Georg-August-Universität
Göttingen
Institut für Informatik

ISSN 1611–1044
Nummer IFI–TB–2006–05

Technischer Bericht

DMMP: A New Dynamic Mesh-based Overlay Multicast Protocol Framework

Jun Lei, Xiaoming Fu, Dieter Hogrefe

**Technische Berichte
des Instituts für Informatik
an der Georg-August-Universität Göttingen**

Juli 2006

Georg-August-Universität Göttingen
Institut für Informatik

Lotzestraße 16-18
37083 Göttingen
Germany

Tel.	+49 (5 51) 39-1 44 14
Fax	+49 (5 51) 39-1 44 15
Email	office@informatik.uni-goettingen.de
WWW	www.ifi.informatik.uni-goettingen.de

DMMP: A new Dynamic Mesh-based overlay Multicast Protocol framework

Jun Lei, Xiaoming Fu, Dieter Hogrefe
Institute for Informatics, University of Goettingen
Lotzestr. 16-18, 37083 Goettingen, Germany
Email: {lei, fu, Hogrefe}@cs.uni-goettingen.de

ABSTRACT

Multicasting provides an efficient way of delivering data from a sender to a group of receivers. It has been gained much attention over the past decade because of an increasing demand for group communication applications such as multimedia streaming. Compared with network layer multicast solutions, recent application-layer multicast and overlay multicast approaches provide a new way of handling multicast without upgrading the infrastructure in a large scale. Meanwhile, they introduce a number of challenges and are still plagued with concerns pertaining to scalability, deployment, heterogeneity and dynamic performance. In this paper we propose a new protocol framework for relieving these issues, so-called the Dynamic Mesh-based Overlay Multicast Protocol or DMMP, which intends to provide an efficient and reliable multicast support by dynamically managing an overlay core comprised of end hosts. Although more analysis and evaluation is necessary, this paper sheds light on several identified design issues with DMMP and initially analyzes its performance.

Keywords---- Overlay, multicast, end host, application layer multicast, media streaming

1. INTRODUCTION

Over the recent years, a deluge of research efforts have been focusing on moving multicast support out of the network core, since the deployment of network layer multicast has been obstructed by both technical and operational reasons. Those reasons include: the lack of an appropriate charging model, the lack of a scalable inter-domain routing protocol and the lack of support in access control and effective network management [1-2]. To solve these issues of IP multicast, various application level multicast solutions have been proposed, which in turn can be largely summarized into two categories, namely, application layer multicast (ALM) and overlay multicast (OM). As a matter of fact, network layer multicast requires changes in IP routers, while ALM and OM approaches rely on network unicast delivery and does not need network layer infrastructure support from intermediate nodes. These non-network layer multicast protocols are classified according to the way of membership management and overlay construction for a multicast group.

In ALM approach, end hosts form a virtual network, and multicast delivery structures are constructed on the top of this virtual overlay network. A basic ALM approach is to form and maintain an overlay for data transmission, where all end hosts in a multicast session are involved without considering the heterogeneities of them, e.g. computation power, bandwidth and access possibilities. For instance, all end hosts join the full mesh construction of ESM (Narada) [4] and multiple connections exist between any two nodes. The main advantage of constructing such a mesh is the easy implementation and being relatively stable, i.e. it can recover quickly from faults. Unfortunately, the mesh maintenance introduces a very high control overhead ($O(N^2$) where N is the number of group members) because each node must keep the information of the whole group or at least most of group. In addition, ESM's sole dependence on the mesh structure results in that it could only be used well in practice into a small or medium-sized group [5]. NICE [6], in contrast, introduces a hierarchical management scheme to create a scalable ALM overlay. In NICE, each member must join the lowest layer of the hierarchy, while a distribution clustering protocol partitions these members in each layer into a set of clusters. Only one node of each cluster can be elected as the leader to be laid at the higher layer. This hierarchical design simplifies the membership management of the application layer multicast and makes it scale better than the full mesh-based structure. Nevertheless, the joining procedure in NICE (where end-to-end latency from the top layer to the lowest layer is measured) leads to a high control overhead (i.e. members at the very top of the hierarchy maintain $O(\log N)$ other members), which not only prolongs the packet delivery, but also is likely vulnerable to single node failures (e.g. failure caused by the node at the highest layer). As described above, ALM approaches address some practical/deployment issues in network layer multicast but there is a general concern about its efficiency. The reason why application layer multicast solutions are inefficient is that the constructed overlay is "randomly" connected without any considerations on the underlying network topology, e.g. two closely connected overlay nodes may be far from each other in the real network.

Observing the weaknesses from ALM approaches, an alternative approach – overlay multicast or OM, by using a kind of “infrastructure-based” solution, has been proposed to implicitly gain the underlying network information. Proposals of such an approach include OMNI [3] and TOMA [7]. Generally, overlay multicast relies on intentionally placed infrastructure nodes (instead of end hosts) in the network like overlay proxies or MSNs to implement the functionalities similar to IP multicast routers. These intermediate nodes can then obtain and utilize the knowledge about the underlying network topology to optimize the overlay routing path. The design issues of OM can be summarized in the following two aspects:

On the one hand, OM approaches employ these fixed or long-term infrastructure-based nodes to simplify membership management and multicast tree construction. This advantage can become a weakness, too, since the assumption of these fixed nodes in the infrastructure limits the extensibility and flexibility of deployment. Before constructing trees to be adaptive to a different metric, the infrastructure must be re-established based on other long-term measurements.

On the other hand, the locations of these intermediate nodes and the status of the overlay links among them are crucial to the overlay construction and maintenance. However, it is not realistic to conjecture the locations of each newly joining member; hence it cannot ensure that the selected node can provide optimal connections for the new member. That is, TOMA and OMNI need dedicated infrastructure deployment and costly server, which could not be adaptive to dynamic network changes and group member changes. Therefore, it is relatively difficult to implement them into the current Internet environment although they are proposed to provide multicast support for group communication applications. Obviously, to develop a practical, efficient and reliable multicast framework is the exclusive way to wide deployment of multicasting services.

