Constructing a Proximity-Aware Power Law Overlay Network

Jianjun Zhang, Ling Liu, and Calton Pu
College of Computing
Georgia Institute of Technology
{zhangjj, lingliu, calton}@cc.gatech.edu

Abstract

Peer-to-Peer (P2P) networks offer a message exchanging overlay for distributed applications such as file sharing, application layer multicast, and publisher/subscriber system. The communication efficiency of the underlying overlay network is thus one of the primary factors that determine the performance of those applications. In this paper, we propose a P2P overlay network aiming at offering the low maintenance overhead of unstructured P2P networks and the scalability and communication efficiency of structured P2P networks. We design a distributed algorithm to construct low-diameter overlay networks with power law topologies. Peers consider both network proximity information and capacity of existing peers when choosing their P2P network neighbors. Using an application layer multicast system as our example, we demonstrate that our system can provide generic, scalable, and low diameter overlay networks for distributed applications that demand efficient P2P communication supports. [1]

1. Introduction

Recent research works in Peer-to-Peer (P2P) networks show a promising paradigm to harness widely distributed, loosely coupled, and inherently unreliable end-hosts for providing distributed services. Consequently, a wide spectrum of P2P networks have been proposed and adopted as communication overlays for large scale distributed applications such as file sharing, publisher/subscriber system, and application layer multicast.

The work presented in this paper addresses the outstanding problem of constructing a generic, efficient, and scalable overlay network. In summary, we want our overlay network to have the following properties.

- The overlay network must be scalable enough to accommodate large number of end-hosts. It should be able to handle network dynamics gracefully without compromising its services.
- The communication latency among any pair of end-hosts should be bounded as the number of end-hosts in the overlay network grows.
- The workloads on each end-host should be matched to its capacity, so that it can avoid being overloaded and introducing bottlenecks into the system.

We proposed a distributed algorithm to construct overlay networks that have the low maintenance overhead of unstructured P2P networks and the efficiency and scalability of structured P2P networks. Compared to the existing works, our algorithm is unique in that it constructs power law topologies that can grow while maintaining low network diameter. Our algorithm uses a build-in mechanism to capture network proximity information of peers. Based on the capacity of each peer, our algorithm gives different weights on network proximity information and peer connectivity information when choosing its P2P neighbors. Less powerful peers are clustered by their network proximity, whereas powerful peers are implicitly assigned with more P2P neighbors and work as the forwarding hub of the overlay networks. Messages exchanged in the overlay network are efficiently forwarded along P2P network links that conform to the underlying IP network topology.

2. Related Works

We observe three classes of P2P networks in the literature. Nevertheless, none of them could offer all the features we ask for. Structured P2P networks [13, 14, 16] offer bounded network diameters by posting a strict regulation on network topologies. However, it is widely recognized that the cost of maintaining predetermined topologies against network dynamics may cause degraded system performance. On the contrary, unstructured P2P networks [3, 4] are known for their simplicity and low maintenance overhead against network dynamics such as peer joining, departure, and failure. Yet due to the randomness of the network topologies, those systems offer no guarantee on communication efficiency. The third type of P2P networks are improving works on unstructured P2P net-

[1] This research is partially supported by NSF CNS, NSF ITR, CERCS Research Grant, IBM Faculty Award, IBM SUR grant, and HP Equipment Grant. Any opinions, findings, and conclusions or recommendations expressed in the project material are those of the authors and do not necessarily reflect the views of the sponsors.

work, represented by [8, 18]. Those systems proposed various mechanisms to regulate unstructured overlay topologies and/or optimize the system performance at the application level. The improved P2P systems are featured with bounded network diameter or the scalability of query processing. Nevertheless, few of them consider the network proximity when constructing overlay networks. For applications like application layer multicast, such inefficient communication services of the P2P network will cause less optimal application performance.

Research works in natural systems [17] and man-made environments [15, 11] discover that the topologies of those systems usually present power law distribution. Later research works [5] show that such topology can grow while maintaining a low network diameter, i.e. the average shortest path between two nodes in term of number of hops.

Because of this scale-free property, a number of P2P systems have been proposed to generate topologies that follow power law distribution. Pandurangan *et al.* [12] proposed a distributed algorithm for building low-diameter P2P networks with bounded degrees. However, they assume a stochastic model for the arrivals and departures of peers. They trace the status of peers and coordinate the connections among them with a central server, which limits the system scalability and reliability. Phenix [18] generates power law topologies of unstructured P2P networks using a preferential attachment mechanism. Nevertheless, neither of them considered the network proximity information in overlay construction.

