Analysis of the NICE Application Layer Multicast Protocol

Suman Banerjee, Bobby Bhattacharjee
Department of Computer Science, University of Maryland, College Park, MD 20742, USA
Emails: {suman,bobby}@cs.umd.edu

UMIACS TR 2002-60 and CS-TR 4380
June 2002

Abstract—**Application layer multicast protocols organize a set of hosts into an overlay tree for data delivery. Each host on the overlay peers with a subset of other hosts. Since application layer multicast relies only on an underlying unicast architecture, multiple copies of the same packet can be carried by a single physical link or node on the overlay. The stress at a link or node is defined as the number of identical copies of a packet carried by that link or node. Stretch is another important metric in application layer multicast, which measures the relative increase in delay incurred by the overlay path between pairs of members with respect to the direct unicast path. In this paper we study the NICE application layer multicast protocol to quantify and study the tradeoff between these two important metrics — stress and stretch in scalably building application layer multicast paths.**

I. Introduction

Multicast is an efficient mechanism to reduce traffic redundancy in the network and is, therefore, an useful service to scale multi-party applications. However, due to the limited success of network-layer multicast solutions, many researchers have suggested implementing the multicast service at the application layer [3], [4], [2], [6], [7], [9], [10], [1]. None of these *Application Layer Multicast* protocols propose any change to the network infrastructure and instead, implement multicast forwarding functionality exclusively at the end-hosts.

The basic idea of application layer multicast is shown in Figure 1. Unlike network layer multicast (Panel 0) where data packets are replicated at routers inside the network, in application layer multicast, data packets are replicated at end-hosts. Logically, the end-hosts form an overlay network, and the goal of application layer multicast is to construct and maintain an efficient overlay for data transmission. Since application layer multicast protocols must send the identical packets over the same link, they are less efficient than native multicast. There are two intuitive metrics of "goodness" defined to evaluate the quality of the application layer multicast data paths. They are:

1) Stress: This metric is defined per link or node of the topology and counts the number of identical packets sent by the protocol over that link or node. For network layer multicast there is no redundant packet replication and hence in this case, the stress metric is one at each link or node of the network.
2) Stretch: This metric is defined per-member and is the ratio of the path length along the overlay from the source to the member to the length of the direct unicast path. Clearly, a sequence of direct unicasts from the source to all the other members (Panel 1, Figure 1) has unit stretch for all members.

Different application layer multicast protocols will create overlay paths that have different stretch and stress. In Figure 1, we show three example application layer multicast overlays on the same topology of routers and hosts. Let us assume that each link on the topology is of unit length. Panel 1 shows the overlay corresponding to a *sequence of direct unicasts* from the source (A) to all the other members. In this case, the stretch to each member is unity (since the direct unicast paths are used). Link $\langle A, 1\rangle$ experiences a stress of 3, while all other links experience unit stress. In general, for a group of N members, using a sequence of direct unicasts is one extreme case where the maximum stress at a link is $O(N)$ (at the data source) and the average stretch of members 1.

Panel 2 shows the overlay corresponding to *ring multicast.* The stretch experienced by the different members are 1 for B, $6/4 = 1.5$ for C and $9/3 = 3$ for D. The stress on each link on the topology is unity. (We consider each link in the topology as two directed links with opposing directions.) Thus, ring multicast is the other extreme case where the maximum stress is 1 while the average stretch at members is $O(N)$.

Finally, Panel 3 shows another configuration of the overlay, which is an intermediate between the two extremes. In this example, the stretch at the members B, C and D are $3/3 = 1, 6/4 = 1.5$ and $3/3 = 1$ respectively. The link $\langle A, 1\rangle$ has a stress of 2, while all other links have unit stress. We can therefore, make a simple observation through this example: *decreasing stretch in an over-*

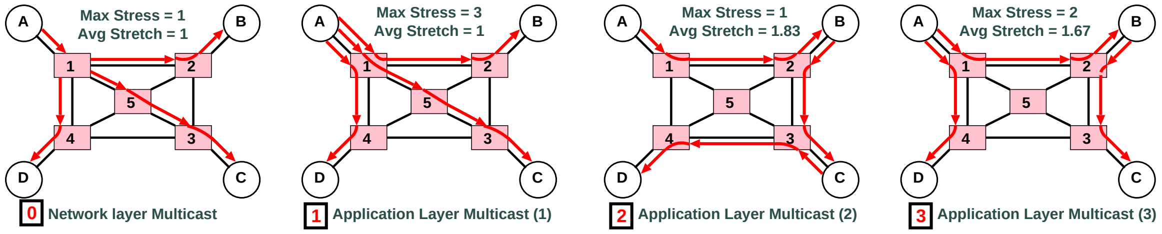

Fig. 1. Network-layer and application layer multicast. Square nodes are routers, and circular nodes are end-hosts.