Strongly motivated, in this paper, we present a new overlay multicast framework which manages a dynamic mesh-based overlay core where involves only participating end hosts without relying on the availability of the OM-aware infrastructure nodes, while providing certain degree of efficiency, reliability and resilience. The remainder of the paper is structured as follows. Section 2 gives a brief overview about DMMP framework. Section 3 further analyzes the properties of DMMP. Then, section 4 discusses the performance metrics used in application level multicast. Finally, section 5 concludes with a brief summary and future work.

2. FRAMEWORK OVERVIEW

Different from aforementioned application level multicast protocols, the Dynamic Mesh-based overlay Multicast Protocol (DMMP) approach presented in this proposal attempts to support large-scale groups without relying on any predefined intermediate nodes. In this approach, the overlay multicast mesh is solely constructed and maintained by end hosts. In addition, DMMP is designed to be suitable for real-time media streaming applications, which represents a substantial group of user applications such as IPTV. Therefore, DMMP tries to tackle the twin requirements of bandwidth and delay. Actually, the conjunction with delay and bandwidth guarantees has not been explicitly considered in prior work. To make DMMP a reality, we decouple the proposed solution into the following parts.

Firstly, DMMP constructs an on-demand overlay core by which it can achieve the optimal performance. Secondly, DMMP distributes the burden of group management and data delivery to a few nodes instead of the source. Thirdly, the self-organizing protocol scales well to at least hundreds of nodes without sacrificing the quality of the overlay network. Fourthly, DMMP achieves the scalability by limiting group management within locality so that it can dramatically reduce the overhead and complexity of the overlay maintenance. The preliminary ideas of DMMP have been proposed as an Internet draft [8] and currently being discussed in the Scalable Adaptive Multicast Research Group (SAMRG) of the Internet Research Task Force (IRTF) [15]. The following subsections discuss more details about DMMP.

2.1 DMMP architecture overview

Essentially, DMMP can be regarded as a hybrid approach of application layer multicast and overlay multicast, which attempts to support one-to-many real-time media streaming applications over the Internet [8]. Before we explore the details of DMMP, several terms need to be clarified [5] [16]:

1. Out-degree: residual degree, namely, the number of the rest outgoing multicast sessions that a node can establish;
2. Free-rider: an end host which is not able to provide extra out-degree for other members, that, it can only receive multicast service;

3. Rendezvous Point (RP): a server or a proxy to assist managing group members and to store some required information (e.g. performance related);
4. Source: the multicast session sender. It can be either a video stored server or some video distributed servers in one service domain, which delivers the data traffic to the source-based multicast group members. DMMP currently is designed to only provide the source-specific multicast mechanism to realize the single source-based overlay multicast;
5. Uptime: the time duration from a node joining in a multicast session to its leaving the session;
6. Super node: some end hosts are selected to construct and maintain a dynamic overlay mesh which is used to manage the multicast group and relay data from the source to receivers.

DMMP employs a two-tier hierarchy consisting of an overlay mesh and some core-based clusters. The key idea behind it is to select a few (presumably higher capability) end hosts (i.e. super nodes) during the multicast initialization phrase and also when group member changes self-organize into an overlay mesh, and dynamically maintain such a mesh. The rest of end hosts select their super nodes according to local policies and service requirements. Based on the mesh, those end hosts sharing the same super node form a core-based cluster. The reason why DMMP limits the membership management in locality is that network situation changes (e.g., multicast members joining/leaving) within a certain cluster will not have any impact on other clusters. In this way, the total control overhead can be alleviated. Fig. 1 depicts a simple example of this hierarchy, in which six super nodes construct the overlay mesh. Based on the mesh, some core-based clusters are shaped to connect the super nodes. Here, a corresponding communication channel between the source and Rendezvous Point (RP) is built by exploiting the existing protocol stacks such as UDP/IP or TCP/IP. The data channels utilize IP unicast according to the underlying IP transport scheme.

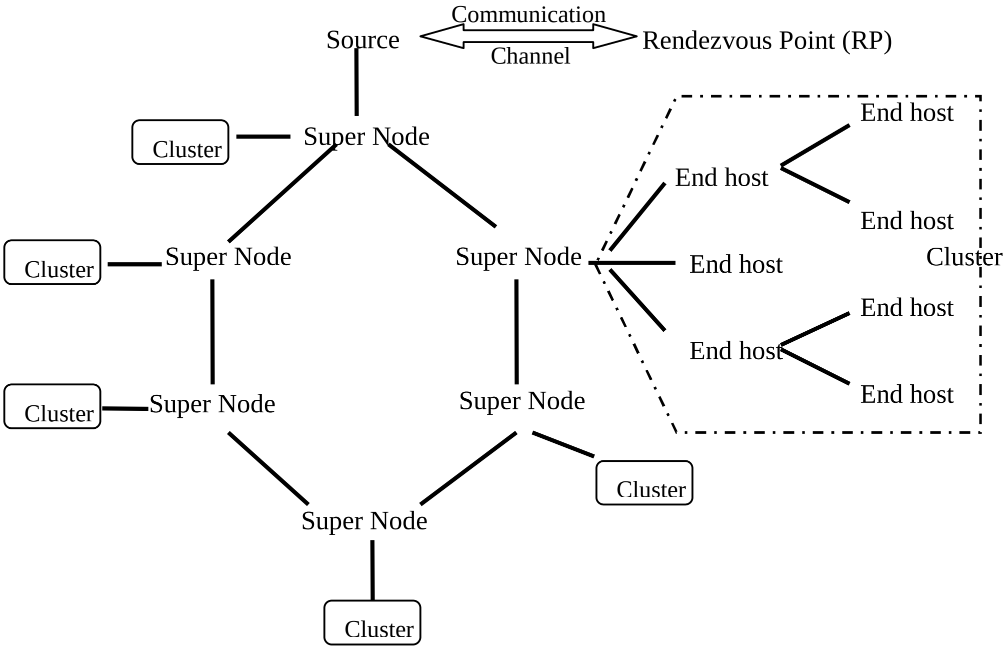


Figure 1 An example of DMMP overlay hierarchy

Specifically, four phrases are involved in the DMMP architecture construction, detailed descriptions could be found in the draft [8]: 1) after initialization, RP will estimate the out-degree of all end hosts, from which it selects some high out-degree nodes to be super nodes; 2) selected super nodes are then connected in a mesh; 3) each non-super node measures the e2e latency to each of the super nodes and one of them with less e2e latency will be chosen; 4)end hosts sharing the same super node will form a core-based cluster. Basically, a source-based DMMP architecture consists of a sender, several receivers, one or many RPs and Dynamic Name Servers (DNSs).