The idea of ranking different peers into different overlay service layers has been exploited by a number of P2P systems. For unstructured P2P network, KaZaA [4] uses the notion of "supernode" and Gnutella v.0.6 [3] has "ultrapeer". In structured P2P network, such peers are referred to as "supernodes" in [21] and are organized into another layer of overlay called "expressway" [19] to accelerate the routing services. Those powerful peers are assigned with more workloads and serve as the "hubs" in the overlay network. In most of the cases [3, 21], ordinary peers have only connections to the supernodes. Sometimes [19], ordinary peers may resort to other ordinary peers when the service of "expressway" is not available. As the result of the predetermined hierarchical architecture, such schemes introduce a few vulnerabilities into overlay networks. First, supernodes are assumed to be stable and possess enough resources to serve their duties. When they are attacked or overloaded, the overlay network might be fragmented if normal peers rely solely on them for services. Secondly, to efficiently route the requests from normal peers, each supernode may keep state information of the normal peers it serves. Supernodes exchange such state information for efficient and accurate routing. Because the state information is usually closely tied to application semantics, it is hard to design a generic and versatile overlay network that can meet the service requirements of different applications. Finally, the system would also be vulnerable when malicious peers assume the role of supernodes and trick other overlay peers into relying on them for services.

3. System Architecture

In our system, we took a different approach to construct efficient and low-diameter overlay network. We use a distributed algorithm to construct an unstructured power law network. Powerful peers are inserted into the same P2P overlay that other peers participate, rather than being explicitly put into a different routing layer. When a new peer joins the overlay, it gathers the information of a number of existing peers as candidates. The new peer decides the likelihood of connecting to a neighbor candidate by evaluating its network proximity and connectivity information. Depending on the capacity of the new peer, it associates different weights to these two types of information in its decision making.

3.1 Topology Construction Algorithm

Each peer in our overlay is an end-host and is uniquely identified by a tuple of four attributes, i.e. ($IP\ address$, $port\ number$, $network\ coordinate$, $capacity$). The network coordinate is measured using mechanisms such as Vivaldi [9] or GNP [1]. The physical network distance between any two peers can be estimated with satisfactory precision using the distance between their network coordinates. Because the performance of an end-host in a distributed environment like P2P network is largely decided by its accessible network bandwidth, we use this information to gauge the capacity of each peer. It can be specified by end users or estimated using certain network probing techniques.

A joining peer i obtains a list of existing peers by contacting a bootstrapping server or recycling its local neighbor cache. The bootstrapping server is an extension of Gnucleus [2]. It records a list of peers that are currently active in the P2P network. The joining peer i sends its own network coordinate to the bootstrapping server together with the query message. When a bootstrapping server receives such a query request, it sorted its cached entries in ascendant order by their network coordinate distances to peer i. It selects a list of peers $L_i = (LD_i, LR_i)$ as the neighbor candidates of peer i, where LD_i is a few peers selected from the top of the sorted cache entry list and LR_i is a few randomly selected ones. In our system, we set $|LR_i| = |LD_i|$ and let $5 \leq |LR_i| + |LD_i| \leq 8$ peers. This is also the default setting used by Gnutella networks [3].

To each candidate $k \in L_i$, peer i sends a probing message as:

$$M_{prob} = \langle\ source = i,\ type = prob,\ TTL = 0,\ hops = 0\ \rangle$$

Each candidate k will send back a responding message

M_{prob_resp}, together with its neighbor list N_k.

$$M_{prob\_resp} = \langle\ source = k,\ type = prob\_resp,\ TTL = 0,\ hops = 0,\ prop = N_k\ \rangle$$

Peer i assembles all the neighbor information contained in the probing replies into a candidate list LC_i. For each unique peer $j \in LC_i$, peer i calculates two types of information. The $frequency$ $f_i(j)$ of peer j records the number of appearances of peer j in LC_i. The $estimated\ distance$ $D(i,j)$ is the network coordinate distance between peer i and peer j. The $normalized\ distance\ estimation$ $d_i(j)$ is defined as:

$$d_i(j) = \frac{D(i,j)}{\mathrm{MAX}_{k \in LC_i} D(i,k)} \tag{1}$$

where $0 < d_i(j) \leq 1$.

