P2P Traffic Localization by Forcing Packet Loss

Hiep Hoang-Van [1] , Koki Mizutani [1] , Takumi Miyoshi [1] , Olivier Fourmaux,[2]

[1] Graduate School of Engineering and Science, Shibaura Institute of Technology,
Saitama, 337-8570, Japan

E-mail: {nb12510,mf13071,miyoshi}@shibaura-it.ac.jp

[2] Laboratoire d'Informatique de Paris 6, UPMC Sorbonne Universités,
Paris, 75005, France

E-mail: olivier.fourmaux@upmc.fr

Received 18 January 2013

Accepted 13 August 2013

Abstract

Recently, peer-to-peer (P2P) traffic is increasing rapidly in volume day by day. One of the main causes is that most of P2P applications including file sharing and streaming applications often form overlay networks for exchanging data that are oblivious to the underlay network topology. As a result, they generate a large amount of inter-domain traffic causing higher cost for internet service providers (ISPs). This raises a problem of traffic localization. To optimize the cross-ISP/AS traffic, existing approaches focus on solving the problem on the application layer where each P2P application must be equipped with an additional protocol to obtain underlay network information from an "oracle" server or an additional locality-aware procedure to estimate location by itself. Therefore some modifications of application software are required for these approaches. In this paper, we propose a novel solution for addressing the problem, called PLS, forcing packet loss to each P2P packet based on geographical location of each destination at the network layer. Since PLS is implemented at a network router, no software modification is required. This proposal can be applied to all types of P2P applications in order to localize the traffic. The experiments evaluated on popular P2P streaming applications show that our proposed method significantly reduces the cross-domain traffic by suppressing the connections with faraway peers.

Keywords: P2P; traffic localization; router aided; packet loss; P2PTV

1. Introduction

According to a recent study, although peer-to-peer (P2P) traffic is declining in percentage of overall Internet traffic, it is still increasing rapidly in volume due to tremendous growing of multimedia content delivery such as video streaming, which accounted for 51 percent of all consumer Internet traffic in 2011 and will up to 55 percent in 2016 [1]. This does not include video exchanged through P2P file sharing. Currently, P2P streaming applications (P2PTV), such as PPStream [2], PPTV (the update version of PPLive) [3], SopCast [4], and Zattoo [5] have become increasingly popular. Therefore, controlling P2P traffic is one of the biggest challenges for internet service providers (ISPs) and research community as well.

The problem is that most of P2P applications often ignore traffic costs at ISPs: peers construct an overlay network and establish their connections based on not the network stability but the resource availability. The overlay network is thus almost independent of the physical network due to a lack of knowledge about the underlay network topology. As a result, P2P applications generate a large amount of unpredictable cross-ISP/AS (Autonomous system) traffic on the Internet. This might temporarily improve performance of P2P application, but affect to available bandwidth of other applications and drive ISPs to pay more for the inter-domain traffic. To reduce the cost for handling cross-ISP/AS traffic, ISPs might try to implement bandwidth throttling or limiting, and/or even blocking P2P systems in their network. In response, P2P systems may change their design and try to hide them from the network. This makes P2P traffic control problem become more challenging. A variety of methods have been introduced, and many works [6,7,8,9,10,11,12,13,14] suggest that the consideration of peer location would reduce inter-ISP/AS traffic and also conserve the bandwidth. This forms the problem of traffic localization.

Almost all of existing approaches focus on solving the problem on the application layer where modifications of P2P systems are required as follows:

- The modification of the application software to integrate a locality-awareness procedure,
- the enhancement of trackers, which guides applications to make better choice in selection of neighbor peers, or
- both of the above.

In this paper, we propose a novel approach for localizing P2P traffic, called PLS (Packet Loss Scheme). In PLS, each packet is forced to drop with a given probability according to geographical locations of the destinations at the gateway routers. The packet loss rate for the connection paths to farther peers are forced to become higher than those to closer peers. By doing that, P2P applications tend to eliminate connection paths to farther peers, and the traffic will be thus localized. Because our proposal is implemented on network routers, it is completely independent of existing P2P applications and can be easily applied to all P2P systems without any software modification.

