Designing Infrastructure-based Overlay Networks for Delay-sensitive Group Communications

Shaoyu Yang, Yoo-Ah Kim and Bing Wang

Computer Science & Engineering Department
University of Connecticut, Storrs, CT, 06269

***Abstract*— Infrastructure-based overlay networks have been proposed to support the quality of service requirements of a wide range of applications. In this paper, we study using infrastructure-based overlay for delay-sensitive low-bandwidth group communications, such as teleconferencing and chat room. In particular, we study where to place overlay nodes (called proxies) to minimize end-to-end delays. We formulate the problem of optimal proxy placement using integer-linear programming and quantify the benefits from using proxies in six real-world networks. We find that, perhaps surprisingly, only two out of the six networks benefit from using proxies. We furthermore use network characteristics to explain these benefits or lack of benefits. Last, for the two networks which benefit from using proxies, we find that a small number proxies (2 to 3) are sufficient to realize most of the performance gains.**

I. Introduction

The Internet has been increasingly used for group-oriented applications such as video-conferencing, online-gaming, chat-room, IPTV, and long-distance learning. These applications have various quality of service (QoS) requirements and often involve a large number of users. The current Internet, however, only provides a single class of best-effort service, with no delay or bandwidth guarantee to the applications. One solution to satisfy the applications' QoS requirements via the best-effort Internet is through infrastructure-based overlay networks (e.g., [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]). In such overlay networks, overlay nodes (referred to as *proxies*) are deployed by a third party to provide more flexible routing inside the network. This infrastructure is scalable and easy to manage since the number of proxies is much smaller than that of end users and the overlay network service provider can directly control the proxies.

In this paper, we study using infrastructure-based overlay networks to support delay-sensitive low-bandwidth group communications, such as teleconferencing and chat room. In particular, we seek to answer the following questions: *What are the benefits from using proxies in realistic networks? Where should we place the proxies to minimize end-to-end delays? How many proxies should be placed to achieve significant benefits?* To answer these questions, we formulate the problem of optimal proxy placement, and explore the benefits from using optimal proxy placement in six real-world single-ISP networks (they are inferred by [12], [13]). Our paper makes the following three main contributions:

- When there is no constraint on the number of proxies, we solve the problem of optimal proxy placement using the shortest-path algorithm. When the number of proxies is limited, we prove that the problem is NP-hard and solve it using integer-linear programming (ILP).
- We quantify the benefits from optimal proxy placement and find that, perhaps surprisingly, only two out of the six networks benefit from using proxies, even when the number of proxies is not limited. We furthermore use network characteristics to explain these benefits or lack of benefits.
- For the two networks which benefit from using proxies, we limit the number of proxies and obtain the optimal proxy placement using ILP. We discover a diminishing gain from using proxies in reducing end-to-end delays and, in particular, 2 to 3 proxies are sufficient to realize most of the performance gains.

At the high level, our results indicate that, in contrast to the tremendous benefits from deploying proxies at the edge of a network (e.g., Akamai [14]), deploying proxy in the core of a network is not necessarily beneficial. Therefore, it is important to rigourously quantify the benefits from using proxies before deployment.

Certain aspects of infrastructure-based overlay network design (e.g., bandwidth provisioning, overlay topology, overlay node location, and content replication) have been considered in [7], [15], [16], [17], [10]. However, all of these studies focus on unicast instead of multicast applications. Furthermore, their optimization goals are not to minimize end-to-end delays. The study of [18] exploits underlying network topology to construct overlay networks for group communication. This study, however, does not rely on any infrastructure support, and hence is in a context different from that in our study. The study of [6] aims to achieve a bounded delay for multicast applications in the presence of proxies. Their focus is on how to form a tree, taking advantage of the capacities of the proxies, not on where to place proxies. Last, the work [19] designs infrastructure-based overlay networks for multicast applications by dividing the problem into three sub-problems: placing proxies, connecting proxies and reserving bandwidths between proxies. The authors solve each sub-problem using heuristics. We consider the first two sub-problems with the goal of minimizing *end-to-end delays* and our integer-linear programming provides *optimal* solutions (we do not consider the last sub-problem since our focus is on applications with low bandwidth requirements).

