Traceroute-Based Fast Peer Selection without Offline Database

Lihang Ying and Anup Basu
Department of Computing Science, Univ. of Alberta, Edmonton, Alberta, Canada
E-mail:{lihang,anup}@ualberta.ca

Abstract

The extreme heterogeneity in the P2P Internet environment causes the connections to candidate peers to vary significantly. Thus, selecting good candidate peers is critical to P2P networking performance. While most research in the literature focus on selecting peers with low latency, high uploading bandwidth, and high serving stability by time-consuming end-to-end measurement, we address how to quickly narrow the scope of good candidate peers. In this paper, we proposed to pick out the close peers from the same location or with low round-trip time from each other, by observing that close peers share routers along the traceroute path to the same destination. Our method only maintains online peers's traceroute records, without any offline database. Experiments with real traceroute records from 352 different IP addresses around the world verify the efficiency of our method.

1. Introduction

Peer-to-Peer (P2P) networking is widely used in applications of file sharing and instant messaging. In P2P, peers communicate directly and share resources with one other. P2P can overcome the server-side bottleneck problem in the centralized client/server architecture. Another advantage of P2P is low cost. P2P is an application-layer solution, which does not need upgrade to an existing network. Besides, P2P utilizes the resources of peers, which greatly reduces the requirements on the capability of a server, and in fact can make servers unnecessary. P2P technologies are being adopted in more and more applications, such as multicasting and online databases.

With the wide adoption of high-speed Internet, multimedia streaming is becoming more and more popular. But one connection of multimedia streaming consumes much more bandwidth than traditional text-based messaging. The server-side bottleneck of traditional client/server solutions constrains the scalability. P2P architectures mitigate the problems in this situation. In recent years, substantial research on application-layer multicasting and streaming with P2P has been conducted. There is also some research on P2P on-demand streaming. But most of these research are on architecture and system design. Few detailed, quantitative and optimal research are oriented towards P2P VOD, such as retrieval strategy, speeding-up initial buffering and packet assignments.

In P2P networks, usually a peer first discovers a list of candidate server peers, then selects an active set from the list, and retrieves data from peers in the set. Due to the extreme heterogeneity in the P2P Internet environment, the connections to candidate server peers vary greatly. As a result, selecting good candidate server peers is critical to P2P networking performance. The selection criteria include low-latency, high uploading bandwidth, and serving stability. Quickly picking good candidate server peers is also important.

Peer selection could be based on random [1], end-to-end measurement [2] [3] [4], and topology [5] [6]. By conducting trace-based analyses, [2] investigated end-to-end measurement-based techniques including round-trip time (RTT) probing, 10KB TCP probing, and bottleneck bandwidth probing. This work reveals that the basic techniques can achieve 40-50% performance improvement, and the basic techniques are limited by relying on eliminating the low-performance peers rather than reliably identifying the best-performing ones. The basic techniques can effectively select peers for adaptive applications, which can update active peers. Combining the basic techniques can better identify a high-performance peer, thus even applications that cannot adapt may benefit.

[3] proposed the methodology of peer selection by machine learning. A decision tree is used to rate peers based on combinations of collected connection information, such as load, bandwidth, and past uploading experience. Then a policy derived by Markov Decision Process is executed to select peers. [5] proposed topology-aware peer selection. It infers an

Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'06)
0-7695-2746-9/06 $20.00 © 2006 IEEE

approximate topology and considers shared network path when selecting peers.

Besides peer selection purely based on networking performance, [7] conducted research on optimal peer selection in a P2P resource economy, where the server peers charge the client for downloading. For downloading, the optimal selection is to minimize the download delay subject to a cost budget constraint. For streaming, the optimal selection is to minimize cost subject to continuous playback.

