

Bandwidth Intensive Application-layer Multicast in Dynamic Environments

Stefan Birrer

Advisor: Fabián E. Bustamante

Department of Computer Science

Northwestern University, Evanston, IL 60201, USA

Email: sbirrer@cs.northwestern.edu

I. INTRODUCTION

High-bandwidth multicast among widely distributed nodes is critical for a wide range of important applications including audio and video conferencing, multi-party games and content distribution. Multicast decouples the size of the receiver set from the amount of state kept at any single node and potentially avoids redundant communication in the network.

The limited deployment of IP Multicast has led to considerable interest in alternate approaches that rely only on end systems [11], [14], [1], [20], [24], [9]. In an end system multicast approach, participating end hosts organize themselves into an overlay topology for data delivery. Each edge in the topology corresponds to a unicast path between two end hosts in the underlying Internet. All multicast-related functionality is implemented at the end hosts instead of at the routers of the underlying network, and the goal of the multicast protocol is to construct and maintain an efficient overlay for data transmission.

Among the proposed end system multicast protocols, tree-based systems have proven to be highly scalable and efficient in terms of physical link stress, state and control overhead, and end-to-end latency [14], [1], [8], [10]. However, normal tree structures have inherent problems in terms of resilience and bandwidth capacity.

Overlay trees composed of autonomous, unpredictable end systems are highly dependent on the reliability of interior nodes, and consequently vulnerable to transient behaviors of the participating end hosts [2]. Measurement studies of widely used peer-to-peer systems have reported median session times ranging from two hours to a minute [6], [12], [10], [23]. Handling this high level of **transiency** while still delivering high application performance at relatively low costs has proven to be a difficult task [22], [10], [18], [7], [17], [21].

Trees are vulnerable to effects arising from bandwidth-limited interior nodes. Measurement studies report a significant degree of **heterogeneity** in peer-to-peer applications [3], [23]. Available bandwidth has been reported to vary from less than 10 Kbps to more than 10 Mbps, with a large portion of the peers having very limited performance [19]. Thus, a major obstacle in multicast applications is to construct a distribution topology that avoids hotspots in the data path while providing

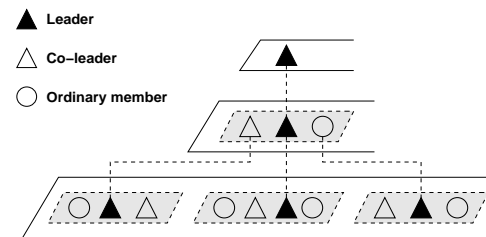


Fig. 1. Nemo's logical organization. Shapes illustrates the role of a peer within a cluster: a leader at a given layer can act as co-leader or ordinary member at the next higher layer.

good end-to-end performance.

In my research, I address these key issues in peer-to-peer multicast: transiency and heterogeneity, while still delivering high application performance. My approach is mainly experimental – designing, building, testing and redesigning as a way of gaining further insights into complex systems.

II. RESEARCH

As a first step to enable large-scale peer-to-peer multicast, we have designed Nemo [4] for resilient multicast specifically addressing high degrees of transiency.

Nemo uses the *implicit approach* to building overlays for multicasting: participating peers are organized into clusters based on network proximity, with every peer being a member of a cluster at the lowest layer. Each of these clusters selects a **leader** that becomes a member of the immediately higher layer. In part to avoid dependency on a single node, every cluster leader recruits a number of co-leaders to form its **crew**. The process is repeated, with all peers in a layer being grouped into clusters, crew members selected, and leaders promoted to participate in the next higher layer. Thus peers can lead more than one cluster in successive layers of this logical hierarchy. Co-leaders improve the resilience of the multicast group by avoiding dependencies on single nodes and providing alternative paths for data forwarding. In addition, crew members share the load from message forwarding, thus improving scalability. Figure 1 illustrates the logical organization of Nemo. Simulation and wide-area experimentation have illustrated the benefit of using crews in

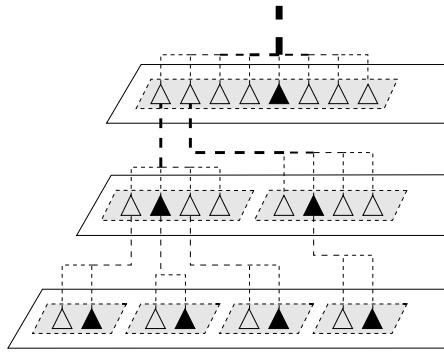


Fig. 2. FatNemo's Topology. The figure illustrates how the tree gets fatter when moving toward the root.

transient environments [4].

Building on the resilience of Nemo, FatNemo [5] addresses the bandwidth constraints of conventional tree-based multicast found in heterogeneous environments through adopting the concept of Leiserson's fat-trees [16] from parallel architecture. Similar to Leiserson's goals, we aim at building a multicast overlay with minimized mean and standard deviation of inter-node communication latency. Thus, we propose to organize participant end systems in a tree that closely resembles a Leiserson fat-tree. To build an overlay fat-tree, FatNemo relies on three main heuristics: (1) higher bandwidth capacity nodes should be placed higher up in the tree, (2) all peers must serve as crew members to maximize load balancing, and (3) the size of clusters should increase exponentially as one ascends the tree.

