IJCAT International Journal of Computing and Technology, Volume 1, Issue 2, March 2014
ISSN : 2348 - 6090
www.IJCAT.org

Self-Improved ODMP Protocol (ODMP-E)

[1] **Mr. Prashant B Kumbharkar** , [2] **Dr. T R Sontakke**

[1] Research Scholor, JJT university, Rajastan -333001

[2]Research Guide , JJT university, Rajastan -333001

Abstract - The aim of this work is to improve performance of ODMP protocol to build a scalable, efficient, reliable and incrementally deployable Protocols for supporting real time media distribution. We presents two self-improvement techniques proposed by the ODMP-E protocol, namely, mesh self-improvement and cluster self-improvement.

***Keywords* – ODMP.**

1. Introduction

The ODMP framework, A two-tier overlay network consisting of a mesh core and several clusters have the potential of supporting large-scale multimedia distribution services. However, the quality of data path can be instable and the average loss rate can also largely increase because of changes in membership dynamically (e.g. new member joining). In this paper, we extend the basic ODMP protocol with self-improvement algorithms to provide a better service hierarchy for delivery, regarding its scalability, efficiency and reliability. The proposed self-improvement mechanisms are actually protocol-independent. They can be
applied to any overlay hierarchy, such as mesh based or tree-based structure. In this paper, we implement them into a ODMP framework to check how the mechanisms about self-improvement can be used to improve the quality of data delivery hierarchy.

The remainder of this paper is organized as follows. Section 1.1 presents two self-improvement techniques proposed by the ODMP-E protocol, namely, mesh self-improvement and cluster self-improvement. Section 1.2 focuses on proving the performance of the ODMP-E protocol. For the analysis of performance, we have considered two aspects through the simulation, i.e., data path quality, control overhead and packet loss. The comparisons in Section 1.2 gives us the correctness of our hypothesis that self-improvement mechanisms enhance the reliability and also improve the efficiency of the data delivery hierarchy. Section 1.3 gives a summary of this paper.

1.1Self-improvement Mechanisms

The ODMP framework depends on a dynamic mesh to distribute the media data to a large number of end hosts. However, the mesh constructed can be sub-optimal, because (i) initial neighbor selection during a member joining in the group have limited topology information; (ii) changes of the members dynamically due to group member joining or leaving; (iii) underlying network conditions, such as traffic load ,routing may be varying. ODMP-E provides two self-improvement mechanisms in order to gracefully enhance the performance of the ODMP framework (e.g., to expand available bandwidth over the network).

1.1.1 Mesh Self-improvement

In ODMP-E, the members of the mesh periodically exchange messages with each other in order to track the dynamic changes of other members in a mesh. As we mentioned in Section 4.5.1, the number of interleaved spanning tree has a great effect on the efficiency of the data delivery. However, because of dynamic membership changes the quality of the established mesh may degrade. ODMP-E permits an incremental improvement of mesh quality by dropping low-performance links and adding additional high-performance links .

Join Additional Mesh Link

Members of Mesh probe each other at random and new links can be added depending on the perceived utility gain. Here, to reflect the mesh quality an utility is defined in Algorithm 1.1.1 following which , members continue to monitor the utility of the existing links, and drop links which are perceived as useless. Our purpose here is to provide a good-quality mesh which can ensure between any pair of mesh members, the paths within the mesh provides better performance compare to the performance provided by the unicast path between them. To illustrate an idea, we suggests an example of adding useful links between a pair of ODMP members in a mesh. Let v be a super node, v selects randomly a non-adjacent super node, say u, and sends a request for u's routing table as well as current available mesh degree. Upon receiving the routing table, v evaluates the utility of the link {u, v}, namely, U(u,v). Suppose as the routing metric latency is used, v measures the RTT (Round-Trip Time) between u and v. In this case, the utility U(u,v) is calculated using the algorithm 1.1.1.

