On the Performance of Overlay Multicast Networks

Doru Constantinescu and Adrian Popescu

Dept. of Telecommunication Systems
School of Engineering
Blekinge Institute of Technology
371 79 Karlskrona, Sweden

Abstract— **The paper reports on a performance study of several Application Layer Multicast (ALM) protocols. Three categories of overlay multicast networks are investigated, namely Application Level Multicast Infrastructure (ALMI), Narada and NICE is the Internet Cooperative Environment (NICE). The performance of the overlay multicast protocols is evaluated with reference to a set of performance metrics that capture both application and network level performance.**

The study focuses on the control overhead induced by the protocols under study. This further relates to the scalability of the protocol with increasing number of multicast participants. In order to get a better assessment of the operation of these protocols in "real-life"-like conditions, we implemented in our simulations a heavy-tailed delay at the network level and churn behavior of the overlay nodes. Our performance study contributes to a deeper understanding and better assessment of the requirements for such protocols targeted at, e. g., media streaming.

I. Introduction

Overlay networks have emerged as a viable solution to the problem of content distribution with multicasting and Quality of Service (QoS) facilities. Overlay networks are networks operating on the inter-domain level, where the edge hosts learn of each other and, based on knowledge of underlying network performance, form loosely coupled neighboring relationships. These relationships are used to induce a specific graph, where nodes represent hosts and edges represent neighboring relationships. Important research questions in this context are protocol scalability, overlay traffic measurements and modeling, efficient data distribution, search and retrieval, load balancing, churn handling, QoS provisioning with multicast or multipath facilities as well as congestion and error control in multicast environments [16].

In general, the schemes for Application Layer Multicast (ALM) can be classified into two main categories: *structured* [4], [20] and *unstructured* [3], [12]. In this paper we focus on unstructured ALM solutions.

A number of research studies have been done on the performance of ALM protocols. Some of them focus on a more theoretical comparison of different ALM solutions such as those presented in [2], while others compare the ALM proposals based on simulation studies [10], [17], [18].

Although several comparative studies have been carried out, we are however not aware, to this day, of any similar simulation study of the ALM protocols analyzed and presented in this paper. We conducted our study by first measuring and modeling the one-way delay process and associated components, as observed in real Internet Protocol (IP) networks. Thereafter, we used these results for the simulation study of the overlay multicast protocols, and further included churn behavior of the participating overlay nodes. This was done in order to have a more realistic set of conditions for the results reported in this paper as well as a strong foundation for future research work.

The remaining of this paper is organized as follows. Section II provides a short description of overlay multicast algorithms followed by details of the simulation environment used in Section III. The paper continues with a presentation of the performance evaluation criteria employed in Section IV. Details about the simulation experiments performed are provided in Section V. The simulation results obtained are reported in Section VI. Finally, Section VII concludes this paper by providing several important observations regarding our performance study.

II. Multicast Algorithms

Today, there are a number of architectural solutions and implementations for multicast overlay networks. These are Peer-to-Peer (P2P) multicast, Overlay Multicast (OM) and Waypoint Multicast (WM) [14]. Each of them has been specifically developed for special conditions and purposes.

Generally, a multicast overlay network has two topologies associated with it, namely the data topology and the control topology. Periodic messages are usually exchanged among the members in the control topology to identify and to recover from situations created by group changes. On the other hand, the data topology, which is usually a subset of the control topology, is composed by data paths involved in data distribution through the multicast group.

There are two classes of algorithms defined for building multicast overlay networks, which are based on using *centralized* or *distributed* topology building algorithms [8], [9]. The centralized algorithms use centralized algorithms for control and data distribution. They can be further partitioned into two classes, which are *full membership knowledge* and with *partial membership knowledge*.

The following three protocols have been selected for further performance study and analysis: Application Level Multicast Infrastructure (ALMI), Narada and NICE is the Internet Cooperative Environment (NICE). It is considered that these protocols are quite representative for the analysis of multicast overlay networks, based on their concept, relevance for the study and complexity. Details about these protocols are provided in [2], [7], [12].

III. Simulation Environment

For the Application Layer Multicast (ALM) simulation study presented in this paper, we used the `myns` simulator [1], further developed for our purposes. The simulation engine is modified to generate heavy-tailed delays for each participating node in a multicast session. Moreover, the runtime scripts are modified to allow for simulating the churn behavior of the participating nodes. In addition, the nodes and agents are adapted to evaluate the overlay performance metrics as described in [7].

