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

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

Abstract—**Multicasting can provide an efficient way of delivering data from a sender to a group of receivers. It has received much attention over the past decade because of an increasing demand for group communication applications such as multimedia streaming. However, native IP multicast has not become widespread largely due to its technical and operational issues. To overcome these obstacles of deployment, various application layer and overlay multicast approaches have been proposed. Compared with IP multicast, they provide a new way of handling multicast without upgrading the infrastructure in a large scale. Nevertheless, they introduce a number of challenges and are still plagued with concerns on scalability, heterogeneity and dynamic performance. In this paper we propose a new protocol framework for addressing these issues, so-called the Dynamic Mesh-based Overlay Multicast Protocol or DMMP, which intends to provide an efficient and resilient multicast support by dynamically managing an overlay core comprised of end hosts. Moreover, DMMP can be used for media streaming which is contracted by a limited resource in stream supplying entities and requires good scalability and reliability. Initial analysis shows that DMMP has the potential to efficiently deliver multicast services and better scalability than NICE.**
Keywords- Overlay, multicast, end host, application layer multicast, media streaming

1. INTRODUCTION

Over the recent years, a lot 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 issues [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.

In ALM approach, end hosts form a virtual network, and multicast delivery structures are constructed on the top of this virtual overlay. 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. Unfortunately, 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. 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 causes a high control overhead, which not only limits the scalability of deployment, but also is likely vulnerable to single node failures (e.g. possible failures caused by the node at the highest layer of hierarchy). As described above, ALM approaches address some practical/-deployment issues in network layer multicast but there is a general concern about its efficiency and scalability.

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 improve multicast efficiency and reduce the resource (e.g, bandwidth). Proposals of such an approach include OMNI [3] and TOMA [7]. The design issues of OM can be summarized in the following two aspects:

On one hand, OM approaches employ some 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. For example, the infrastructure must be re-established based on other long-term measurements before constructing new multicast trees to adapt to the requirements imposed by a different metric.

On the other hand, TOMA and OMNI need dedicated infrastructure deployment and costly server, which could not be adaptive to dynamic network changes and group member changes such as new members join. 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 resilient multicast framework is the essential way towards wide deployment of multicasting services.

Additionally, the explosive growth of multimedia services and applications over Internet necessitates streaming media to a large population of users. However, with current media streaming technology, it is hard to develop a complex on-demand media streaming system due to the following two key challenges [8]. First, the total number of concurrent clients the system can support is limited by the resources of the streaming supplying entity. The limitation mainly comes from (1) the server's processing power; (2) memory size and (3) out-bound network bandwidth. Nonetheless, the first two issues are outside of our scope, as the objective of this paper is to provide an efficient and resilient overlay multicast framework. Second, current media streaming proposals have limitations in reliability and scalability. The reliability concern arises from the fact that only one entity is responsible for all clients. The scalability issue results from the situation that adding Internet-scale potential users requires adding a commensurate amount of resources to the supplying server. Meanwhile, aforementioned proposals could not explicitly support real-time media streaming applications in a large scale.

Motivated by these previous studies, in this paper we present a new overlay multicast framework which manages a dynamic mesh-based overlay core and only involves participating end hosts without relying on the availability of the OM-aware infrastructure nodes, while providing certain degree of efficiency, reliability and resilience. The proposed framework can be applied to media streaming applications having hard real-time requirements, since it addresses two key issues of current media streaming systems. The properties of DMMP can be summarized as follows.

1. DMMP considers the *heterogeneous capacities* of group members by evaluating their available bandwidth during runtime. In this framework, high-capacity nodes which are able and willing to make more contributions to the network are expected to get better performances. This may help maximizing the usage of available bandwidth for the overlay tree.
2. DMMP also considers the *end-to-end delay* for end hosts. When constructing the overlay multicast tree, high-capacity nodes are given priority to stay at the higher level of the tree. In return, this allows us to produce the tree as short as possible and hence the overall delay could be reduced.
3. DMMP considers the transient nature of end hosts and tries to prevent incapable or short-lived nodes from staying close to the center of the multicast tree. Consequently, the DMMP overlay structure is relatively *stable and resilient* to dynamic network changes. The failure of a single node may result in a transient instability in a small subset of participants, but it will not cause a catastrophe in the whole overlay.