The rest of this paper is organized as follows. In section II, we introduce the related work. Our proposed scheme is explained in detail in Section III, and then Section IV shows an implementation of the proposed method. Section V gives experimental results. Conclusions and future work are provided in Section VI.

2. Related work

The idea of traffic localization techniques was originally proposed for P2P file sharing applications. Plissonneau et al. [15] showed their analysis on eDonkey file sharing, and reported that most of traffic traversed nationwide or international networks. It is also noted that about 40 percent of the traffic could be localized if locality-aware peer selection mechanisms were integrated. Karaginanis et al. analyzed Bittorent trace logs and found that about 50 percent of the files could be downloaded from peers at the same ISP [16].

The work introduced by Bindal et al. is probably closest to ours [8]. They proposed biased neighbor selection scheme applying to Bittorent, which selects only *k* peers from outside of ISP and (35 - *k*) peers in the same ISP, where *k* is parameter. The idea of biased neighbor selection can be implemented in two ways: modifying trackers and clients and the use of P2P traffic shaping devices. The former certainly requires software modification, whereas the latter, similar to ours method, does not require any modification of software. However, to apply this idea to other applications such as P2PTV, the peer list format of the applications has to be known in advance. It means that the shaping device will be dependent on the P2P applications.

Aggarwal et al. [6] proposed that ISPs and P2P users should cooperate together for improving the performance of traffic localization problem. The ISPs, by knowing clearly the network information such as a physical topology and geographical information of peers, can offer an “oracle” service to the P2P users to help them make a better selection of neighbor peers. In response, P2P users can query the oracle and use the suggestion to help ISPs manage their traffic more efficiently. This scheme requires the trust and good cooperation between the ISPs and the P2P applications. In addition, each P2P application must be equipped with an additional module to communicate with the oracle.

Deriving from the oracle idea, P4P is a very famous framework [14]. P4P also proposed that network providers and P2P system should not try to improve network efficiency independently, but cooperate with each other. In the P4P architecture, each network provider, e.g. an ISP, maintains an iTracker in its own network. An iTracker provides the p-distance interface, representing the logical distance and costs among PIDs (aggregation nodes) from coarse-grained to fine-grained based on physical network information. Contrary to the original oracle, the use of p-distance interface by ISPs and P2P applications is very flexible due to simple and extensible of interface design. The Internet Engineering Task Force (IETF) is going to make a standardization called ALTO (application-layer traffic optimization) [13], [7] based on oracle-based system [6] and P4P [14].

Choffnes and Bustamate introduced another approach for traffic localization [9]. They claimed that the presence of the oracle service provided by ISPs is redundant since all the necessary information for peer selection is already gathered by content distribution networks (CDNs). By using DNS redirection, they assumed that two peers are recognized as being close to each other if they are redirected to a similar set of replica servers. This idea was implemented as a java plugin to Azureus Bittorent client, called Ono. Of course, a lot of software modifications are required for this scheme.

Other approaches including IDMaps [10] and GNP [12] proposed mechanisms for underlay network information estimation by end system. The distance between two peers can be estimated based on some special hosts called tracers in IDMaps or Landmarks in GNP, whose locations have been computed in advance. The existence of tracers and Landmark makes it hard to achieve high accuracy in a large-scale network. In addition, such measurement-based approaches require to probe the network by active or passive measurement. However, measuring real traffic is sometimes very difficult when some overlay networks are running on one node simultaneously.

As described above, most of existing P2P locality-aware mechanisms require several software modifications of P2P systems. This is sometimes very hard due to closed design or license problem of commercial software. Miyoshi et al. [11] proposed P2P-DISTO framework for P2P traffic localization without any modification of existing application software. According to the geographical location of peers, the packets that were sent to or received from farther peers were inserted longer additional delay than closer peers. This paper is similar to the P2P-DISTO idea but tries to realize it with a different mechanism. Instead of adding delay, we discard the packets with an adaptive probability at network routers. Our method requires neither any software modification nor collaboration between ISPs and P2P system. Therefore, it can be applicable as a solution for reduction of the cross-ISP/AS traffic to wide variety of P2P applications.