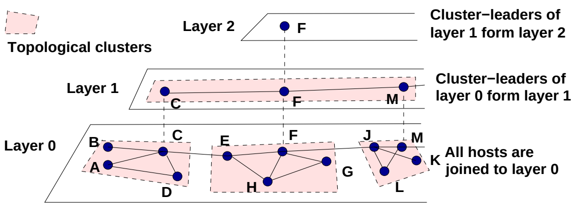

Fig. 2. Hierarchical arrangement of hosts in NICEṪhe layers are logical entities overlaid on the same underlying physical network.

lay leads to increased stress and vice versa.

In this paper, we study the relationship between the stress and the stretch metrics using the NICE application layer multicast protocol [1] as a representative protocol.

II. Scalable Application Layer Multicast

In this section, we summarize the NICE protocol to create a scalable application layer multicast overlay as presented in [1]. The protocol arranges the set of end hosts into a hierarchy; the basic operation of the protocol is to create and maintain the hierarchy. The hierarchy implicitly defines the multicast overlay data paths. The member hierarchy is crucial for scalability, since most members are in the bottom of the hierarchy and only maintain state about a constant number of other members. The members at the very top of the hierarchy maintain (soft) state about $O(\log N)$ other members. Logically, each member keeps detailed state about other members that are *near* in the hierarchy, and only has limited knowledge about other members in the group. The hierarchical structure is also important for localizing the effect of member failures. While constructing the NICE hierarchy, members that are "close" with respect to the distance metric are mapped to the same part of the hierarchy: this allows us to produce trees with low stretch.

The NICE hierarchy is created by assigning members to different levels (or layers) as illustrated in Figure 2. Layers are numbered sequentially with the lowest layer of the hierarchy being layer zero (denoted by L_0). Hosts in each layer are partitioned into a set of clusters. Each cluster is of size between k and $3k-1$, where k is a constant, and consists of a set of hosts that are close to each other. Further, each cluster has a cluster leader. The protocol distributedly chooses the (graph-theoretic) center of the cluster to be its leader, i.e. given a set of hosts in a cluster, the cluster leader has the minimum maximum distance to all other hosts in the cluster.

Hosts are mapped to layers using the following scheme: All hosts are part of the lowest layer, L_0. The clustering protocol at L_0 partitions these hosts into a set of clusters. The cluster leaders of all the clusters in layer L_i join layer L_{i+1}. This is shown with an example in Figure 2, using $k = 3$. The layer L_0 clusters are [ABCD], [EFGH] and [JKLM][1]. In this example, we assume that C, F and M are the centers of their respective clusters of their L_0 clusters, and are chosen to be the leaders. They form layer L_1 and are clustered to create the single cluster, [CFM], in layer L_1. F is the center of this cluster, and hence its leader. Therefore F belongs to layer L_2 as well.

For ease of exposition only in this section, we consider the case where all clusters has the same size, k. (The constant factor does not affect the analysis.) Then, the following properties hold for the distribution of hosts in the different layers:

- A host belongs to only a single cluster at any layer.
- If a host is present in some cluster in layer L_i, it must occur in one cluster in each of the layers, $L_0, \ldots, L_{i-1}$. In fact, it is the cluster-leader in each of these lower layers.
- If a host is not present in layer, L_i, it cannot be present in any layer L_j, where $j > i$.
- The size of each cluster is k, and the leader of the cluster is its graph-theoretic center.
- There are $M = \log_k N$ layers, and the highest layer has only a single member.

In the next section, we analyze the stress and stretch metrics of the overlay trees generated by this protocol.

[1] We denote a cluster comprising of hosts $X, Y, Z, \ldots$ by $[XYZ\ldots]$.

