Efficient Peer-to-Peer Overlay Construction

Lionel M. Ni and Yunhao Liu

Department of Computer Science
The Hong Kong University of Science and Technology
Kowloon, Hong Kong, China SAR
ni@cs.ust.hk

Abstract*

In unstructured peer-to-peer (P2P) systems, the mechanism of a peer randomly joining and leaving a P2P network causes topology mismatch between the P2P logical overlay network and the physical underlying network, causing a large volume of redundant traffic in the Internet. In order to alleviate the mismatching problem, we introduce several distributed algorithms to optimize the overlay, while retaining the search scope. Our simulation study shows that this approach can effectively solve the mismatch problem and significantly reduce P2P traffic and response time.

1. Introduction

Popularized by Napster, peer-to-peer (P2P) has been a well known term recently. Large amount of files are shared in these P2P systems, such as Gnutella, KaZaA, BitTorrent, with most of the contents being provided by the P2P users themselves. Today, millions of users (peers) join the network by connecting to some of active peers in the P2P overlay network.

There are mainly three different architectures for P2P systems: centralized, decentralized structured, and decentralized unstructured [15]. In centralized model, such as Napster [5], central index servers are used to maintain a directory of shared files stored on peers so that a peer can search for the whereabouts of a desired content from an index server. However, this architecture creates a single point of failure, and its centralized nature of the service also makes systems vulnerable to denial of service attacks [8]. Decentralized P2P systems have the advantages of eliminating reliance on central servers and providing greater freedom for participating users to exchange information and services directly between each other. In decentralized structured models, such as Chord [23], Pastry [20], Tapestry [26], and CAN [17], the shared data placement and topology characteristics of the network are tightly controlled based on distributed hash functions. Although there are many discussions on this model, decentralized structured P2P systems are not practically in use in the Internet.

We focus on decentralized unstructured P2P systems, such as Gnutella [2] and KaZaA [4]. File placement is random in these systems, which has no correlation with the network topology [25]. Unstructured P2P systems are most commonly used in today's Internet. The most popular search mechanism in use is to blindly "flood" a query to the network among peers (such as in Gnutella) or among supernodes (such as in KaZaA). A query is broadcast and rebroadcast until a certain criterion is satisfied. If a peer receiving the query can provide the requested object, a response message will be sent back to the source peer along the inverse of the query path, and the query will not be further forwarded from this responding peer. This mechanism ensures that the query will be "flooded" to as many peers as possible within a short period of time in a P2P overlay network. A query message will also be dropped if the query message has visited the peer before.

Studies in [22] and [21] have shown that P2P traffic contributes the largest portion of the Internet traffic based on their measurements on some popular P2P systems, such as FastTrack (including KaZaA and Grokster) [1], Gnutella, and DirectConnect. Measurements in [18] have shown that even given that 95% of any two nodes are less than 7 hops away and the message time-to-live (TTL=7) is preponderantly used, the flooding-based routing algorithm generates 330 TB/month in a Gnutella network with only 50,000 nodes. A large portion of the heavy P2P traffic caused by inefficient overlay topology and the blind flooding is unnecessary, which makes the unstructured P2P systems being far from scalable [19]. One of the important reasons for this problem is that, the mechanism of a peer randomly choosing logical neighbors without any

* This research is supported by the Hong Kong RGC Grant HKUST6264/04E.

knowledge about the underlying physical topology causes topology mismatch between the P2P logical overlay network and physical underlying network.