As LC_i serves as a sampling of peers in the P2P network, $f_i(j)$ is the sampling of the degree of each candidate j, and $d_i(j)$ is the estimation of the actual network distance between peer i and peer j.

We define the $Connection\ Preference$ of peer i to peer j as:

$$P_i(j) = \gamma \cdot PF_i(j) + (1-\gamma) \cdot PD_i(j) \tag{2}$$

Here, $PD_i(j)$ denotes the $Distance\ Preference$ of peer i connecting to peer j. It is defined as the probability that peer i chooses peer j as its P2P network neighbor, based on the network distance between them. It is defined as:

$$PD_i(j) = \frac{\frac{1}{d_i(j)} - \alpha}{\sum_{k \in LC_i} \frac{1}{d_i(k)} - \alpha} \tag{3}$$

where $-\infty < \alpha \leq 1$.

Similarly, $PF_i(j)$, the $Degree\ Preference$ of peer i to peer j, is the probability that peer i chooses peer j as its P2P network neighbor based on the frequency of peer j. The more incident edges peer j has in the P2P network, the more likely it has higher frequency in the candidate lists of other peers. We define $PF_i(j)$ as:

$$PF_i(j) = \frac{f_i(j) - \beta}{\sum_{k \in LC_i} f_i(j) - \beta} \tag{4}$$

where $-\infty < \beta \leq 1$.

By choosing different values for parameters α, β, and γ, we can tune the overlay network according to the requirement of different applications. For an overlay networks supporting applications that are sensitive to communication latency, we can use larger values of α and γ. On the contrary, for an overlay network that emphasizes more on load balancing, a larger value for β and a lower value for γ is more preferable.

The values of parameter α, β, and γ can be mathematically calculated using techniques like the one used in [7], providing that we know the number of peers and the power law distribution parameters. However, in a distributed environment like P2P network, it is hard to predict the exact number of peers and their behavior.

In our system, we use the capacity information of each peer to decide the values of those parameters. We define $Resource\ Level$ r_i to reflects the capability peer i possesses. It is defined as the proportion of peers that have less capacities than peer i in the overlay network. It satisfies constraint $0 \leq r_i \leq 1$. Specifically, we set the preferential parameters as $\alpha = 1 - r_i$, $\beta = r_i$, and $\gamma = r_i^{-ln(r_i)}$. Such assignments exactly reflect our design rationale: the capacity of a peer should be used to decide the properties of its connections in the overlay network. More powerful peers should connect to other peers that are equally powerful and care less for the network proximity, whereas peers with limited resources should connect to peers that are closer to them and avoid being overloaded.

The definition of connection preference is revised as:

$$\begin{aligned} P_i(j) \quad = \quad & r_i^{-\ln(r_i)} \cdot \frac{f_i(j) - r_i}{\sum_{k \in LC_i} f_i(j) - r_i} + \\ & (1 - r_i^{-\ln(r_i)}) \cdot \frac{\frac{1}{d_i(j)} - (1 - r_i)}{\sum_{k \in LC_i} \frac{1}{d_i(k)} - (1 - r_i)} \end{aligned} \tag{5}$$

We considered two approaches when calculating the resource level value of a peer. The first one is to use some statistical information like the one presented in Saroiu *et al.* [15], which measured the bandwidth distribution for Gnutella P2P networks. The second approach is the one we actually adopted. In this approach, the resource level r_i of peer i is approximated by counting the proportion of peers that have less or equal capacities than peer i in its neighbor candidate list LC_i. Although this approximation may introduce estimation errors, it avoids the reliance on the statistical information that may become outdated as the network technologies evolve.

After peer i selects its neighbor list N_i and sets up its outgoing edges, it sends a backward connection request to each peer $k \in N_i$ in the following format.

$$M_{back\_req} = \langle source = i,\ type = back\_req,\ TTL = 0,\ hops = 0,\ prop = c_i - \sigma|N_i|\ \rangle$$

The request is piggybacked with the capacity c_i of peer i and the size of its neighbor list N_i. Node k sets up a back link when $c_k - \sigma|N_k| \leq c_i - \sigma|N_i|$. Otherwise, with probability p_b, a backward connection is setup. The value of p_b controls the ratio between the number of outgoing links and the number of incoming links of each peer. In our implementation, we set it to 0.5. Parameter σ maps the average workload for handling one out-going link to the unit of capacity. Its value is decided by the specific ap-

plication our overlay network supports. In our simulation, we set it to 1.

