# THash: A Practical Network Optimization Scheme for DHT-based P2P Applications

Yi Sun, Richard Yang, Xiaobing Zhang, Yang Guo, Jun Li, Kavé Salamatian

▶ **To cite this version:**

Yi Sun, Richard Yang, Xiaobing Zhang, Yang Guo, Jun Li, et al.. THash: A Practical Network Optimization Scheme for DHT-based P2P Applications. IEEE Journal on Selected Areas in Communications, 2013, 31 (9), pp.379-390. 10.1109/JSAC.2013.SUP.0513034 . hal-00876543

**HAL Id: hal-00876543**

**https://hal.science/hal-00876543v1**

Submitted on 20 Dec 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THash: A Practical Network Optimization Scheme for DHT-based P2P Applications

Yi Sun, Y. Richard Yang, Xiaobing Zhang, Yang Guo, Jun Li, and Kave Salamatian

***Abstract*—P2P platforms have been criticized because of the heavy strain that they can inflict on costly inter-domain links of network operators. It is therefore mandatory to develop network optimization schemes for controlling the load generated by a P2P platform on an operator network. While many research efforts exist on centralized tracker-based systems, in recent years multiple DHT-based P2P platforms have been widely deployed and considered as commercial services due to their scalability and fault tolerance. Finding network optimization for DHT-based P2P applications has thereby potential large practical impacts. In this paper, we present THash, a simple scheme that implements a distributed and effective network optimization for DHT systems. THash uses standard DHT put/get semantics and utilizes a triple hash method to guide the DHT clients to choose their sharing peers in proper domains. We have implemented THash in a major commercial P2P system (PPLive), using the standard ALTO/P4P protocol as the network information source. We conducted experiments over this network in real operation and observed that compared with Native DHT, THash reduced respectively by 47.4% and 67.7% the inter-PID and inter-AS traffic, while reducing the average downloading time by 14.6% to 24.5%.**

***Index Terms*—Network optimization, DHT, peering guidance matrix (PGM), ALTO/P4P, PPLive.**

## I. Introduction

P2P platforms have been frequently considered as commercial platforms for content distribution [1]. But one major obstacle to the further development of P2P system is the inability to control the spread of the traffic and the utilization of network resources, in particular costly inter-domain link capacities. This is therefore mandatory for a content distribution P2P platform to provide solutions for controlling the spread of the load they generate and to maintain the locality of traffic. We are calling this problem P2P network optimization. The aim of this paper is to propose a practical scheme to address the network optimization problem in DHT-based P2P content distribution platforms.

Manuscript received February 27, 2012; revised July 16, 2012.

Y. Sun, Y. Guo, and J. Li are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P.R. China (e-mail: {sunyi, guoyang, lijun}@ict.ac.cn).

Y. R. Yang is with Yale University, New Haven, CT, USA (e-mail: yry@cs.yale.edu).

X. Zhang is with Shanghai Synacast Media Tech. Co. Ltd. (PPLive), Shanghai, P.R. China (e-mail: davidzhang@pplive.com).

K. Salamatian is with the University of Savoie France, Paris, France (e-mail: kave.salamatian@univ-savoie.fr).

This work was supported by the National Basic Research Program of China under Grant Nos.2012CB315802 and No.2011CB302702, the National Science Foundation of China under Grants No.61003266, No.61070188, No.61132001, No.61070187, and No.61100176, and the NSFC-ANR pFlower project under Grant No. 61061130562.

Digital Object Identifier 10.1109/JSAC.2013.SUP.0513034

Up to now, many research efforts targeted tracker-based centralized P2P applications, *e.g.*, [2]–[6]. However, in recent years, DHT-based schemes have been increasingly utilized because of decentralization, fault tolerance and scalability properties, *e.g.*, [7] (used in BitTorrent, eMule, and Thunder), Chord [8], and CAN [9]; currently, the DHT mode of Thunder contributes as high as 3% of the total traffic of China Telecom.

Network optimization entails favoring content downloading from local rather than distant peers. This optimization should be done taking into account the status of the network and the constraint of the network operator. In order to access this information, we can use ALTO/P4P [5] as the source of information relative to network status. Indeed, network optimization might deteriorate the user performance as it might result in limiting the scope of data exchanges by forbidding the user to download from other network regions that have high bandwidth. There is therefore a fundamental trade-off between network optimization and user performance.

Dealing with this trade-off is relatively easier in a tracker-based system, as the centralized tracker will consider both global bandwidth matching and locality when computing the neighbors for a peer. The problem becomes more difficult for DHT systems that are distributed. Moreover, a generic DHT system supports only "put" and "get" semantics; a "put" simply appends values to a list and a "get" retrieves a given number of top elements from the list. Therefore, a practical optimization scheme for DHT-based P2P systems should not change too much the semantics of "put" and "get" in order to maintain a high level of backward compatibility and an easy deployment. This can be achieved by adding clever manipulation scheme on top of DHT or by extending the semantic of DHT to enable the data transformation needed for network and bandwidth optimization.

In this paper, we present a simple scheme, named THash, which implements effective network optimization for DHT systems. An innovative notion Peering Guidance Metric (PGM) is proposed to consider not only the ISP network efficiency but also the application state and requirements. THash is following the standard DHT put/get semantics and implements distributed network optimization under the guidance of PGM by favoring peering with local peers.

However, any network optimization scheme is only efficient when there are enough potential local peers. An effective way to increase the number of potential peers is to enable NATed peers, *i.e.*, peers behind a NAT router, which currently account for a large proportion in typical P2P applications, to join the system as separate nodes. We will also present in the paper how to achieve this for DHT systems.

