The Impact of Heterogeneous Bandwidth Constraints on DHT-Based Multicast Protocols

Ashwin R. Bharambe[†], Sanjay G. Rao [§], Venkata N. Padmanabhan [‡], Srinivasan Seshan [†], and Hui Zhang[†]
[†]Carnegie Mellon University [§]Purdue University [‡]Microsoft Research

***Abstract*— We consider support for bandwidth-demanding applications such as video broadcasting using DHTs. We focus our investigations by considering the impact of heterogeneity in the outgoing bandwidth capabilities of nodes on Scribe, a representative and relatively mature DHT-based multicast protocol. We expose important issues that arise due to the mismatch between the ID space that underlies the DHT and the outgoing bandwidth constraints on nodes.**

I. Introduction

While DHTs were originally developed with applications like peer-to-peer file sharing in mind, there has been considerable interest in recent years in applying DHTs to overlay multicast applications [3], [7], [10], [13], [18]. In DHT-based approaches, the focus is on maintaining a structure based on a virtual id space, and enabling scalable and efficient unicast routing based on the node identifiers - the unicast routes are then used to create multicast distribution trees. This approach is in contrast to *performance-centric approaches* such as [4], [8], [11], [16], where the primary consideration while adding links to the overlay topology is application performance.

Two principal reasons have been advocated for a DHT-based approach. First, DHTs provides a generic primitive that can benefit a wide range of applications, among them overlay multicast. Second, the same DHT-based overlay can be used to simultaneously support and maintain a large number of overlay applications and multicast trees. This could help achieve lower overheads as compared to constructing and maintaining several seperate overlays. While DHT-based approaches have these potential advantages, a key unknown is application performance. Achieving good performance with DHTs is an active and ongoing area of research.

In this paper, we explore issues in enabling high-bandwidth broadcasting applications using DHTs. Our exploration is guided by design lessons we have learnt from our experience deploying an overlay-based broadcasting system [5]. In particular, we focus our investigation by considering the implications of a key issue - heterogeneous *outgoing* bandwidth constraints of nodes in the overlay. Such heterogeneity arises due to the presence of hosts behind various access technologies like cable modem, DSL and Ethernet, as summarized in Figure 1.

Event	Low Speed 100Kbps (deg. 0)	Medium Speed 1.5Mbps (deg. 2)	High Speed 10Mbps (deg. 10)	Avg Deg
Sigcomm [5]	22%	2%	76%	7.64
Slashdot [5]	74%	4%	22%	2.28
Gnutella [17]	65%	27%	8%	1.34

Fig. 1. Constitution of hosts from various sources. "deg" refers to our model of how many children nodes in each category can support. Sigcomm and Slashdot refer to two different broadcasts with an operationally deployed broadcasting system based on overlay multicast. Gnutella refers to a measurement study of peer characteristics of the Gnutella system.

We present an initial evaluation of Scribe [10], a representative and relatively mature DHT-based protocol for overlay multicast. Our experiments show that imposing bandwidth constraints on Scribe can result in the creation of distribution trees with high depth, as well as a significant number of *non-DHT links,* i.e., links that are present in the overlay tree but are not part of the underlying DHT. Trees with high depth are undesirable as larger the number of ancestors for a node, higher the frequency of interrupts due to the failure or departure of ancestors, and ultimately poorer the application performance. Non-DHT links are undesirable because they restrict the benefits of the route convergence and loop-free properties of DHT routing, and incur maintenance costs in addition to that of the DHT infrastructure. We find that a key cause for the issues observed is the mismatch between the id space that underlies the DHT structure and node bandwidth constraints. We discuss potential ways to solve the problem. and conclude that the issues are not straight-forward to address.

II. Evaluation Framework

Our evaluation is motivated by video broadcasting applications. Such applications involve data delivery from a single source to a set of receivers. Further, they are non-interactive, and do not place a tight constraint on the end-to-end latency. We assume a constant bit rate (CBR) source stream, and assume only nodes interested in the content at any point in time are members of the distribution tree and contribute bandwidth to the system.