2.2 Features of DMMP

After having taken a glimpse at the basic primitives of DMMP, its main features, to emphasize the necessity of such a design, are presented as follows.

- **Support end hosts with heterogeneity**

It can be used in heterogeneous environments like the Internet as the upper level nodes in the overlay hierarchy are chosen dynamically. In DMMP, it is possible that only a small number of end hosts are selected to construct the overlay mesh when there are a large proportion of free-riders in the group.

- **Dynamic mesh-based approach**

Tree-based overlays are regarded as the most efficient solution for data distribution in a stable network. Nevertheless, lacking of redundant links, an essential characteristic of the tree structure, is not efficient for dynamic scenarios. To address the inherent fragility of trees in dynamic networks, an overlay mesh which connects nodes by multiple paths is introduced in this approach to support efficient and reliable applications.

- **Efficient data distribution tree**

Because overlay multicast networks are built on top of a general unicast infrastructure, the problem of efficiently managing network resources is somehow more difficult than classical networks. Overlay links are built and maintained depending on how well they are utilizing the underlying links. To take advantage of the underlying network, an efficient data delivery tree is constructed upon a mesh and core-based clusters.

- **Adaptive and resilient to dynamic network changes**

The transient nature of the end hosts introduces some issues regarding reliability in an overlay multicast tree. The ungraceful departure of nodes or unexpected network failures may result in data outages on the downstream nodes or even crashes to the whole multicast system. Nevertheless, DMMP is designed to be robust to handle these problems by preventing incapable or transient nodes from staying at the center of the multicast tree. Also it has to note that DMMP could be extended as a pure OM solution when some known designated infrastructure nodes can be used.

Previously, there have been several approaches to devise the mesh-based overlay multicast. One example is Scattercast [17], a mesh-based application level infrastructure for content distribution, in which the mesh is first built among multicast proxies by a neighbor discovery protocol. Another example is a mesh-based data dissemination overlay called Bullet [16]. However, to the best of our knowledge, DMMP is the first proposal supporting all the above four features for real-time media streaming applications over the Internet.

Thus, our main contributions can be summarized as follows.

1. DMMP considers the *heterogeneous capabilities* of group members by investigating their available bandwidth. In this framework, high out-degree nodes which are able to and willing to make more contributions to the network are likely to get better performances. So far, they are entitled to having better performances and more responsibilities. Another benefit that DMMP offers makes it an attractive option as this principal we propose here might further augment the available bandwidth for the overlay tree.
2. In addition to supporting heterogeneous end hosts with considering their available bandwidth, DMMP also considers the end-to-end delay for end hosts. While constructing the overlay multicast tree, high-capacity nodes have the priority to stay at the higher level of the tree: this allows us to produce the tree as short as possible and hence the *overall delay* can be reduced. Furthermore, it is beneficial to achieve the fast convergence during tree initialization phrase and after dynamic changes.
3. Then, we address the transient nature of end hosts by periodically pushing high-capacity nodes to the higher level of the tree using some comparison mechanism. That is, DMMP prevents incapable or transient nodes from staying at the center of the multicast tree. In this case, the DMMP overlay structure is *relatively stable and resilient* to dynamic network changes. Although the failure of a single node may result in a transient instability in a small subset of participants, no single-node failure would lead to a catastrophe in any part of the overlay multicast tree.

3. FURTHER ANALYSIS OF DMMP PROPERTIES

As known, media streaming is a bandwidth-constraint service and available bandwidth resources may be insufficient for multicast sessions during runtime [9]. It is why we need to consider the out-degree bound in streaming applications, which can be easily observed from the available bandwidth. For example, on the assumption that the bit rate of media is B and the outbound bandwidth of an end host i is b(i), the total number of sessions it can establish is b(i)/B which is also the maximum degree of the end host. The knowledge of available bandwidth in overlay routing is nowadays regarded as acquirable, based on recent advances in avail-bandwidth measurement techniques and tools [10-12]. If an end host's maximum degree is less than two (<2), the end host can only perform as a leaf node because it can only receive data from incoming sessions.

One important aspect of DMMP is the ability to support the heterogeneity of the node capability: currently we take out-degree as the primary capability. Usually, only a small number of nodes can serve in the overlay mesh (non-leaf) nodes, while a large number of hosts can only work as leaf nodes. One question immediately arises: how many non-leaf nodes are required when constructing the overlay multicast tree for a given sized group and a given topology of network? To answer this question, we firstly estimate the required non-leaf nodes which can provide extra out-degrees for other nodes in terms of different group numbers, to form the multicast tree in each cluster.

The rest of this section will be organized as follows. Section 3.1 will give an analysis on how it is possible to support end hosts with different capacities. The issue of overall delay optimization with respect to tree depth and out-degree will be discussed in section 3.2. In section 3.3, we show the importance of uptime deployment since DMMP is able to be adaptive and resilient to dynamic network changes. Meanwhile, we will consider the instability caused by dynamic network changes, e.g. end hosts may join/leave the group at will. Another metric in the protocol design, namely the convergence time, will be finally discussed, which covers the session start procedure and dynamic network change scenarios.

3.1. Required number of non-leaf nodes for tree construction

For convenience of discussion, we use the one cluster case as an example to explore the possibility of constructing the overlay multicast tree in DMMP. We assume that *m* end hosts participating in the cluster in which the percentage α of end hosts are non-leaf nodes which have average *n* out-degrees. That is, $(1-\alpha)\cdot m$ end hosts could only be performed as leaf-nodes as they can hardly provide extra out-degree for other nodes. These leaf-nodes can just receive the services instead of making some contributions to the network. Thus, according to the overlay construction mechanism of DMMP they are striven to be placed at the bottom as possible. Moreover, super nodes have also a constraint that they can directly connect no more than *k* end hosts as its immediate children. This operation guarantees that the multicast tree within each cluster satisfies bandwidth constraints of media streaming applications.