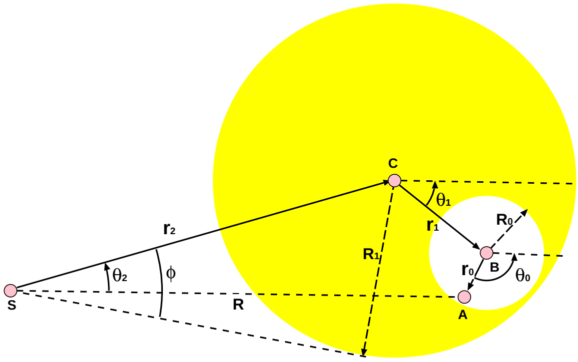

Fig. 3. Stretch for an arbitrary member, A, for the NICE protocol. The circle indicates the cluster radius and does not imply that the structure of the cluster is exactly circular.

III. Analyzing Stress and Stretch

Through the example in Section I we observed that both the stress and stretch metrics can vary between 1 and $O(N)$, depending on the application layer multicast protocol used and the structure of the underlying topology.

In this section, we analyze the stress and stretch metrics for the NICE protocol. We quantify both the average and maximum values of the two metrics. The analysis can be summarized as follows: the maximum and average stretch and the average stress for the NICE protocol are functions of the cluster size parameter, k only, while the maximum stress depends on both k and N, the size of the group. We show the exact relationship between these metrics later in this section.

Model: Since we are interested in the asymptotic nature of the metrics, we assume a very large member population that is densely and uniformly distributed in the network. This assumption can be expressed mathematically as follows: For any member, u and any real number, $r > 0$, let $\nu(u, r)$ denote the number of members within a distance r of u. Then, there exists constants c_1 and $c_2 > 1$ such that $c_1\nu(u, r) \leq \nu(u, 2r) \leq c_2\nu(u, r)$. This is the same assumption made by Plaxton et. al. [8] to quantify the stretch along overlay paths. We also assume (for the sake of simplicity) that the distance between members are Euclidean. For a large set of uniformly distributed members, the clusters created by the NICE protocol in each layer will have similar properties, i.e. will have the same cluster radius (in a graph-theoretic sense). Additionally, all clusters have the same number of members, k, as defined by the protocol.

Stretch: Consider a member, A located at an arbitrary point in the space, that belongs to layer L_0 of the hierarchy and no other higher layer (see Figure 3). Let, B be the leader of the L_0 cluster to which A belongs. B therefore, belongs to layer L_1. Also, let C be the leader of the L_1 cluster to which B belongs. C belongs to layer L_2. In this example, we assume that there are only three layers L_0, L_1 and L_2.

The direct unicast path length from the source, S, to A is R. The path length between S and A on the overlay is $r_2 + r_1 + r_0$. The stretch for member A is therefore, given by $(r_2 + r_1 + r_0)/R$. It is easy to observe that $R = \sum_{i=0}^{2} r_i \cos\theta_i$, where θ_i is as marked in Figure 3.

Generalizing for $M(= \log_k N)$ layers, for a member, X, that belongs to layers $L_0, \ldots, L_j$ and no other higher layer, the stretch, s_X is given by:

$$s_X = \frac{\sum_{i=j}^{M} r_i}{\sum_{i=j}^{M} r_i \cos\theta_i} \tag{1}$$

The stretch is maximum for members that belong to layer L_0 only, and therefore it is sufficient to calculate the stretch for members in layer L_0 only. In the rest of this section we calculate the stretch for these members only.

Since the member population is large and network is densely populated with the members, we now make a *fluid approximation* as follows. Let ρ denote the density of members per unit area. Let $R_0, R_1, \ldots, R_M$ denote the radius of clusters in the layers $L_0, L_1, \ldots, L_M$ respectively. Clearly, $r_i \leq R_i$. The number of members in a cluster is k. Let ρ_i denote the density of members that belong to layer L_i. It follows that the size of a cluster in layer L_i is proportional to $\rho_i {R_i}^2$ and must be equal to k, according to the stated invariants. The number of members at layer L_i is given by N/k^i. Therefore the density of members at layer L_i is given by $\rho_i = \rho/k^i$. Hence,

$$\frac{\rho}{k^i} {R_i}^2 \propto k$$

which implies

$$R_i = R_{i-1}\sqrt{k} \tag{2}$$

We first consider the "far" members such that the first hop on the data path is at least a distance μR_{M-1} away from the source, i.e. $r_2 \geq \mu R_1$ in Figure 3, with say $\mu \geq 2$.