The outgoing bandwidth limit of each host determines its *degree* or *fanout* in the overlay multicast tree, i.e.,

the maximum number of children that it can forward the stream to. We categorize hosts as being behind: (a) constrained links such as cable and DSL (few hundred Kbps); (b) intermediate speed links such as T1 lines (1.5 *Mbps*); and (c) high-speed links (10 *Mbps* or better). Given typical streaming video rates of the order of several hundred kilobits per second [5], we quantize the degrees of the low, medium, and high speed hosts to 0, 2, and 10. The degree 0 nodes are termed *non-contributors*. For higher speed connections, the degree is likely to be bounded by some policy (in view of the shared nature of the links) rather than the actual outgoing bandwidth. Figure 1 summarizes the constitution of hosts seen from measurement studies [17] and real Internet broadcast events [5].

The *Average Degree* of the system is defined as the total degree of all nodes (including the source) divided by the number of receivers (all nodes but the source). In this paper, we focus on regimes with an average degree greater than 1 which indicates that it is feasible to construct a tree.

III. Pastry/Scribe

While there have been several DHT-based proposals for multicast in recent years [9], [13], [18], [10], we choose to focus on Scribe. Scribe is one of the more mature proposals among DHT-based approaches with well-defined mechanisms to honor per-node degree constraints. A more recent follow-up work SplitStream [3] builds on top of Scribe and considers data delivery along multiple trees, rather than a single tree to improve the resiliency of data delivery. While we draw on some of the extensions proposed in Splitstream, we only consider single tree data delivery in this paper. We discuss some of the implications of multiple-tree solutions in Section VIII.

Scribe is built on top of the Pastry DHT protocol [14], and is targeted at settings which involve support of a large number of multicast groups. Each group may involve only a subset of the nodes in the Pastry system, but members in Pastry not part of a particular multicast group may be recruited to be forwarders in any Scribe tree. In this paper however, our evaluation assumes all participating members in Pastry are also part of the Scribe tree.

Each node in Pastry is assigned a unique 128-bit `nodeId` which can be thought of as a sequence of digits in base 2^b (b is a Pastry parameter.) A Pastry node in a network of N nodes maintains a routing table containing about $\log_{2^b} N$ rows and 2^b columns. The entries in the r^{th} row of the routing table refer to nodes whose `nodeId`s share the first r digits with the local node's `nodeId`. The routing mechanism is a generalization of hypercube routing: each subsequent hop of the route to the destination shares longer *prefixes* with the destination `nodeId`.

Scribe utilizes Pastry's routing mechanism to construct multicast trees in the following manner: each multicast group corresponds to a special ID called `topicId`. A multicast tree associated with the group is formed by the union of the Pastry routes from each group member to the `topicId`. Messages are multicast from the root to the members using reverse path forwarding [6].

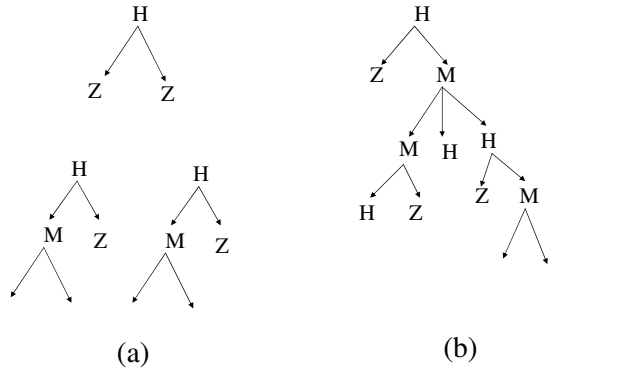

Fig. 2. Issues with heterogeneous degree constraints. H,M, and Z represent nodes of high, medium and zero (non-contributor) degrees respectively; (a) Entire subtrees (bottom) could be rejected when the subtree connected to the source (top) is saturated with non-contributors. (b) Depth can be poor with heterogeneous degree constraints.

A key issue with Scribe is that the number of children of a node A in the Scribe tree can be as high as the *in-degree* of the node in the underlying Pastry infrastructure – that is, the number of nodes in Pastry which use A as the next hop when routing towards the `topicId`. In general, this may be greater than is permitted by the node's bandwidth constraints. In order to tackle this overloading of nodes, the authors of Scribe/SplitStream have proposed two mechanisms:

• *Pushdown:* Whenever an overloaded node A receives a request from a potential child X, it can drop an existing child C, if X is found to be more "desirable" as a child than C. The orphaned node (either C or X) can contact one of the children of A as a potential parent, and this process goes on recursively. Choosing the criteria to determine which child of A (if any) that X should displace is an important issue. We discuss further in Section V.
• *Anycast.* If all nodes in the system have non-zero degree constraints, pushdown is guaranteed to terminate since leaf nodes will always have capacity. However, in the presence of non-contributor (degree 0) nodes, pushdown could end at a leaf that does not have capacity. This is tackled by an anycast procedure which provides an efficient way to locate a node with free capacity [3].