The rest of the paper is organized as follows. Section II describes the problem setting and Section III presents our approaches. In Section IV, we apply our approaches to real-world networks to quantify delay reduction from using proxies for group communications. Finally, Section V concludes the paper and presents future work.

II. Problem Setting

Consider an infrastructure-based overlay network. We represent this network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of nodes (routers) in the network and $\mathcal{E}$ is the set of physical links connecting the nodes. Let d_l be the delay of link l. We assume that the underlying routing protocol determines the default IP path based on metrics other than the delay (e.g., the weights of links), and therefore the default IP path between two nodes may not necessarily be the path with the minimum delay. We do not consider bandwidth constraints of the links since the applications we consider have low bandwidth requirements (e.g., teleconferencing and chat room, with bit rate of tens of or hundreds of Kbps). As to be shown in Section III, even in this setting, the problem of optimal proxy placement is NP-hard.

The infrastructure-based overlay network is used to support a set of delay-sensitive low-bandwidth group communications. Let S denote the set of sources. Each source $s \in S$ has a group of receivers, denoted as R_s. A source and its corresponding receivers form a multicast group. A source or a receiver is associated with an access router in the network. Then data from a source first reach its access router and then reach the receiver via the access router of the receiver. Multiple receivers may have the same access routers. Since the route between an end user (a source or a receiver) and its access router is fixed (and hence has a constant delay), for simplicity, we only consider delays inside the overlay network. In the following, the source of a multicast group refers to the access router of the source; similarly, the receivers refer to the access routers of the receivers. Let $D_p(u, v)$ denote the delay on the default IP path from router u to router v. Then $D_p(u, v)$ is simply the sum of delays of links on the default IP path. As described earlier, $D_p(u, v)$ may not be the minimum delay from router u to router v.

Our goal is to place proxies at appropriate places so that we can minimize delays experienced in group communications. Suppose that we are allowed to choose at most N_V proxies from V and at most N_E pairs of proxies can be maintained as *overlay links* (an overlay link is a network path connecting two proxies). We assume that proxies are located at the routers. The proxies can be used as relays to forward data from a source to a receiver. That is, source s can send data to a proxy first and then the proxy forwards the data to another proxy or to receiver r directly. Let $D(s, r)$ represent the delay from source s to receiver r in the overlay network. It depends on the path from s to r. When the path from s to r is the default IP path, we have $D(s, r) = D_p(u, v)$. If the path from s to r is $(s, o_1, ..., o_k, r)$, where o_i is a proxy, $i = 1, \ldots, k$, we have $D(s, r) = D_p(s, o_1) + \sum_{i=1}^{k-1} D_p(o_i, o_{i+1}) + D_p(o_k, r)$.

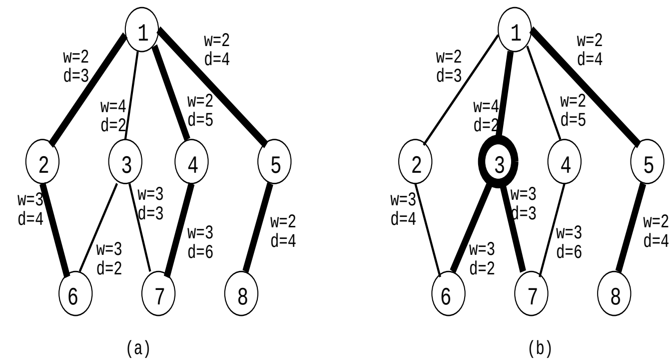

Fig. 1. Illustration of default IP paths, overlay paths and placement of proxies. Link weights and delays are marked on links. (a) Default IP paths (in thick lines) from the source to the receivers. (b) Paths from the source to the receivers (in thick lines) after placing a proxy at router node 3.