Quickly starting to play after a user clicks to demand a video, i.e., shortening initial buffering time, is one critical factor for QoS based VOD. However, the P2P architecture has drawbacks that are not good for shortening initial buffering time. A peer needs time to pick out good peers from the candidate list. End-to-end measurement approaches select peers by time-consuming probing. We try to narrow the scope of candidate peers close to the client (In this paper, a peer is any client participating in a P2P network. The client is also a peer; however, it is running on the local machine. Readers may choose to think of themselves as the client which connects to numerous peers.) without time-consuming probing. Thus, the chance of picking out good peers from the candidate list increases. (When we say "closeness" of two peers, there are two meanings: first, the two peers are close to each other physically; second, the link between the two peers has low RTT or high bandwidth.) Differing from topology-based methods, we try to pick the close peers without inferring topology. The main idea of our work is to select peers sharing the same routers along the path of traceroute to the same destination.

The remainder of this paper is organized as follows: Section 2 outlines the motivations of our work. Section 3 describes the details of traceroute-based peer selection. Experiments and analysis are given in Section 4. The work is concluded in Section 5.

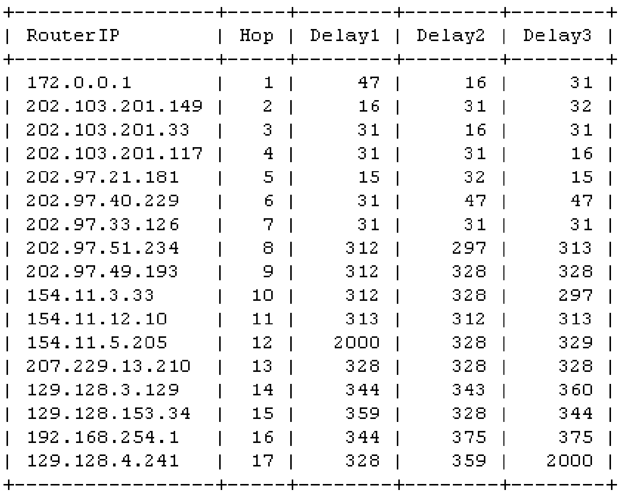
2. Motivation

Peer-to-peer live streaming has gained great success in China, such as Coolstreaming [8], PPLive [9], and PPStream [10]. PPLive claims to successfully support live broadcasting with 500,000 concurrent users. On the other hand, the Internet infrastructure environment in China is much more complex than in developed countries. The quality of links within China varies a lot. Furthermore, the links between different ISP are much slower than the links within the same ISP. Intuitively, the links within a given city and the same ISP are preferred to get good links and reduce the traffic of backbone network. Therefore, to improve P2P applications in the similar networking environment as China, it is necessary to distinguish peers from different ISPs and different locations.

There are several approaches to distinguish peers from different ISP and different locations [11]. First, we could utilize a table of IP address designation. However, this table usually is too coarse and out-of-date. Second, IP location databases, which could be based on whois database and collects IP address locations (For convenience, we use IP to present IP address later on.) manually or through reports from users, could be used to find the locations of peers. However, the first drawback of this approach is that the coverage of such databases is limited. For example, the QQWry [12] database covers China well, but it has limited coverage outside China. IP2Location [13] covers US adequately, but it does not work well in China. Second, it is time-consuming to query such large databases because the databases include all IP subnets around the world. Another straightforward method to pick close peers is to match IP with same prefix. Although this simple scheme has already improved peer selection, it has obvious disadvantages. First, it would mistakenly pick peers from the different locations with the same IP prefix as the client. Second, the peers from the same city and the same ISP with different IP subnets to the client could not be picked out. This paper proposes to automatically and efficiently distinguish peers from different ISP and different locations.