The `myns` simulator incorporates the Georgia Tech - Internetwork Topology Models (GT-ITM) topology generator and implements the Narada and NICE ALM protocols. Further, an implementation of a sequence of directed unicasts at application level is also provided, which can be used as a template for further implementations of various ALM protocols in the `myns` simulating engine.

Simulator operation is achieved through script parsing. The runtime scripts can either be manually configured as for the case of simpler topologies containing only a few nodes, or more complex GT-ITM generated topologies, parsed by the simulation engine as in the case of topologies requiring several hundred/thousands of nodes. In addition, the ALMI protocol architecture have been implemented and integrated in the `myns` simulation engine [7]. The implementation closely follows the definition of ALMI protocol operation as described in [12], [15].

IV. Performance Evaluation Criteria

The evaluation criteria used in our experiments is one in which we compare ALMI, Narada and NICE with reference to the following parameters [5], [6], [9], [14]:

- *Protocol scalability* in terms of how well the protocol behaves with increasing number of group members. We evaluate the scalability with reference to the following metrics: *control overhead*, *efficiency*, *link stress*, *link stretch*, *resource usage*, and *time to first packet*; all of them functions of the number of nodes in the overlay network.
- *Protocol dynamics* in terms of protocol adaptability with different group sizes and node behavior. This can be evaluated by looking at the data delivery path in terms of, e. g., the number of control messages generated in the case of high churns for various group sizes. We evaluate the protocol dynamics with reference to the following metrics: *control overhead*, *efficiency*, *link stress*, *link stretch*, *losses* and *stability*; all of them functions of the churn generated in the overlay network.
- *QoS provisioning capability* can be evaluated by emulating "real-world"-like conditions for the protocol data delivery. This criteria can be evaluated by enforcing that each data packet has a certain delay at the network layer. This delay is drawn from a heavy-tailed distribution as observed in real network conditions. Further, we use the solution of emulating different types of applications such as to create "real-world"-like conditions. We evaluate the QoS provisioning capability by looking at the delay performance in the overlay network (in terms of the link stretch), and the corresponding delay of the direct IP path between the same nodes.

Further details and the definition of these performance metrics are presented in [7].

V. Simulation Experiments

Two sets of experiments were performed for the purpose of this paper. In the first set of experiments, the multicast behavior of the ALM protocols was simulated. The following group sizes were considered, for each protocol: 16, 32, 64, 128, 256, 512, and 1 024. For every group size, the simulation run was configured with a duration of 1 500 s, including a "warm-up" period of 300 s [11].

During the warm-up period no data packets were transmitted, but only control messages. This was done with the purpose of allowing the overlay multicast protocols to come to a steady state before the overlay multicast session was initiated. For every group size, all participating overlay nodes (including the one selected to act as source) were initialized and randomly attached to the topology nodes, within the first 100 s of simulation time. The topologies were generated with GT-ITM, and each topology was a two-level hierarchical transit-stub topology, containing 1 250 nodes and about 6 000 links [19]. Each generated topology was configured to have 5 transit domains with 10 nodes each, 2 stub domains per transit node, and 12 nodes per stub domain. The average node degree for the generated topologies was between 3 and 4.

The second set of experiments was configured to have similar conditions as those explained above, the main difference, however, being that a number of the active overlay nodes were now configured to undergo churn. Further details on the simulation environment, the simulation characteristics and the churn model used for these experiments are provided in [7].

Each protocol, group size, and churn parameter was simulated 32 times in both set of experiments. In general, each simulation run completed within several minutes, except for the simulation runs involving larger group sizes (e. g., 1 024 nodes) in the Narada protocol, which lasted much longer (in the order of several hours each).

VI. Simulation Results

In this section we evaluate the ALM protocols with regard to the performance evaluation criteria defined in Section IV. It is also mentioned that the reported results are presented as averages over all simulation runs including the 95 % Confidence Intervals (CIs) for the respective metric.

For the evaluation of the protocol dynamics, a series of experiments have been done that included churn behavior of the active overlay multicast nodes. This section reports the main results obtained in these experiments. Due to the extent of these simulations and space limitations, we only report in this paper selected simulation results.

TABLE I

Identifying churn experiments.