Given a set of multicast groups, our problem is to find the optimal placement of proxies and an overlay path between each source and receiver pair to minimize the end-to-end delays. In particular, we consider two objective functions — minimizing the total delay over all source and receiver pairs, i.e.,

$$\text{minimize:} \sum_{s \in S} \sum_{r \in R_s} D(s, r), \tag{1}$$

and minimizing the sum of the maximum delays, i.e.,

$$\text{minimize:} \sum_{s \in S} \max_{r \in R_s} D(s, r). \tag{2}$$

We now illustrate how proxies and overlay links can be used to reduce the delay using an example. Fig. 1 shows an overlay network with 8 routers. Each link in the network is marked with a delay and a weight. The routing protocol determines the default IP path to be the path with the minimum weight. We assume that node 1 is the source, and nodes 6, 7 and 8 are receivers. The thick lines in Fig. 1(a) indicate the default IP paths from the sender to all the receivers. For example, the default IP path from node 1 to node 6 is path $(1, 2, 6)$ with a total weight of 5 and a total delay of 7. The default IP path from node 1 to node 7 is path $(1, 4, 7)$ with a total delay of 11. To reduce the delays, we can place a proxy at node 3 (See Fig. 1 (b)). By using the proxy as a relay, node 6 can now receive data over path $(1, 3, 6)$ and node 7 can use path $(1, 3, 7)$. As a result, the delays from node 1 to nodes 6 and 7 are reduced from 7 to 4, and from 11 to 5, respectively.

III. Optimal Placement of Proxies

In this section, we present the algorithms to find an optimal placement of proxies to minimize end-to-end delays. We first consider a special case where there are no constraints on the number of proxies and overlay links. This case provides the maximum benefits from using proxies without considering the

$$\text{minimize: } \sum_{s \in S} \sum_{r \in R_s} D(s,r) \quad (3)$$

or

$$\text{minimize: } \sum_{s \in S} \max_{r \in R_s} D(s,r) \quad (4)$$

subject to:

$$x_v \in \{0,1\}, v \in V \quad (5)$$

$$x_e \in \{0,1\}, e \in E \quad (6)$$

$$x_e \le x_u, x_e \le x_v, e=(u,v) \in E \quad (7)$$

$$\sum_{v \in V} x_v \le N_V \quad (8)$$

$$\sum_{e \in E} x_e \le N_E \quad (9)$$

$$f_r^s(u,v) \in \{0,1\}, s \in S, r \in R_s, (u,v) \in E \quad (10)$$

$$0 \le f_r^s(u,v) \le x_e, s \in S, r \in R_s,$$
$$e=(u,v) \in E, u \ne s, v \ne r \quad (11)$$

$$0 \le f_r^s(s,v) \le x_v, s \in S, r \in R_s \quad (12)$$

$$0 \le f_r^s(v,r) \le x_v, s \in S, r \in R_s \quad (13)$$

$$\sum_{v \in V \setminus \{s\}} f_r^s(s,v) = 1, s \in S, r \in R_s \quad (14)$$

$$\sum_{v \in V \setminus \{s\}} f_r^s(v,s) = 0, s \in S, r \in R_s \quad (15)$$

$$\sum_{v \in V \setminus \{r\}} f_r^s(v,r) = 1, s \in S, r \in R_s \quad (16)$$

$$\sum_{v \in V \setminus \{r\}} f_r^s(r,v) = 0, s \in S, r \in R_s \quad (17)$$

$$\sum_{u \in V} f_r^s(u,v) = \sum_{w \in V} f_r^s(v,w), s \in S, r \in R_s,$$
$$v \in V \setminus \{s,r\} \quad (18)$$

$$D(s,r) = \sum_{(u,v) \in E} f_r^s(u,v) D_p(u,v),$$
$$s \in S, r \in R_s \quad (19)$$

Fig. 2. Integer-linear programming formulation for optimal proxy placement.

cost of overlay deployment. We then consider the general case where we can use only N_V proxies and N_E overlay links. In this case, we prove that the problem is NP-hard for both objective functions (1) and (2). We formulate the problem as an integer linear programming (ILP), which is used to obtain optimal proxy placements in real networks (see Section IV-B).

