ISP-Friendly Peer Matching Algorithms

Cheng-Hsin Hsu, Nitin Chiluka and Mohamed Hefeeda
School of Computing Science, Simon Fraser University, Surrey, BC, Canada

ABSTRACT

In peer-to-peer (P2P) systems, a receiving peer needs to be matched with multiple sending peers, because peers have limited capacity and reliability. We study the following peer-matching problem: given a set of potential senders for an object requested by a receiver, find the subset of senders that will minimize the load on the backbone network and achieve the best performance for the receiver. We propose two new algorithms to solve this problem: ISP-friendly and network-based. Our ISP-friendly algorithm leverages information about the Autonomous System (AS) graph and different types of relationship among ASes, and finds the closest senders in terms of AS proximity, i.e., the number of AS hops that the data from senders have to cross to reach the receiver. Our network-based algorithm finds the closest senders in terms of network proximity, i.e., number of IP hops. Our initial results indicate that a significant reduction in the number of ASes that P2P traffic has to traverse from senders to receivers is achievable.

Categories and Subject Descriptors: C.2[Computer Systems Organization]: Computer-Communication Networks

General Terms: Design

1. OVERVIEW

Peer-to-peer (P2P) file-sharing systems, such as BitTorrent and Gnutella, have attracted millions of users and generate tera bytes of Internet traffic every day. P2P video streaming (live and on-demand) has also seen wide deployment. In addition, there have been several proposals for designing peer-assisted content distribution networks (CDNs). In all of these systems, a receiving peer needs to be *matched* with multiple sending peers, because peers have limited capacity and reliability. Efficient peer matching is critical to the performance of P2P systems, because it could shorten download times in file-sharing systems and enhance video quality in streaming systems. Most importantly, efficient peer matching is a major factor in reducing the cost of carrying the P2P traffic across various Internet Service Providers (ISPs). For example, previous works indicate that the random peer matching algorithm used in current BitTorrent may unnecessarily increase the bandwidth consumption on inter-ISP links. Inter-ISP links are costly for ISP operators, and they are more susceptible to congestion. Increasing the cost on ISPs may (actually does, in many cases) lead them to react adversely by shaping the P2P traffic or even blocking P2P traffic. These adverse reactions will degrade the performance of P2P file-sharing and streaming systems, and will make it harder for peer-assisted CDNs to provide high-quality, commercially viable, distribution services. Therefore, it is in the best interest of both ISPs and P2P systems to employ efficient, ISP-friendly, peer matching algorithms.

We study the following peer-matching problem: given a set of potential senders for an object requested by a receiver, find the subset of senders that will minimize the load on the backbone network and achieve the best performance for the receiver. This problem is fairly general and algorithms for solving it can be used in many P2P systems. For example, in BitTorrent-like networks, a tracker can use our algorithms to return matching senders for a requesting client. In peer-assisted CDNs, where peers help a few dedicated servers in distributing content, servers performing sender-receiver matching can benefit from our algorithms. There are several approaches for solving the peer matching problem. The simplest and widely used one is to randomly match senders with receivers. We propose two new algorithms: ISP-friendly and network-based, which are briefly described in the following.

ISP-Friendly Peer Matching. This algorithm finds the closest senders to a given receiver in terms of Autonomous System (AS) proximity, i.e., the number of AS hops that the data from senders have to cross to reach the receiver. The basic idea is to leverage information about the AS graph and different types of relationship among ASes. For example, previous works, e.g., [1], infer fairly accurately the relationships between ASes, which could be customer-to-provider (c2p), peer-to-peer (p2p), provider-to-customer (p2c), and sibling-to-sibling (s2s). Using this relationship information, the AS path that between any two end points can be inferred [2]. The inference algorithm in [2] returns the shortest AS path that conforms to the known practices of BGP policy routing.

Our peer matching algorithm builds *offline* a two-dimensional matrix that contains shortest paths among all ASes. Then, it uses this matrix online to match peers. Since there are currently more than 25,000 ASes in the Internet, this matrix will be very large. We employ several optimizations to compress this matrix without compromising the accuracy. For example, it is known that many customer ASes have only one (c2p) link to a provider AS, i.e., they are degree-1 ASes . Traffic to/from such ASes does not have options for the last/first AS hop. Thus, degree-1 ASes can easily be combined with their AS providers, because the shortest AS paths to/from them are the same as the shortest AS paths to their providers (plus one hop). This optimization reduces the matrix size without losing any information and without increasing the running

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

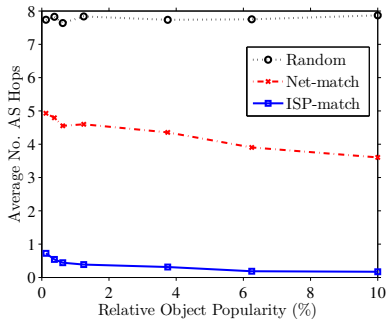

Figure 1: Sample results: Comparison among our proposed algorithms and the currently used random algorithm.

time of the online peer matching process.