The above algorithm is similar to the one proposed in [91], however, different in design function we extend this algorithm with gain considerations: v calculates how a performance of its routes could be significantly change if link {u,v} is added. i.e, we propose an utility threshold to evaluate whether the adding link between them is desirable. Such a threshold depending on the number of existing super nodes, and the available degree of mesh for u and v. If U(u,v) is above the threshold, v should send a request, "AddLinkMsg" in our implementation, to u if u still have available mesh degree.

Algorithm 1.1: Utility for Mesh Self-Improvement

For for member v **do**
U(u,v) = 0
CL = current latency in u and v within the mesh improvement
NL = new latency between u and v within the mesh if new edge u-v added
if NL < CL then
U(u,v) + =(CL – NL)/CL
end if
end for

The message flow is shown in Fig 1.1.

Simply, u will not refuse for the addition of the link but at the same time if there is a super node s, which adds a link {u, s}, possibly ,the super node u breaks the connection to a non-super node .

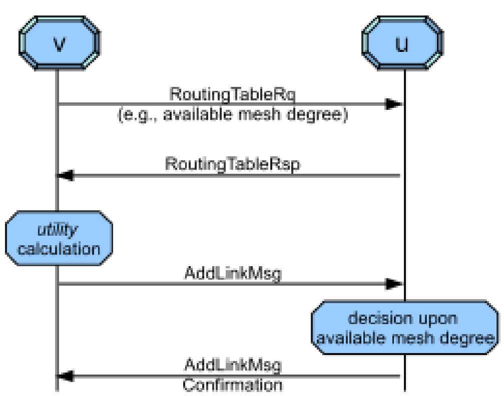

Fig 1.1 : Example of additional mesh links

If it fails, since u's consensus threshold increases, u immediately drops a link anyway.

Deletion of Mesh Link

Besides periodical adding the links to a mesh, each member of mesh periodically considers to drop an inefficient link. Dropping an link is easy than adding an extra link as it may require no message exchange. For example, v updates the last routing updations received from neighbors and computes the consensus cost of every link, such as {u,v} as described in [91]. However, each group member can be selected as a source in Narada. In our case, we consider only one source and therefore, it is useful to focus on optimization of the routes from receivers to the source. In our implementation, it is not feasible to drop a link which is within the shortest path between v and the source, or between u and the source. Otherwise, it effects on all optimal links of v and u. As a complimentary to [91], we propose the approach as given. Suppose c is the minimum value of the calculated consensus costs and If c is less than the consensus threshold that relies on v's current available degree of mesh and the number of super nodes, the corresponding link is dropped {u,v}. To drop the link, v notifies u by sending a "DropLinkMsg", and both of their super nodes update their internal database, in particular, the routing tables. If a "DropLinkMsg" is somewhere lost, v sends a message again when it receives a next refresh message from u.

1.1.2 Cluster Self-improvement

Here we suggest the self-improvement mechanisms as a cluster member having a higher capacity than its parent node can be promoted. Usually , the direct children and their parent can swap their positions. After promotion, the former child becomes the new parent and its former parent may become the current child. Still, there are some factors which affect the mechanism:

- the involved number of nodes in the promotion
- the participated nodes reliability .
- after promotion, the existing tree re-construction may be

complicated since the promoted child may not have enough bandwidth to accept all existing end hosts as its children. We describe a idea by taking the following example shown in Fig 1.2, suppose that node 1.1.2 has high capacity than its parent node 1.1 based on the capacity comparison. Then, node 1.1.2 sends a promotion request to node 1.1. After some authorization check, node 1.1 acknowledges the request and sends back a status report which contains the address of node s. Here, it is necessary that node 1.1 waits till node s has received a breakup request. Otherwise, the join request from 1.1.2 may arrive earlier, which causes a loop in the overlay tree.

Then, node 1.1 breaks the connections with node s and 1.1.2. However, node 1.1 keeps node s as its backup parent in case node 1.1.2 is leaving or unreachable. Moreover, node s considers node 1.1 as it temporary child. At the same time, node 1.1.2 contacts node s and notifies node 1.1 to be its child. Once node 1.1 receives the notification and rejoins the tree as the child of node 1.1.2, it may break the connection with node 1.1.1 if node 1.1.2 still has available capacity. In the following example, node 1.1.2 can support at least three children. Therefore, after the first swap, the node 1.1.1 requests to join as one child of node 1.1.2.