In order to compose a complete overlay multicast tree, the number of multicast sessions for leaf nodes should be no less than $(1-\alpha)\cdot m$. Then, we have

$$k + n\cdot m\cdot\alpha > (1-\alpha)\cdot m + \alpha\cdot m\,, \qquad (1)$$

It is clear that $k \geq n$ because the super nodes selected based on the selection mechanism of DMMP (as depicted in section 6.2 of [8]) should provide more out-degrees than other cluster members.

Thus,

$$k + n\cdot m\cdot\alpha > n(1+m\cdot\alpha)\,, \qquad (2)$$

To make (1) come into existence α should satisfy the following inequality:

$$n(1+m\cdot\alpha) > (1-\alpha)\cdot m\,,$$

that is,

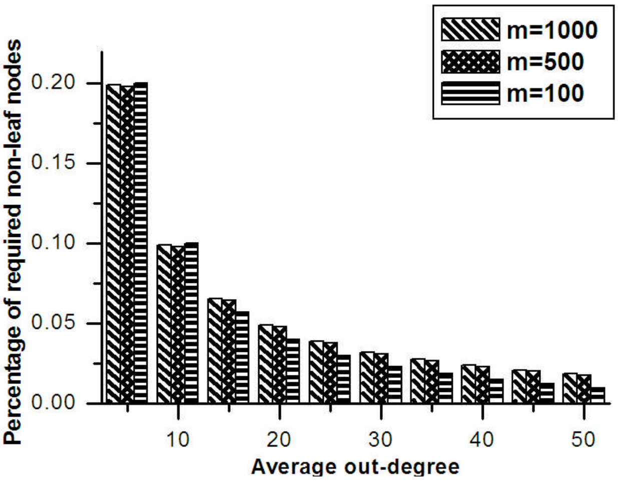


Figure 2 Percentage of required non-leaf nodes vs. average out-degree

$$\alpha > \frac{m-n}{m \cdot n}. \qquad (3)$$

Letting *m*=1000, *n*=20, we can interpret the condition (3) as follows. When α is larger than 0.049, it would be possible to complete the construction of the overlay multicast tree while satisfying the bandwidth requirements if there are 1000 end hosts in the cluster and the average out-degree of non-leaf nodes is 20. That is, (in theory) it would be possible to form the overlay multicast tree for 1000 end hosts if there are no less than 50 end hosts with average out-degree 20. In this case, a large number of free-riders (nearly 950 leaf nodes) exist in the network, which is quite accord with the common situation over the today's Internet.

Note that α relies on the total number of participants (*m*) and the average out-degree of non-leaf nodes (*n*), we show the exact relationship between the out-degree and the percentage of non-leaf nodes in Fig. 2.

In Fig. 2, the percentage of required non-leaf nodes is not high (<0.20) which means no more than 20% non-leaf nodes with average out-degree five can totally support 1000 nodes to construct the multicast tree. Besides, the percentage of required non-leaf nodes reduces dramatically from average-degree 5 to 10. This indicates that less than 0.1 nodes with average out-degree no less than 10 can satisfy the bandwidth requirements for multicast tree construction. Although the percentages of different groups are similar shown in Fig. 2, the exact numbers of required non-leaf nodes in terms of different group numbers are quite different. To better illustrate, the comparison of the number of required non-leaf nodes in accordance with different group numbers i.e., *m*=1000, *m*=500, *m*=100, is shown in Fig. 3.

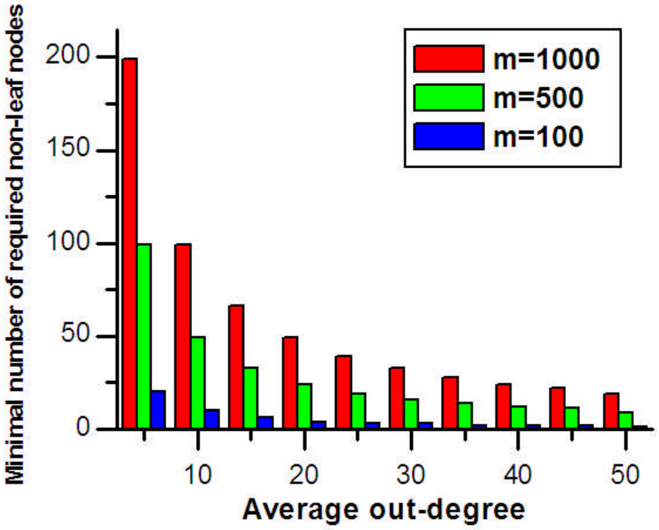


Figure 3 Required number of non-leaf nodes vs. out-degree

Fig. 3 illustrates the relationship between the minimal number of required non-leaf nodes and the average out-degree in terms of different groups.

We roughly analyzed the necessary ratio of non-leaf nodes to leaf nodes regarding different group numbers. We also found that it should be possible to construct the DMMP-aware overlay multicast tree over today's Internet if a small number of end hosts possess of some extra out-degrees. That is, DMMP multicast tree can be constructed in each cluster to satisfy bandwidth constraints of media streaming applications when there are end hosts have different capabilities (e.g. bandwidth support).

Since DMMP targets at providing an efficient and reliable multicast solution for large-scale media streaming applications, it should optimize the overall delay as well besides satisfying the bandwidth requirement. The question arises concerning how to optimize the overall delay of the overlay multicast tree. It is known that the overall delay from the top to the bottom of the tree is somehow proportional to the depth of the tree. Here, we intend to construct the multicast tree as short as possible so that the overall delay from certain super node to end hosts within each cluster will be reduced. A detailed analysis on the issue of tree depth is provided in the following section.