0733-8716/13/$31.00 © 2013 IEEE

We have completely implemented THash over PPLive [10], a leading P2P based online video service with more than 200 million users in China that offers both live and video-on-demand streaming of TV programs/shows, movies, drama, sports, news and entertainment. We obtained the network status information and operator guidelines through an ALTO/P4P feed. We conducted experiments over PPLive network with more than 28000 real users and showed the benefits of THash. In particular, we showed that compared with Native DHT, THash reduced respectively by 47.4% and 67.7% the inter-PID and inter-AS traffic, while reducing the average downloading time by 14.6% to 24.5%.

## II. Background Information

This section provides some background information that is needed for the rest of the paper.

### A. DHT Model

We consider a DHT-based network that uses standard DHT semantics. Specifically, the DHT network is used for publishing and querying resource information. Peers in the DHT network can play three roles:

**Publishing Peer (PP):** a peer who wants to publish information to announce its ownership of a specific resource.

**Requesting Peer (RP):** a peer who sends a request to search for a specific resource.

**Indexing Peer (InP):** a peer who maintains the publication information (publishing peer list) of a specific resource.

In a traditional DHT network, a PP computes a hash using the resource file identifier (file-id), and then invokes the "put" method of the DHT to insert the data into the corresponding InPs; a RP applys the same hash function as the PP on a given file identifier and then invokes the "get" method of the DHT to fetch from the InPs a list of $k$ peers that have the required resource.

A major drawback of classical DHT-based P2P schemes is that the returned peer list does not consider network optimization. Thus, the RP may download from peers that are not network efficient; where for network optimization, it is desirable that the RP connects to close-by PPs.

### B. ALTO/P4P as Network Information Source (NIS)

Ensuring the locality of traffic in P2P network has motivated the proposition of an Internet Draft at IETF for achieving Application Layer Transport Optimization (ALTO). The proposed approach named ALTO/P4P [5] is a simple and flexible framework that enables ISPs and application developers to cooperate in order to optimize application communications and to reduce resource consumption.

Specifically, a network provider (ISP) implementing the ALTO/P4P network information framework deploys a server (called iTracker) that provides its network information relative to topology, congestion status, cost and routing policies through the ALTO/P4P protocol. This information, which is called "my-Internet view" of an ISP, consists of two maps: the Network Map and the Cost Map. The Network Map divides an ISP network into multiple regions called PID domains identified each by a PID domain number (PIDN). The Cost Map defines the cost, referred as p4p-distance, between each pair of PID domains.

In order to obtain the PIDN and ASN that its IP address belongs to, each peer can query the iTracker, for example at the time it obtains its IP address. This query will also return the two Network and Cost Maps.

TABLE I
An Example PGM

| | PID1 | PID2 | PID3 | Intra-AS Percentage |
|---|---|---|---|---|
| PID1 | 75% | 10% | 15% | 90% |
| PID2 | 18% | 70% | 12% | 85% |
| PID3 | 10% | 10% | 80% | 90% |

### C. PPLive

In order to evaluate our proposed scheme in a real environment, we have implemented it into the PPLive environment [10]. PPLive (also referred as PPTV) is a leading online video service platform, founded in 2004 and offering more than 120 TV station live streaming and VoD of thousands of TV shows and programs. The PPLive Company claims to have more than 200 million user installations and an active monthly user base of 104 million, *i.e.*, a 43% penetration of Chinese internet users. The average viewing time per person per day over PPLive is more than 2.5 hours, the highest stickiness among all China websites. PPLive provides a hybrid CDN-P2P cloud platform and uses tracker-based scheme as well as DHT for its content distribution.

## III. THash Scheme

In this section, we describe the THash scheme that implements the network optimization along with taking care of application performance.

### A. Peering Guidance Matrix (PGM)

The ISPs provide through ALTO/P4P their perspectives through network and cost maps. In order to have a complete view of the network an application has to integrate it states and requirements into these maps. For this purpose, we define for each existing content resource and for each particular ISP a matrix called Peering Guidance Matrix (PGM). The PGM specifies for a client in a given PID, the relative proportion of downloading peers that it should use in each PID domain.

Table I shows an example PGM for a specific resource for an ISP with 3 PID domains. Following this PGM a RP of PID1 should select up to $90\%*75\%=67.5\%$ peers from its own PID, $90\%*10\%=9\%$ peers from PID2, $90\%*15\%=13.5\%$ peers from PID3 and $1-90\%=10\%$ peers from other ASes. It is noteworthy that in order to ensure fast download as well as system robustness, we should not restrict to choose peers only within a single domain and leave opportunity for clients to exchange data with peers outside. The precise derivation of the PGM values that are optimizing network load induced by the P2P application will be described in Section III-C.

### B. Triple Hash Implementation of PGM

In classical DHT network, the resource publishing and lookup (DHT put/get semantics) are only based on hashing the resource file identifier (file-id). However, in to optimize the network usage, we have to integrate the PGM into the process. This is achieved by the THash scheme we are defining in this section.

The THash scheme is based on a triple hash method that follows the standard DHT put/get semantics, while implementing the network optimization by adhering with the indications of PGM. Moreover, THash addresses a set of practical challenges, including avoidance of bottlenecks at network information source servers and NAT traversals.

*1) Resource Publishing in THash:* When a PP wants to announce its ownership of a resource file in THash, it has to compute the following three hash values and send them to the InPs defined by the P2P application:

$$\textit{Key1} = Hash(\textit{file-id}) \tag{1}$$

$$\textit{Key2} = Hash(\textit{file-id} + \textit{ASN}) \tag{2}$$

$$\textit{Key3} = Hash(\textit{file-id} + \textit{ASN} + \textit{PIDN}) \tag{3}$$

InPs should store and manage these three keys along with the ASN and PIDN of the PP, which can be retrieved from the PP's IP address.