3. Proposed Scheme

As we know, P2P systems build their all protocols on the underlay network for the communication among peers, called overlay network. Consider the overlay network in which a querying peer receives a list of available candidates from a root server, trackers, or other peers. Without a locality-aware mechanism, the querying peer then randomly selects a set of peers to contact with. Such kind of selection sometimes leads to suboptimal choices. To increase the downloading speed, P2P applications including P2PTV currently tend to eliminate delayed peers from the list of available peers. From the viewpoint of the applications, longer round-trip time (RTT) leads to longer delay. From this observation, Miyoshi et al. proposed P2P-DISTO [11] that adjusts RTT by inserting additional delay into each packet depending on the geographical location of its destination in order to localize the traffic. However, to delay the traffic, the routers need to hold the packets for a certain time before forwarding them to the destination. P2P-DISTO therefore requires a very large buffer memory to hold many packets at any time. This makes it hard to deploy P2P-DISTO in a real router.

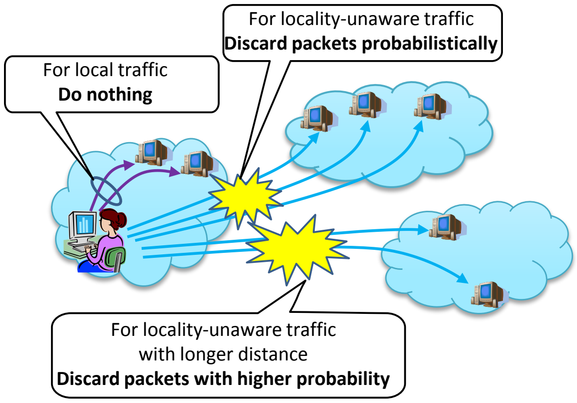
For locality-unaware traffic
Discard packets probabilistically
For local traffic
Do nothing
For locality-unaware traffic
with longer distance
Discard packets with higher probability

Fig. 1. The concept of PLS.

Communication performance in P2P networks would be measured in terms of not only latency but also packet loss probability. We thus propose PLS to adjust packet loss probability for each packet depending on the geographical location of its destination. Figure 1 illustrates the concept of the PLS scheme. PLS does nothing for local traffic, but forces packet loss for locality-unaware traffic. By discarding packet probabilistically, the P2P traffic can be localized for the following reasons: (1) Packet loss might cause a failure of transmitting the hand shaking packets with farther peers, and closer peers are therefore likely to be selected instead; (2) If no failure occurs in the hand shaking phase, the peer might download video data from another peer located in a different ISP/AS. However, the subsequent packet loss will make this connection unstable. Moreover, P2P streaming applications tend to close unstable connections to improve the quality of the video. Therefore, cross-ISP/AS connections will be reduced; in other words, the traffic can be localized. Since PLS does not require large buffer memory, it can be more easily deployed in a real router.

As mentioned in previous sections, PLS can be implemented outside of the existing applications. Therefore, we introduce a router-aided approach. Figure 2 shows an architecture of the proposed router. A traffic classification module and a packet loss module are complemented with the common routing function. The traffic classification module classifies input traffic into two types: P2P traffic and the other traffic. The classification ensures that only P2P traffic goes into packet loss module whereas the normal traffic is forwarded directly to common routing function. This is to avoid degrading the quality of service (QoS) of non-P2P traffic flowing through the PLS router.

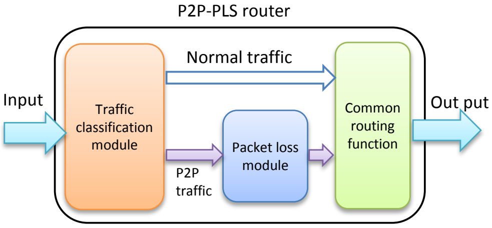

Fig. 2. PLS router architecture.

In the packet loss module, the destination of each packet is first resolved by using several IP-to-geographic-location mapping services. The packet might be discarded with a given probability depending on the location of the destination of packet. If the destination is in the same area as the PLS router, for example in the same AS, ISP, or country, the packet is normally forwarded to the common router function, in other words the packet loss rate is zero. Otherwise, the module drops the packet with a defined rate of packet loss. The packet loss rate can be changed according to the distance.