3.2. The analysis on the tree depth

To reduce the overall delay of the tree, a mechanism could be used in DMMP is to assign nodes with larger out-degree to the higher level in each cluster. This seems reasonable because those nodes are likely to get better performance if they are willing to contribute more to the overlay applications. In this case, more end hosts could attach to the tree at each level; hence the tree depth would be shortened. It is assumed that the delay from high level to the lower level is proportional to the depth of the tree. The following subsections discuss several issues related to the tree depth, including overall delay optimization, the worst and the best cases of tree depth problem.

3.2.1. Overall delay optimization

In most cases, a super node connects directly with non-leaf nodes which have comparatively higher available bandwidth. We suggest considering the out-degree of the super node and the number of non-leaf nodes, namely, $k \propto \alpha \cdot m$, where k is the out-degree of the super node.

If $k > \alpha \cdot m$, then some leaf-nodes, $k - (\alpha \cdot m)$ will be selected as the immediate children of the super node as well. However, they can not provide additional sessions for downstream nodes. To complete the multicast tree, non-leaf nodes should provide enough sessions for the downstream nodes.

Thus,

$$\alpha \cdot m \cdot n \geq (1-\alpha) \cdot m - [k - (\alpha \cdot m)],$$

$$\alpha \cdot m \cdot n \geq m - k. \qquad (4)$$

Combining this with the condition (3), the rest of the non-leaf nodes can be attached to the tree at the second level. In this case, the most optimal result for tree-depth should be 2.

If $k < \alpha \cdot m$, some non-leaf nodes can not directly connect the super node and the rest of them will be placed at the lower level. Then, all non-leaf nodes should provide sessions for leaf nodes, that is to say, the following inequality should be satisfied:

$$\alpha \cdot m \cdot n \geq (1-\alpha) \cdot m,$$

$$\alpha \cdot (n+1) \geq 1. \qquad (5)$$

When the condition (5) is satisfied, α is more than $1/(n+1)$. Reconsidering predefined condition (3), we need to combine two conditions for α. Now, consider

$$\frac{1}{n+1} \sim \frac{m-n}{m+n}. \qquad (6)$$

It is certain that $\frac{1}{n+1} < \frac{m-n}{m \cdot n}$, if $m, n > 0$. Similarly, the most optimal result for the tree-depth would be 2 too.

As discussed above, the proposed mechanism of tree construction ensures all leaf nodes are placed at the bottom of the multicast tree as possible. In reality, some leaf nodes, however, may have already occupied the positions at the higher level of the tree. To explain the tree depth problem more explicitly, we show the worst case and best case of the tree depth issue. In the following subsections, we discussed the issue on the tree depth concerning the best case and the worst case.

3.2.2. The worst case of tree depth problem

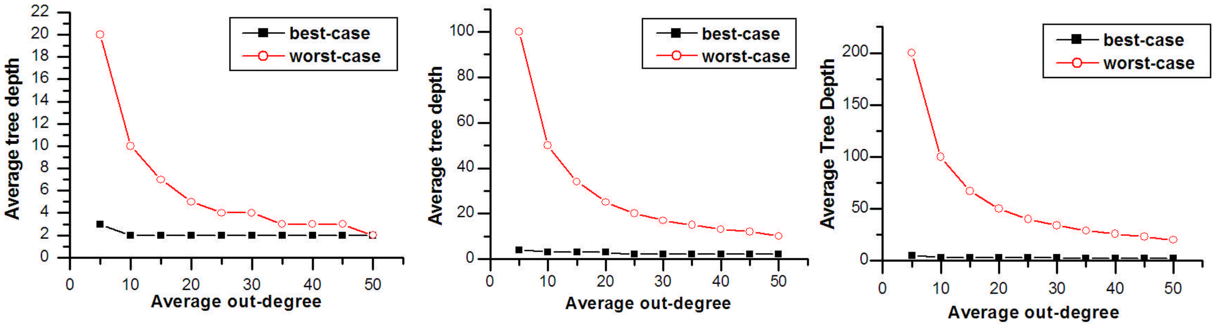
In the worst case, leaf nodes will try to attach to the tree at the higher level so that only one non-leaf node can stay at each level. Theoretically, at least one non-leaf node should stay at each level of the tree; otherwise, it is impossible to support multicast sessions for downstream nodes. For example, the mechanism attaching to the tree without invitation allows nodes to join in the tree once they receive an answer from one of the group members [13]. This approach would create deep graphs with a high worst- case tree depth ($m(1-\alpha)/(n-1)$) but fast join operations and less cost of tree construction.

3.2.3. The best case of tree depth problem

In contrast to the worst case, the best situation is that all nodes with higher out-degree try to occupy the positions at the higher level of the multicast tree so that all leaf nodes can only be placed at the bottom level. For example, the approach of attaching to the tree with best invitation supposes that a newly joining node waits for all responses from the requested nodes until it finds the best one [13]. This approach would create wide graphs with a low worst-case tree depth ([$\log_n^{m(1-\alpha)}$]) but slow join operations and high cost of tree construction as well.

We make use of the above analysis to derive the result of tree depth issue concerning the best case and worst case in terms of *m*=1000, 500 and 100.

In Fig. 4, the tree depth decreases dramatically between out-degree 5 and 10 concerning the worst case. The difference between the best case and the worst case is large, especially when the group is large. For the best case, it is, however, not so optimal in the reality because some leaf nodes may have already occupied the positions at the higher level of the tree. For example, some high out-degree nodes could be just attached to the initial tree at the lower level if some early leaf nodes have already taken up the higher position of the tree. How to deal with this situation? A self-refinement mechanism is proposed that each newly joining member tries to switch to the higher level of the tree.



(a) (b) (c)

Figure 4 Average tree depth when (a) *m*=100 (b) *m*= 500 and (c) *m*= 1000

However, as long as we aim to get better performance by the self-refinement mechanism, the overhead and the cost of tree construction will be increased. In order to balance the tradeoff between the cost of tree construction and the performance, we need to investigate a new tree construction mechanism for DMMP. For example, a hybrid method of attaching to tree with best invitation and attaching to the tree without invitation might alleviate the cost of construction by sacrificing some performances. These design options and their implications will be studied in the later stage of the project.