Each particular DHT network has different method, mainly based on recursive search, to find the InPs storing the keys [7]–[9]. In THash, the keys for each resource are stored in three groups of InPs: one group for each hash key. Even if THash scheme triggers three separate recursive searches relative to each key for the InPs storing it, the complexity of the scheme remains in the same order as in native DHT, $O(\log N)$, moreover these searches are run in parallel. However, THash has to maintain 3 different keys (*Key1*, *Key2* and *Key3*, resulting in a tripled storage requirement in worst cases. Considering the current rapid reduction in the cost of the storage, we believe that the tradeoff between storage cost and benefit of THash is very favorable to its usage.

*2) Resource Searching in THash:* The above-described triple hash method for resource publishing in THash enables the P2P client to control the P2P traffic balance between different PID and AS domains by choosing the domain he want to download from by adding the ASN and PIDN in the key derivation. A RP selects its peers in three steps: first, the RP launch a search with *Key3* containing its own PIDN and ASN, obtaining peers in its own PID domain; in a second stage, the RP launches a set of parallel searches with *Key3* derived using the other PIDN given in the PGM. These queries return some extra peers from the same AS but in other PIDs. If the above procedures do not meet the intra-AS peer ratio indicated in the PGM, the RP launches a query with *Key2* only and obtain additional peers in the same AS. In the last stage, the RP uses *Key1* to retrieve a list of peers regardless of the AS and PID domains. The RP take from the lists resulting from each step a number of peers that depends on the PGM specified probabilities. As each InP maintains the corresponding PGM corresponding to the resources it manages, the RP only needs to tell in the query the total number of peers it needs and InPs can calculate how many peers they should return according to the instructions in the PGM. The above process is depicted in Fig. 1.

The complexity of the above searching process is $O(\log MN)$, where *N* is the number of peers in the DHT network and *M* is the number of PID domains of interest. However, it is expected that *M* is much smaller than *N*, so that the complexity remains the same order as that in native DHT, $O(\log N)$. This is confirmed on our PPLive traces where the value of *M* remains negligible compared to *N* (usually 4–5 orders of magnitude smaller than *N*).

*3) Peer Reselection in THash:* In a realistic setting, nodes are dynamic and change states. We need therefore a mechanism to prune poorly behaving peers with low upload capacities. This is done by asking in the peer searching process described in the previous section, slightly more nodes that really needed. The RP thereafter connects to a subset, validating the PGM constraint, of peers in the resulting list. Therefore in operation, a badly behaving peer can be replaced by another peer in the list still validating the PGM constraint. If there is no remaining candidate in peer list, the RP can launch another round of resource search similar to what described above to complete the list.

However, in some extreme cases, for example if the link between two domains breaks, the distribution of peers can deviate from PGM guidance. In such situations, the above peer reselection can solve the issue as the peers in the failed domain are replaced by peers in other domains following PGM constraints. This shows that THash can ensure that RPs always connect with PPs with satisfactory upload performance.

### C. Derivation and Distribution of PGM

Previous sections assumed the availability of the PGM without describing how to compute and distribute it. Here we will fill this gap by presenting how the PGM is obtained.

*1) Basic PGM:* As illustrated in Section II-B, the p4p-distance in the Cost Map expresses the costs or distances between peers in different domains from the network point of view. ISPs assign the p4p-distance in a variety of ways, for example deriving it from OSPF weights and BGP preferences or assigning it according to a financial cost or congestion status. In this paper we will assume that p4p-distance are derived following [6]. Yet our approach is agnostic of the particular p4p-distance algorithm and can be used directly with any approach. While p4p-distances are from the ISP perspective, we have to integrate the application perspective to compute the PGM. In particular, the total number of PPs in one domain should also be considered. The more copies of the resource of interest exist in one domain, the higher should the probability of the RP selecting peers in this domain. Let's denote the p4p-distance from PID *i* to PID *j* calculated by the ISP as $p_{ij}$ and the total number of the PPs for the resource of interest in domain *j* by $n_j$. Then a simple method to compute the PGM entry $w_{ij}$ is

$$w_{ij} = c_{i1} \cdot \frac{\frac{1}{p_{ij}}}{\sum_j \frac{1}{p_{ij}}} + c_{i2} \cdot \frac{n_j}{\sum_j n_j} \tag{4}$$

that is a weighted average by two weight coefficients $c_{i1}$ and $c_{i2}$, of the proportion of the PPs in domain *j* and the