There are two drawbacks of forcing packet loss method. With packet recovery supported protocols such as TCP, packet loss makes an impact on network throughput due to re-transmitting the missing packets. With other protocols such as UDP providing no recovery for lost packets, packet loss results in degrading application performance. Therefore, the reasonable packet loss rate and the QoS should be taken into account.

4. Implementation of PLS

We describe the implementation of PLS in detail in this section. We set up a desktop PC as a PLS router. The detailed hardware configuration is as follows: Intel Core i7-2600 3.4 GHz CPU, 12GB memory, two 1-Gbps Ethernet network interface cards, operated under Linux Ubuntu 12.04, kernel 3.2.0-29 generic.

Published by Atlantis Press
Copyright: the authors

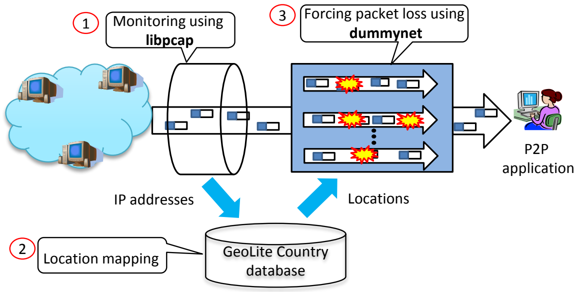

Fig. 3. The implementation of PLS.

For the traffic classification module, many methods have been already proposed. For instance, ISPs usually use deep packet inspection and session-based classification with 5 tuples (IP addresses, port numbers, and protocol type) to block P2P systems. More recently, Valenti et al. introduced Abacus [17], a behavioral classification method for P2P traffic. Abacus showed an accurate classification result of P2P applications relying only on the count of packets and bytes that peers exchange during a small time windows. Therefore, we can utilize such kinds of classification methods above to implement the module in PLS. In this study, we skip the packet classification module and only focus on implementation of packet loss module.

Figure 3 illustrates the implementation of packet loss module in the following three steps:

- Packet monitoring: We used *libpcap* [18], a well-known packet capture library to monitor all packets travelling through the router. The header of each packet was analyzed to read their source and destination IP addresses.
- Location mapping: the obtained IP addresses were then mapped to their location using IP-to-location service. In this study, we utilized GeoLite Country database, a free IP-to-country database created by MaxMind [19] for simplicity. After this step, we have a map of IP addresses to their corresponding countries.
- Forcing packet loss: for simulating packet loss in a real network, *dummynet* [20] was used. Dummynet is a flexible tool for simulating packet filtering, bandwidth management, packet delay, and packet loss. It is originally implemented in FreeBSD but also available for other frameworks including Linux and Windows. By using ipfw firewall, which is the main user interface for dummynet, we can create pipes between sender and receiver peers, and the packets are carried through these pipes. Depending on the geographical location of the destination obtained in the second step, each pipe can be configured with a different packet loss rate.

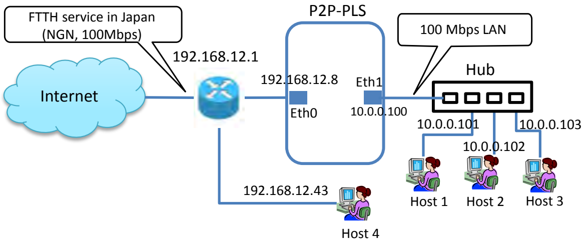

Fig. 4. Network environment and configuration.

5. Experimental Results

Figure 4 shows our experimental network environment and configuration. For the Internet connection, we signed the contracts with FLETS HIRAKI NEXT, a 100 Mbps optical access service on the next generation network (NGN), and plala HIKAKI Mate with FLETS as an ISP in Japan. PLS is configured as a gateway router. There are 2 types of measurement hosts. Host 4 is connected directly to the main switch to evaluate the traffic in normal case without applying packet loss scheme. Other three hosts are connected to the PLS router to evaluate the traffic when the proposed method is applied. The hardware configurations of all the measurement hosts are the same as follows: Intel Core i5-2440 CPU 3.1 GHz, 4GB of memory, a 100 Mbps network interface card, operated under 64-bit Windows 7.

Wireshark [21], a well-known packet sniffer application, was installed on all the measurement hosts for capturing traffic and generating statistical information. To ensure that no non-P2P traffic is generated due to the skip of traffic classification module, only P2P applications and Wireshark are permitted to run on the measurement hosts.