3.3 The impact of uptime

Compared to the network layer multicast, overlay multicast approaches usually are more susceptive to dynamic network changes, e.g., nodes leave the group just after a short time, which are also called transient nodes. DMMP is able to tackle this problem by periodically pushing high-capacity nodes to higher levels of the tree. Then, it is very likely that super nodes and their immediate children are the high-capacity nodes after a certain time. They are placed at the top level of the tree since they have relatively higher capacity, which is an indication of being more stable than low level nodes. In addition, the newcomers who have higher capacities could climb from the bottom to a higher level after some switching phrases. For example, newcomers at the lower level could switch to the higher level if their capacities exceed (over a predefined threshold) their current parent. Here, an appropriate threshold will be defined to avoid unnecessary switching since if the child has a smaller bandwidth support, it will be ultimately placed below the parent.

How can DMMP achieve the above objective? We combine uptime (its definition seen in section 2.2.1) with out-degree as the capacity of the node, to strengthen the maintenance of the overlay hierarchy. The main goal of doing so is to reduce the impacts of frequent changes on the management of memberships, e.g. transient nodes leave the group. Hence, only a small portion of the tree will be affected and needs to be re-constructed after the dynamic changes.

To illustrate the problem in a simple way, it is assumed that the capacity of each host is linear distributed. Explicitly, we assign the capacity of each end host c_k as follows:

$$c_k = b_k + \frac{b_k}{m} \cdot t_k, \tag{7}$$

where $1 \le k \le m$, m is the total number of cluster members, b_k is the out-degree of node k.

Once an end host participates in the multicast session, t_k starts to calculate the uptime for this member. When it leaves the group, the t_k will be reset as 0.

Upon the expression (7), nodes either with definitely higher bandwidth support or having joined in the multicast service for a long time may have the higher capacity. Moreover, a node is encouraged to contribute more bandwidth resource or longer service time in tradeoff for better service quality. Instead of applying the mechanism that end hosts with higher out-degree will be assigned to the higher level of the tree, those nodes with higher capacity will be entitled higher priority to stay at the higher level of the tree.

To show the importance of the uptime explicitly, we apply the expression (7) into the experiment of Fig. 5. In this case, some node may have higher capacity at the beginning due to its higher out-degree (i.e. out-degree=15). When it leaves the group at time *t*=30, other nodes, e.g. whose out-degree=10, can exceed its capacity in the end. Moreover, some node may join in the multicast session later but it can also exceed the lower degree node after a certain time. Actually, this is a kind of self-refinement used to achieve the efficiency and reliability of the multicast tree. For example, node with out-degree of 12 joins the group at time *t*=30 and exceeds the node with out-degree 10 at time *t*=90 due to its higher bandwidth support. Seen from this simple example, stable nodes with higher out-degree are likely placed at the higher level regardless of dynamic changes.

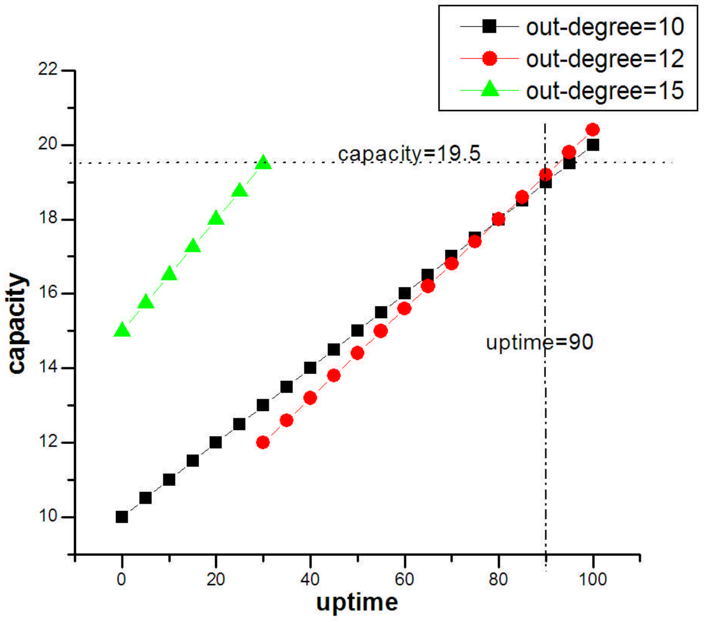


Figure 5 An example of the uptime and the capacity of nodes

In above three sections, we have shown the possibility of constructing the overlay multicast tree in DMMP and discussed some performance related factors, e.g. the out-degree, the tree depth and the uptime. Next, we discuss the convergence time which is one of required metrics for protocol designs.

3.4 The analysis on the convergence time

The proposed approach of multicast tree construction should be beneficial not only to satisfy the service requirements, but also to shortening the convergence time [14]. On the assumption that N nodes take part in the multicast group, they have totally n different levels of capacities.

Let's suppose *m* = 10 and *n* = 4, where five nodes in this group are free-riders, two nodes have only one out-degree and the rest three nodes respectively have two, three and four out-degrees. If we only select one super node to lead the multicast tree, one of the nodes with highest capacity will be selected as the super node. Here, the node with capacity *c* = 4 will lead constructing the multicast tree.

Fig. 6 shows two possible constructions, in which five nodes can only act as leaf nodes and each of the rest nodes occupies one certain capacity. Each circle represents one ode and the number inside is the capacity of this node. On the left side (a), it is one of the possible best cases, where all leaf nodes are placed at the bottom level as possible. Comparatively, the worst case is shown in the right side (b). According to the proposed mechanism, the worst case should be that all low-capacity nodes are placed at the high level of the tree.

We assume that each level takes the similar convergence time (*T*) for the tree since in DMMP each node is assumed to find its parent by requiring the existing higher level nodes. Thus, it will take 2*T* for (a) to complete the construction while it will be 4*T* for (b) to reach the convergence. Nevertheless, the heterogeneities of underlying end hosts and different methods of the overlay tree construction have a great impact on the convergence time, which will be investigated in the next step of research work.