A. Unlimited number of proxies

When there are no constraints on the number of proxies and overlay links, we can find an optimal proxy placement to minimize the delay between each source and receiver pair using the shortest path algorithm as follows. Given a network graph $\mathcal{G}$, we construct *a logical graph* $G=(V,E)$, which is a complete graph where $V=\mathcal{V}$. Each edge in E is a *logical link*, connecting a pair of nodes via the default IP path determined by the underlying routing protocol. The delay on the logical link $e=(u,v) \in E$ is $D_p(u,v)$, i.e., the delay on the default IP path from router u to v. We can now find the shortest-delay path from source s to receiver r in the logical graph G using the delays on the logical links (e.g., using Dijkstra's algorithm). Any intermediate node on the shortest-delay path from source s to receiver r is a proxy. This is because if the shortest-delay path from source s to receiver r thus found is $(s, v_1, \ldots, v_k, r)$, that is, s and r are not connected directly by their default IP path, the intermediate nodes $v_1, \ldots, v_k$ forward data for source s and hence are proxies. This solution gives the maximum benefit from using the overlay network when we are allowed to use as many proxies as necessary.

B. Limited number of proxies

Once we restrict the number of proxies that we can use, the problem of optimal proxy placement becomes NP-hard by a reduction from the SET COVER problem. The proof is found in the Appendix. We next formulate the problem as an integer linear programming (ILP), which can be solved by optimization tools (e.g., CPLEX [20]).

Our ILP formulation is shown in Fig. 2. For simplicity, we assume that one unit of data is sent from a source to the receivers in a multicast group. We consider a complete logical graph $G=(V,E)$ as in Section III-A. Let x_v represent whether a vertex $v \in V$ is chosen as a proxy: $x_v=1$ if v is chosen as a proxy, and $x_v=0$ otherwise. Similarly, let $x_e=1$ represent that an edge $e \in E$ is selected as an overlay link connecting two proxies in the overlay network, and $x_e=0$ otherwise. Then, for an edge $e=(u,v)$, we have Constraint (7) $x_e \le x_u$ and $x_e \le x_v$. That is, x_e can be an overlay link only if both u and v are proxies. We can choose at most N_V proxies and N_E overlay links, which gives Constraints (8) and (9), respectively.

We now construct a multicast tree for each multicast group using overlay links. We assume a single path is used to deliver data from a source to a receiver. Let $f_r^s(u,v)$ represent whether we use the overlay link (u,v) to send data from s to r where $s \in S$, $r \in R_s$. That is, $f_r^s(u,v) \in \{0,1\}$. Then we have Constraint (11) $0 \le f_r^s(u,v) \le x_e$ for $u \ne s$, $v \ne r$, and $e=(u,v)$ as we can use the link only when it is selected as an overlay link. Note that the above constraint does not apply to the logical link of (s,r) since a source and receiver pair is always allowed to use the default IP path between them. Furthermore, we allow source s to send to a proxy node v even if the link (s,v) is not an overlay link, which leads to Constraint (12), $0 \le f_r^s(s,v) \le x_v$. Similarly, we have Constraint (13), $0 \le f_r^s(v,r) \le x_v$ for receiver r. Constraints (14)-(18) ensure that exactly one unit of flow is sent from s to r, which determines a unique path from s to r.