In this experiment, we evaluated on existing P2PTV applications because of their current popularity. We selected two types of applications, PPTV for performing on-demand video, and SopCast for performing live streaming video. The consideration of peer's locality on the two applications was reported in technical report of NAPA-WINE project [22], where SopCast is almost unaware of peer's location and PPTV is not strong as well. We set each application to run one-by-one on the measurement hosts. On SopCast, a live Chinese channel, CCTV-2, was selected to play. On PPTV, we watched an on-demand drama.

We configured the PLS where every packet exchanging with foreign peers will be lost with probability of 10, 20, and 30 percent on hosts 1, 2, and 3 respectively. Meanwhile, we do not need to rebuild the source code whenever we change the packet loss rate.

5.1. Results with PPTV

The experiment with PPTV was conducted in October 2012. An on-demand drama video was selected for playing on the measurement hosts. Since each host has been configured with a different packet loss rate, we measured the amount of downloaded data and the number of peers on each host, and computed their ratio by country as metrics for comparing. Each packet loss mode from 0 to 30 percent was performed five times, and the average value of the downloaded data quantity and the number of peers were calculated as the final results.

The left side of Fig. 5 shows the downloaded data distribution, where the vertical axis presents the country-by-country ratio of downloaded data. The top three countries when the packet loss rate is zero are shown individually, and the others are bundled together into one group. Without packet loss, downloaded data from China and Japan account for major portion of total, which is about 40 percent for each country. With any packet loss mode of PLS, the ratio of downloaded data from Japan increases significantly, whereas the overseas traffic from China and other countries decreases. The packet loss rate of 20 percent presents the best performance, where the ratio of downloaded data from Japan accounts for approximately 80 percent of the total. This proves that PPTV tends to receive data pieces from peers in Japan that have much lower packet loss rate than foreign peers. In other words, our proposal clearly realizes P2P traffic localization on PPTV.

The right side of Fig. 5 presents the neighbor peer distributions, where the vertical axis shows the country-by-country ratio of the number of peers that the measurement host exchanged data with. The results indicate that the number of peers has changed very little, and almost independent of packet loss rate. This can be explained as follows: PPTV first connects to some root servers to obtain a list of available online peers, and then forms an overlay network to exchange data pieces with a subset of those peers. Because PLS cannot intervene to this phase, the neighbor peer distribution is therefore pretty stable.

Figure 6 gives an example of temporal change of throughput by country. From this figure, without the proposed packet loss method, most of traffic comes from China, and the amount of traffic from Japan accounts for minor portion. After applying the packet loss method with the rate of 20 percent, PPTV changes to connect with peers in Japan to download the data. Therefore, the amount of traffic from Japan significantly increases and accounts for major portion.

5.2. Results with SopCast

All experiments with SopCast were performed in November 2012. A live Chinese channel, CCTV-2, was chosen for the experiment on the measurement hosts. We found that SopCast is different from PPTV in the number of neighbor peers. There are very few peers: only four or five neighbor peers which are exchanging video data are found in SopCast. It means that we cannot sometimes find any Japan peer and therefore cannot localize the traffic inside Japan. To avoid this, we have to check whether Japan peers exist or not before running the experiment.

Figures 7 and 8 demonstrate the results for SopCast. The results are similar to those of PPTV: the amount of downloaded data from Japan increases

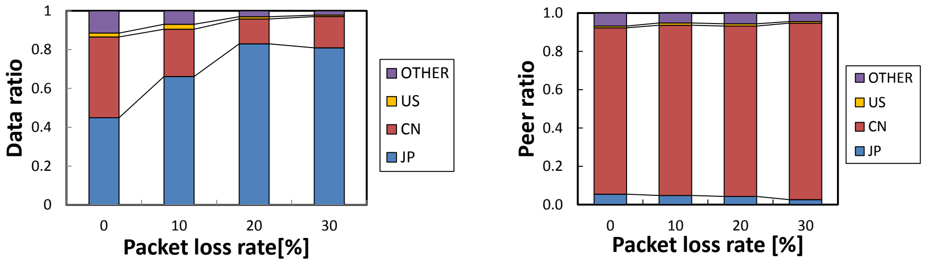

Fig. 5. Downloaded data and neighbor peer distribution by country for PPTV when forcing packet loss with different rate.