3.2 Overlay Optimization and Maintenance

Our overlay construction algorithm builds efficient unstructured P2P networks from start point and guides each new peer to select its neighbors based on its capacity and network proximity information. Once the overlay is constructed, peers in our system behave like the ones in normal unstructured P2P network such as Gnutella [3]. Due to the limit of space, we skip the details of our overlay optimization and maintenance algorithms. It should be pointed out that the simplicity of our unstructured overlay network gives us enough design space to accommodate various optimization techniques like the ones used in [8, 18].

4 Experimental Evaluation

We have implemented a discrete event simulation to evaluate the mechanisms presented in this paper. To simulate the IP networks, we used the Transit-Stub graph model from the GT-ITM topology generator [20] to generate 10 network topologies. Each topology consists of 5050 routers. Links are assigned latency values following a uniform distribution on different ranges according to their types: U(15ms, 25ms) for intra-transit domain links, U(3ms, 7ms) for transit-stub links, and U(1ms, 3ms) for intra-stub links. Peers are randomly attached to the stub domain routers and organized into overlay networks using the algorithm given in Section 3. Capacities of peers are generated using the distribution given in Table 1, which summarized the measurement results reported in [15].

Capacity level	Percentage of peers
1×	20%
10×	45%
100×	30%
1000×	4.9%
10000×	0.1%

Table 1. Peer Capacity Distribution

4.1 Topology Evaluation

We simulate the construction of overlays from ground-up. Peers join our overlay with intervals following an exponential distribution Expo(1s). They use the distributed algorithm given in Section 3 to choose their neighbors. Figure 1 plots the log-log degree distribution of an overlay network of 5×10^3 peers. It shows a clear power law distribution.

As we discussed in Section 1, applications like application layer multicast demands proximity-awareness of overlay networks. We compared the overlay networks constructed using our algorithm with the ones randomly generated using a centralized algorithm proposed in [7]. We simulated the joining process of 1×10^3 peers. We compared the average network distance of each peer to its neighbors. The average we measured in our proximity-aware overlay network is 62.75ms. The average in random power-law network is 339.58ms.

4.2 Improvement of Application Performance

We use an application layer multicast system as an example to show how a proximity-aware overlay network can improve the performance of applications. Application layer multicast has been proposed as an alternative for wide-area multicast services. In this approach, end-hosts form multicast groups and implement multicast functionalities. Multicast data are replicated on end-hosts and propagated over the unicast edges connecting the multicast group members. Compared to IP multicast, application layer multicast systems are less efficient because they may send data multiple times over the same IP network link. Moreover, end-hosts usually have different capacities and constraints such as network connectivity, forwarding capacities, and availabilities. The workload distribution among those heterogeneous end-hosts consequently affects the overall system performance.

The application layer multicast protocol that we used is an implementation of the truncated reverse path broadcasting algorithm. Its functionality is similar to the DVMRP [10] IP-multicast protocol. We use overlay networks and end-hosts to implement the polling and pruning processes of multicast group management, instead of using the IP network devices such as routers. Because of the limits of space, we skip the details of the protocol.

We simulated multicast groups consisting of 1×10^3 to 1×10^4 peers. We used the routing weights generated by the GT-ITM package to simulate the IP unicast routing. IP multicast systems are simulated by merging the unicast routes into shortest path trees. We measured two metrics that are usually used to evaluate application layer multicast systems. $Relative\ Delay\ Penalty$ is defined as the ratio between the average application layer multicast delay and the average IP multicast delay. $Link\ Stress$ is the ratio between the number of IP messages generated by an application layer multicast tree and the number of IP messages generate by the equivalents IP multicast tree.