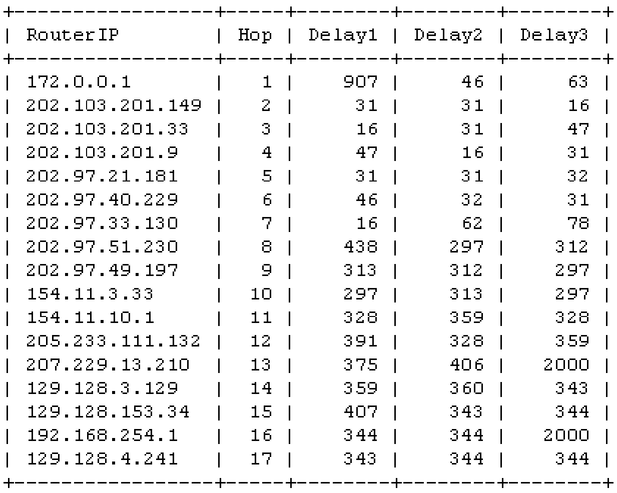
According to [15], "Traceroute [14] is widely used to detect and diagnose routing problems, characterize end-to-end paths through the Internet, and discover the underlying network topology. Traceroute identifies the interfaces on a forwarding path and reports round-trip time statistics for each hop along the way." Figure 1 is an example of traceroute from two different IPs (222.84.110.115, 219.159.216.235) of the same Internet Service Provider (China Telecom) in the same city (Liuzhou, China) to the same destination (IP:129.128.4.241, University of Alberta). In the figure, "RouterIP" means the router's IP along the path; "Hop" means the hop along the path from the source to a router; "Delay1", "Delay2", and "Delay3" mean the round-trip time between the source and a router by three tests. Although the two IPs belong to different IP subnets, they share many routers. By contrast, we can determine that the two IPs are close according to the sharing of routers, which is based on the observation that the chance that all of the routers along the path are changed is low, although there are multiple paths between two nodes.

In the next section, details of how to utilize traceroute to distinguish peers from different locations and different ISP will be discussed.



```
+-----------------+-----+--------+--------+--------+
| RouterIP        | Hop | Delay1 | Delay2 | Delay3 |
+-----------------+-----+--------+--------+--------+
| 172.0.0.1       |   1 |     47 |     16 |     31 |
| 202.103.201.149 |   2 |     16 |     31 |     32 |
| 202.103.201.33  |   3 |     31 |     16 |     31 |
| 202.103.201.117 |   4 |     31 |     31 |     16 |
| 202.97.21.181   |   5 |     15 |     32 |     15 |
| 202.97.40.229   |   6 |     31 |     47 |     47 |
| 202.97.33.126   |   7 |     31 |     31 |     31 |
| 202.97.51.234   |   8 |    312 |    297 |    313 |
| 202.97.49.193   |   9 |    312 |    328 |    328 |
| 154.11.3.33     |  10 |    312 |    328 |    297 |
| 154.11.12.10    |  11 |    313 |    312 |    313 |
| 154.11.5.205    |  12 |   2000 |    328 |    329 |
| 207.229.13.210  |  13 |    328 |    328 |    328 |
| 129.128.3.129   |  14 |    344 |    343 |    360 |
| 129.128.153.34  |  15 |    359 |    328 |    344 |
| 192.168.254.1   |  16 |    344 |    375 |    375 |
| 129.128.4.241   |  17 |    328 |    359 |   2000 |
+-----------------+-----+--------+--------+--------+
```

(a)



```
+-----------------+-----+--------+--------+--------+
| RouterIP        | Hop | Delay1 | Delay2 | Delay3 |
+-----------------+-----+--------+--------+--------+
| 172.0.0.1       |   1 |    907 |     46 |     63 |
| 202.103.201.149 |   2 |     31 |     31 |     16 |
| 202.103.201.33  |   3 |     16 |     31 |     47 |
| 202.103.201.9   |   4 |     47 |     16 |     31 |
| 202.97.21.181   |   5 |     31 |     31 |     32 |
| 202.97.40.229   |   6 |     46 |     32 |     31 |
| 202.97.33.130   |   7 |     16 |     62 |     78 |
| 202.97.51.230   |   8 |    438 |    297 |    312 |
| 202.97.49.197   |   9 |    313 |    312 |    297 |
| 154.11.3.33     |  10 |    297 |    313 |    297 |
| 154.11.10.1     |  11 |    328 |    359 |    328 |
| 205.233.111.132 |  12 |    391 |    328 |    359 |
| 207.229.13.210  |  13 |    375 |    406 |   2000 |
| 129.128.3.129   |  14 |    359 |    360 |    343 |
| 129.128.153.34  |  15 |    407 |    343 |    344 |
| 192.168.254.1   |  16 |    344 |    344 |   2000 |
| 129.128.4.241   |  17 |    343 |    344 |    344 |
+-----------------+-----+--------+--------+--------+
```