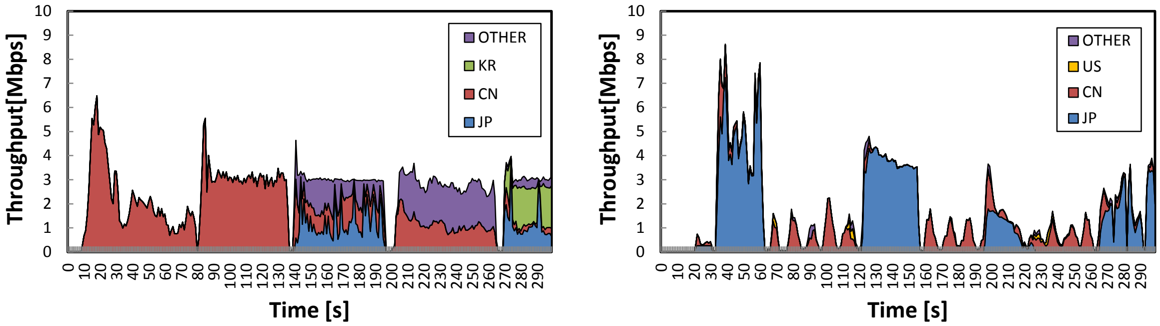

Fig. 6. Temporal change of throughput for PPTV without and with packet loss method; the left side shows the result in normal case without packet loss method, and the right side shows the result when forcing 20% packet loss to overseas traffic.

sharply according to the packet loss rate; whereas the neighbor peer distributions are fairly stable.

6. Conclusions

This paper proposed a router-aided approach for optimizing the traffic of P2P streaming applications. By forcing packet loss at the routers, traffic could be localized without any modifications of application software. This proposal can be applied for designing new traffic-shaping devices that support localization function. We also introduced an implementation of the PLS router utilizing libpcap, dummynet, and GeoLite country database. The experimental results proved that the proposed method successfully realized traffic localization on P2PTV applications. In particular, once the packet loss method is applied, both PPTV and SopCast change to download video data pieces from peers in Japan and tend to eliminate the connection paths to foreign peers.

Several challenges remain at the current implementation. In the future, we will improve the PLS using not only country information of peers but also other information such as ISP, AS, and the quality of connection links to achieve finer-grained results. Furthermore, the quality of video after applying the method should be examined carefully.

Acknowledgment

This study was partly supported by a Grant-in-Aid for Young Scientists (B) No. 23760344 from the Japan Society for the Promotion of Science (JSPS).

References

1. Cisco System, Inc. “Cisco visual networking index: forecast and methodology, 2011-2016,” *White paper*, (2012)
2. PPStream. http://www.pps.tv/

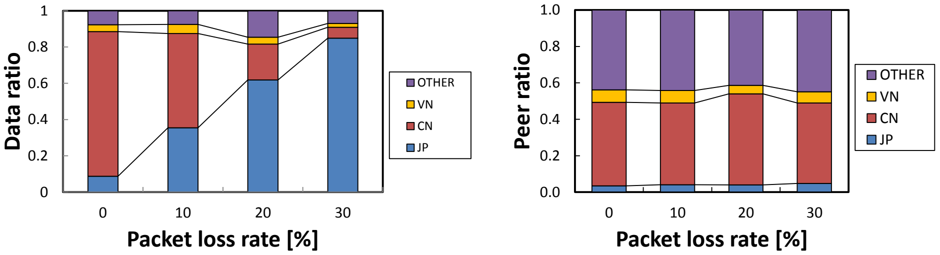

Fig. 7. Downloaded data and neighbor peer distribution by country for SopCast when forcing packet loss with different rate.