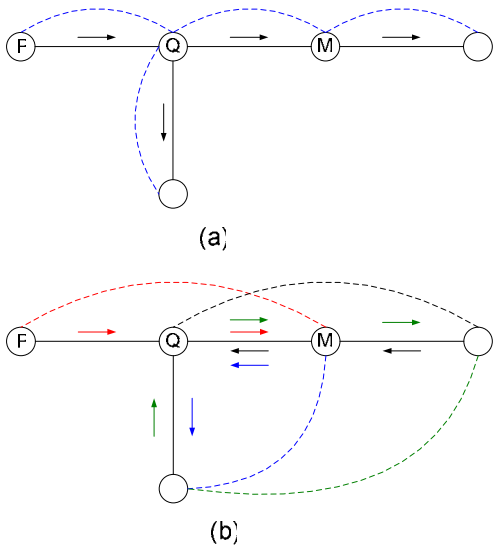

Figure 1: Topology mismatch problem

In a P2P system, all participating peers form a P2P network over a physical network. A P2P network is an abstract, logical network called an overlay network. Maintaining and searching operations of a Gnutella peer are specifically described in [3]. When a new peer wants to join a P2P network, a bootstrapping node provides the IP addresses of a list of existing peers in the P2P network. The new peer then tries to connect with these peers. If some attempts succeed, the connected peers will be the new peer's neighbors. Once this peer connects into a P2P network, the new peer will periodically *ping* the network connections and obtain the IP addresses of some other peers in the network. These IP addresses are cached by this new peer. When a peer leaves the P2P network and then wants to join the P2P network again (no longer the first time), the peer will try to connect to the peers whose IP addresses have already been cached. This mechanism, although is simple to implement, causes mismatch problem. Let us see an example shown in Figure 1, where solid lines represent physical links and dotted lines represent logical links.

When peer *P* inserts a query message to the network, in the case of an efficient P2P overlay as shown in Figure 1(a), there is no message duplications. However, when topology mismatch problem occurs, as shown in Figure 1(b), a message from peer *P* will incur many unnecessary message duplications, especially if the physical distance between peer *Q* and *M* are very far, both the traffic cost and the query response time will be increased significantly.

Aiming at alleviating the mismatch problem, reducing the unnecessary traffic, many methods have been introduced. In this paper, we will firstly overview some traditional approaches and discuss the reason why these approaches do not work, and then introduce several recently proposed algorithms.

2. Traditional Approaches

There are several traditional topology optimization approaches. End system multicast, Narada, is proposed in [7], which first constructs a rich connected graph on which to further construct shortest path spanning trees. Each tree rooted at the corresponding source using well-known routing algorithms. This approach introduces large overhead of forming the graph and trees in a large scope, and does not consider the dynamic joining and leaving characteristics of peers. The overhead of Narada is proportional to the multicast group size. This approach is infeasible to large-scale P2P systems.

Researchers have also considered to cluster close peers based on their IP addresses (e.g., [9, 16]). We believe there are two limitations for this approach. First, the mapping accuracy is not guaranteed by this approach. Second, this approach may affect the searching scope in P2P networks.

Recently, researchers in [24] have proposed to measure the latency between each peer to multiple stable Internet servers called ``landmarks". The measured latency is used to determine the distance between peers. This measurement is conducted in a global P2P domain and needs the support of additional landmarks. Similarly, this approach also affects the search scope in P2P systems.

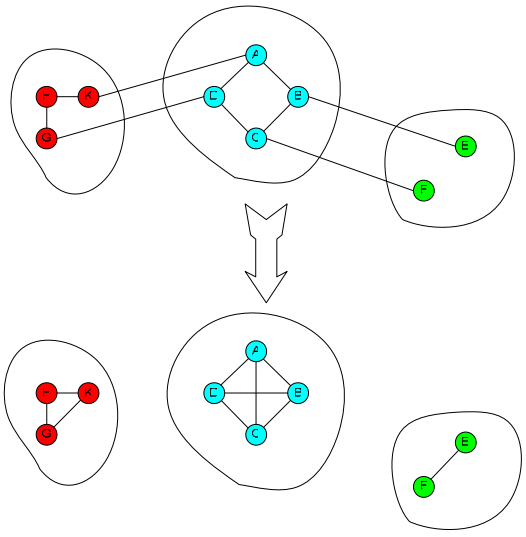

Figure 2: before optimization, queries can visit all of the peer, while after optimization, all queries can only visit a small group of live peers.

Using an example shown in Figure 2, we explain why these proximity based approaches will shrink query search scopes. In Figure 2, peers *A*, *B*, *C*, *D* locate in the same AS, peers *E*, *F* and *H*, *G*, *K* belong to other ASs, respectively . It is safe to assume that the physical distance between *A* and *B* or *E* and *F* are much smaller than that of *K* and *A* or *C* and *F*, as illustrated in Figure 2. Using above discussed approaches, when peers successfully obtain or estimate the distance between each pair of them, and optimization policy for each node is to connect the closest peers while retaining the original number of logical neighbors, a connected graph may be broken into three components. As a result, before optimization, queries can visit all of the peer, while after optimization, all queries can only visit a small group of live peers in the system, and the search scope of queries is significantly reduced.