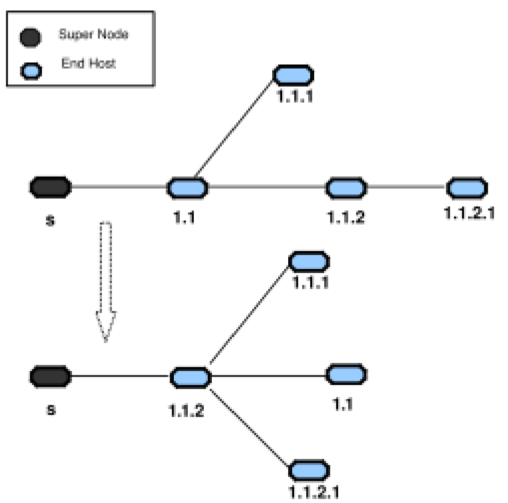

Fig: 1.2 Example of Cluster self improvement

As shown in Fig 1.3. the message flow of the promotion example, Because of our strategy of constructing the overlay hierarchy, nodes at the bottom of the overlay are either leaf nodes ,transient nodes or new comers. Cluster improvement algorithm described above allows the newcomers who have higher capacities can “climb” from the bottom level to a higher level after some switching stages. In real facts, this is important for new customers which have higher capacity and willing to share their resources can get better quality of service. For example, a new comer at the lower level could switch with its parent if its capacity exceeds the current parent (over a predefined threshold). However , an appropriate threshold (defined in Sec 1.2.1) should be chosen to avoid unnecessary switching as if the child has a smaller bandwidth support, it will be ultimately placed below the parent. The main aim of this is to reduce the effects of frequent changes in the overlay so that only a small part of a overlay multicast tree will be affected and needs to be re-constructed after the dynamic changes.

1.1.3 Extended Messages

We need to extend the messages defined in paper 3 to support self-improvement mechanisms. Table 1.1 lists the ODMP-E messages. In this ,Routing TableReq and RoutingTableResp messages are used to get the information for

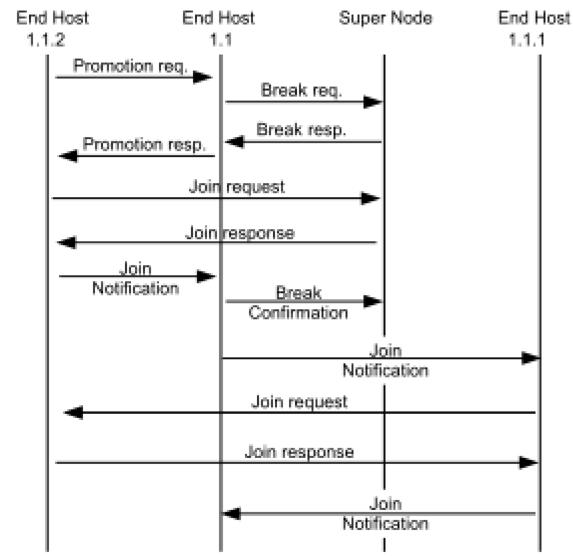

Fig: 1.3 Message flow of self improvement

evaluating random links. If the adding of links is decided by one member in a mesh, it sends AddLinkMsg for adding the evaluated link to another member in a mesh. Similarly, DropLinkMsg is used to request for dropping unuseful links. Note the first six messages are extended to support mesh self-improvement mechanisms. Therefore, they are only exchanged between members in a mesh.
For the self-improvement mechanisms in the cluster, the last six messages are defined. Meanwhile, Promotion Request and Promotion Response messages are used to determine whether the swap between parent node and the child node is allowed. Once a promotion is confirmed, the child node tries to break the existing links with other nodes through Break Request and Break Response messages. additionally, Join Notification is used to notify the partitioned nodes to rejoin the group.
In addition, these partitioned nodes can request the new parent for joining the group. When such a joining procedure fails, they can follow the initial join phase [1].