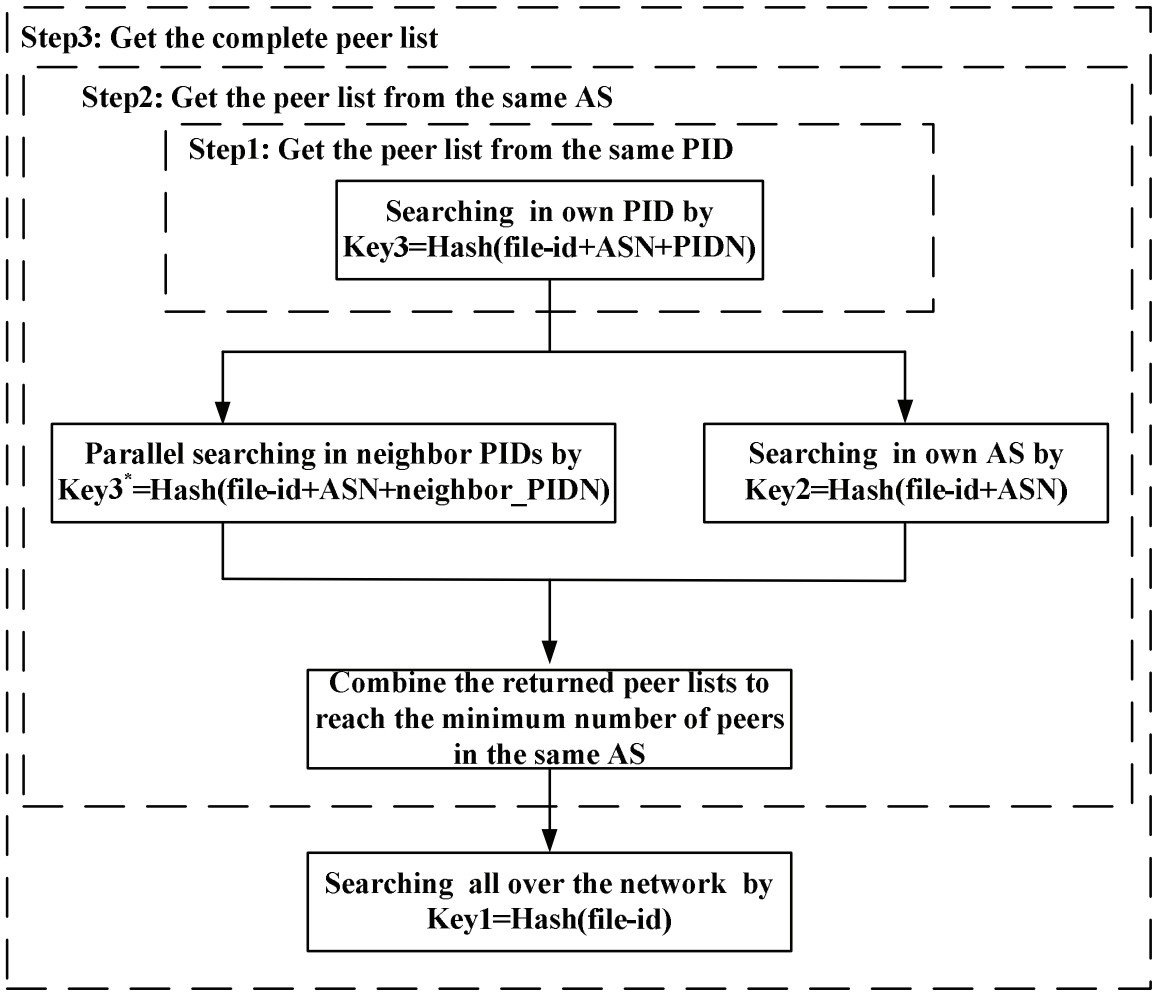


Fig. 1. Resource searching process for RP.

normalized inverse p4p-distance $p_{ij}$. The content of the $i^{th}$ row in the PGM is obtained by normalizing $w_{ij}$ (Eq. (5)).

$$a_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \tag{5}$$

The remaining parameters to determine are weighting coefficients $c_{i1}$ and $c_{i2}$. These can be assigned using different strategies, *e.g.*, the network operator can decide to statically assign these values according to its experience or policies; or the coefficient can be set dynamically according to the network conditions. We propose to set these values such that we give a higher weighting to the more informative elements. Let's assume that for each PID of an AS domain containing *k* PIDs, we define two $k$-vectors containing the distance $p_{ij}$ to other domains and the number of copies of the resource $n_j$. All of these values can be put in a matrix $B_i = (b^i_{mn})_{k\times 2}$. The coefficients relative to the informative value of each element are derived as:

$$c_{i1} = \frac{1-\overline{H}_{i1}}{2-\overline{H}_{i1}-\overline{H}_{i2}} \quad c_{i2} = \frac{1-\overline{H}_{i2}}{2-\overline{H}_{i1}-\overline{H}_{i2}} \tag{6}$$

where $\overline{H}_{ij}$ is the normalized entropy of the $j^{th}$ column of matrix $B_i$ relative to PID *i* that is obtained using the below formula:

$$\overline{H}_{ij} = -\frac{1}{\log k}\sum_{m=1}^{k}\frac{b^i_{mj}}{\sum_{l=1}^{k} b^i_{lj}}\log\frac{b^i_{mj}}{\sum_{l=1}^{k} b^i_{lj}} \tag{7}$$

In order to further enhance to performance and to optimize other network criterion, the above basic PGM computation can be easily extended to include other metrics, for example the aggregated upload capacity of different domains, *i.e.*, to make more likely connecting to peers in the PIDs which have larger upload bandwidth and are near PID *i*. However in a real network, the overhead of collecting in real time the available bandwidth of all DHT peers and to distribute it to other nodes is challenging. Meanwhile to the availability of such measurement service, we already proposed a peer reselection mechanism which ensures that a THash client only connects with peers with adequate upload bandwidth.

*2) Distribution of PGM:* As described above, the PGM is derived using the network costs provided by ISP server and the numbers of PPs in the different domains of the network. However, the last input is available only at the InPs. Therefore, one might ask the InP to compute the PGM. However, this is not really suitable as InPs can be unreliable both respect to availability and security. Frequently InPs are also P2P user nodes, thus we cannot assume that they will always be

available. Moreover, the p4p-distance, usually reflecting the ISP policies and topology, is rather sensitive and it will be desirable, from a security view, not to let common P2P users have this information. For these reasons, we propose to use an additional server named the Application Optimization Engine (AOE) to compute the PGM. The AOE is maintained by ISP itself or a trustful third party and collects network information from the ISP server (named iTracker) as well as the number of PPs from InPs to calculate the PGM.

