

# What a mesh: understanding the design tradeoffs for streaming multicast

Animesh Nandi<sup>◊</sup>, Bobby Bhattacharjee<sup>‡</sup>, Peter Druschel<sup>◊</sup>

<sup>◊</sup>Max Planck Institute for Software Systems    <sup>‡</sup>Rice University    <sup>†</sup>University of Maryland

## ABSTRACT

Cooperative end-system multicast (CEM) is a promising paradigm for Internet video distribution. Several CEM systems have been proposed and deployed, but the tradeoffs inherent in the different designs are not well understood. In this work, we provide a common framework in which different CEM design choices can be empirically and systematically evaluated. Based on our results, we conjecture that all CEM systems must abide by a set of fundamental design constraints, which we express in a simple model.

## 1. INTRODUCTION

Research in cooperative end-system multicast (CEM) has focused either on the design of new protocols or on comparisons of complete systems. Several CEM approaches including single-tree (e.g. [4]), multi-tree (e.g. [3]), mesh-based (e.g. [12]), and hybrids (e.g. [1, 11, 10]) have been proposed. Prior research has led to a number of partially verified “communal hypotheses”, e.g. that mesh-based systems must incur high latencies and that tree-based systems are not resilient to churn. Yet, the networking and systems community still lacks a fundamental understanding of the CEM design space.

Gaining such an understanding is critical: the bandwidth required for streaming high quality video will remain near the limits of broadband network capabilities for the foreseeable future. From the system- and network-designer’s perspective, the CEM protocol should efficiently utilize all available bandwidth. From the end-user’s perspective, the protocol should have perfect continuity (i.e. streaming quality), low startup delay, and preferably low lag. Unfortunately, no single protocol meets all of these goals.

We conduct an in-depth and systematic empirical comparison of different CEM data delivery techniques, with the goal of understanding the inherent tradeoffs in CEM designs. Our approach differs from previous works that have compared CEM design choices qualitatively (e.g. [6]) or analytically (e.g. [2]), and with those that have compared specific CEM protocols empirically (e.g. [7]). It is not our intent to recommend any single approach or protocol. Instead, we explore the CEM design space, cleanly identify the tradeoffs that apply to these systems, tease out different components that are responsible for different aspects of observed behavior, and partition deployment scenarios into regions where different systems excel.

A systematic comparison of CEM systems is non-trivial. These systems deliver data over a diversity of data topologies (tree, mesh, hybrids) which are constructed and maintained using different control and transport protocols. The overall performance depends both on the properties of the data topology (how well it is able to use existing resources, how well it can withstand failures), and on the control protocol (how quickly the data topology is built/healed).

By necessity, existing system implementations couple the data- and control-planes and often use different transport protocols.

## 2. METHODOLOGY

Our methodology revolves around a common framework in which different CEM data delivery techniques can be faithfully (i.e. unbiasedly) compared. To factor out the effects of the transport protocols and the control plane, we re-implement, from scratch, representative single-tree, multi-tree, mesh-based, and hybrid protocols in their entirety using the SAAR anycast primitive [9]. SAAR implements a decentralized anycast service for overlay neighbor acquisition, and can efficiently support multicast overlays with diverse structures.

SAAR was developed exclusively as an efficient control mechanism (not in conjunction with any dataplane). However, it may still introduce an unintended bias in favor of a particular dataplane structure. Hence, we also experiment with an idealized control plane with perfect knowledge and a configurable response time, which allows us to control for any biases introduced by SAAR.

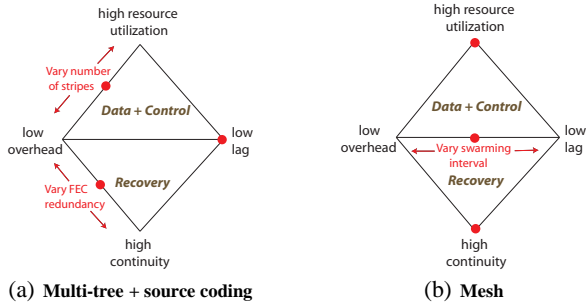
It was not clear to us, a priori, which specific dataplanes ought to be implemented to provide a representative sampling of the many overlay protocols that have been proposed. Instead of implementing every known protocol, we have meticulously implemented three basic data planes: single-tree, multi-tree, and mesh-based delivery. One (and sometimes a hybrid) of these three paradigms form the basis for every protocol in the literature.

Along with the base protocols, we have implemented a range of recovery strategies like ephemeral forwarding [1, 10], randomized forwarding [1], and mesh recovery [11]. We experiment with tree-based systems augmented with these recovery techniques. We also experiment with hybrid protocols that augment mesh-based systems with tree backbones to lower latency. By combining these base protocols and recovery techniques, we cover the major CEM protocols and approaches that have been published.

Our implementations can be executed on Planetlab, Emulab, Modenet, or deployed on the general Internet. The unmodified implementations can also be run on top of a network emulator, which executes the actual protocol code atop an emulated network with a given distribution of link delays and bandwidths.

## 3. EXPERIMENTAL EVALUATION

We evaluate the CEM design choices under diverse operating conditions, including different levels of node churn, packet loss, and stream rates. Moreover, to model the resource availability in real deployments, we rely on an empirical distribution of node bandwidths, which was obtained by measuring broadband hosts in Europe and the US [5]. In addition to exhaustive emulations, we verify the observed trends using the wide-area Planetlab testbed.