1.2 Performance Evaluation

The established mesh core may not be optimal due to the frequent member joining/leaving as observed and implies the quality of path length may be also affected. Therefore, we wish to implement a mesh self-improvement mechanisms to enhance the mesh performance. Due to this reason, we implement the self-improvement in each cluster.

However ,In this section, we proves the effectiveness of the two self-improvement mechanisms through measuring the data path length, loss rate ,stress and control overhead.

Messages	From	To	Operation
RoutingTableReq.	Mesh Member	Mesh Member	Links Evaluation
RoutingTableResp.	Mesh Member	Mesh Member	
AddLinkMsg	Mesh Member	Mesh Member	Adding Links
AddLinkMsg Confir-mation	Mesh Member	Mesh Member	
DropLinkMsg	Mesh Member	Mesh Member	Dropping Links
DropLinkMsg Confir-mation	Mesh Member	Mesh Member	
Promotion Req.	Cluster Member	Cluster Member	Promotion
Promotion Resp.	Cluster Member	Cluster Member	
Break Req.	Cluster Member	Cluster Member	Break Connection
Break Resp.	Cluster Member	Cluster Member	
Break Confirmation	Cluster Member	Cluster Member	
Join Notification	Cluster Member	Cluster Member	Member Join

1.2.1 Simulation Setup

We use approach to build the underlying network topologies in the following simulation:

1) NED-oriented topology generation

For the performance evaluation of ODMP-E , following parameters are configured

- Target Overlay Terminal Num: Values between 128 and 2,048.
- Threshold for Promotion: threshold = 100,000 bit = 100 kb. As the multimedia session sends data at a constant bit rate of 128 and 256 kbps.
- we set low bandwidth values for the end hosts. It is possible to support up to 2 Mbps bit rate.
- Refresh Mesh Interval: 2.5 seconds. It defines the period for refreshing the mesh core.
- Refresh Timer: 3.5 seconds. It is used to periodically trigger the message exchanges among cluster members.
- Utility Interval: 5.0 seconds, which is a period used to consider adding/deleting random links.

In the following comparisons, The entire set of members join in the first 200 sec, and run the simulation for another 1,800 sec to allow a topology to be stabilized. i.e, the performance of ODMP-E will be evaluated using two phases: joining phase and stabilization phase.

1.2.2 Topology : Scenario

As Narada does not scale well (support up to 512 end hosts), in this scenario we only perform the experiments and compare with NICE. We measured the router stress, link stress and control overhead regarding different group sizes.

Fig 1.4 shows the comparison of router stress with various group sizes. ODMP-E has much less router stress (around 20%) as Compared with NICE, in addition , the performance of ODMP-E is quite stable regardless of the group size changes. When the group size becomes larger ,the performance of ODMP-E still keeps stable, but the router stress caused by NICE increases .

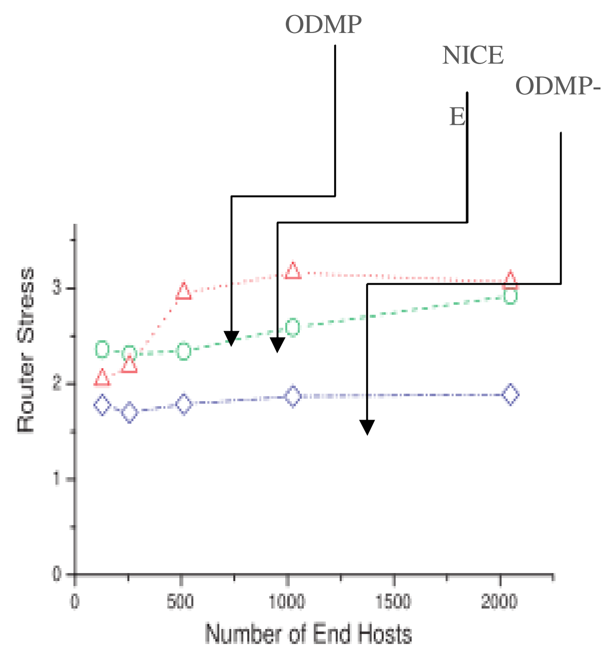

Fig :1.4 Comparison of Router Stress

In a stabilized circumstances, the self-optimized ODMP-aware mesh and clusters can provide high efficiency of data delivery by adding efficient links and deleting inefficient links and as well as promoting high capable nodes near the overlay core. In contrast ,, when the group size becomes large NICE has even deeper layered hierarchy and more separated clusters within each layer, which enlarges the possibility of redundant transmission over the routers.