Nodes `Off`-state	Churn 10 %	Churn 30 %	Churn 50 %
3 minutes	churn$_{10-3}$	churn$_{30-3}$	churn$_{50-3}$
5 minutes	churn$_{10-5}$	churn$_{30-5}$	churn$_{50-5}$
7 minutes	churn$_{10-7}$	churn$_{30-7}$	churn$_{50-7}$

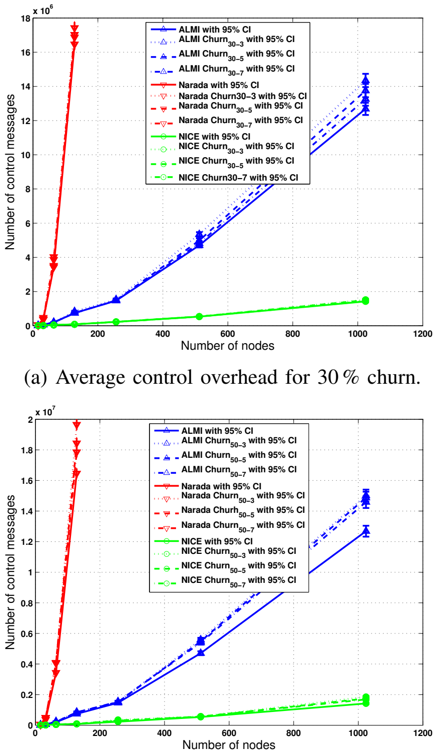
Moreover, when concerning a particular churn simulation parameter, we refer to the churn parameter classification described in Table I. In this way, a given experiment with specific churn parameters is easily identified. A more complete set of results for the performance study reported here are provided in [7].

A. Control Overhead

The average control overhead for the overlay multicast protocols, when the overlay nodes undergo churn with 30 %, and 50 %, is illustrated in Figures 1(a) and 1(b), respectively. The figures show the results obtained in these cases, when the respective fraction of the overlay nodes leave, and rejoin the overlay multicast. For illustration purposes, the cases with no churn are also included in the figures.

We observe a similar behavior for all protocols, namely that the number of control messages increases with increasing churn rates. We also note that all protocols exhibit a more oscillating behavior at higher churn rates, which can be explained by the higher variance obtained in these cases. This is believed to be a direct result of the overlay multicast tree maintenance, and the high variance of the results corroborate this observation.

These results are considered to be reasonable since with increasingly higher churn rates, up to half of the nodes in the overlay can undergo churn and the protocols become very busy maintaining the multicast delivery tree. Furthermore, the general shape of the control overhead averages in both Figure 1(a) and 1(b) match the shape of the case with no churn. This similarity indicates that the general behavior of the protocols does not alter with churn.



(a) Average control overhead for 30 % churn.

(b) Average control overhead for 50 % churn.

Fig. 1. Selected simulation results for average control overhead.

However, the values obtained for the control overhead are much higher in the case of churn and with higher variance that increases with higher churn rates. This adversely affects the overall performance for all protocols since much more control traffic is needed in this case for maintaining the overlay multicast tree.

We further note that, at high churn rates (e. g., churn$_{50-3}$ from Figure 1(b)), Narada may become overwhelmed with control messages and consumes much resources in order to maintain the mesh. At small group sizes, however, Narada seems to be more robust to higher churns than the other two protocols.

A similar result is observed for ALMI where a similar behavior is noted. NICE is observed to be more robust to churn than the other two protocols. It is noted, however, that also in this case, the control overhead variance increases noticeably with higher churn rates, with the consequence of an oscillating behavior of the protocol.

B. Overlay Efficiency

Figure 2 illustrates the overlay efficiency as a function of churn in the case of churn$_{50-3}$. For comparison purposes, the efficiency in the case of no churn is also plotted.

We observe for all protocols that the efficiency metric η increases with the group size despite the fact that almost half the nodes leave and rejoin the overlay multicast session. We note that regardless of the churn behavior, all overlay protocols are still more efficient than overlay unicast transmissions.

We also observe that for smaller group sizes, Narada performs better than both ALMI and NICE. Similar to the case of no churn, its performance drops with increasing group size but even at the largest group size simulated it is still more efficient than overlay unicast.

NICE shows a stable evolution in its efficiency since this continuously increases with the overlay multicast group size, in spite of the fact that the churn rates are quite high.

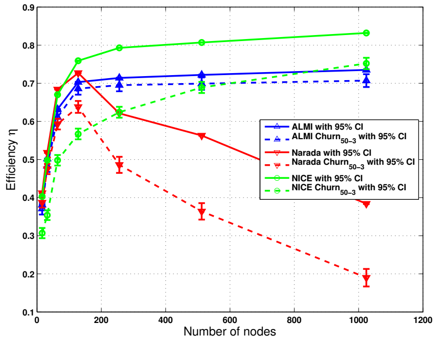


Fig. 2. Average efficiency for churn$_{50-3}$.