3. Distributed Approaches to Topology Mismatch Problem

We have proposed several approaches to solve overlay topology mismatch problems in P2P systems [10-14]. They are scalable and completely distributed in the sense that they do not require global knowledge of the whole overlay network when each node is optimizing the organization of its logical neighbors. For example, in Adaptive Connection Establishment (ACE) [14], every single peer builds an overlay multicast tree between itself (source node) and the peers within a certain diameter from the source peer, and then optimizes the neighbor connections that are not on the tree, while retaining the search scope, as illustrated in Figure 3. Our simulations show that ACE can significant improve the performance of P2P systems. We also show that a larger diameter leads to a better topology optimization rate and a higher overhead due to extra information exchanging.

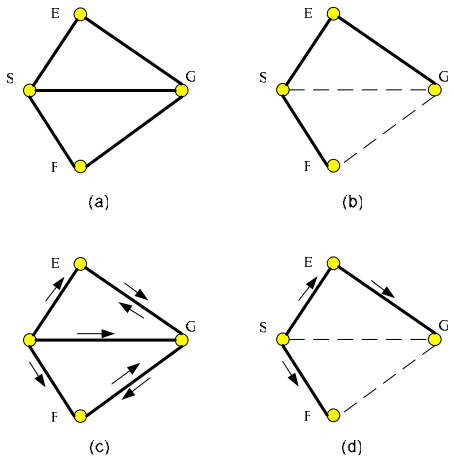

Figure 3: ACE

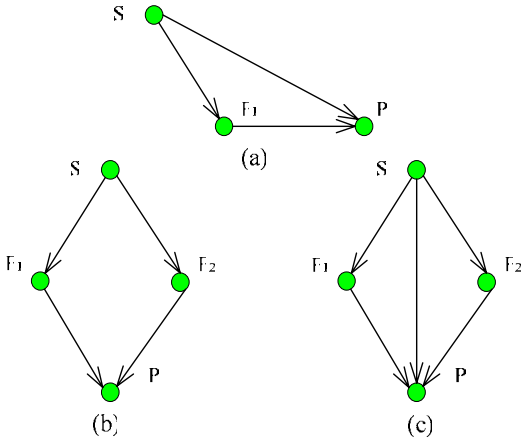

Figure 4: TTL2 detector of LTM

To improve the convergence speed of ACE, we propose a location-aware topology matching (LTM) scheme [10]. In LTM, each peer issues a detector in a small region so that the peers receiving the detector can record relative delay information, as shown in Figure 4. Based on the delay information, a receiver can detect and cut most of the inefficient and redundant logical links, as well as add closer nodes as its direct neighbors. Our simulation studies show that the total traffic and response time of the queries can be significantly reduced by LTM without shrinking the search scope. Our study shows that only one tenth of the original traffic cost is necessary to cover the same number of peers, and the average response time is reduced by approximately 80%.

To further remove traffic overhead, we propose a scalable bipartite overlay (SBO)[11]. The SBO which employs an efficient strategy for distributing optimization tasks in peers with different colors, as shown in Figure 5. In SBO, each joining peer is assigned a color so that all peers are divided into two groups of white or red colors, respectively. Each peer is only connected with peers in its opposite color. Each white peer probes neighbor distances and reports the information to the red neighbors. Each red peer computes efficient forwarding paths. A white peer that is not on forwarding paths of a red peer then tries to find a more efficient red peer to replace this red neighbor. Our evaluations show that SBO achieves approximately 85% reduction on traffic cost and about 60% reduction on query response time.

4. Summary

Peer-to-Peer (P2P) computing has emerged as a popular model aiming at further utilizing Internet information and resources, complementing the available client-server services. Without assuming any knowledge of