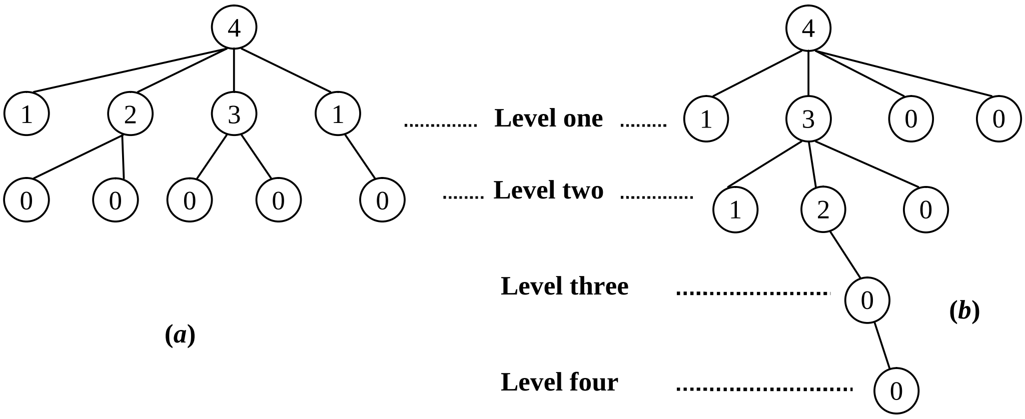


Figure 6 Possible tree constructions: (a) the best cases; (b) the worst cases

Besides, we need to find out how long will it take for a new joining member to attach into the tree as end hosts may join the group at will, which is another indication of convergence. So far we take Fig. 7 as an example to explore how a newcomer (i.e. node X) can join in the multicast tree.

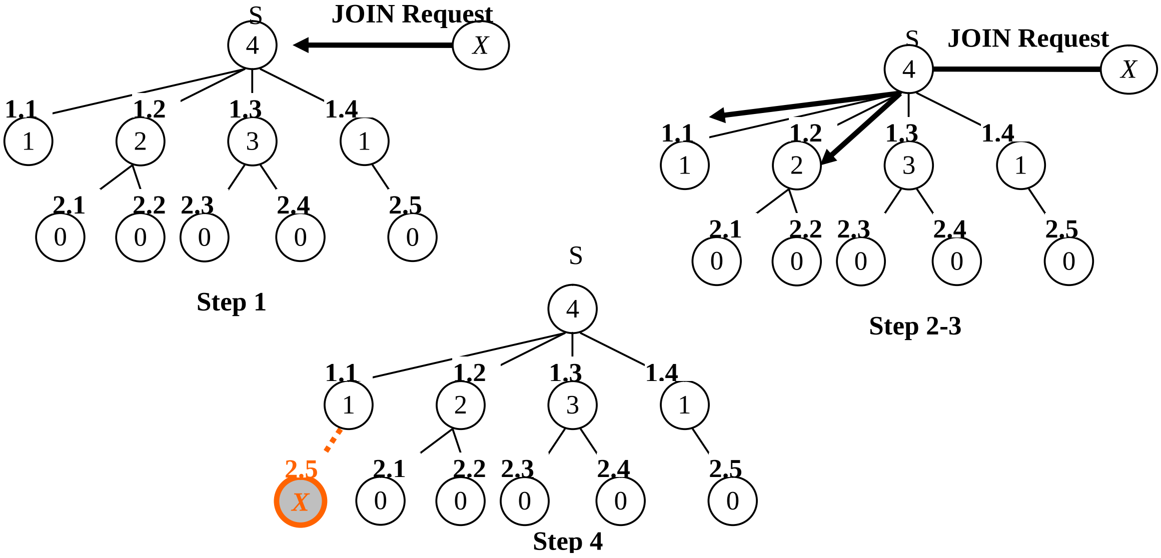


Figure 7 An example of a newcomer joining in the multicast tree

Step 1 Firstly, node X sends a JOIN request to the super node S to be its intermediate node.
Step 2 The super node will check the rest of its out-degree. Unfortunately, S, in Fig. 7, has no other possible sessions for X.
Step 3 Then, S will check its intermediate children and transfer the JOIN request to one of its children with the largest out-degree. In this example, S will transfer the request to node 1.1 and 1.3 because both of them have additional out-degrees.
Step 4 If there are multiple potential parents at the same depth, it chooses the best one in terms of their uptime. Suppose node 1.1 has larger uptime than node 1.3, the newcomer X will accept the JOIN response from the node 1.1. Based on the message exchange procedure, the newcomer finally attaches to the node 1.1 and joins in the multicast tree. Therefore, it takes 2T for X to join in the multicast session.

From this example, we also find that the depth of the multicast tree largely impacts the time of joining procedure. Besides, it turns out that it is important to take the tree depth into consideration during the overlay multicast tree construction. The analysis on the node leaving procedure is similar to the above example, but is omitted for simplicity reasons.

4. DISCUSSIONS ON PERFORMANCE METRICS

Depending on applied application level multicast protocols and the structure of the underlying topology, performance measures, including stress and stretch metrics, can vary greatly. In this section, we analyze the stress, stretch and overhead metrics which are usually used evaluating application level multicast approaches.

Model: Since we are interested in the asymptotic nature of the metrics, we assume a large number of end hosts are densely and uniformly distributed in the network. For a large set of uniformly distributed end hosts, the clusters created by the DMMP protocol will have similar properties, i.e. will have the similar number of cluster members, m, as defined by the protocol.

Stress: Stress is defined as the number of copies of an identical packet sent over a single link. Stretch: as another performance metric. According to its definition, we only consider the maximum stress of DMMP:

$$\text{Maximum stress} = \max(\boldsymbol{S_n}, \boldsymbol{k_s})$$

where $\boldsymbol{S_n}$ is the total number of super nodes and $\boldsymbol{k_s}$ is maximum average degree of the super node. Regarding the maximized stress of NICE, $k\log(_k^N)$ (*N* is the total number of group members and *k* is the cluster size, e.g. *k*=100), the maximum stress of DMMP is greatly smaller than NICE's since the value of $\boldsymbol{S_n}$ is not large (its maximum size is smaller than hundred) and $\boldsymbol{k_s}$ the maximum out-degree of the super node, will not very large (e.g. tens).