The space and time complexity of PGM deployment can be easily derived. If THash is going to be largely deployed, each ISP would have to deploy its own AOE server, just like the P4P iTracker that each ISP has to deploy. The overall space complexity for PGM storage is *O(nm)*, where *n* is the number of the contents, and *m* the number of ISPs. It Means that the space is linear both in the number of content and the number of ISPs. Nonetheless, it will be hard to go beyond a *O(nm)* complexity as the logic of ISP optimization is to control each content independently and to define a PGM for each content. This means a storage space at least *O(n)* per ISP's AOE. As seen in Table I, the PGM only specifies the peering guidance for different PID domains within a single ISP. So each ISP's AOE has to only maintain its own PGMs. Therefore, the time for generating PGM does not depend on the number of ISPs.

Another challenge is that the DHT network may encompass a large number of InPs. The AOE can become a bottleneck if all these InPs interact directly with it for the computation of PGMs. In addition to this, synchronizing the content of the matrix among the different InPs for the same resource can be difficult.

In order to avoid making the AOE a bottleneck, we strictly limit in THash the number of InPs that can interact with the AOE. Recall that *Key1* is the hash value of the file-id of the resource. Therefore, InPs for *Key1* maintain the most complete publication information for the entire DHT network. We therefore restrict the communication with the AOE to only *Key1*-level InPs. This shortcut reduces strongly the communication overhead. More precisely let's suppose there are *n* ISPs in the network and in average each ISP has *m* PID domains. Then the proportion of *Key1*-level InPs to the overall number of InPs is given by

$$\frac{\textit{Key1 InPs}}{\text{total}} = \frac{1}{1+n+n*m} \tag{8}$$

Usually, *n* and *m* are relatively large (typically in order of $10^2$ to $10^3$) and the shortcut can therefore reduce highly the communication overhead. If a *Key1*-level InP finds that the distribution of a specific resource varies drastically (by monitoring the number of PPs in each AS and PID domain), it initiates a connection with the AOE and uploads the current distribution information of the resource to it. The AOE computes the new PGM according to the p4p-distance given by the iTracker and the new resource distribution information and replies with the updated PGM. In addition to the above InP-initiated updating process, the AOE can also itself update the PGM when it observes a significant variation of the ISP network state. In this case it will actively initiate connections with *Key1*-level InPs. Moreover, we assume that the PGM has a validity lifetime. This means that *Key1*-level InPs should query periodically the AOE to ensure PGMs' freshness.

While assessing the variation of the network information (say p4p-distance) is relatively easy using the information provided by the iTracker, deciding on the variation of the resource distribution is subtler. Several approaches can be used for this purpose. In this work we have used a simple technique that consist of comparing the $L_1$ norm of the difference between distribution of nodes among different domains at consecutive time and and to launch a PGM update if this $L_1$ distance exceeds a given threshold (5%).

The last remaining action is to for *Key1*-level InPs getting the latest PGM to transmit it to other InPs that manage the same resource. One can build on the update process that exists anyway in the current DHT-based P2P systems for synchronizing the peer list among the InPs, and take advantage from these opportunities to also exchange the PGMs among InPs maintaining the same key values. Another approach consists of asking *Key1*-level InPs receiving the matrix directly from AOE to share it with other *Key2* and *Key3*-level InPs storing the same resource, by deriving the *Key2* and *Key3* of the content they manage for all available PID and AS domains. With these keys, the *Key1*-level InPs can launch searching requests in the DHT network to find the relevant *Key2* and *Key3*-level InPs for these keys and transfer the new matrix to them.

### D. NAT Traversal

The mechanisms introduced above enable the clients to select appropriate peers following the guidance of the PGM. However, the performance of THash will be badly affected if there are not enough peers in the selected PID or AS domains to attain the PGM specification. Thus, it is desirable to increase the number of peers in each domain of the DHT network.

An effective way to increase the number of peers is to allow the peers behind the NATs [11] (called NATed peers) to join the system as individual nodes. Currently, NATed peers account for a large proportion of the peers in typical P2P applications. A recent investigation by PPLive showed that more than 50% of the system users were behind NATs. Therefore, including NATed peers as individual nodes into the DHT network will enhance the performance of THash.

But unfortunately, NAT devices block P2P streaming from peers that are behind different private networks. In order to overcome the problem, several NAT traversal methods have been proposed, *e.g.*, STUN [12] and TURN [13]. All these methods are based on the availability of a globally reachable server to help the NATed peers establishing connection. However, it is not appropriate to deploy a centralized server in a DHT network where all the peers are distributed and self-organized. One might say that iTracker and AOE are also centralized, however we should distinguish here between the different points of views of ISPs and application providers. The iTracker and AOE servers are deployed by the ISPs or trustful third party organizations. Due to security consideration, an ISP wants to have a full control of the information it provides to users and do not let the users to interact with it. Because of this an ISP will prefer a centralized scheme. But the NAT traversal servers should be maintained by the P2P

application providers. Due to cost and reliability consideration, in the DHT mode the P2P application providers are usually not inclined to deploy any centralized servers such as AppTracker and NAT traversal servers.

In this part, we will describe a NAT traversal method for the DHT network that does not need the support of a centralized server. Moreover, we were careful about the dynamic characteristic and the low processing capacity of the DHT client, and we particularly considered the overhead and reliability in our proposed mechanism.

