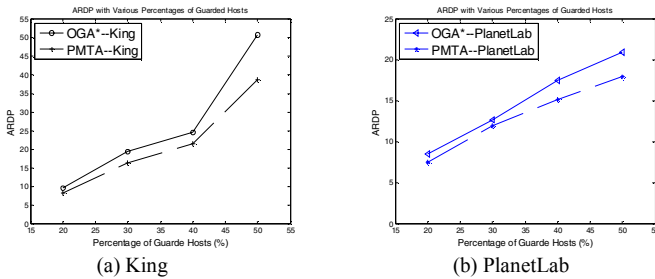


Fig.9. ARDP under Various Percentages of Guarded Hosts (node degree=2)

Also we notice that different percentages of guarded hosts will lead to different performances of overlays. We present our results about *ARDP* under various percentages of guarded hosts in Fig.9 when the outgoing degree of nodes is two. *ARDP* increases when the percentage of guarded hosts increases, in which we can see *ARDP* of PMTA is lower than that of OGA* under any percentage of guarded hosts. PMTA performs better than OGA*, especially in the environments where high percentages of guarded hosts exist.

To provide a more complete picture of the performance advantages afforded by PMTA, we study CDF of overlay latencies. In Fig.10, we show the results on an overlay with 50% guarded nodes when outgoing degree is only two. As expected, we can see that PMTA produce latency distribution with very thin tail as well as good mean value, compared with OGA*. As we can see, with the King dataset, over 80% overlay latencies of OGA* are more than 6000ms, while all overlay latencies of PMTA are less than 6000ms. PMTA not only reduce *ARDP*, but also reduce the absolute overlay latencies. The average overlay latency of PMTA is 23-54% lower than that of OGA* under different parameters among over 1000 experiments.

From previous section, average potential of PMTA-Overlay is higher than average potential of OGA*-Overlay. So PMTA results in less hops in tree overlays, which can reduce overlay latencies and *RDP*.

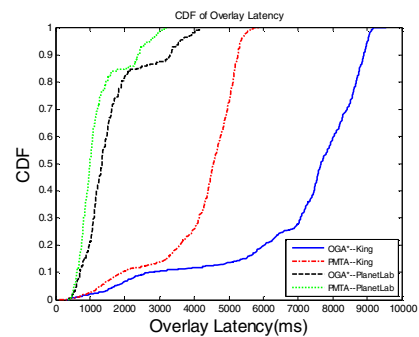


Fig.10. CDF of Overlay Latencies (node degree=2, 50% guarded hosts)

4) Supply Capability (SC)

Figs 11 to 13 show *SC* of overlays built by PMTA and OGA*. PMTA reduces *SC* with various parameters, including various node degrees and various percentages of guarded hosts among different sizes of overlay. Fig.11 gives a picture that PMTA can reduce *SC* among different sizes of overlays. As we can see, with 50% guarded hosts, PMTA produces around 55% reduction in *SC*.

PMTA lowers *SC* in overlays with various node degrees, as reported in Fig.12. Also *SC* is reduced by PMTA in overlays with various percentages of guarded hosts, as illustrated in Fig.13. These figures indicate that PMTA is adaptive to all overlays under different conditions, especially in the environments with low node degree and high percentage of guarded hosts.

The reduction of *SC* is also derived from the improvement of potentials. The positions with high potentials produce lower *SC* than the positions with low potentials. So PMTA results in lower *SC* than OGA*.

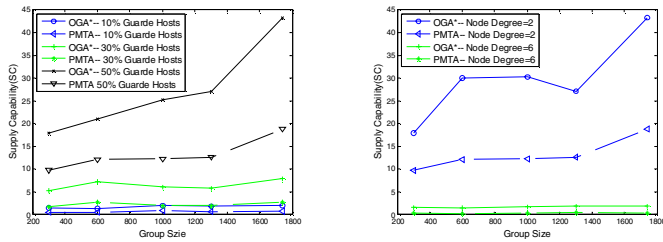
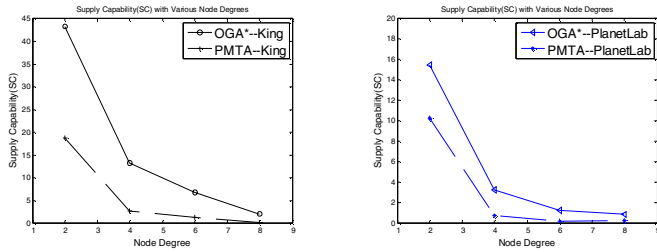
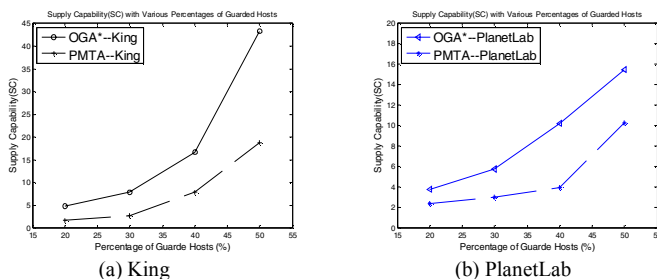