IV. Issues with heterogeneous constraints

Our evaluation of Scribe focuses on the following concerns that arise with heterogeneous degree constraints:

• *Rejections:* The tree constructed by a protocol could attain sub-optimal configurations, as for example shown in Figure 2(a). Here, the system as a whole has sufficient bandwidth resources to enable connectivity to all nodes. However, the subtree rooted at the source is saturated with

non-contributors, and the bandwidth resources of nodes in the disconnected subtrees remains unutilized. Nodes in the disconnected subtrees are eventually *rejected,* or forced to exit the multicast session.

• *High Depth:* An optimal configuration in terms of depth is one where the nodes that contribute the most (i.e. highest degree) form the highest levels, with lower degree nodes at lower levels. In the absence of mechanisms that explicitly favor construction of such trees, a protocol could produce trees of high depth such as shown in Figure 2(b). We believe that the depth metric is important as it significantly influences application performance. In general, in an overlay multicast application, the performance seen by a node depends on two factors: (i) the frequency of interruptions due to the failure of an ancestor, or due to congestion on an upstream link; and (ii) the time it takes a protocol to recover from the interruptions. The frequency of interruptions a node experiences in turn depends on the number of ancestors the node has, or the depth of the node.

• *Non-DHT Links:* While the two concerns above apply to *performance-centric* protocols as well, DHT-based designs need to deal with additional concerns with regard to preserving the structure of the DHT. In particular, while the pushdown and anycast operations described in Section III help Scribe cope with heterogeneous node bandwidth constraints, they may result in the creation of parent-child relationships which correspond to links that are not part of the underlying Pastry overlay. We term such links as *non-DHT* links. We believe these non-DHT links are undesirable because: (i) the route convergence and loop-free properties of DHT routing no longer apply if non-DHT links exist in significant numbers; and (ii) such links require explicit per-tree maintenance which reduces the benefits of DHTs in terms of amortizing overlay maintenance costs over multiple multicast groups (and other applications).

V. Techniques Evaluated

We present two variants of the pushdown algorithm that we evaluated in Scribe. The first policy, *Preempt-ID-Pushdown* is based on the policy implemented in [3], and is not optimized to minimize depth in heterogeneous environments. The second policy, *Preempt-Degree-Pushdown*, is a new policy that we introduced in Scribe to improve depth in heterogeneous environments.

• *Preempt-ID-Pushdown:* When a saturated node A receives a request from a potential child X, X preempts a child C of A if X shares a longer prefix with the `topicID` than C. Further, the orphaned node (X or C) contacts a child of A and continues the pushdown if the orphaned node shares a prefix match with the child. However, if no child of A shares a prefix with the orphaned node, we continue with the pushdown operation by picking a random child of A.[1] An anycast operation is employed if a leaf node is reached without a parent being found.

• *Preempt-Degree-Pushdown:* Here, node degree is the primary criterion in the pushdown. When a saturated node A receives a request from a potential child X, X preempts the child (say C) of A which has the lowest degree, provided X itself has a higher degree than C. The orphaned node (X or C) picks a random child of A that has a degree equal to or greater than itself and continues the pushdown. An anycast operation is employed if a leaf node is reached without a parent being found.

While *Preempt-Degree-Pushdown* can improve the depth of trees produced by Scribe compared to *Preempt-ID-Pushdown,* it can lead to the creation of a larger number of non-DHT links given that the id is no longer a key criterion in pushdown. Further, *Preempt-Degree-Pushdown* itself cannot create perfectly balanced trees - for example, if node A has a lower degree than node X, there is no mechanism in place for X to displace A. Doing so would require further deviation from the DHT-structure, and the creation of additional non-DHT links. In fact, we believe it is not easy to construct trees with both low depth, as well as a low fraction of non-DHT links. We discuss this further in Section VII.