(b)

Figure 1. An example of traceroute from two different IPs from the same city (Liuzhou, China) and the same ISP (China Telecom) to the same destination (IP:129.128.4.241, University of Alberta). (a) From IP:222.84.110.115; (b) From IP: 219.159.216.235. Note: 2000 means that the round-trip time is equal to or large than 2000ms.

3. Traceroute-Based Peer Selection

3.1. Tracker-Based Achitecture and Burden on Tracker

Although totally decentralized DHT-based P2P systems [16] are currently popular, they are inefficient for networking. Especially, it is time-consuming to find peers through multiple hops when it is critical to shorten startup time for VOD. Therefore, we argue that using reliable Tracker servers like BitTorrent [1] to manage the connections of peers is more suitable for P2P VOD. To balance load and avoid single point failure, multiple Trackers are used to manage the connections. Totally decentralized architecture without delaying startup time would be discussed in the future work.

With a tracker-based architecture, it is important to reduce the burden of trackers in order to require fewer trackers. For example, a tracker in BitTorrent only maintains whether a peer is online. By default, a peer reports every minute if it is still online. Our traceroute-based peer selection scheme does not add significant burden to a tracker because of the following measures:

- Traceroute record from a client is only reported to trackers once at the beginning.
- Traceroute-based peer selection happens only once when a client requires peer list at the beginning.
- Trackers only keep the online peers' traceroute record. Furthermore, only the first 6 hop routers and routers with the average RTT less than 150ms to the client are kept. Suppose the size of source IP is 4 bytes. The size of router IP is 4 bytes. The size of hop (with the range 0-6) is 1 bytes. The size of RTT (with the range 0-150ms) is 1 bytes. Then, the total size of each entry is 4+4+1+1=10 bytes. Assume one tracker could manage 10,000 online peers. The size of online peers' traceroute records is: 6 * 10 * 10,000 = 600,000 bytes, which are relatively small and efficient for querying.

3.2. Report Traceroute

If only one destination is used, clients close to the destination would lack routers along the path for finding close peers. Thus, a client would traceroute to two destinations. The traceroute record to the destination with more hops of routers to the client is reported to Trackers. The two destinations are chosen as far as possible, such as one in China and another one in North America. Then, at least one traceroute record has enough router entries.

To avoid waiting for traceroute when a client starts, a client keeps record of the last traceroute. If updated traceroute is not available yet, the last record will be used. This strategy works based on the observation that most routers along the path do not change frequently. This way, only the first use of the program need to wait traceroute.

3.3. Peer Selection Algorithm

Intuitively, the less the hop of a shared router to a client, the closer the two peers are. The less the round-trip time between a shared router and a client, the closer the two peers are. Peer selection is conducted from the router with less RTT or less hops to the client.

Algorithm 1: Traceroute-based Peer Selection Algorithm

```
Client:
$D_1$ = The traceroute record to the first destination;
$D_2$ = The traceroute record to the second destination;
$D_{selected}$ = The traceroute record of the destination with more hops of routers to the client;

Send the message of requesting peer list to trackers with selected traceroute record $D_{selected}$ listed by hop or RTT in ascending order;

Tracker:
$R_{max}$= maximal round-trip time for peer selection
$H_{max}$= maximal hop for peer selection

Only keep the online peers' traceroute record with the first $H_{max}$ hop routers and routers with RTT less than $R_{max}$ to the client;

For (each RouterIP in traceroute record, whose round-trip time to the client is less than $R_{max}$ or hop to the client is less than $H_{max}$)
{
        Add peers with the same RouterIP, whose RTT
        to the client is less than $R_{max}$ or whose hop to the client is less than $H_{max}$;

        If (the number of candidate is larger than required candidate num)
        {
                Break;
        }
}
If (the number of candidate is less than required candidate num)
{
        Randomly select more peers into candidate.
}
return candidate to the client.
```