ALMI exhibits also a stable performance with comparable results as in the case of no churn. However, the variance is much higher in this case, but the use of the Minimum Spanning Tree (MST) seems to be a good choice for this protocol. We further observe that the efficiency metrics are lower when the overlay nodes undergo churn than in the case of no churn, and the metric decreases with higher churn rates.

A general observation regarding all protocols is the higher variances, which increase with group size as observed in our experiments. The variance for higher churn rates is observed to be larger than one order of magnitude as compared to the case without churn. This is an expected results since our simulation scenarios test the protocols under extreme conditions [7].

C. QoS Provisioning Capability

For assessing the QoS provisioning of the protocols we looked at the average delay for data packets in the overlay multicast tree. In Figure 3 we show the results from three simulations runs; simulation without churn, churn$_{10-3}$, and churn$_{50-5}$.

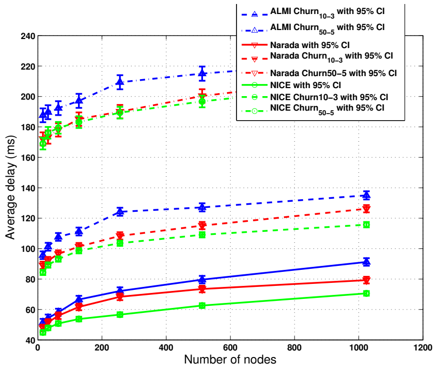


Fig. 3. Average mean delay for three different simulation sets.

It is observed that ALMI has the highest delays in all three cases, and NICE the lowest delays. Also, in the case of churn, all three protocols show an increase in the average delay, and for larger group sizes and higher churn rates the delay passes over 200 ms. This increase is a direct consequence of churn since the rebuilding of the data paths due to churn leads to suboptimal paths.

All three protocols try to refine the distribution tree when the participating members fail or leave and, with increasing churn rates, a best cost overlay delivery tree (regardless of the metric used) is increasingly harder to achieve. The oscillations in the tree rebuilding mechanisms directly and adversely impact on the delay, as observed in our simulations.

The fact that NICE and Narada use source-specific trees for data delivery explains their better performance with regard to delay, in comparison with ALMI, which uses a (shared) MST. The fact that the MST in ALMI is calculated by the session controller (i. e., centralized approach) impacts on ALMI's delay bottleneck as observed in our simulations.

On the other hand, although Narada and NICE perform better than ALMI, a clear difference between their delay performance is observed. NICE has lower delay than Narada in all simulation scenarios, which is a direct result of the hierarchical approach used by NICE where the cluster members select their neighbors based on minimum end-to-end (e2e) latency of the corresponding unicast path. The lower performance observed for Narada is a consequence of the overlay group size and its inherent larger control overhead required for multicast tree maintenance.

VII. Concluding Remarks

Based on our experiments, several significant conclusions can be drawn about the selected ALM protocols. From our simulations, we observe that the protocols that use source-specific, distributed group/tree management show better performance. Our research focus is on real-time streaming applications and, from this perspective, such an approach seems to be the better solution.

Another important observation is regarding the resources required and consumed by an ALM protocol. We observe that all three protocols show higher efficiency than the overlay unicast approach, and the best performance with regard to resource usage is given by a MST approach. Further, it is observed that the distributed mesh induces the highest protocol overhead, which in turn, limits the scalability of the protocol in terms of number of users.

Another fundamental aspect is the inherent churn behavior of ALM-based protocols. Consequently, the selected ALM protocols are configured to emulate this behavior in order to observe the protocol dynamics (i. e., tree refinement) under extreme conditions. Tree refinement is absolutely necessary in order to provide a close to optimal protocol performance.

However, this also involves increased control overhead, and this must taken into consideration in order to have a scalable protocol. Shared trees are able to provide low cost trees, while source-specific trees provide lowest delay. From our simulation results, a hierarchical source-specific approach seems to be a good compromise, but a hybrid approach (i. e., hierarchical multiple shared trees) can be the optimal solution.

This paper was devoted to a performance study of overlay multicast networks. Three fundamental categories of overlay multicast networks and associated algorithms were considered in this study. The evaluated overlay multicast protocols are ALMI, Narada and NICE.

The performance of these overlay multicast protocols was evaluated through a comprehensive simulation study with reference to a detailed set of performance metrics that captured both application and network level performance. A particular interest was given to the issues of scalability, protocol dynamics and delay optimization as part of a larger problem of performance-aware optimization of the overlay networks.