*1) Bootstrapping:* The problem of bootstrapping in presence of NAT is not particular to our proposed scheme. We can reuse the solution used by the underlying DHT system that we are using. We are therefore assuming that before beginning communication over the DHT network, each new NATed node knows at least one public peer (IP address and port number) that has globally reachable IP addresses. These public peers act as mediator servers that are used to retrieve a group of 3 "relay" nodes that are ready to help in NAT traversal.

After joining the DHT network, a new node sends a bootstrap message to the mediator servers it knows. The mediators will send back to the node a list of other public peers (*<IP address, UDP port, Node ID>*) from its stored information (*e.g.* $k$-bucket in Kademlia). After collecting information of more public peers, the new NATed node selects 3 public peers from the lists received from the mediator servers to negotiate whether they can act as its relay. The reason for selecting 3 relays is to increase the reliability as the likelihood of 3 nodes simultaneously failing is very low. We have assumed that the NATed node use the PID and AS numbers of the potential relays to ensure that the node prefer a relay in the same PID domain to a relay in another PID but the same AS domain to a relay in another AS domain. The NATed node is exchanging periodically keep-alive messages with its relays. When a relay is about to quit the DHT network, it notifies its relayed peers so they can search for other substitutes.

*2) NAT Traversal Procedures:* Once the relays are found, the NATed peer connects to them and therefore punches a "hole" into the NAT that is used to communicate with other peers. There are two types of packets in the DHT network: control messages and data packets. Control messages are small sized packets that are used to publish and search and data packets are used to transfer contents mainly in form of long packet streams. We propose for each type of packets specific NAT traversal mechanisms.

All control messages used for P4P/DHT, like peer list update, PGM distribution, *etc.*, are transferred to the NATed peer by the relays through the hole punched in the NAT. Since the relays have globally reachable IP addresses, any node in the DHT network can communicate with them. As the control message is small sized and not frequent, the above relaying scheme is simple, fast and places an acceptable burden on the relays.

The above relaying mechanism is not suitable for NAT traversal of data, as the generated load on the relays can become prohibitive. Instead, we use another mechanism that consists for a node wanting to send data to a NATed peer, to first send a connection request to one of the NATed peer's relays containing its IP address and port number. The relay forwards the connection request to the NATed peer, that can upon received the relayed request, connects to the requesting peer by punching the NAT and sending subsequent data packets directly to the requesting peer. Indeed, the connection time for two endpoints is a little higher than that of the simple control messages relaying. However, the additional load on the relay is highly reduced by letting the two endpoints to exchange large amount of data directly without the help of the relay.

## IV. Performance Evaluation

In order to validate the proposed scheme and to assess its performance in realistic operations, we have deployed THash with the help of 3 major Chinese ISPs (China Telecom, China Unicom, CERNET) over a real operational P2P network, PPLive. We present here this deployment and the performance results achieved.

### A. Experimental Deployment

As stated above, we have conducted experimental deployment inside the P2P network of a major actor of P2P video streaming in China, PPLive. The P2P network was studied in the network of the three major Chinese ISPs (China Telecom, China Unicom, CERNET). These three operators, which cover more than 85% of the Chinese Internet users [14], have implemented an ALTO/P4P service that is accessible from an iTracker server based at Yale University. We had indeed during the experiment nodes that were not belonging to any of the above three ISPs. We all categorized these nodes into an "other networks" category while computing the PGM.

We selected the Kademlia protocol as the underlying DHT mechanism. We are referring in the forthcoming the original implementation of Kademlia as Native DHT. We added THash approach as described before into Kademlia. We have also implemented THash and Native DHT as plugins to the PPLive clients. PPLive released new version of their software to support our experiment. All users who had updated their clients joined our experiment. According to our statistic data, more than 28,000 nodes downloaded the updated version of PPLive and joined our experiment platform.

The experiment lasted for 6 hours from 16:00 to 22:00 on Jul 8th, 2011. We logically divided the experiment platform nodes into three DHT networks according to the node identification number (Node ID). The first DHT network (consisting of nodes with *Node ID % 3=0*) utilized Native DHT for resource publication and searching. The second DHT network (consisting of nodes with *Node ID % 3=1*) used THash scheme and the last one (consisting of nodes with *Node ID % 3=2*) also used THash but with NAT traversal.

PPLive assigned for our experiments five test channels over which five separate files were distributed. The sizes of these files were respectively 369MB, 222MB, 508MB, 235MB and 226MB. Users in these five channels used DHT approaches, either Native DHT or THash, for resource sharing without the help of PPLive trackers.

An AOE server was deployed by us and calculated the PGM matrices for the five resources. This server is shared by the different ISP networks in our test. The computation

details of basic PGM were as described in Section III-C, considering not only the p4p-distance but also the distribution of the resources. Upload capacity aware PGM is not tested as upload capacity were not available. We obtained the Cost Map from the ALTO/P4P server at Yale University [15] and obtained the resource distribution information from the *Key1* InPs in the network.

Every node in the DHT network maintained a simple and a complete log. The simple log recorded basic data such as the publishing and searching delay, the replied peer list for a resource and the amount of traffic exchanged with each peers. The complete log kept more detailed information such as debugging logs. A log server we implemented collected the performance statistics. During our test, we required every node to submit its simple log to our log server every 15 minutes. In this manner we were able to have the complete information of the resource shared by peers in the DHT networks.