VI. Evaluation Details

We use the original Scribe and Splitstream implementation [15] for our experiments. In the Scribe implementation, Scribe-level links were maintained separately from the underlying Pastry links. Thus, if Pastry changed its routing table (due to its own optimizations), the Scribe level link would appear to be a non-Pastry (i.e. non-DHT) link afterwards. In order to avoid such over-counting, we associate a DHT or non-DHT flag with a Scribe link *only when it is first established.* [2]

Our experiments use a Poisson arrival pattern and a Pareto-distributed stay time for clients. These choices have been motivated by group dynamics characteristics observed in overlay multicast deployments [5] and Mbone measurements [2]. Our experiments last for a duration of 1000 seconds, and assume a mean arrival rate of 10 joins per second. Further, our experiments assume nodes have a mean stay time of 300 seconds, a minimum stay time of 90 seconds, and a parameter of $\alpha = 1$ in the Pareto distribution. This corresponds to a steady state group size of

[1] This is a slight departure from [3], where an anycast operation is employed if no child of A shares a prefix with the orphaned node. We have observed better performance in depth in homogeneous environments with our optimization. The intuition is that pushdown tends to do better at filling up nodes higher in the tree, while anycast tends to choose parents at more random locations in the tree.

[2] It is possible that Pastry route table changes can transform a initial non-DHT Scribe link into a DHT link. However, the probability of this happening is very small.

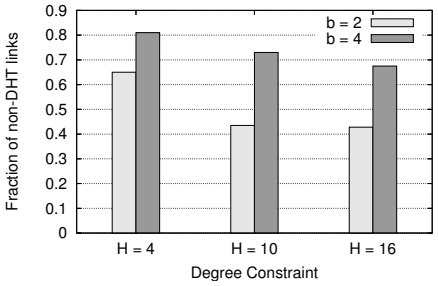

Fig. 3. Fraction of non-DHT links (mean over the session) in homogenous environments for various values of node degree and b, the base of the node IDs in Pastry.

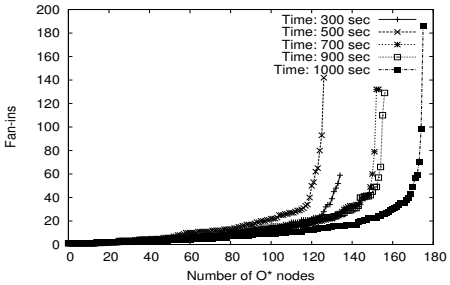

Fig. 4. Distribution of fan-in/in-degree of $\mathbf{0}*$ nodes in Pastry. The Y-Axis is the in-degree of Pastry routing tables. The X-Axis is the number of $\mathbf{0}*$ nodes that have an in-degree less than a particular value. Each curve presents the distribution at different times during the simulation. There exists a sharp skew – indicating a small number of nodes with high in-degree – which persists throughout the simulation.

about 3000 members. Finally, given that our focus is on bandwidth-sensitive and non-interactive applications, we simply consider a uniform-delay network model throughout this paper.

VII. Empirical Results

We present the results of experiments with Scribe with both homogeneous and heterogeneous degree constraints.

Homogeneous Environments: We assume that all nodes have a degree H. Figure 3 plots the fraction of non-DHT links within the Scribe tree as a function of H. There are 3 sets of bars, each set corresponding to a different value of H. Each set consists of bars of 2 shades, correspoding to different values of b, the base of the node IDs in Pastry. Each bar represents the mean of three runs. We find the fraction of non-DHT links is high and over 40% for all configurations we evaluate.

We discuss two factors that contribute to the creation of non-DHT links in Figure 3. Consider a `topicID` of 00...00. Let $\mathbf{0}*$ represent the nodes whose IDs match the topicID in the first digit (that is, the first digit is 0 and the rest of the digits are arbitrary). A join or reconnect request from any node in Scribe should be routed in the first hop to a $\mathbf{0}*$ node, since we would like to match at least the first digit of the `topicID`. So, if there were no pushdown operations, given the reverse-path nature of tree

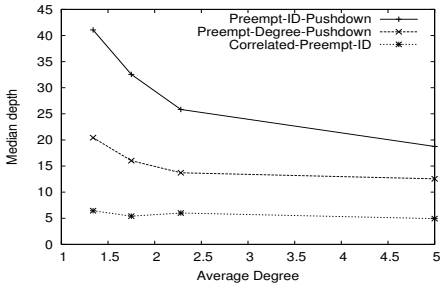

Fig. 5. Depth Vs. Average Degree in heterogeneous settings. We compute mean depth of a node during the session, and compute median across the nodes. The fraction of non-contributors is fixed at 50%.