The performance study presented in this paper represents the essential foundation for our future work. The goal is to implement an ALM protocol designed for real-time media streaming and content distribution. Our protocol uses QoS routing information gathered from the overlay multicast members, while e2e reliability of data delivery is provided by using a rate-based congestion control mechanism and Forward Error Correction (FEC).

The scalability and performance of the protocol will be analyzed by extensive simulations and by deployment and testing it in real networks by using the PlanetLab environment [13].

References

[1] Banerjee, S.,"myns Simulator",
`http://www.cs.umd.edu/~suman/research/myns/`

[2] Banerjee, S. and Bhattacharjee, B., "A Comparative Study of Application Layer Multicast Protocols", *preprint*, Department of Computer Science, University of Maryland, USA, 2002.

[3] Banerjee, S., Bhattacharjee, B. and Kommareddy, C., "Scalable Application Layer Multicast", *In Proceedings of the ACM SIGCOMM 2002*, pp. 205–217, Pittsburgh, USA, 2002.

[4] Castro, M., Druschel, P., Kermarrec, A.-M. and Rowstron, A., "Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure", *IEEE Journal of Selected Areas in Communications*, Vol. 20, No. 8, pp. 1489–1499, 2002.

[5] Chalmers, R. C. and Almeroth, K. C., "Modeling the Branching Characteristics and Efficiency Gains of Global Multicast Trees", *In Proceedings of the IEEE INFOCOM 2001*, Vol. 1, pp. 449–458, Anchorage (Alaska), USA, 2001.

[6] Chu, Y.-h., Rao, S. and Zhang, H., "A Case for End System Multicast", *In Proceedings of the 2000 ACM SIGMETRICS*, pp. 1–12, Santa Clara, USA, 2000.

[7] Constantinescu, D., "Overlay Multicast Networks: Elements, Architectures and Performance", *PhD dissertation*, Blekinge Institute of Technology, Karlskrona, Sweden, 2007.

[8] El-Sayed, A., "Application Level Multicast Transmission Techniques Over the Internet", *PhD dissertation*, Institut National Polytehnique de Grenoble, Grenoble, France, 2004.

[9] El-Sayed, A., Roca, V. and Mathy, L., "A Survey of Proposals for an Alternative Group Communication Service", *IEEE Network*, Vol. 17, No. 1, pp. 46–51, 2003.

[10] Fahmy, S. and Kwon, M., "Characterizing Overlay Multicast Networks and Their Costs", *IEEE/ACM Transactions on Networking*, Vol. 15, No. 2, pp. 373–386, 2007.

[11] Law, A. M. and Kelton, W. D., "Simulation Modeling & Analysis, Third Edition", *McGraw-Hill*, 2000.

[12] Pendarakis, D., Shi, Y. S., Verma, D. and Waldvogel, M., "ALMI: An Application Level Multicast Infrastructure", *In Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS 2001)*, pp. 49–60, San Francisco, USA, 2001.

[13] PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services,
`http://www.planet-lab.org/`

[14] Popescu, A., Constantinescu, D., Erman, D. and Ilie, D., "A Survey of Reliable Multicast Communication", *In Proceedings of the 3rd Euro-NGI Conference on Next Generation Internet Networks (NGI 2007)*, pp. 111–118, Trondheim, Norway, 2007.

[15] Shi, Y. S., "Design of Overlay Networks for Internet Multicast", *PhD dissertation*, Sever Institute of Technology, Washington University, Saint Louis, USA, 2002.

[16] Shi, Y. S. and Turner, J. S., "Multicast Routing and Bandwidth Dimensioning in Overlay Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 8, pp. 1444–1455, 2002.

[17] Shi, Y. S. and Turner, J. S., "Issues in Overlay Multicast Networks: Dynamic Routing and Communication Cost", *Technical Report*, Washington University in Saint Louis, WUCS-02-14, USA, 2002.

[18] Wu, S. and Banerjee, S., "Improving the Performance of Overlay Multicast with Dynamic Adaptation", *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC 2004)*, pp. 152–157, Las Vegas, USA, 2004.

[19] Zegura, E. W., Calvert, K. L. and Battacharjee, S., "How to Model an Internetwork", *In Proceedings of the IEEE INFOCOM 1996*, Vol. 2, pp. 594–602, San Francisco, USA, 1996.

[20] Zhuang, S. Q., Zhao, B. Y., Joseph, A. D., Katz, R. H. and Kubiatowicz, J. D., "Bayeux: An Architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination", *In Proceedings of the ACM NOSSDAV 2001*, pp. 11–20, New York, USA, 2001.