Fig. 11. Supply Capability (SC)



(a) King (b) PlanetLab
Fig. 12. SC under various Node Degrees (50% guarded hosts)



(a) King (b) PlanetLab
Fig. 13. SC under various Percentages of Guarded Nodes (node degree=2)

V. CONCLUSION

In this paper, we first proposed a novel multicast tree construction protocol called Potential-based Multicast Tree Algorithm (PMTA). The key concept of PMTA is potential, which helps to build a balanced tree in the presence of guarded hosts. Secondly, we give a solution to overcome joining failures in the tree construction process. We evaluated PMTA under various network conditions such as high percentage of guarded hosts and different node degree constraints using real Internet delay measurements data. All evaluation results show that PMTA performs much better than OGA and its variant OGA* in several aspects such as the average potential, *RD*P, *SC* and *RR*. PMTA reduces *ARDP* by 26% and average overlay latencies by 23-54%. The results suggest that PMTA is more practical to be used in real Internet environment with a large number of guarded hosts existing.

As for the future work, we plan to apply the concept of potential to the overlay protocols other than multicast tree construction in presence of guarded hosts.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (No.60473087, No.60673184, and No.90412012) and National Basic Research Program of China (973 Program) (No. 2007CB310806). Thanks to Dr. Yongqiang Xiong from

Microsoft Research Asia for his comments and suggestions.

REFERENCE

- [1] Y. Chu, S. Rao, and H. Zhang. A Case For End System Multicast. *ACM Sigmetrics*, pages 1.12, 2000.
- [2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In *Proc. of ACM SIGCOMM '02*, 2002.
- [3] P. Francis. Yoid: Extending the Internet Multicast Architecture. Unrefereed report, Apr. 2000. <http://www.aciri.org/void>.
- [4] D. Tran, K. Hua, and T. Do. Zigzag: An Efficient Peer-to-Peer Scheme for Media Streaming. *Proc. of IEEE INFOCOM 03*, 2003.
- [5] W. Wang, H. Chang, A. Zeitoun, and S. Jamin. Characterizing Guarded Hosts in Peer-to-Peer File Sharing Systems. *IEEE GLOBECOM Global Internet and Next Generation Networks Symposium*, Nov. 2004.
- [6] A. Ganjam and H. Zhang. Connectivity Restrictions in Overlay Multicast. *Proc. of NOSSDAV*, Jun. 2004.
- [7] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. in *Proc. Middleware*, Heidelberg, Germany, Nov. 2001, pp. 329–350.
- [8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2001.
- [9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content Addressable Network. In *Proc. of ACM SIGCOMM 01*, 2001.
- [10] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, Jan. 2004.
- [11] Y. Chu, S. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture. *Proc. of ACM SIGCOMM '01*, 2001.
- [12] W. Wang, C. Jin and S. Jamin. Network Overlay Construction under Limited End-to-End Reachability. In *Proceedings of IEEE Infocom*, 2005.
- [13] Sylvia Ratnasamy, Mark Handley, Richard Karp, and Scott Shenker. Application-level Multicast using Content-Addressable Networks. In *Proceedings of Third International Workshop on Networked Group Communication (NGC)*, London, UK, November 2001.
- [14] Network Coordinate Research Group at Harvard, <http://www.eecs.harvard.edu/syrah/nc/>
- [15] A. Basu, A. Lin, and S. Ramanathan. Routing using potentials: A dynamic traffic-aware algorithm. In *Proceedings of ACM Sigcomm*, Aug 2003.
- [16] Y. Chu, A. Ganjan, T. Ng, S. Rao, K. Sripanidkulchai, J. Zhan and H. Zhang. Early Experience with an Internet Broadcast System Based on Overlay Multicast. *Usenix Annual Technical Conference*, 2004.
- [17] K. P. Gunnadi, S. Saroiu and S. D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In *Proceedings of Sigcomm IMW*, Nov 2002.
- [18] A. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure Overlay Services. In *Proc. of ACM SIGCOMM 02*, 2002.
- [19] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: An Application Level Multicast Infrastructure. In *Proceedings of the 3rd Usenix Symposium on Internet Technologies & Systems (USITS)*, March 2001.