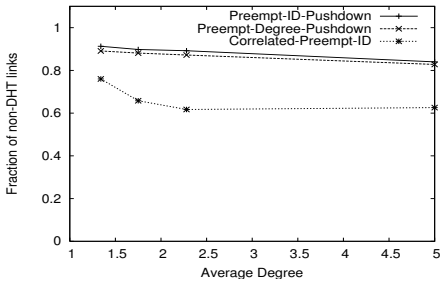

Fig. 6. Fraction of non-DHT links Vs. Average Degree in heterogeneous settings. The fraction of non-contributors is fixed at 50%.

construction in Scribe, all parents in a Scribe tree would be $\mathbf{0}*$ nodes.

A first factor leading to the creation of non-DHT links is that the total bandwidth resources at the $\mathbf{0}*$ nodes may not be sufficient to support all nodes in the tree. Let b be the base of the node IDs in Pastry, and AD be the average degree of the nodes in the system. Then, the $\mathbf{0}*$ nodes represent a fraction $\frac{1}{2^b}$ of the total nodes of the system, and we expect them to only be able to support a fraction $\frac{AD}{2^b}$ of the nodes in the system. Thus, we expect to see $1-\frac{AD}{2^b}$ links that have non-$\mathbf{0}*$ nodes as parents. Such links are likely to be non-DHT links. This is because: (i) these links must have been created by pushdown operations as described above; and (ii) there are no explicit mechanisms in place to prefer choosing DHT links during a pushdown.

From this discussion, we expect the number of non-DHT links to be equal to $1 - \frac{H}{2^b}$ in a homogeneous environment, where all nodes have a degree H (as the average degree $AD = H$). While this partially explains Figure 3, the fraction of non-DHT links is significantly higher than our estimate. In particular, if $H \geq 2^b$, then we would not expect to see any non-DHT links. However, even when $H = 16$ and $b = 2$ so that $H \gg 2^b$, non-DHT links constitute over 40% of the links in the tree. We believe this is due to a second factor that contributed to the creation of non-DHT links, as we discuss in the next paragraph.

Figure 4 plots the CDF of the fan-ins of the $\mathbf{0}*$s in the system at various times during the simulation. The fan-in of a node is the number of other nodes in the system that have this node as a neighbor in Pastry. We see that there is a significant skew in the fan-ins of the $\mathbf{0}*$s. Due to

the skew, Scribe join requests hit the $\mathbf{0}*$s non-uniformly, causing a much larger number of pushdowns, and hence non-DHT links. This also results in poor utilization of the available bandwidth resources at many of the $\mathbf{0}*$ nodes.

We have investigated potential factors that may have led to the skew. For instance, we considered whether it resulted from the uniform delay model used in our simulations. Preliminary experiments indicate that the skew exists even with topologies with non-uniform delays generated using the GeorgiaTech simulator reported in [10]. We believe that the skew arises due to Pastry's join and repair mechanisms in which a new node picks up routing table entries from other nodes in the system. While this reduces join/repair times and overheads, it makes nodes that joined earlier far more likely to be picked as neighbors as compared to other nodes. We defer to future work an examination of how fundamental the skew is to the design of Pastry/Scribe, and whether it can be eliminated using simple heuristics.

Heterogeneous Environments: Our experiments with heterogeneous environments were conducted with 50% of the nodes being non-contributors (degree 0), and for various average degree values. Changing the average degree value results in a different fraction of nodes of medium (degree 2) and higher (degree 10) degree. Figure 5 compares the depth of the Scribe multicast tree created with *Preempt-ID-Pushdown* and *Preempt-Degree-Pushdown* in heterogeneous environments. The depth is computed as follows: we compute the mean depth of a node by sampling its depth at different time instances, and then compute the medians across the nodes. The optimal median depth for any of the plotted configurations (not shown in the graph) is about 4. The top 2 curves correspond to *Preempt-ID-Pushdown* and *Preempt-Degree-Pushdown*. *Preempt-ID-Pushdown* performs significantly worse than optimal. This is expected given that there are no mechanisms in place that optimize depth in heterogeneous environments. *Preempt-Degree-Pushdown* performs better than *Preempt-ID-Pushdown* but is still far from optimal, consistent with discussions in Section V.