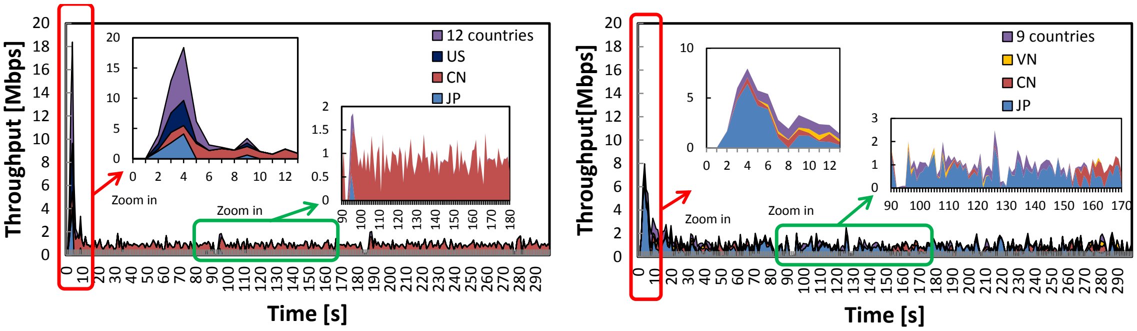

Fig. 8. Temporal change of throughput for SopCast without and with packet loss method; the left side shows the result in normal case without packet loss method, and the right side shows the result when forcing 20% packet loss to overseas traffic.

3. PPTV. http://www.pptv.com/
4. SopCast. http://www.sopcasst.com/
5. Zattoo. http://zattoo.com/
6. V. Aggarwal, A. Feldmann, and C. Scheideler, “Can isps and p2p users cooperate for improved performance?,” *ACM SICCOMM Comput. Commun. Rev.*, **37**, 29–40 (2007).
7. R. Alimi, R. Penno, and Y. Yang, “Alto protocol,” *Internet draft, draft-ietf-alto-protocol-10.txt*, (2011).
8. R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, “Improving traffic locality in bittorrent via biased neighbor selection,” *Proc. IEEE Int. Conf. Distributed Comput. Syst. (ICDCS 2006)*, (2006).
9. D. Choffnes and F. Bustamante, “Taming the torrent - a practical approach to reducin cross-isp traffic in peer-to-peer systems,” *Proc. ACM SIGCOMM*, 363-374 (2008).
10. P. Francis, “A global internet host distance estimation service,” *IEEE/ACM Trans. Net.*, **9**, 525-540 (2001).
11. T. Miyoshi, Y. Shinozaki, and O. Fourmaux, “A p2p traffic localization method with additional delay insertion,” *Proc. 4th Int. Conf. Intelligent Networking and Collaborative Syst. (INCoS2012)*, 148-154 (2012).
12. T. Ng and H. Zhang, “Predicting internet network distance with coordinates-based approaches,” *Proc. IEEE INFOCOM*, (2002).
13. J. Seedorf, S.Kiesel, and M.Stiemerling, “Traffic localiza-tion for p2p-applications: the alto apprach,” *Proc. IEEE Int. Conf. Peer-to-Peer Comput. (P2P2009)*, 171-177 (2009).
14. H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu, and A. Sil-berschatz, “P4p: provider portal for applications,” *Proc. ACM SIGCOMM 2008*, 351-362 (2008).
15. L. Plissoneau, J. Costeux, and P.Brown, “Detailed analysis of edonkey transfers on adsl,” *Proc. 2nd Conf. Next Generation Internet Design and Engineering (NGI 06)*, 256-262 (2006).
16. T. Karaginannis, P. Rodriguez, and K. Papagiannaki, “Should internet service providers fear peer-assisted content distribution?” *Proc. Internet Management Conf. (IMC 2005)*, 63-76 (2005).
17. S. Valenti and D. Rossi, “Fine-grained behavioral clas-sification in the core: the issue of flow sampling,” *Proc. 7th Int. Wireless Commun. and Mobile Comput. Conf. (IWCMC2011)*, 1028-1032 (2011).

18. Tcpdump and libpcap pulbic repository. http://tcpdump.org/.
19. MaxMind and GeoIP, IP address location technology. http://www.maxmind.com/app/ip-location/.
20. Dummynet. http://info.iet.unipi.it/˜luigi/dummynet/.
21. Wireshark. http://www.wireshark.org/.
22. A. Horvath, M. Telek, D. Rossi, P. Veglia, D. Ciullo, M. Garcia, E. Leonardi, and M. Mellia, “Dissecting pplive, sopcast, tvants,” *NAPA-WINE project, Tech. Rep.*, (2009).