The router stress produced by ODMP is relatively stable. Fig 1.5 shows comparison of link stress between NICE, ODMP and ODMP-E. Usually, ODMP-E can achieve much better performance than ODMP. Also, because of the gradually optimized overlay hierarchy the link stress of ODMP-E is quite stable. Such an observation exactly results that the ODMP-E protocol is very important and helpful to improve the efficiency and stability of the data delivery in the ODMP framework.

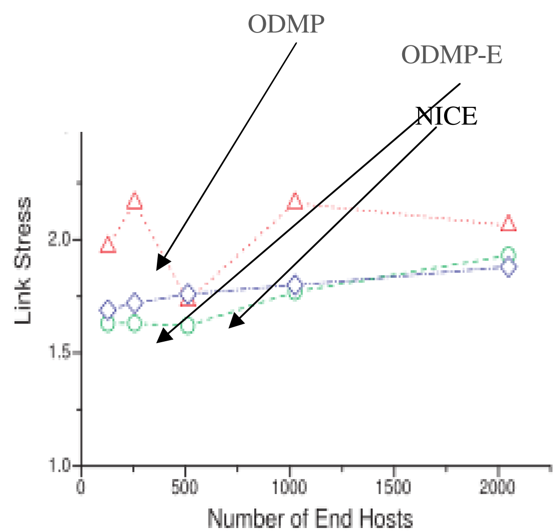

Fig1.5 : Comparison of link stress

ODMP-E has comparatively less link stress though its underlying network as compared with NICE, when configd with 5,000 routers. Specifically when the group size is less than 1,000, the router stress of the ODMP-E protocol is up to 15% less than that of NICE. When the group size is larger than 1,500, the performance of NICE is a bit better than ODMP-E.

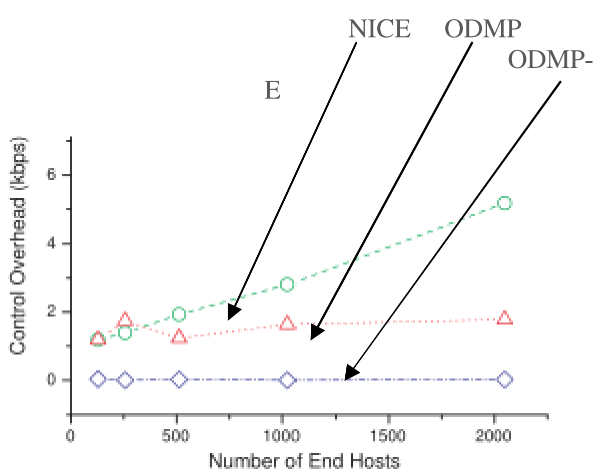

Fig 1.6:Comparison of Control Overhead

NICE overlay hierarchy have control overhead is much higher than both ODMP and ODMP-E. As the control overhead of NICE is become serious in a highly dynamic environment. The main reason causes such an phenomenon is that maintenance of multi-layered hierarchy is very costly especially when the group size is large. Since the optimized overlay hierarchy can reduce the message exchanges for joining the multicast session, the control overhead causes by ODMP-E is even less than ODMP. Suppose high capacity nodes have been promoted to the high level of the hierarchy, it is easier for newcomers to find available parents to join the group.
Which concludes:
1) ODMP-E can achieve more efficient data delivery than NICE and ODMP.
2) ODMP-E can largely enhance the performance of ODMP in a relatively stable environment.
3)control overhead caused by the ODMP and ODMP-E protocols is stable and much less than that of NICE, even when the group size becomes large.