From Equation 1, we can provide a simple bound for the stretch $s_{X,f}$ of a "far" member X, that belongs to layer L_0 only, and no other higher layer, based on the following observations. For $0 \leq i < M$, $r_i \leq R_i$ and the minimum value of $\cos\theta_i$ is -1. Given r_M, the minimum value of $cos\theta_M$ is $\cos\phi$, where $\phi = \sin^{-1}\frac{R_{M-1}}{r_M}$ (see Figure 3). $\cos\phi$ is minimum when ϕ is maximum, i.e. r_M is minimum, i.e. $r_M = \mu R_{M-1}$. The maximum value of the numerator is given by $r_M + \sum_{i=0}^{M-1} R_i$. The minimum value of the denominator is given by $r_M \cos\Phi - \sum_{i=0}^{M-1} R_i$, where $\Phi = \sin^{-1}\frac{1}{\mu}$.

Thus an upper bound of the stretch of a "far" member, X is given by:

$$s_{X,f} \leq \frac{r_M + \sum_{i=0}^{M-1} R_i}{r_M \cos\Phi - \sum_{i=0}^{M-1} R_i} \tag{3}$$

Let us choose $\mu = 2$, i.e. $\Phi = \frac{\pi}{6}$. Noting, $M = \log_k N$ and using Equation 2, it follows that $\sum_{i=0}^{M-1} R_i = R_0(\sqrt{N}-1)/(\sqrt{k}-1)$. Therefore, dividing both numerator and denominator of Equation 3 by r_M, we have:

$$s_{X,f} \leq \frac{1 + \frac{R_0}{r_M}\frac{\sqrt{N}-1}{\sqrt{k}-1}}{\frac{\sqrt{3}}{2} - \frac{R_0}{r_M}\frac{\sqrt{N}-1}{\sqrt{k}-1}} \tag{4}$$

Since $r_M \leq \mu R_{M-1} = 2R_0 k^{(M-1)/2}$, Equation 4 implies

$$s_{X,f} \leq \frac{1 + \frac{1}{\sqrt{Nk}}\frac{\sqrt{N}-1}{\sqrt{k}-1}}{\frac{\sqrt{3}}{2} - \frac{1}{\sqrt{Nk}}\frac{\sqrt{N}-1}{\sqrt{k}-1}} \tag{5}$$

Finally, for asymptotically growing N, simplifying Equation 5 we conclude

$$s_{X,f} \leq 2\frac{k - \sqrt{k} + 1}{k\sqrt{3} - \sqrt{3k} - 2} \tag{6}$$

Note that by choosing a large μ, Φ can be made small and $\cos\Phi \to 1$, which leads to an even tighter bound.

Now, we examine the "near" members, i.e. those members for which the first hop distance from the source is $\leq \mu R_{M-1}$. Let Δ be the maximum stretch for these members. Then, the maximum stretch at members is bounded by $\max(\Delta, s_{X,f})$.

If, N_n and N_f are the number of "near" and "far" members, and $s_{X,n}$ and $s_{X,f}$ be the respective bound on the stretch for these members, the average stretch of all members is bounded as:

$$\bar{s} \leq \frac{1}{N}(N_n.s_{X,n} + N_f.s_{X,f}) \tag{7}$$

where, $s_{X,n} \leq \Delta$. Note that, $N_n/N \leq 1$ and $N_f/N \leq 1$. The average stretch at the members is thus bounded by a constant that depends on the cluster size, k.

Stress: To calculate the average stress on links and nodes in the network, we make the same fluid approximation, which we briefly outline due to space constraints. Let $\mathcal{N}$ denote the number of links (nodes) in the network. The number of links that connect cluster leaders of layer L_i to their respective cluster members is given by $\mathcal{N}/k^i$. These links carry $\leq k.i$ replicated copies of a data packet. Then the average stress can be computed as:

$$\bar{\lambda} \leq \frac{1}{\mathcal{N}} \sum_{i=0}^{M=\log_k \mathcal{N}} \frac{\mathcal{N}}{k^i} ki = \frac{k^2}{(k-1)^2} + O(\frac{\log \mathcal{N}}{\mathcal{N}})$$