**Figure 1: Constraint triangles for CEM protocols. A red dot on a vertex means that the protocol optimizes the associated metric. A red dot on an edge connecting two vertices means that the protocol can trade off between the two metrics, by varying the indicated protocol parameter. We assert that no system can optimize all three metrics in any triangle.**

We find that mesh-based systems utilize available resources effectively and deliver high continuity under a wide range of conditions, but they inherently incur high lag and join delay. The mesh configuration (swarming interval and block size) can be optimized to reduce the lag and startup delays to some extent, but beyond a point, the increased overhead negatively impacts continuity. Although pure tree-based systems have low lag and startup delay, they must rely on sophisticated recovery mechanisms to improve their streaming quality under churn and packet loss. We observe that in resource-constrained scenarios, no hybrid tree-mesh system can simultaneously match the near-perfect streaming quality of a pure mesh and the low lag and startup delay of a pure tree-based system. However, when resources are abundant, then tree-based systems with recovery can achieve low lag, low startup delay, and high streaming quality even under adverse conditions. A full description of our results can be found in [8].

## 4. DESIGN CONSTRAINTS

In this section, we distill our observations and reasoning into a simple model that identifies design constraints and fundamental tradeoffs for CEM systems. The model is based on a set of constraints that we assert no CEM design can violate. We have depicted these constraints in Figure 1 as a pair of inter-related constraint triangles. We conjecture that these are, in fact, *impossibility* triangles, in that CEM systems (and indeed any streaming system) can choose to optimize at most two properties from each triangle, but *never* all three. A protocol may provide parameters that allow a trade-off between two (or more) properties in a triangle.

### 4.1 The constraint triangles

**The Data + Control triangle states that no dataplane design can simultaneously achieve all three of low lag, high global resource utilization, and low overhead on the data path.** For example, compared to single-tree systems, multi-tree systems (Figure 1) utilize resources better, but this comes at the cost of increased stripe tree maintenance overhead. Meshes provide essentially perfect utilization, but must incur either high overhead (due to frequent swarming exchanges) or high lag. The underlying reason behind this triangle is as follows: to achieve high resource utilization, a dataplane must be *dynamic*, i.e., be able to use upload bandwidth of all nodes even during periods of high churn. Such a dataplane cannot maintain statically computed paths; the price for this must be paid in terms of coordination overhead on the data path. This

overhead can be amortized but doing so necessarily increases lag.

**The Recovery triangle states that it is impossible to simultaneously achieve low overhead recovery, low lag and high continuity.** Reactive recovery strategies either incur high lag (since the receiver must detect a missing packet or heartbeat) or high overhead (lag can be reduced by increasing heartbeat frequency). Proactive recovery strategies have relatively low lag but must perform “blind” repairs (without a-priori knowledge of what data was lost). Proactive repair strategies that provide high continuity (without increasing lag) necessarily incur high overhead.

The constraint triangles imply that existing or future hybrid systems that combine trees and meshes cannot *fundamentally* improve performance, because each component of the hybrid is subject to the constraint triangles.

## 5. CONCLUSION

We conduct a systematic empirical comparison of CEM dataplane design alternatives. The goal is to understand the inherent tradeoffs of different design alternatives, in the quest for the optimal CEM system. Our empirical results demonstrate the inherent tradeoffs of CEM designs. Although some of these tradeoffs were expected, this is the first work that systematically explores the design space to demonstrate that these tradeoffs are inherent. Finally, we condense our findings into a simple model that identifies what we conjecture to be fundamental constraints that no CEM design can violate. In particular, the model asserts that no CEM design can simultaneously achieve all three of low overhead, low lag, and high continuity. A full description of our results and model can be found in [8].

## 6. REFERENCES

- [1] S. Banerjee, S. Lee, B. Bhattacharjee, and A. Srinivasan. Resilient multicast using overlays. In *SIGMETRICS 2003*.
- [2] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg. Epidemic live streaming: Optimal performance trade-offs. In *SIGMETRICS 2008*.
- [3] M. Castro, P. Druschel, A.M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream: High-bandwidth multicast in a cooperative environment. In *SOSP 2003*.
- [4] Y. 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 *USENIX 2004*.
- [5] M. Dischinger, A. Haeberlen, K.P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *IMC 2007*.
- [6] J. Liu, S.G. Rao, B. Li, and H. Zhang. Opportunities and challenges of peer-to-peer Internet video broadcast. In *IEEE Special Issue on Recent Advances in Distributed Multimedia Communications, 2007*.
- [7] N. Magharei, R. Rejaie, and Y. Guo. Mesh or multiple-tree: A comparative study of live p2p streaming approaches. In *INFOCOM 2007*.
- [8] A. Nandi, B. Bhattacharjee, and P. Druschel. Understanding the design tradeoffs for cooperative streaming multicast. In *Technical report MPI-SWS-2009-002, Max Planck Institute for Software Systems, April 2009*.
- [9] A. Nandi, A. Ganjam, P. Druschel, T.S.E. Ng, I. Stoica, H. Zhang, and B. Bhattacharjee. SAAR: A shared control plane for overlay multicast. In *NSDI 2007*.
- [10] V. Venkataraman, K. Yoshida, and P. Francis. Chunkspread: Heterogeneous unstructured end system multicast. In *ICNP 2006*.
- [11] F. Wang, Y. Xiong, and J. Liu. mTreebone: A hybrid tree/mesh overlay for application-layer live video multicast. In *ICDCS 2007*.
- [12] X. Zhang, J. Liu, B. Li, and T.S.P. Yum. Coolstreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming. In *INFOCOM 2005*.