To summarize, our contributions of this paper are as follows: Firstly, we propose DMMP, an efficient and resilient overlay multicast framework. Then, we analyze its basic properties. Lastly, we compare DMMP with another application layer multicast protocol NICE using stress as the evaluation metric, which shows DMMP has the potential to support large multicast groups with lower stress.

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 and 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

In order to overcome two challenges mentioned in Section 1, media streaming task in DMMP is accomplished by the following two phases:

- DMMP constructs an on-demand overlay core (or mesh) by which it can achieve the optimized performance. On one side, DMMP distributes the task of group management and data delivery to a few nodes (which constructs the on-demand overlay core) instead of the source. On the other side, it alleviates the risk of one entity dependent reliability.
- Based on the structured mesh, several clusters are formed to connect with selected mesh members. DMMP applies the concept of locality (e.g., clusters) into the group management so that it can dramatically reduce the control overhead and complexity of the overlay maintenance.

In addition, as required in real-time media streaming services a sequence of media packets should be transmitted with minimal communication delay and maximum bandwidth support. Therefore, DMMP tries to meet both requirements of bandwidth and delay.

Let us explain how to construct DMMP overlay hierarchy by an example, detailed description could be found in [15]. In Fig. 1, 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. Basically, a source-based DMMP framework consists of a sender, several receivers, one or many Rendezvous Points (RP is a server or a proxy to assist managing group members and to store some required information (e.g. performance related)) and Domain Name Systems (DNSs). The deployment of RPs is flexible according to the application requirements.

Assuming that it is the first time to construct the overlay hierarchy, then:

Step 1 When obtaining a list of group members from the RP, the source will select an application-specific number of end hosts as super nodes. Those end hosts are actually used to manage the multicast group and relay data from the source to receivers. To illustrate the super node selection mechanism in a simple way, we assume that the capacity of each host is linear distributed. Thus, we assign the capacity of each end host c_i as follows:

$$c_i = b_i + \frac{b_i}{N} \cdot t_i , \qquad (1)$$

where $1 \le i \le N$, N is the total number of group members, b_i is the available bandwidth of node i. For media streaming

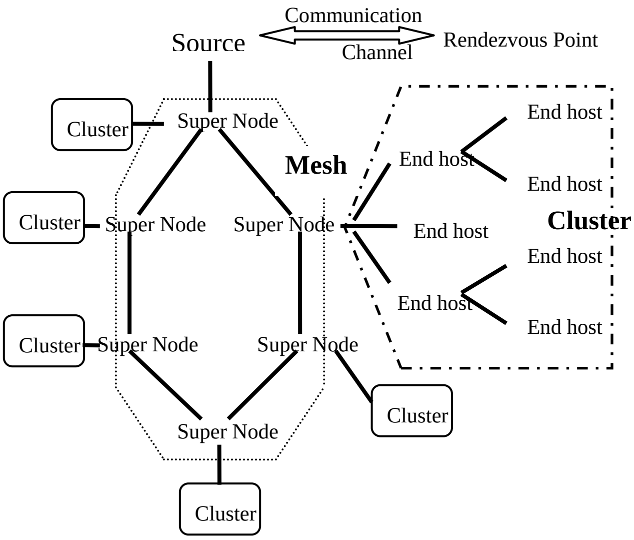


Figure 1 An example of DMMP overlay hierarchy

systems, available bandwidth resources may be insufficient for multicast sessions during runtime [11], which is why we consider the available bandwidth during the super node selection. Obviously, if an application has additional requirements on end-to-end delay or loss rate, those metrics could be jointly considered in expression (1) during the overlay hierarchy construction. Moreover, t_i starts to calculate the time duration from a node joining in a multicast session to its leaving the session, or is called *uptime*. In the initialization stage, nodes with higher bandwidth support will be selected as super nodes since the current *uptime* is zero.

After selection, super nodes self-organize into an overlay mesh rooted at the source. Due to the space limit, we don't address the issue of overlay mesh construction which is mainly motivated from [4].