Stretch: Stretch is also called Relative Delay Penalty (RDP), defined as below:

$$\text{RDP} = \frac{\textit{overlay delay}}{\textit{unicast delay}}$$

Consider a member, *X* located at an arbitrary point in the space, that belongs to one certain cluster $\boldsymbol{L_s}$ of the overlay hierarchy. Let *S* be the super node of $\boldsymbol{L_s}$ to which *X* belongs. Then, we assume that average tree depth of DMMP could be calculated as a combination of the worst case and the best case (seen in section 3.2):

$$d_{avg} = A(\frac{m(1-\alpha)}{n-1}) + B(\log_n^{m(1-\alpha)}),$$

in which *A*, *B* are two parameters used in synthetically computing the average tree depth. We briefly outline due to space constrains. To simplicity, we just assume *A* = *B* = 0.5, α =0.9 and each level of the tree depth represents the unit delay. Thus, the average stretch within each cluster could be described as:

$$Stretch_{avg} = 0.2s_n + 2(1-\log_n^{10s_n}) \text{ and } s_n << n.$$

Then,

$$Stretch_{avg} = 0.2s_n + 2.$$

Protocol overhead: Corresponding to the structure of DMMP overlay hierarchy, protocol overhead could be divided into two components: mesh management overhead and cluster maintenance overhead. In DMMP, super node needs to exchange update information with each other at a constant frequency of $\boldsymbol{c_1}$. Besides, each super node takes charge of one cluster in which *m* cluster members construct a core-based tree. Thus, mesh management overhead on each super node is $\boldsymbol{O(c_1\ (ns + m))}$. To maintain each cluster and to be adaptive to dynamic changes, cluster members periodically (at the constant frequency of $\boldsymbol{c_2}$) report their status to their neighbors. In this way, the super node will have at maximum $\boldsymbol{O(c_2 k)}$.

Therefore, the overall overhead on super node is $\boldsymbol{O(c_1\ (ns + m) + c_2 k)}$.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel Dynamic Mesh-based overlay Multicast protocol (DMMP) framework to overcome some dynamic efficiency and deployment issues with media streaming applications over the Internet. Under this framework, some selected nodes in a physical region self-organize into an overlay mesh, which is dynamically maintained according to their resource availability and performance. By systematic analysis, we verified that it is possible to construct such an overlay hierarchy for DMMP although there are a large number of free-riders in the network. Secondly, the tree depth has a great impact on the performance of the applications, and hence we construct multicast tree within each cluster as short as possible. To address the instability and unreliability traits of end hosts, DMMP periodically pushes high-capacity nodes to the higher level of the tree. Moreover, we roughly estimated the convergence time of DMMP by a simple example. Furthermore, we analyzed the three important metrics for application level multicast, namely, stress, stretch and protocol overhead.

We are currently implementing the DMMP framework and plan to evaluate its performance and scalability through experiments in PLANETLAB and simulations. During simulations, we plan to make some comparisons with other approaches such as ESM, OMNI and TOMA or Scattercast in terms of aforementioned performance metrics. Furthermore, open issues pertaining to DMMP will be also studied, such as security, end-to-end Quality-of-Service (QoS) provisioning and environment friendliness (e.g., ability to traverse NATs and firewalls).

ACKNOWLEGEMENT

We thank Ruediger Geib, Nicolai Leymann and Dr. Xiaodong Yang for their valuable input that helped improve this work. We also thank John Buford, Yuji-UG-Imai and Yangwoo Ko for helpful discussions in the SAMRG on exploiting super node selection.

REFERENCES

1. K. Almeroth, “The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment”, IEEE Network, Jan./Feb. 2000.
2. C. Diot, B. Levine, J. Lyles, H. Kassem, and D. Balensiefen, “Deployment issues for IP multicast service and architecture”, IEEE Network, Jan. 2000.
3. S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, “OMNI: An Efficient Infrastructure for Real-time Applications”, Computer Networks, 50(6): 826-841, 2006.
4. Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, “A Case for End System Multicast”, IEEE JSAC, Vol. 20, No.8, October 2002.
5. S. Banerjee and B. Bhattacharjee, “Analysis of the NICE Application Layer Multicast Protocol”, UMIACS Technical Report TR 2002-60 and CS-TR 4380, June 2002.
6. S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable Application Layer Multicast”, SIGCOMM’02, August 2002.
7. L. Lao, J.-H. Cui and M. Gerla, “TOMA: A Viable Solution for Large-scale Multicast Service Support”, IFIP Networking 2005, May 2005.
8. J. Lei, X. Fu, X. Yang and D. Hogrefe, “A Dynamic Mesh-based overlay Multicast Protocol (DMMP)”, Internet draft (draft-lei-samrg-dmmp-00.txt), work in progress, June 2006.
9. G. Tan, S.A. Jarvis, D.P. Spooner, “Improving Fault Resilience of Overlay Multicast for Media Streaming”, IEEE International Conference on Dependable Systems and Networks (DSN-2006), June 25-28, 2006, Philadelphia, USA
10. M. Jain, C. Dovrolis, “End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput”, IEEE/ACM Transaction on Networking, 11 (4) 537-549, 2003.
11. N. Hu, P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques”, IEEE JSAC, 21(6): 879-894, 2003.
12. J. Strauss, D. Katabi, F. Kaashoek, “A measurement study of available bandwidth estimation tools”, in Proceedings of ACM SIGCOMM Conference on Internet Measurement, 2003, pp. 39-44.

13. G. Carsten, K.-H. Vik, P. Halvorsen, “Multicast Tree Reconfiguration in Distributed Interactive Applications”, 2nd IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME'06).
14. H. Ural, Z. Keqin, “An efficient distributed QoS based multicast routing algorithm”, IPCCC 2002.
15. IRTF Scalable Adaptive Multicast Research Group, http://www.samrg.org/
16. D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat, “Bullet: High bandwidth Data Dissemination using an Overlay Mesh”, in SOSP’03, Bolton Landing, New York, USA, 2003.
17. Y. Chawathe, “Scattercast: An adaptive Broadcast Distribution Framework”, PhD thesis, University of California, Berkeley, December 2000.