Figure 6 shows the fraction of non-DHT links from our simulations for *Preempt-Degree-Pushdown,* and *Preempt-ID-Pushdown.* The fraction of non-DHT links is over 80% for a range of average degrees. We believe both factors that we discussed with homogeneous environments – insufficient resources at $\mathbf{0}*$ nodes, and the skew in the in-degree of Pastry – have contributed to the creation of non-DHT links. Further, as discussed, even if the skew could be completely eliminated, we would still expect to see $1 - \frac{AD}{2^b}$ non-DHT links due to insufficient resources at $\mathbf{0}*$ nodes, where AD is the average degree of the nodes in the system.

A third important factor that could cause non-DHT links in heterogeneous environments is that it may be desirable to use non-$\mathbf{0}*$ nodes as parents to minimize the depth of trees. For example, in an environment with nodes of degree H, L, and 0 ($H > L$), the optimal depth tree requires having all nodes of degree H at the highest levels in the tree, and thus as interior nodes. However, only a fraction $\frac{1}{2^b}$ of nodes of degree H are likely to be $\mathbf{0}*$ nodes. Thus, optimizing for tree depth in Scribe could potentially result in a larger fraction of non-DHT links due to the need to use non-$\mathbf{0}*$ nodes of degree H as interior nodes. Consequently, we would expect *Preempt-Degree-Pushdown* to have a higher fraction of non-DHT links as compared to *Preempt-ID-Pushdown*. However, both policies perform similarly. We believe this is because the other two factors causing non-DHT links dominate in our experiments.

Summary: Our experiments with Scribe indicates trees produced have a high depth, and a large fraction of non-DHT links. There are three factors that cause the creation of non-DHT links with Scribe. First, the bandwidth resources of nodes that share a prefix with the `topicId` may not be sufficient to sustain all nodes in the system. Second, minimizing depth of trees in Scribe requires utilizing higher degree nodes, even though they may not share a prefix with the `topicId`. The third factor is a skew in the in-degree of Pastry. We believe the skew is a result of specific heuristics employed in Pastry, and can potentially be minimized. However, we believe the first two factors are fundamental to the mismatch of node bandwidth constraints and node ids with DHT-based designs. Further, simple analysis shows that the first factor alone could lead to the creation of $1 - \frac{AD}{2^b}$ non-DHT links, where AD is the average degree of the system, and b is the base of the node IDs in Pastry.

VIII. Feasibility of Potential Solutions

We sketch potential solutions and consider their ability to address the issues raised in the previous section:

ID-Degree Correlation: A natural question is whether changing the random id assignment of DHTs, and instead employing an assignment where node ids are correlated to node bandwidth constraints can address the issue. To evaluate the potential of such techniques, we consider *Correlated-Preempt-ID* heuristic, where nodes with higher degrees are assigned `nodeId`s which share longer prefixes with the `topicId`. Figure 5 shows that this policy indeed is able to achieve depths close to the optimal depth of 4, while Figure 6 shows it can significantly lower the fraction of non-DHT links. However, while such a solution could work in scenarios where the DHT is primarily used for a specific multicast group, disturbing the uniform distribution of DHT `nodeId`s can be undesirable, and can adversely affect routing properties of DHTs [1]. Further, DHT's are particularly useful in scenarios where there is a

shared infrastructure for a wide variety of applications including multicast sessions. In such scenarios, it is difficult to achieve a correlation between node id and node degree assignments across all trees.

Multiple Trees: Another question is whether the issues involved can be tackled using the multi-tree data delivery framework used to improve the resiliency of data delivery and for bandwidth management [3], [11]. In this framework, 2^b trees are constructed, with the `topicIds` of every tree beginning with a different digit. Each node is an interior node in the one tree where it shares a prefix with the `topicId`, and is a leaf node in the rest. We note that a direct application of the multi-tree approach cannot solve the problem - if nodes belong to multiple degree classes to begin with, then, each of the trees will continue to have nodes of multiple degree classes, and the issues presented in this paper continue to be a concern.

Multiple Trees with Virtual Servers [12]: One potential direction for solving the issues with DHTs is to combine the multi-tree data delivery framework with the concept of virtual servers proposed in [12]. The idea here is that a node can acquire a number of ids proportional to its degree, and then use the multi-tree data delivery framework above. A concern with this approach is that we are not completely concentrating the resources of a higher degree node in one tree, rather, we are distributing it across several trees, thereby giving up on the policy of interior disjointness. The performance implications would need to be carefully evaluated.