### *B. Performance Metrics*

We considered the following performances metrics:

- *Publishing delay:* The delay between a PP sending a resource publication request and receiving the confirmation of its registration from the corresponding InPs. Usually, a PP records its publication information on a series of InPs.
- *Searching delay:* The delay between a RP sending a resource query and receiving the peer list from the corresponding InPs. Generally, the searching process is recursive, and for the THash parallel queries are run.
- *Intra-domain peer ratio:* The ratio of the number of intra-domain peers to the total number of the peers in the peer list. More specifically, this metric can either be intra-PID peer ratio or intra-AS peer ratio. This value should be increased for network efficiency.
- *Inter-domain traffic:* The total P2P traffic traversing different domains for transmitting the five resources in our test is another network efficiency metric that has great influences on the scalability of the system. More specifically, this metric can be measured either inter-PID traffic or inter-AS traffic.
- *Downloading performance speedup:* This metric measures the enhancement on the downloading performance resulting from using the THash scheme. It is defined as the ratio of the average downloading time of THash to the average downloading time of Native DHT for a specific resource.
- *Peer reselection ratio:* This metric measures the stability of the system and is defined as the ratio of number of peers updated to the total number of transmission peers in each periodic updating interval.
- *PGM update overhead:* We use PGM update frequency to estimate the overhead for PGM updating since the size of the PGM is very small.

### *C. Experimental Results*

Fig. 2(a) depicts the variation of the average publishing delay for THash and Native DHT. As can be seen, the publishing delay does not differ too much between the two schemes. As is shown in Fig. 2(b), the relative ratio of the publishing delay of the pure THash to the Native DHT varies between 88%$\sim$109%. Implementing NAT traversal causes a slight rise in the average publishing delay (about 15% in Fig. 2(b)). However, this delay is mainly due to the distributed NAT traversal process itself and not related to the THash scheme. Therefore, we can conclude that THash does not have a negative impact on this important DHT network performance metric. The variation in different hours is related to network load variations that affect the same way in THash and Native DHT.

Fig. 3(a) plots the variation of the average searching delay for THash and Native DHT. As expected, we observe that the use of THash causes an increase in the average delay for searching resources. The Fig. 3(b) shows that the relative ratio of the average searching delay of THash to Native DHT varies from 109% to 124%. The reason of this increase is that in THash a RP may send a series of parallel queries for searching in different domains simultaneously and this increases the total number of resource requests in the DHT network, consequently increasing message queuing and processing delays. Fortunately, in THash we do not need to wait for the arrival of all the reply messages before we can initialize the session and begin to download the data. Once the first reply message arrives, the RP can initiate the data transmission immediately with the peers contained in that reply message. Indeed the same is applicable to native DHT, but the difference is that in THash, we have to run several queries in parallel (searching by different level keys for the same resource), where in native DHT a single query (only searching by file-id) is run. Therefore, it is meaningful to compare the native DHT query delay with the delay of the first reply in THash. We show in Fig. 3(a) with the bars filled with cross lines the delay of the first reply for THash. It can be seen that this delay is even lower than the average delay for Native DHT. Overall, the relative ratio of the searching delay of the first reply in THash to Native DHT average varies from 82%$\sim$97% and it is stable over time. Finally, Fig. 3(a) and 3(b) show that the join of the NATed peers into the DHT network increases the average searching delay, because NAT traversal is mandatory when requesting a peer list from a NATed InP. However, the delay of the first reply in THash does not differ too much whether or not including NATed peers. The reason is that in THash a RP inquiries many InPs with different level keys simultaneously and it can always be firstly answered by the InPs with the public IP addresses.

Fig. 4 shows the proportion of intra-domain peers averaged over each one of the five resources for THash and Native DHT. We can clearly see that the network optimization introduced by THash remarkably limits inter-AS and inter-PID traffic. In average the proportion of intra-PID and intra-AS peer for THash are 25.9% and 29.2% higher than those of Native DHT. We have observed that in every 1 hour time interval, the THash scheme only generated in average respectively 52.6% and 32.3% of inter-PID and inter-AS traffic compared to Native DHT scheme. Over the six hours of test and only for these five resources, THash saved respectively 0.272 TB inter-PID traffic and 0.318 TB inter-AS traffic (see Fig. 5). In addition, Fig. 4 and Fig. 5 show that by including NATed peers into the

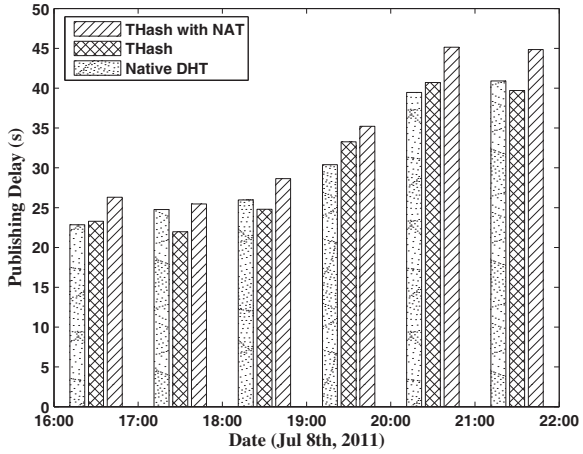


Fig. 2(a). Average publishing delay.

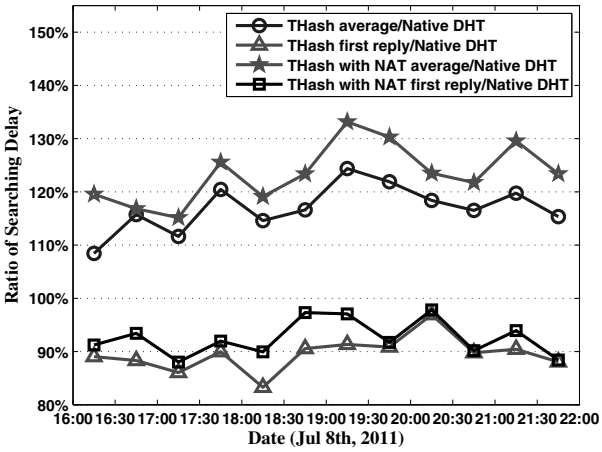


Fig. 3(b). Ratio of searching delay for different approaches.