4. Experiments and Analysis

We used the pcVOD real-world system [17] to collect the traceroute records from 352 different IPs from Asia, North America and Europe to two different destinations, 222.89.109.150 (Zhengzhou, China) and 129.128.4.241 (University of Alberta, Edmonton, Canada). To evaluate the accuracy of selecting peers from the same city and the same ISP, we observe the data from the four cities with multiple IPs as listed in Table 1. IPs from the same city are from different subnets. For example, the 21 IPs from Changsha Telecom, China are listed in Table 2. As we can see, there are 14 subnets.

Table 1: Cities with multiple IPs in our traceroute records

City and ISP	IP number
Changsha Telecom, China	21
Liuzhou Telecom, China	8
Hangzhou Netcom, China	17
Shanghai Telecom, China	11

We randomly pick one IP from the same city as the client. Then, the peer selection algorithm is used to select peers from the 352 IPs. Detection rate is the number of selected IPs, from the same city as the client, divided by the total number of IPs from this city. False alarms are the number of selected IPs, which are not from the same city as the client.

4.1. Hop-Based Peer Selection

To analyze what the proper threshold of maximal hop from peers to shared routers should be, we observe with different maximal hops as Table 3 shows. As we can see from Table 3, with increasing maximal hop, the overall detect rate increases. Before maximal hop increases to 7, there are no significant false alarms. When maximal hop is 6, the detect rate is quite high without false alarm. It verifies the efficiency of our method to automatically select peers from the same location and from different IP subnets.

Table 2: IP Subnets and IPs from Changsha Telecom China

IP Subnets	IP
218.77.56.*	218.77.56.140
	218.77.56.203
218.77.58.*	218.77.58.85
218.77.59.*	218.77.59.147
	218.77.59.231
220.168.41.*	220.168.41.246
	220.168.41.89
220.168.61.*	220.168.61.155
220.168.62.*	220.168.62.125
220.168.63.*	220.168.63.70
222.240.120.*	222.240.120.179
222.240.121.*	222.240.121.54
222.240.24.*	222.240.24.116
	222.240.24.210
	222.240.24.34
222.240.25.*	222.240.25.67
222.240.26.*	222.240.26.175
	222.240.26.192
	222.240.26.208
222.240.27.*	222.240.27.149
61.150.138.*	61.150.138.70

Table 3: Detect rate and false alarms with different maximal hop

Max Hop	Overall Detect rate	Overall False alarms
2	35.8%	0
3	86.8%	0
4	86.8%	0
5	90.6%	0
6	90.6%	0
7	92.4%	11
8	98.1%	32

Table 4: RTT-based peer selection from a client in Changsha Telecom, China

Max RTT	Detect number	False alarms
200	100%	5
150	90%	4
100	45%	4

4.2. RTT-Based Peer Selection

Table 4 shows the detect rate and false alarms of RTT-based peer selection by a client from Changsha Telecom, China. We can see that RTT-based peer selection algorithm is not as efficient as hop-based peer selection algorithm to select peers from the same city and same ISP. Specifically, there are significant false alarms.

4.3. Combined Peer Selection

Although RTT-based algorithm is not as efficient as hop-based algorithm to select peers from the same city and the same ISP, it could be used to select more peers when there are not enough peers from the same city and the same ISP. This strategy is based on the observation that if both of their RTT to the same router are low, that the chance of two peers with low RTT between each other is higher. The experiment below verifies the intuition.