Recall that $D_p(u,v)$ denotes the delay on the default path from node u to v, and $D(s,r)$ denotes the delay from source s to receiver r in the overlay network. Then $D(s,r)$ can be computed as $\sum_{(u,v) \in E} f_r^s(u,v) D_p(u,v)$, which gives Constraints (19). Our objective is to minimize the sum of

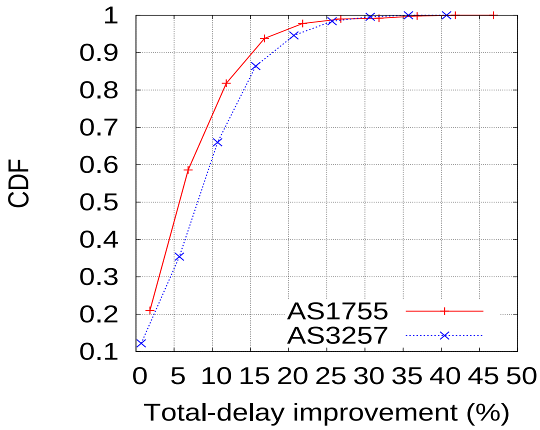

Fig. 3. CDF of the total-delay improvement for AS3257 and AS1755 from 500 settings.

$D(s, r)$ over all source and destination pairs or the sum of the maximum $D(s, r)$ in each multicast group.

IV. Benefits from optimal proxy placement

We explore the benefits from optimal proxy placement in six ISP maps, inferred using end-to-end measurements [12], [13]. These ISPs are in the US, Europe and Australia. Their AS numbers and names are listed in Table I, along with the number of routers and links in each ISP map. Each link in these maps is annotated with weights and delays. The default IP path between a pair of nodes are determined using link weights as the cost metric. These are the only network maps with both delay and weight information in the public domain that we are aware of. We use two performance metrics to quantify delay reduction from using optimal proxy placement. The first is the *total-delay improvement*, i.e.,

$$\sum_{s\in S}\sum_{r\in R_s} D_p(s,r) / \sum_{s\in S}\sum_{r\in R_s} D_o(s,r) - 1,$$

where $D_o(s, r)$ is the delay from source s to receiver r after the optimal proxy placement. Recall that $D_p(s, r)$ is the delay from source s to receiver r on the default IP path (see Section II). The second metric is the *max-delay improvement*, i.e.,

$$\sum_{s\in S}\max_{r\in R_s} D_p(s,r) / \sum_{s\in S}\max_{r\in R_s} D_o(s,r) - 1.$$

In the following, we first explore the delay improvements when not limiting the number of proxies (i.e., the maximum amount of delay improvements that can be achieved from the overlay networks). We then restrict the number of proxies and explore the benefits from a limited number of proxies.

A. Unlimited number of proxies

When not limiting the number of proxies, for each network, we randomly generate 500 settings. Each setting contains two multicast groups. In each multicast group, a node is randomly selected as the source and 1000 receivers are uniformly associated with 20 nodes that are selected randomly. Table I lists the delay improvements for these six networks. Perhaps surprisingly, only two out of the six networks, namely AS3257 and AS1755, have significant total-delay improvements when using proxies. For max-delay improvement, we only observe significant improvement in AS1755. Fig. 3 plots the CDF (Cumulative Distribution Function) of the total-delay improvements in AS3257 and AS1755 from the 500 settings. In AS1755, 25% of the settings have a total-delay improvement over 10%; the maximum improvement being 42%. In AS3257, 35% of the settings have a total-delay improvement over 10%; the maximum improvement being 37%.

We now investigate why using proxies leads to dramatically different delay improvements in different networks by analyzing the characteristics of the networks. In particular, we consider the following three statistics:

- Correlation coefficient of the link delays and weights. We expect benefits from using proxies when the correlation between the link delays and weights is low (when the correlation is high, the default IP path from a source to a destination may coincide with their shortest-delay path and using proxies does not help to reduce delays).
- Coefficient of variation (CV) of the delay-weight ratios (i.e., the ratio of delay over weight) of all the links in the network. We use this statistic since we expect larger delay improvement when the delay-weight ratios have a higher variation (when the delay-weight ratios of all the links are the same, using proxies does not reduce delay since the default IP path is the shortest-delay path).
- Network sparsity. Intuitively, we expect benefits from using proxies when the network is relatively dense (so there is sufficient amount of path redundancy). We define the sparsity of a network as $\ln(|\mathcal{E}|)/\ln(|\mathcal{V}|)$, where $|\mathcal{E}|$ and $|\mathcal{V}|$ are respectively the number of links and routers in the network. For a tree-like graph, this statistic is close to 1; while for a complete graph, this statistic is close to 2.