The per-node bandwidth constraint is critical for bandwidth-demanding applications and can be stated as the number of full-rate streams a peer is able to support, i.e. its *out-degree*. By organizing peers based on their out-degrees [11], we aim to reduce the bandwidth constraints of links higher up in the tree. Since estimating available bandwidth is a time-consuming process [13], [15], peers initially join the tree based on proximity. Once in the tree, every leader checks its highest owned cluster for better suited leaders in terms of bandwidth availability, and transfers leadership if such a peer exists. This assures that high out-degree peers will gradually ascend to higher layers, thus helping shape the tree as a bandwidth optimized fat-tree. Figure 2 illustrates how a fat-tree is resembled on the overlay.

In cooperative environments, peers contribute resources in order to receive a service. Conventional multicast trees, however, are not well suited for these environments as interior nodes share most of the forwarding responsibility, while leaf nodes only consume without providing resources. Continuing our work, I'm exploring how performance-based multicast can be applied to cooperative environments and I'm also building new applications, such as wide-area file systems and large-scale multi-player games, to illustrate the advantage of these novel protocols and to better understand their limitations.

REFERENCES

- [1] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proc. of ACM SIGCOMM*, August 2002.
- [2] M. Bawa, H. Deshpande, and H. Garcia-Molina. Transience of peers & streaming media. In *Proc. of HotNets-I*, October 2002.
- [3] A. R. Bharambe, S. G. Rao, V. N. Padmanabhan, S. Seshan, and H. Zhang. The impact of heterogeneous bandwidth constraints on DHT-based multicast protocols. In *Proc. of IPTPS*, February 2005.
- [4] S. Birrer and F. E. Bustamante. Resilient peer-to-peer multicast without the cost. In *Proc. of MMCN*, January 2005.
- [5] S. Birrer, D. Lu, F. E. Bustamante, Y. Qiao, and P. Dinda. FatNemo: Building a resilient multi-source multicast fat-tree. In *Proc. of IWCW*, October 2004.
- [6] F. E. Bustamante and Y. Qiao. Friendships that last: Peer lifespan and its role in P2P protocols. In *Proc. of IWCW*, October 2003.
- [7] M. Castro, M. Costa, and A. Rowstron. Performance and dependability of structured peer-to-peer overlays. In *International Conference on Dependable Systems and Networks*, June/July 2004.
- [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*, March 2003.
- [9] M. Castro, A. Rowstron, A.-M. Kermarrec, and P. Druschel. SCRIBE: A large-scale and decentralised application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communication*, 20(8), October 2002.
- [10] Y.-H. Chu, A. Ganjam, T. S. E. Ng, S. G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang. Early experience with an Internet broadcast system based on overlay multicast. In *Proc. of USENIX ATC*, June 2004.
- [11] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang. A case for end system multicast. *IEEE Journal on Selected Areas in Communication*, 20(8), October 2002.
- [12] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, modeling and analysis of a peer-to-peer file-sharing workload. In *Proc. of ACM SOSP*, December 2003.
- [13] M. Jain and C. Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. August 2002.
- [14] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole Jr. Overcast: Reliable multicasting with an overlay network. In *Proc. of the 4th USENIX OSDI*, October 2000.
- [15] K. Lai and M. Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *USENIX Symposium on Internet Technologies and Systems*, pages 123–134, March 2001.
- [16] C. E. Leiserson. Fat-trees: Universal networks for hardware-efficient supercomputing. *IEEE Transactions on Computers*, 34(10):892–901, October 1985.
- [17] J. Li, J. Stribling, R. Morris, M. F. Kaashoek, and T. M. Gil. A performance vs. cost framework for evaluating DHT design tradeoffs under churn. In *Proc. of IEEE INFOCOM*, March 2005.
- [18] R. Mahajan, M. Castro, and A. Rowstron. Controlling the cost of reliability in peer-to-peer overlays. In *Proc. of IPTPS*, February 2003.
- [19] T. S. E. Ng, Y. hua Chu, S. G. Rao, K. Sripanidkulchai, and H. Zhang. Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems. In *Proc. of IEEE INFOCOM*, April 2003.
- [20] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using content-addressable networks. In *Proc. of NGC*, November 2001.
- [21] S. Ratnasamy, S. Shenker, and I. Stoica. Routing algorithms for DHTs: Some open questions. In *Proc. of IPTPS*, March 2002.
- [22] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling churn in a DHT. In *Proc. of USENIX ATC*, December 2004.
- [23] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang. The feasibility of supporting large-scale live streaming applications with dynamic application end-points. In *Proc. of ACM SIGCOMM*, August/September 2004.
- [24] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination. In *Proc. of NOSSDAV*, June 2001.