IX. Summary and Discussion

In this paper, we have considered the impact of heterogeneity in the outgoing bandwidth constraints of nodes on overlay multicast using Scribe. Our results indicate that trees produced by Scribe tend to have a *large depth,* as well as a significant fraction of *non-DHT links.* The key reason for this is the mismatch between the id space that underlies the DHT structure and node bandwidth constraints. We have not found obvious or satisfactory solutions to address the problem, leading us to believe the issues involved are not trivial.

Our work has been motivated by lessons we learnt from deploying an overlay-based broadcasting system [5]. Beyond the particular issue of bandwidth heterogeneity considered in this paper, our experience also highlights the importance of considering factors such as heterogeneity in node stabilities, as well as connectivity restrictions due to entities such as NATs and firewalls. While these concerns pertain to both *performance-centric* and *DHT-based* designs, we believe they are more challenging to address in the DHT context given the structure imposed by DHTs. Although there has been significant progress in improving the performance of DHTs, with regard to delay-based metrics such as *Relative Delay Penalty*(RDP) [4], we believe that it would be important to address the challenges posed by heterogeneity before a compelling case can be made for using DHTs to support bandwidth-demanding broadcasting applications.

Acknowledgments: We thank Anthony Rowstron and Miguel Castro for access to, and for clarifications regarding the Scribe code.

References

[1] A.Bharambe, M.Agrawal, and S.Seshan. "Mercury: Supporting scalable multi-attribute range queries". In *ACM Sigcomm* (2004).

[2] Almeroth, K. C., and Ammar, M. H. Characterization of mbone session dynamics: Developing and applying a measurement tool. Tech. Rep. GIT-CC-95-22, 1995.

[3] Castro, M., Druschel, P., Kermarrec, A., Nandi, A., Rowstron, A., and Singh, A. SplitStream: High-bandwidth Content Distribution in Cooperative Environments. In *Proceedings of SOSP* (2003).

[4] Chu, Y., Rao, S., and Zhang, H. A Case for End System Multicast. In *Proceedings of ACM Sigmetrics* (June 2000).

[5] Chu et. al. Early deployment experience with an overlay based internet broadcasting system. In *USENIX Annual Technical Conference* (June 2004).

[6] Deering, S. Multicast Routing in Internetworks and Extended LANs. In *Proceedings of the ACM SIGCOMM* (Aug. 1988).

[7] I.Stoica, D.Adkins, S.Zhuang, S.Shenker, and S.Surana. "Internet Indirection Infrastructure". *"IEEE/ACM Transactions on Networking"* (April 2004).

[8] Jannotti, J., Gifford, D., Johnson, K. L., Kaashoek, M. F., and Jr., J. W. O. Overcast: Reliable Multicasting with an Overlay Network. In *Proceedings of the Fourth Symposium on Operating System Design and Implementation (OSDI)* (Oct. 2000).

[9] Liebeherr, J., and Nahas, M. Application-layer Multicast with Delaunay Triangulations. In *IEEE Globecom* (Nov. 2001).

[10] M. Castro, P. Druschel, A. K., and Rowstron, A. Scribe: A large-scale and decentralized application-level multicast infrastructure. In *IEEE Journal on Selected Areas in Communications Vol. 20 No. 8* (Oct 2002).

[11] Padmanabhan, V., Wang, H., and Chou, P. Resilient Peer-to-peer Streaming. In *Proceedings of IEEE ICNP* (Nov. 2003).

[12] Rao, A., Lakshminarayanan, K., Surana, S., Karp, R., and Stoica, I. "Load Balancing in Structured P2P Systems". In *Proceedings of the Second International Workshop on Peer-to-Peer Systems (IPTPS)* (2003).

[13] Ratnasamy, S., Handley, M., Karp, R., and Shenker, S. Application-level Multicast using Content-Addressable Networks. In *Proceedings of NGC* (2001).

[14] Rowstron, A., and Druschel, P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)* (2001).

[15] Rowstron, A., Castro, M., et. al. SimPastry (Scribe) Implementation, v3.0a.

[16] S. Banerjee, B. B., and Kommareddy, C. Scalable Application Layer Multicast. In *Proceedings of ACM SIGCOMM* (Aug. 2002).

[17] Saroiu, S., Gummadi, P. K., and Gribble, S. D. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking (MMCN)* (January 2002).

[18] S.Q.Zhuang, B.Y.Zhao, J.D.Kubiatowicz, and A.D.Joseph. Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination. In *Proceedings of NOSSDAV* (Apr 2001).