Table I presents the above statistics for the six networks. First, we observe that the correlation coefficient for AS1755 (i.e., 0.06) is close to zero, indicating little correlation between link delays and weights, which explains the benefits from using proxies in this network. Secondly, we observe the highest CV (of the delay-weight ratios) in AS3257 and AS1755, consistent with the results of higher benefits in these two networks. Last, the AS1221 is much sparser than the other networks, which is consistent with its low delay improvements.

B. Limited number of proxies

We next focus on AS3257 and AS1755, the two networks with benefits from using proxies. For these two networks, we restrict the number of proxies and explore the benefits from a limited number of proxies. This is motivated from the observation that, when not limiting the number of proxies, a large number of proxies may be used to achieve the maximum delay improvement. Fig. 4 plots the CDF of the number of proxies used to achieve the maximum delay improvement when not limiting the number of proxies, obtained from the

TABLE I

BENEFITS FROM OPTIMAL PROXY PLACEMENT IN SIX ISP MAPS (NOT LIMITING THE NUMBER OF PROXIES).

AS	Name	# of Routers	# of Links	Total-delay improvement	Max-delay improvement	Corr. coef.	CV	Sparsity
3257	Tiscali (Europe)	164	328	13.3%	3.2%	0.17	1.01	1.14
1755	Ebone (Europe)	88	161	11.8%	11.8%	0.06	0.96	1.14
1239	Sprint (US)	323	972	4.4%	2.7%	0.37	0.63	1.19
3967	Exodus (US)	80	147	4.3%	2.9%	0.18	0.72	1.14
6491	Abovenet (US)	145	376	3.7%	2.4%	0.22	0.76	1.19
1221	Telstra (Australia)	115	153	2.3%	2.4%	-0.13	0.82	1.06

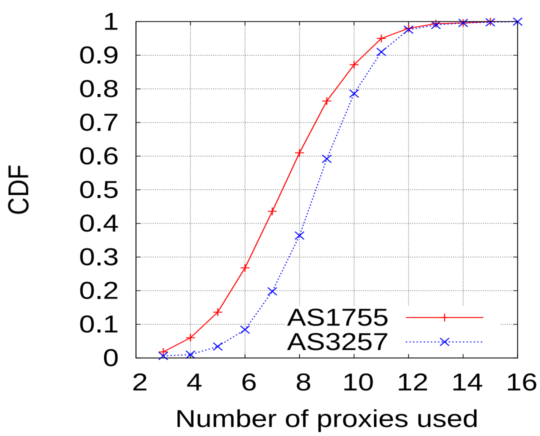

Fig. 4. CDF of the total number of proxies used to achieve the maximum delay improvement (when not limiting the number of proxies).

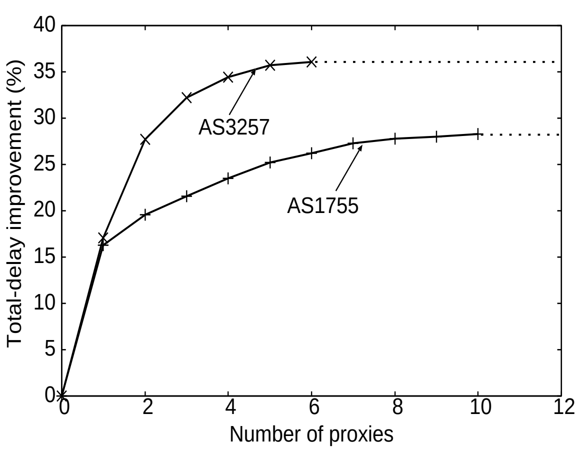

Fig. 5. Total-delay improvements when increasing the number of proxies.