1.3 Summary

Self-improved ODMP protocol (ODMP-E) is proposed to periodically optimize the established ODMP-aware overlay mesh and clusters.

To improve the quality of service delivery hierarchy, we define a utility threshold in the mesh self-improvement to prove whether the adding link between super nodes is desirable. This threshold can drastically reduce the cost of unnecessary link changes.

ODMP-E has been validated to further enhance the performance of ODMP in terms of scalability, reliability and efficiency. The simulation results have proved that self-improvement mechanisms can largely help optimizing the ODMP clusters in reducing control overhead and the packet loss .

2. Conclusion

For link stress, ODMP-E can achieve very competitive performance .The performance of ODMP-E is better than that of NICE with much less router stress and much less control overhead. Overall, we conclude thatODMP-E can assist the ODMP framework to be scalable, stable and efficient in supporting large-scale media distribution services.

References

[1] ODMP: A new Overlay Dynamic Multicast Protocol framework , International Journal of Scientific and Research Publications, Volume 3, Issue 7, July 2013 ISSN 2250-3153

[2] J. Lei, X. Fu, and D. Hogrefe, "DMMP: A New Dynamic Mesh-based overlay Multicast Protocol Framework ," IEEE Consumer Communications and Networking Conference - Workshop on Peer-to-Peer Multicasting (P2PM 2007), 2007.

[3] Y. Cui, B.-C. Li, and K. Nahrstedt, "oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks," IEEE Journal on Selected Areas in Communications, Special Issue on Recent Advances in Service Overlay Networks,Vol. 22, No. 1, pp.91–106, 2004.

[4] X. Hei, C. Liang, Y. Liu, and K. Ross, "A Measurement Study of a Large-scale P2P IPTV System," IEEE Transactions on Multimedia, Vol. 9, No. 8, pp.1672-1687, 2007

[5] F. Wang, J. Liu, and Y. Xiong, "Stable Peers: Existence, Importance, andApplication in Peer-to-Peer Live Video Streaming," IEEE INFOCOM, 2008.

[6] Y.-H. Chu, S.-G.Rao, and H. Zhang, "A case for End System Multicast," SIGMETRICS,2000.

[7] H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider Portal for Applications," in Proc. of Sigcomm, 2008

[8] M. Castro, M.-B.Jones, A.-M.Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman, "An Evaluation of Scalable Application-level Multicast Built Using Peer-to-peer Overlays," in Proc. of IEEE INFOCOM'03, San Francisco, California, USA, 2003.

[9] S. Banerjee and B. Bhattacharjee, "Analysis of the NICE Application Layer Multicast Pro-tocol," UMIACS Technical Report TR 2002-60 and CS-TR 4380, 2000.

[10] ns-2 simulator, http://www.isi.edu/nsnam/ns/.

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

[12] J. Lei, L. Shi, and X. Fu, "An experimental analysis of joost peer-to-peer vodservice,"Technical Report No. IFI-TB-2007-03 Institute for Computer Science,University of Goet-tingen, Germany, ISSN 1611-1044, 2007.

[13] O. Babaoglu, H. Meling, and A. Montresor, "Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems," 22nd IEEE International Conference on Distributed Computing Systems (ICDCS'02), 2002.

[14] S. Guha, N. Daswani, and R. Jaine, "An Experimental Study of the Skype Peer-to-Peer VoIP System," in Proc. of IPTPS'06, 2006.

[15] C. K. Yeo, B. S. Lee, and M. H. Er, "Application layer multicast architecture for media streaming," in Proceedings of a 7th IASTED International Conference on Internet and Multimedia Systems and Applications, pp. 455–460, Honolulu, Hawaii, USA, August 2003.

First Author
Mr. Prashant B Kumbharkar
ME (Comp.Engg)
Assoc. Professor ,Siddhant College Of Engg, Pune
Exp:19Yrs

Second Author
Dr. Trimbak R Sontakke
ME,PhD (E&Tc)
Ex-Director ,SGGSNanded
Exp:40Yrs