From Figure 2 $\sim$ Figure 3, we can see that application layer multicast systems show significant improvement in both metrics when they are implemented over our overlay networks: the relative delay penalty is close to 1, which is its theoretical upper bound; the link stress is about 2/3 of the ones over random power law topologies. We attribute such improvements to the fact that our algorithm successfully incorporates network proximity information into the overlay topologies. Multicast payloads are forwarded along much shorter path, (recall the result in Sec-

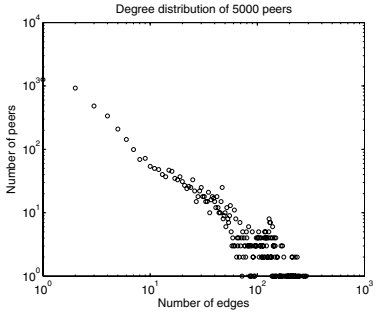

Figure 1. log-log degree distribution of proximity-aware overlay network

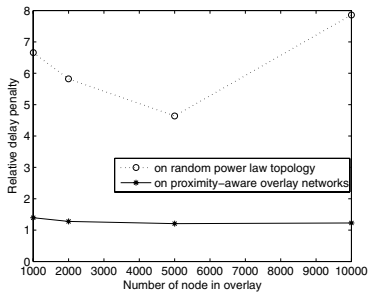

Figure 2. Relative delay penalty of application layer multicast application

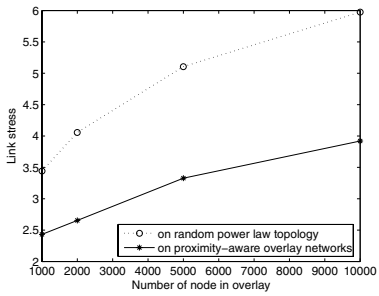

Figure 3. Link stress of application layer multicast application

tion 4.1), and thus incur less IP traffic in the underlying IP network.

5 Conclusion

Research works such as RON [6] have been designed to build generic overlays independent of the applications using them. Optimization techniques such as [8] can be used to improve the performance of the overlay networks at the application level. Our system is different from those works in a few aspects. First, our system distinguishes the distance of peers and constructs overlay networks that incorporate network proximity information. Second, our algorithm builds "scale-free" power law topologies and assigns peers with different number of P2P connections according to their capacities. Compared to structured P2P systems [14, 13, 16] and their optimizations [19, 21], our system is more resilient to network dynamic and is easier to implement. Our algorithm is fully distributed and based on only local information. It makes few assumptions on the underlying network as well as the peer activity pattern.

References

[1] E. Ng. GNP software 2003. http://www-2.cs.cmu.edu/ eugeneng/research/gnp/software.html.

[2] Gnucleus. The Gnutella web caching system. http://gnucleus.sourceforge.net.

[3] The Gnutella RFC. http://rfc-gnutella.sourceforge.net.

[4] Sharman networks LTD. KaZaA media desktop. http://www.kazaa.com.

[5] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *32nd Symposium on Theory of Computing*. ACM, 2000.

[6] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resillient overlay networks. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP)*, 2001.

[7] T. Bu and D. Towsley. On distinguishing between internet power law topology generators. In *IEEE INFOCOM*, New York, NY, June 2002. IEEE.

[8] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P systems scalable. In *ACM SIGCOMM*, Karlsruhe, Germany, August 2003. ACM.

[9] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *ACM SIGCOMM*, Portland, Oregon, USA, August 2004. ACM.

[10] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, 8(2), May 1990.

[11] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM*, pages 251–262. ACM, 1999.

[12] G. Pandurangan, P. Raghavan, and E. Upfal. Building low-diameter p2p networks. In *Proceedings of the 42nd Annual IEEE Symposium on the Foundations of Computer Science*, 2001.

[13] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proceedings of SIGCOMM*. ACM, 2001.

[14] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329–??, 2001.

[15] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of Peer-to-Peer file sharing systems. In *Proceedings of MMCN*, San Jose, CA, August 2002.

[16] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings SIGCOMM*, pages 149–160. ACM, 2001.

[17] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[18] R. H. Wouhaybi and A. T. Campbell. Phenix: Supporting resilient low-diameter peer-to-peer topologies. In *IEEE INFOCOM*, Hong Kong, China, March 2004. IEEE.

[19] Z. Xu, M. Mahalingam, and M. Karlsson. Turning heterogeneity into an advantage in overlay routing. In *Proceedings of INFOCOM*, 2003.

[20] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an internetwork. In *IEEE Infocom*, volume 2, pages 594–602. IEEE, March 1996.

[21] B. Y. Zhao, Y. Duan, L. Huang, A. D. Joseph, and J. D. Kubiatowicz. Brocade: Landmark routing on overlay networks. In *1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*.