500 generated settings. We observe that a maximum of 16 proxies are used to achieve the maximum delay improvement in these two networks. A natural question is: how many proxies are required to achieve most of the performance gains?

To answer the above question, for both AS3257 and AS1755, we randomly choose 5 settings (from the 500 settings) in which the total-delay improvement is at least 20% when not limiting the number of proxies. In each setting, we increase the number of proxies, N_V, from 1 until the increment does not lead to any additional gain. The number of overlay links, N_E, is not restricted. For each value of N_V, the optimal proxy placement is obtained by solving the ILP (see Fig. 2) using CPLEX [20]. For all the settings, we observe a diminishing gain from increasing the number of proxies on reducing end-to-end delays. Two examples are shown in Fig. 5, which plots the results for one setting in AS3257 and one setting in AS1755. We observe a significant delay reduction (compared to not using proxies) when using 2 to 3 proxies; the delay reduction is less significant afterwards. The results for other settings are similar. In summary, we observe that using 2 proxies obtains 55% to 78% of the maximum gains while using 3 proxies obtains 72% to 89% of the maximum gains. The above results indicate that a small number of proxies (2 or 3) can achieve most of the performance gains.

We further explore the characteristics of the nodes chosen as proxies in the optimal proxy placement. This study may help designing heuristic algorithms for optimal proxy placement (for networks too large to be solved by ILP). We next report the results for the optimal proxy placement when $N_V = 3$. In AS1755, a node located at London, United Kingdom is chosen as a proxy in all of the 5 settings we examined. Its degree is 5 (the average node degree in AS1755 is 3.7). In AS3257, two nodes, one located at Frankfurt, Germany and one located at Milan, Italy, are chosen as proxies in all of the 5 settings. Their node degrees are respectively 12 and 6 (the average node degree in AS3257 is 4.1). The above indicates that, despite the choice of multicast groups, certain nodes are chosen as proxies in the optimal proxy placement. They tend to have relatively high degrees (above average). Further investigation of proxy characteristics is left as future work.

C. Discussion

All the above results are for single-ISP networks. In general, these results indicate that using proxies inside a single-ISP network does not provide significant benefits — even for the two networks with benefits, less than 5% of the settings have total-delay improvement over 25%. For a network consisting of multiple ISPs, weights on peering links may be purposely set to reflect routing policies, not correlated with the actual delays. In these cases, using proxies may provide more delay reduction than that in a single-ISP network. However, we are not aware of any such data in the public domain to validate this conjecture.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we studied where to place proxies in infrastructure-based overlay networks to minimize end-to-end delays for delay-sensitive low-bandwidth group communications. We solved the problem of optimal proxy placement using the shortest-path algorithm when there is no constraint on the number of proxies. When the number of proxies is limited, we proved that the problem of optimal proxy placement is NP-hard and formulated it using integer-linear programming (ILP). We then quantified the benefits of using proxies in six real-world networks and used network characteristics to explain such benefits or lack of benefits. Last, we discovered a diminishing gain from adding proxies on reducing end-to-end delays. In particular, we found that 2 to 3 proxies are sufficient to realize most of the gains. As future work, since it may not be feasible to obtain the optimal solution using ILP directly for large networks, we plan to develop approximate algorithms for proxy placement.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their insightful comments. We also would like to thank Jun-Hong Cui, Swapna Gokhale, Jim Kurose and Don Towsley for helpful discussions. This work is partly supported by UCONN Large Grants FRS 449250 and FRS 449251.

REFERENCES

[1] Y. Chawathe, S. McCanne, and E. A. Brewer, "RMX: Reliable multicast for heterogeneous networks," in *Proc. IEEE INFOCOM*, March 2000.

[2] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. J. W. O'Toole, "Overcast: Reliable multicasting with an overlay network," in *Proceedings of USENIX Symposium on Operating Systems Design and Implementation*, October 2000.