Step 2 During the cluster creation procedure, each non-super node will firstly consult its local cache for super node candidates. If there are no suitable candidates, it queries the RP immediately. Then, the requestor caches these new received candidates, from which it chooses the best one based on e2e latency measurements.

Step 3 Those non-super nodes sharing the same super node will then form a local cluster. The cluster formation is initiated by the super node which is responsible for informing the RP and contacting the source. Classically, certain numbers (due to the super node's available bandwidth) of end hosts with larger capacity are selected as its immediate children.

Step 4 If the capacity of a super node is exhausted, it responds to new requestors with its immediate children and an indication of rejection. These requestors then send *Join* requests to the list of candidate parents received from the super node. In this case, requestors with higher out-degree are likely to be accepted as the children. If there are multiple acceptances, the end host attaches to the one which is "near" to it due to the e2e latency.

Step 5 The iteration will continue until all end hosts confirm their positions, and at the same time the control hierarchy is initially constructed for the overlay multicast group.

Essentially, DMMP can be regarded as a hybrid approach of application layer multicast and overlay multicast, which attempts to support one-to-many media streaming applications having hard real-time requirements [8]. The preliminary ideas of DMMP have been proposed as an Internet draft [9] and are currently being discussed in the Scalable Adaptive Multicast Research Group (SAMRG) of the Internet Research Task Force (IRTF) [10].

In the next section we present further analysis on three main properties of DMMP. 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 how DMMP is able to be adaptive and resilient to dynamic network changes.

3. FURTHUER ANALYSIS ON DMMP PROPERTIES

One important property of DMMP is the ability to support the heterogeneity of the node capacities. Currently, we take out-degree as the primary capacity, which is noted as the number of the outgoing multimedia sessions that a node can establish. 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. Meanwhile, the knowledge of available bandwidth in overlay routing is nowadays regarded as acquirable, based on recent advances in available bandwidth measurement techniques and tools [12]. However, in a heterogeneous environment like Internet, only a small number of end hosts can provide extra out-degree, while a large number of them can only receive data from incoming sessions, so-called 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.

3.1 Required number of non-leaf nodes for tree construction

Construction of an overlay multicast tree can be modeled as a degree-constrained spanning tree problem. For the convenience of our discussion, one cluster case is taken as an example to explore the possibility of constructing the DMMP-aware overlay multicast tree. We assume that m end hosts participating in the cluster in which the percentage α of end hosts are non-leaf nodes. Out-degrees for i ($1 \leq i \leq [\alpha \cdot m]$) non-leaf node is n_i. That is, $[(1-\alpha) \cdot m]$ end hosts could only perform as leaf-nodes as they can hardly provide extra out-degree for other nodes. These leaf-nodes are planned to be placed at the bottom as possible because they can just receive the services instead of making some contributions to the network. Unless otherwise stated, in the remaining sections, above notations are kept in the same meaning.

Observed from [16], it has a feasible solution to compose an overlay multicast tree in a cluster if and only if $n_i \geq 1$ and $\sum_{i=0}^{m} n_i \geq 2m$. Then, we have

$$(1-\alpha)\cdot m + \sum_{i=0}^{m\cdot\alpha} n_i \geq 2m, \quad (2)$$

since $[(1-\alpha)\cdot m]$ end hosts have only 1 out-degree. We assume the distribution of n_i is an arithmetical series, that is,

$$n_i = [n_1 + d\cdot(i-1)].$$

To simplify the inequality (2), we choose d =0.5. Thus,

$$\alpha\cdot n_1\cdot m + \alpha\cdot m \cdot (\alpha\cdot m - 1) \cdot 1/4 \geq m \cdot (1+\alpha),$$

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

$$m\cdot\alpha^2 - (4\, n_1 - 5)\alpha - 4 \geq 0. \quad (3)$$

We interpret the relationship between the minimal number of required non-leaf nodes and the minimal out-degree in terms of different group in inequality (4). As long as the following inequality is satisfied,

$$\alpha \geq (4/m)^{1/2} - (4n_1 - 5)/2m \quad (4)$$

the overlay tree can be constructed. Let take m = 1000, n_1 = 2 as an example, $\alpha \leq 0.06324$. That is, it is possible to form the overlay multicast tree for 1000 end hosts if there are at least 64 non leaf-nodes with minimal out-degree 2. In this case, a large number, nearly 930 leaf nodes exist in the network, which is quite accord with the common situation over the today's Internet.