Based on our collected traceroute records, the RTT of all links between China and North America are larger than 150ms, while RTT of the most links within North America are less than 150ms. We randomly pick one IP from Canada as the client. As Table 5 shows, with hop-based peer selection algorithm with maximal hop 6, only 3 peers from Canada could be selected. Table 6 shows RTT-based peer selection algorithm with maximal RTT 150ms could select more peers from North America without peers from China, which illustrates that RTT-based peer selection algorithm could be used to select peers with lower RTT to the client.

Table 5: Hop-based peer selection from a client in Canada

Max Hop	Number of Selected IP from Canada(outside China)	Number of Selected IP from China
6	3	0

Table 6: RTT-based peer selection from a client in Canada

Max RTT (ms)	Number of Selected IP from Canada/USA (outside China)	Number of Selected IP from China
200	11	2
150	10	0
100	10	0

5. Conclusions and Future Work

In this paper we proposed an algorithm for selecting peers that are close to the same location or with low round-trip time from each other, following the observation that close peers share routers along the traceroute path to the same destination. Oriented to P2P VOD application, a tracker-based architecture is used. Trackers only keep the online peers' traceroute records. Based on experiments with real traceroute records from 352 different IP addresses around the world, hop-based peer selection algorithm with maximal hop 6 could effectively select peers from the same city and the same ISP without significant false alarms. RTT-based algorithm with maximal RTT 150ms could be used to select more peers when there are not enough peers from the same city and the same ISP. Introducing our strategy into a totally decentralized architecture would be considered in our future work.

6. References

[1] B. Cohen, Incentives Build Robustness in BitTorrent, http://bitconjurer.org/BitTorrent/.

[2] T. S. Eugene Ng, Y.H Chu, S.G. Rao, K. Sripanidkulchai, and H. Zhang, "Measurement-Based Optimization Techniques for Bandwidth-Demanding Peer-to-Peer Systems," IEEE INFOCOM'03, 2003.

[3] D.S. Bernstein, Z. Feng, B.N. Levine, and S. Zilberstein, "Adaptive Peer Selection," 2nd International Workshop on Peer-to-Peer Systems, Berkeley, CA, USA, 2003.

[4] L. Xiao, Y. Liu, and L.M. Ni, "Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment," IEEE Transactions on Computers, Vol. 54, No. 9, Sep 2005.

[5] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "PROMISE: Peer-to-Peer Media Streaming Using CollectCast," ACM Multimedia, Berkeley, CA, 2003.

[6] Z. Xu, C. Tang, and Z. Zhang, "Building Topology-Aware Overlays using Global Soft-State," 23rd International Conference on Distributed Computing Systems (ICDCS), 19-22 May, 2003.

[7] M. Adler, R. Kumary, K. Rossz, D. Rubensteinx, T. Suel and D.D. Yaok, "Optimal Peer Selection for P2P Downloading and Streaming," IEEE INFOCOM'05, 2005

[8] X. Zhang, J. Liu, B. Li, and T.S.P. Yum, "CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming," IEEE INFOCOM'05, 2005.

[9] http://www.pplive.com

[10] http://www.ppstream.com

[11] V.N. Padmanabhan and L. Subramanian, "An Investigation of Geographic Mapping Techniques for Internet Hosts," ACM SIGCOMM'01, 2001.

[12] http://www.cz88.net/fox/

[13] http://www.ip2location.com/

[14] V. Jacobson, "Traceroute software," 1989, ftp://ftp.ee.lbl.gov/traceroute.tar.gz

[15] Z.M. Mao, J. Rexford, J. Wang and R.H. Katz, Towards an Accurate AS-Level Traceroute Tool, ACM SIGCOMM 2003, 2003.

[16] B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, and J.D. Kubiatowicz, "Tapestry: A Resilient Global-scale Overlay for Service Deployment," IEEE Journal on SAC, Vol. 22, No. 1, Pgs. 41-53, 2004.

[17] L. Ying and A. Basu, "pcVOD: Internet Peer-to-Peer Video-On-Demand with Storage Caching on Peers," The Eleventh International Conference on Distributed Multimedia Systems DMS'2005, Banff, Canada, 2005.