[3] Y. Chawathe, S. McCanne, and E. A. Brewer, *An Architecture for Internet Content Distributions as an Infrastructure Service*, 2000. Unpublished, http://www.cs.berkeley.edu/yatin/papers/.

[4] S. Shi, J. S. Turner, and M. Waldvogel, "Dimensioning server access bandwidth and multicast routing in overlay networks," in *Proceedings of NOSSDAV'01*, June 2001.

[5] S. Shi and J. S. Turner, "Routing in overlay multicast networks," in *Proc. IEEE INFOCOM*, June 2002.

[6] N. Malouch, Z. Liu, D. Rubenstein, and S. Sahu, "A graph theoretic approach to bounding delay in proxy-assisted, end-system multicast," in *Proceedings of the Tenth IEEE International Workshop on Quality of Service (IWQoS)*, (Miami Beach, FL), May 2002.

[7] Z. Duan, Z.-L. Zhang, and Y. T. Hou, "Service overlay networks: SLAs, QoS, and bandwidth provisioning," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 870–883, 2003.

[8] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *Proc. IEEE INFOCOM*, April 2003.

[9] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz, "OverQoS: An overlay based architecture for enhancing Internet QoS," in *USENIX NSDI*, 2004.

[10] T. V. Nguyen, F. Safaei, P. Boustead, and C. T. Chou, "Provisioning overlay distribution networks," *Elsevier Journal of Computer Networks*, pp. 103–118, 2005.

[11] L. Lao, J.-H. Cui, and M. Gerla, "TOMA: A viable solution for large-scale multicast service support," *In Proceedings of IFIP Networking 2005, Waterloo, Ontario, Canada*, May 2-6 2005.

[12] D. W. Ratul Mahajan, Neil Spring and T. Anderson, "Inferring link weights using end-to-end measurements," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, 2002.

[13] R. M. Neil Spring and D. Wetherall, "Measuring ISP topologies with Rocketfuel," in *Proc. ACM SIGCOMM*, August 2002.

[14] http://www.akamai.com.

[15] Z. Li and P. Mohapatra, "The impact of topology on overlay routing serive," in *Proc. IEEE INFOCOM*, March 2004.

[16] H. Zhang, J. Kurose, and D. Towsley, "Can an overlay compensate a careless underlay?," in *Proc. IEEE INFOCOM*, April 2006.

[17] J. Han, D. Watson, and F. Jahanian, "Topology aware overlay networks," in *Proc. IEEE INFOCOM*, March 2005.

[18] M. Kwon and S. Fahmy, "Topology aware overlay networks for group communication," in *Proceedings of NOSSDAV'02*, May 2002.

[19] L. Lao, J.-H. Cui, and M. Gerla, "Multicast service overlay design," *In Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'05), Philadelphia, PA, USA*, July 2005.

[20] http://www.ilog.com/products/cplex/.

APPENDIX
NP-HARDNESS PROOF

Proof: We prove that the problem of optimal proxy placement is NP-hard for both objective functions (1) and (2) by a reduction from the SET COVER PROBLEM to our problem. In the SET COVER PROBLEM, we are given a set of elements, denoted as M, and a collection of subsets of elements, denoted as C. We want to check if we can select k subsets in C such that all elements in M are covered. This problem can be reduced to our problem as follows. For each element i in M, we create two nodes s_i and r_i. We want to send data from s_i to r_i for all i. For each subset $X \in C$, we create a node t_X. If an element i belongs to X, then we create an edge between s_i and t_X, and also an edge between r_i and t_X. Furthermore, on both edges, the delay and weight are set to one. For all the other pairs of nodes, the edge connecting a pair of nodes has the delay and weight of much larger than one.

It is easy to see that we can choose k proxies such that the total delay (or the sum of the maximum delays) is exactly $2|M|$ if and only if there are k subsets to cover all elements in the instance of the SET COVER PROBLEM. ■