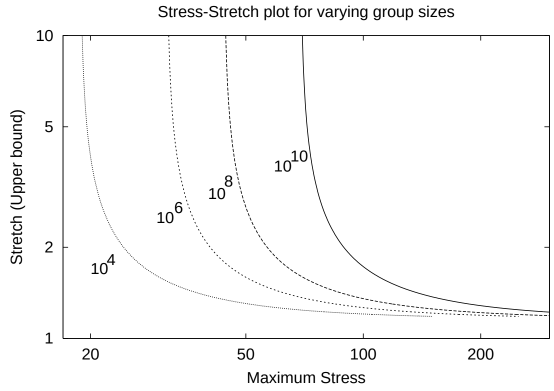

Fig. 4. Stretch vs Stress for the NICE protocol as the group size is varied. This is based on the uniform and dense distributed members on an Euclidean space.

Thus $\bar{\lambda} = k^2/(k-1)^2$ for asymptotically large $\mathcal{N}$. The maximum stress occurs at links close to the source for the NICE protocol. This can be calculated using a similar analysis and is given by:

$$\lambda_{\max} = k \log_k N \tag{8}$$

Stress vs Stretch: In Figure 4, we plot the upper bound of stretch at far members against the maximum stress, as derived by the fluid-based analysis for the dense distribution of a large number of members. The plots are obtained by choosing different values of k and calculating the corresponding values of $s_{X,f}$ (Equation 6) and $\lambda_{\max}$ (Equation 8). Both the axes are plotted in the logarithmic scale. As the number of members in the group increase, the shape of the plot is unchanged. However, the plots increasingly translate towards higher maximum stress. This is because the stretch does not depend on the group size, while the maximum stress increases with increasing group size.

Through detailed simulations, we have also studied the tradeoffs between the stress and stretch metric for randomly distributed group members on realistic topologies. The results conform to the analytic results obtained here.

Topology-aware techniques: Our analysis was based on the assumption of a large member population densely distributed in the network. However, in practice, the distribution of members in the network may not be dense, and the uniformity assumption may not hold. Gupta [5] defines a centralized topology-aware tree building algorithm that guarantees $O(1)$ stretch between *any pair* of the members on the tree. However, the stress at the members can be as large as $O(N)$.

As a part of our work, we have defined a simple modification to this algorithm, which can simultaneously guarantee $O(logN)$ stretch between any pair of members and $O(1)$ stress at the members if the underlying topology is known. Note that the bounds for these topology-aware centralized algorithms hold irrespective of member population size and the distribution of members in the network.

IV. Summary

Our work studies the tradeoff between stress and stretch for application layer multicast overlays, in particular with respect to the NICE application layer multicast protocol. This study quantifies the relationship between the two metrics and how the k parameter can be used to effectively tradeoff between these two metrics.

References

[1] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In *Proceedings of ACM SIGCOMM*, August 2002.

[2] Y. Chawathe. Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service. *Ph.D. Thesis, University of California, Berkeley*, December 2000.

[3] Y.-H. Chu, S. G. Rao, and H. Zhang. A Case for End System Multicast. In *Proceedings of ACM SIGMETRICS*, June 2000.

[4] P. Francis. Yoid: Extending the Multicast Internet Architecture, 1999. White paper http://www.aciri.org/yoid/.

[5] A. Gupta. Steiner points in tree metrics don't (really) help. In *Symposium of Discrete Algorithms*, January 2001.

[6] J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O'Toole. Overcast: Reliable Multicasting with an Overlay Network. In *Proceedings of the 4th Symposium on Operating Systems Design and Implementation*, 2000.

[7] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: An Application Level Multicast Infrastructure. In *Proceedings of 3rd Usenix Symposium on Internet Technologies & Systems*, March 2001.

[8] C. G. Plaxton, R. Rajaraman, and A. W. Richa. Accessing nearby copies of replicated objects in a distributed environment. In *ACM Symposium on Parallel Algorithms and Architectures*, 1997.

[9] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using content-addressable networks. In *Proceedings of 3rd International Workshop on Networked Group Communication*, November 2001.

[10] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. Katz, and J. Kubiatowicz. Bayeux: An architecture for scalable and fault-tolerant wide-area dat a dissemination. In *Eleventh International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001)*, 2001.