From above analysis, it should be possible to construct the DMMP-aware overlay multicast tree to satisfy bandwidth constraints of media streaming applications although end hosts have different capacities (e.g. available bandwidth support). Since DMMP targets at providing an efficient and resilient multicast solution for large-scale media streaming applications, it should optimize the overall delay besides satisfying the bandwidth requirement.

3.2 Analysis on the tree depth

We believe there is a need to reduce the *overall delay* of the multicast tree, which can be easily observed from the time-constraints of media streaming systems, for instance, a packet arriving after its scheduled play back time is useless and considered as lost. The question arises concerning how to optimize the overall delay of DMMP-aware multicast tree. The overlay construction within each cluster has a great impact on the e2e delay of each node. Henceforce, the objective of reducing the overall delay can be regarded as constructing the multicast tree within each cluster as short as possible. It is also noted that the overall delay from the top to the bottom of the tree is somehow proportional to the depth of the tree. In DMMP, a mechanism is proposed that nodes with larger capacity would be assigned to the higher level in each cluster. This seems reasonable because more end hosts could attach to the tree at each level, and the tree depth would be shortened. In addition, those nodes staying at the higher level are likely to get better performance if they are willing to contribute more to the overlay applications.

In reality, it is, however, not so optimal since some leaf nodes may have already occupied the positions at the higher level of the tree. In this case, some high out-degree nodes could only be attached to the initial tree at the lower level. To explain the tree depth problem more explicitly, in the following subsections we discuss the issue concerning two cases.

3.2.1. The worst case of tree depth problem

In the worst case, all leaf nodes will attempt to join the tree at the higher level or they have already occupied these places. 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. One example 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 $[(1-\alpha)\cdot m/(n-2)]$ but fast join operations and less cost of tree construction. Here, n represents the average out-degree of non-leaf nodes.

3.2.2. The best case of tree depth problem

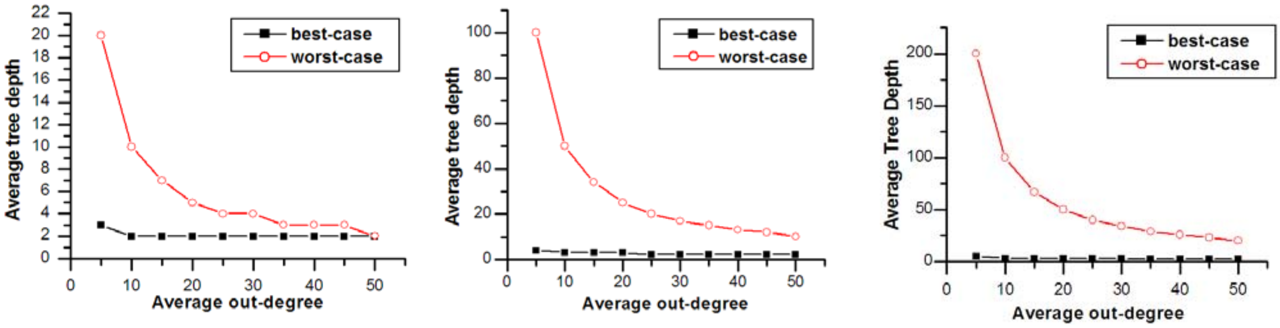
In contrast to the worst case, the best situation is that all nodes with higher out-degree try to occupy the positions at the higher level of the multicast tree so that all leaf nodes can only be placed at the bottom level. For example, the approach of attaching to the tree with best invitation supposes 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 ($[\log_{n-1}^{m(1-\alpha)}]$) but slow join operations and high cost of tree construction as well.