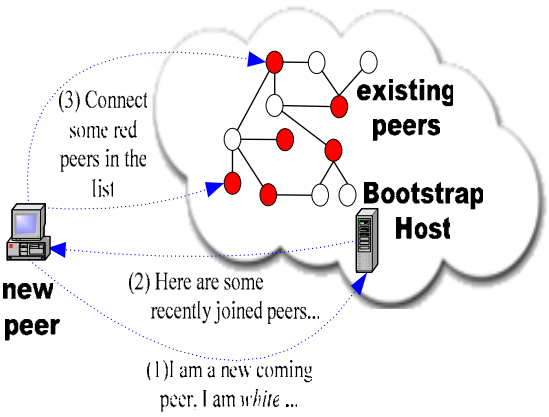

Figure 5: Bootstrapping a new peer in SBO

the underlying physical topology, the conventional P2P mechanisms are designed to randomly choose logical neighbors, which causes a serious topology mismatch problem between the P2P overlay network and the underlying physical network. This mismatch problem incurs a great stress in the Internet infrastructure and adversely restraints the performance gains from the various search or routing techniques. In order to alleviate the mismatch problem and reduce the unnecessary traffic and response time, three schemes are recently proposed, namely, Adaptive Connection Establishment (ACE), Scalable Bipartite overlay (SBO), and Location-aware Topology Matching (LTM) techniques. All of them achieve the above goals without bringing any noticeable extra overheads. Moreover, these techniques are scalable because the P2P overlay networks are constructed in a fully distributed manner where global knowledge of the network is not necessary. ACE is the simplest one, but the convergent speed is relatively slow. SBO, with the same overhead, has a better performance than ACE. The convergent speed of LTM is the fastest, but it needs the support of NTP[6] to synchronize the peering nodes.

References

[1] Fasttrack, http://www.fasttrack.nu

[2] Gnutella, http://gnutella.wego.com/

[3] The Gnutella protocol specification 0.6, http://rfc-gnutella.sourceforge.net

[4] KaZaA, http://www.kazaa.com

[5] Napster, http://www.napster.com

[6] NTP: The Network Time Protocol, http://www.ntp.org/

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

[8] O. D. Gnawali, "A Keyword-Set search system for peer-to-peer networks," in *Master's thesis, Massachusetts Institute of Technology, June*, 2002.

[9] B. Krishnamurthy and J. Wang, "Topology Modeling via Cluster Graphs," *Proceedings of SIGCOMM Internet Measurement Workshop*, 2001.

[10] Y. Liu, X. Liu, L. Xiao, L. M. Ni, and X. Zhang, "Location-Aware Topology Matching in Unstructured P2P Systems," *Proceedings of IEEE INFOCOM*, 2004.

[11] Y. Liu, L. Xiao, and L. M. Ni, "Building a Scalable Bipartite P2P Overlay Network," *Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS)*, 2004.

[12] Y. Liu, L. Xiao, L. M. Ni, and Y. Liu, "Overlay Topology Matching in Unstructured P2P Systems," *Proceedings of the Second International Workshop on Grid and Cooperative Computing (GCC)*, 2003.

[13] Y. Liu, Z. Zhuang, L. Xiao, and L. M. Ni, "AOTO: Adaptive Overlay Topology Optimization in Unstructured P2P Systems," *Proceedings of IEEE GLOBECOM*, 2003.

[14] Y. Liu, Z. Zhuang, L. Xiao, and L. M. Ni, "A Distributed Approach to Solving Overlay Mismatch Problem," *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS)*, 2004.

[15] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-peer Networks," *Proceedings of the 16th ACM International Conference on Supercomputing*, 2002.

[16] V. N. Padmanabhan and L. Subramanian, "An Investigation of Geographic Mapping Techniques for Internet Hosts," *Proceedings of ACM SIGCOMM*, 2001.

[17] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-addressable Network," *Proceedings of ACM SIGCOMM*, 2001.

[18] M. Ripeanu, A. Iamnitchi, and I. Foster, "Mapping the Gnutella Network," *IEEE Internet Computing*, 2002.

[19] Ritter, Why Gnutella Can't Scale. No, Really, http://www.tch.org/gnutella.html

[20] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *Proceedings of International Conference on Distributed Systems Platforms*, 2001.

[21] S. Saroiu, K. P.Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An Analysis of Internet Content Delivery Systems," *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, 2002.

[22] S. Sen and J. Wang, "Analyzing Peer-to-peer Traffic Across Large Networks," *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, 2002.

[23] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," *Proceedings of ACM SIGCOMM*, 2001.

[24] Z. Xu, C. Tang, and Z. Zhang, "Building Topology-aware Overlays Using Global Soft-state," *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, 2003.

[25] B. Yang and H. Garcia-Molina, "Efficient Search in Peer-to-peer Networks," *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS)*, 2002.

[26] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-resilient wide-area location and routing," *Technical Report UCB//CSD-01-1141, U.C.Berkeley*, 2001.