To illustrate, we collected AS data from CAIDA for ASes in North America (US, Canada, and Mexico). We found that there are 9,031 in North America, of which there are 3,233 degree-1 ASes (35%). Given that the AS paths are fairly stable [2], the cost of running our algorithm in the background is not significant. Nevertheless, we are currently developing more optimizations including: (i) merging multi-homed ASes with their providers, and (ii) incrementally updating the shortest paths matrix when there are some changes in AS paths, instead of recomputing all of its entries. Our initial analysis for merging multi-homed ASes indicate that the matrix size can be dramatically decreased; only 1,144 ASes are left after applying this optimization on ASes in North America, and the computation of the matrix only took a few minutes. The main goal of this poster is to show that peer matching based on AS relationship is beneficial.

Network-based Peer Matching. This algorithm finds the closest senders to a given receiver, in terms of network proximity, i.e., number of IP hops. We achieve this goal by choosing senders whose IP addresses share the longest prefix with the IP address of the receiver. The rationale is that the longer the shared IP prefix between senders and receivers, the more likely that they are within the same network. We perform longest prefix matching using a binary trie data structure, somewhat similar to the ones used for IP lookup in core routers. The main advantage of the peer matching based on their IP addresses is that it is almost *free*: it does not rely on any external source of information. However, it is not always the case that IPs with shared prefixes are near to each other in the network sense, especially if the shared prefixes are small.

2. INITIAL RESULTS

We have implemented the proposed peer matching algorithms along with current random algorithm in Java for preliminary performance evaluation. We consider the average number of AS hops from each receiver to its senders as our main performance metric. Smaller number of AS hops reduces the traffic amount on inter-ISP links, which are more susceptible to network congestion that leads to poor application performance. Smaller number of AS hops also reduces the operational costs on ISPs, since the majority of these inter-ISP links are expensive customer-to-provider links.

We generate the AS graph for North America using data from CAIDA and described in [1]. Then, we compute the shortest AS paths. We create more than 120,000 peers and distribute them over the ASes. The number of peers assigned to an AS is proportional to its size. We estimate the AS size using IP ranges that belong to each AS, which we obtain from the AT&T AS traceroute project. Peers are assigned IPs from real IP ranges. We simulate multi-sender download sessions for files of different popularities. The popularity of a file is adjusted by controlling the number of peers that have that file. For a given file, we vary the number of peers storing it from less than 0.1% (rare file) to 10% (fairly popular file). For each download session, a receiver is randomly selected. Then n number of senders are chosen by one of the peer matching algorithms. We then compute the number of AS hops traversed by the traffic from the chosen senders to the receiver. We vary n between 2 and 20 to cover wide ranges of P2P systems, and for each value of n we repeat the experiment 1,000 times and report the average values.

A sample of our results is shown in Fig. 1. The figure clearly shows that the currently used random algorithm imposes high load on the Internet, because the average number of ASes traversed is almost 8. Our simple network-based matching algorithm reduces the average number of hops to less than 4.5. This is a significant performance gain, given that it comes at very little cost as this algorithm uses just the IP addresses of peers. Finally, our ISP-friendly peer matching algorithm dramatically decreases the average number of AS hops to less than one. This is because it returns peers within the receiver's AS or very close ASes in most download sessions. Therefore, it significantly reduces the load on the expensive inter-ISP links. In addition, close-by peers usually have short and less variable round trip delays and thus are expected to yield better application-level performance for the P2P systems using our matching algorithms.

We should mention that the number of AS hops is a coarse performance metric, because ASes vary significantly in size. However, keeping the traffic within the same AS, even if it is large, is desirable as links within ASes are usually over provisioned, and they tend to cost less than inter-ISP links.

3. CONCLUSIONS AND FUTURE WORK

We presented two algorithms to solve the problem of matching senders with receivers in P2P content distribution networks in order to reduce the load on ISPs and to enhance the performance of P2P applications. Our initial results indicate that a significant reduction in the number of ASes that P2P traffic has to traverse from senders to receivers is achievable. We are currently pursuing several directions to optimize the performance of our peer matching algorithms and to rigorously evaluate their performance. For example, we are refining the results returned by the ISP-friendly peer matching algorithm based on the IP prefixes of peers. Comparing IP prefixes within the same AS could be more accurate than comparing them in different ASes. In addition to the small AS hops, this refinement could reduce the number of IP hops, and therefore could reduce the load on the internal links within the same AS as well. We are also designing a more efficient AS-level path inference algorithm that supports incremental updates. The new algorithm will further reduce computational complexity of AS path inference. Finally, we are implementing our peer matching algorithms in a real peer-assisted content distribution network, called pCDN. pCDN is being developed by our group for the Canadian Broadcasting Corporation (CBC), which is the largest Internet content provider in Canada with millions of online users. pCDN 1.0 is currently being tested on CBC network for the release to the public in the near future. More information on pCDN and the peer matching algorithms can be found at [3].

4. REFERENCES

[1] X. Dimitropoulos and et al. AS relationships: Inference and validation. *ACM SIGCOMM Computer Communication Review (CCR'06)*, 37(1):29–40, January 2007.

[2] Z. Mao and et al. On AS-level path inference. In *Proc. of ACM SIGMETRICS (SIGMETRICS'05)*, pages 38–49, Banff, Canada, June 2005.

[3] Network Systems Lab Home Page. http://nsl.cs.sfu.ca/wiki.