Accordingly, 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. 2, 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 huge, especially when the group is large. In some cases, early-joined leaf nodes may have already taken up the higher position of the tree. How to cope with this situation? A self-refinement mechanism is proposed in DMMP by periodically comparing their capacities [15]. Upon expression (1), nodes either with definitely higher bandwidth support or having joined in the multicast session for a long time will be switched to or kept staying at the higher level of the tree. As a result, these leaf nodes are replaced by high capacity nodes.

3.3. Resilient to dynamic network changes

One cause for current multimedia streaming services which cannot guarantee required QoS occurs mainly from unstable network status. Compared with IP 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.

How can DMMP achieve the above objective? The node failure (e.g., leave the group accidentally) is detected by noticing periodically missing REFRESH message between the node and its relatives (e.g., parent, sibling). Once the



(a) **(b)** **(c)**

Figure 2 Average tree depth when (a) N=100 (b) N= 500 and (c) N= 1000

suspect node is confirmed to "death", one of its children with higher capacity will replace its place, and other children will correspondingly change their positions. Accordingly, the information will be updated in the source. However, if the leaving node is a super node, it will be even more difficult since all its cluster members are partitioned from the tree. One possible solution is that each immediate child of the super node must find a backup parent list. Once the super node leaves, these children try to contact with their alternative parents to rejoin the tree [9].

If one non-leaf node leaves the group, its downstream nodes will be unavoidably affected. We believe that two possible means can alleviate such impacts: one is to reduce the possibility of failures; the other is to reduce the number of possible affected nodes. In practice, however, the first way might be very difficult since end hosts may join/leave the group at will. For the second, DMMP proposes a proactive mechanism by periodically pushing high-capacity nodes to higher levels of the tree. Meanwhile, we combine *uptime* with *available out-degree* as the capacity of each node, which is depicted in expression (1) to strengthen the maintenance of the overlay hierarchy. Thereby, it is very likely that super nodes and their immediate children are the high-capacity nodes after a certain time. They have relatively higher capacity, which is an indication of having more bandwidth support and being more stable. In addition, the newcomers who have higher capacities could "climb" from the bottom to a higher level after some switching stages. For example, a newcomer at the lower level could switch with its parent if its capacity exceeds (over a predefined threshold) the 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.

To summarize, stable nodes with higher bandwidth support are likely placed at the higher level of the DMMP-aware tree regardless of dynamic changes. Moreover, a node is encouraged to contribute more resources or longer service time to the network in tradeoff for better service quality. These design options and their detailed implementations will be studied in the future stage of the proposal.

4. DICUSSIONS ON PERFORMANCE METRICS

Towards validating our theory presented above, we describe initial results consisting of a series of performance evaluations on a proposed model. Meanwhile, we focus our efforts on analyzing the *Stress* metric for DMMP (Note that we have also investigated other metrics, such as *stretch*, *overhead* [15], which are not presented here due to the space limit). Generally, we would like to keep the stress on all links as low as possible. For instance, the stress for any network level multicast tree is one.

Model: Since we are interested in the asymptotic nature of the metric, we assume a very large number of end hosts are densely and uniformly distributed in the network. Thus, the DMMP-aware clusters will have similar properties, i.e. will have the similar number of cluster members, k.

Stress: We refer to the number of identical copies of a packet carried by a physical link as the *stress* of a physical link [4]. It mainly measures the additional load on a network link, and therefore it closely related with the efficiency of resource utilization and scalability of the protocol. DMMP builds the data delivery plane directly on top of the overlay hierarchy as shown in Figure 1. A distance vector protocol runs on top of the core mesh and within each cluster data are forwarded top-down across the DMMP-aware tree. Thus, the number of links that connect super nodes of cluster C_i to their respective cluster members is given by p_i which is no more than their out-degrees. The number of packet copies is determined by the number of downstream nodes ($\le n_i$). The average stress could be:

$$\bar{\lambda} \le \frac{\sum_{i=1}^{L} p_i + L[\sum_{i=1}^{[k\cdot\alpha]}(n_i - 1) + k\cdot(1-\alpha) + 1]}{N}. \qquad (5)$$

where N is the total number of links (nodes) in the network, L notes the number of super nodes and n_i is the out-degree of non-leaf nodes.

Let us make fluid approximation on expression (5). Then, the average stress of DMMP is about

$$\bar{\lambda} \le \frac{p+1}{k} + 1,$$

Here, p is the average out-degree of super nodes and $p \le k$ for asymptotically large group N.

Then, we assume

$$p = k \cdot (1-\alpha), \quad (6)$$

because the number of leaf nodes in the DMMP is much larger than the number of non-leaf nodes. Reconsidering the condition in (4)

$$\bar{\lambda} \leq 2 + \frac{1}{k} - \alpha, \quad (7)$$

Figure 3 presents the initial comparison results of the average stress between DMMP and NICE, a well-known protocol which assumes to have a good scalability.

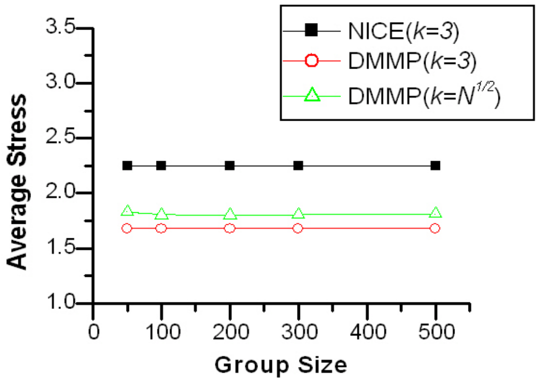


Figure 3 The comparison on the average stress

Meanwhile, the value of k in DMMP usually varies with the value of N. But the value of k in NICE is predefined (usually k=3) and will not change corresponding to the group size. In contrast to NICE (the average stress is $k^2/(k-1)^2$ [5]), the stress of DMMP keeps at fairly small values (always below 1.9) regardless of the group size. It means DMMP could achieve better performance in terms of resource efficiency and scalability. However, if we set $k= N^{1/2}$, the value of stress is larger than the first setting k = 3. We believe it might be caused by improper value of k, which also implies choosing values of k has a great impact on the performance of DMMP and should be taken into consideration in a DMMP implementation when the value of N changes.

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. Through analysis, we conclude that it is possible to construct such an overlay hierarchy for DMMP although there are a large number of leaf nodes in the network. Secondly, the tree depth has a great impact on the overall latency of the multicast tree, and hence we construct multicast tree within each cluster as short as possible. To address the instability and unreliability aspects of end hosts, DMMP periodically pushes high-capacity nodes to the higher level of the tree. Finally, our preliminary analytical results show good stress for large multicast groups. Our analysis is based on the assumption of a large member population densely distributed in the network. However, in practice, the uniformity assumption may not hold.

Thus, we are currently implementing the DMMP framework and evaluating its performance and scalability through simulations using OMNeT++ [14]. We plan to compare DMMP with some other approaches such as NICE, ESM, OMNI and TOMA in terms of performance, scalability and overhead. Furthermore, open issues pertaining to DMMP will be also studied, such as security, end-to-end Quality-of-Service (QoS) provisioning.

ACKNOWLEDGMENT

We thank Ruediger Geib, Nicolai Leymann and Xiaodong Yang for their valuable input that helped improving 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, Special Issue on Overlay Distribution Structures and their Applications, Vol. 50, No. 6, April 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] Mohamed M. Hefeeda, Bharat K. Bhargava, "On-Demand Media Streaming Over the Internet," ftdcs, p. 279, The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03), 2003.

[9] 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, June 2006.

[10] IRTF, Scalable Adaptive Multicast Research Group, http://www.samrg.org/

[11] 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 2006, Philadelphia, USA

[12] 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.

[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] OMNeT++, http://www.omnetpp.org/index.php.

[15] J. Lei, X. Fu, I. Juchem, D. Hogrefe, "Architectural thoughts and requirements considerations on video streaming over the Internet", Technical Report No. IFI-TB-2005-06, Institute for Informatics, University of Goettingen, Germany, ISSN 1611-1044, Nov. 2005.

[16] M. Yang, Z. Fei, "A Proactive Approach to Reconstructing Overlay Multicast Trees", IEEE INFOCOM'04, 2004.