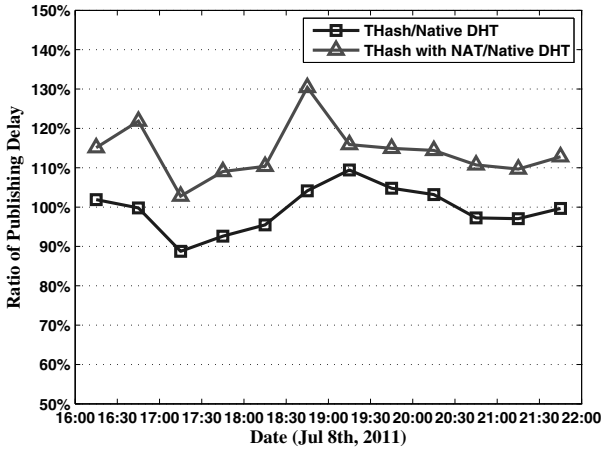


Fig. 2(b). Ratio of publishing delay for different approaches.

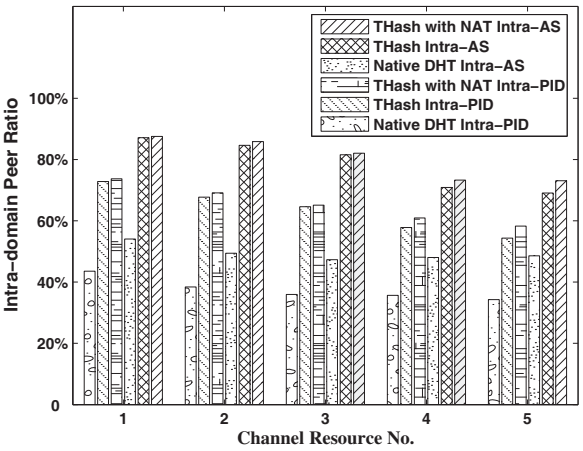


Fig. 4. Intra-domain peer ratio.

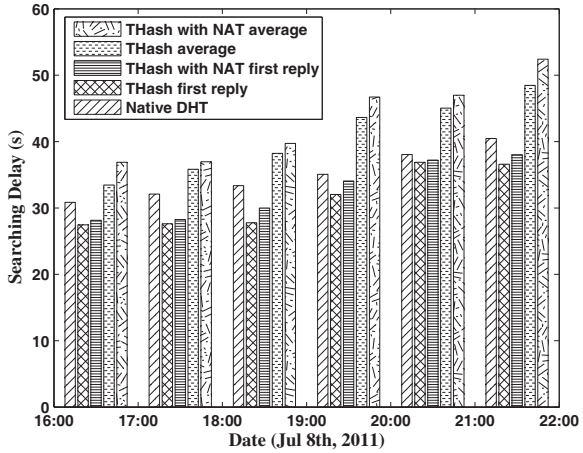


Fig. 3(a). Average searching delay.

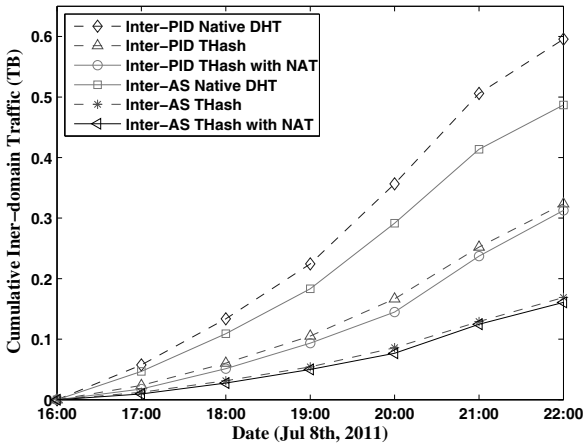


Fig. 5. Inter-domain traffic.

DHT system we can further increase the intra-domain peer ratio (2.0% for intra-PID and 1.7% for intra-AS to the pure THash) and reduce the inter-domain traffic (0.011TB inter-PID traffic and 0.008TB inter-AS traffic), thus improving the network efficiency.

Fig. 6 compares the downloading performance speedup of THash compared to Native DHT. What is noteworthy is that different resources have different distributions in the DHT

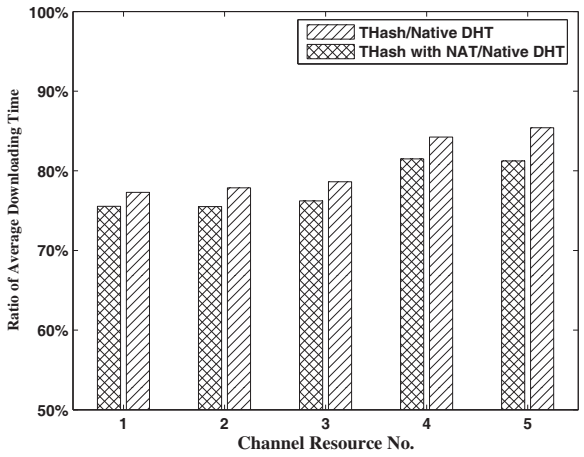


Fig. 6. Downloading performance speedup.

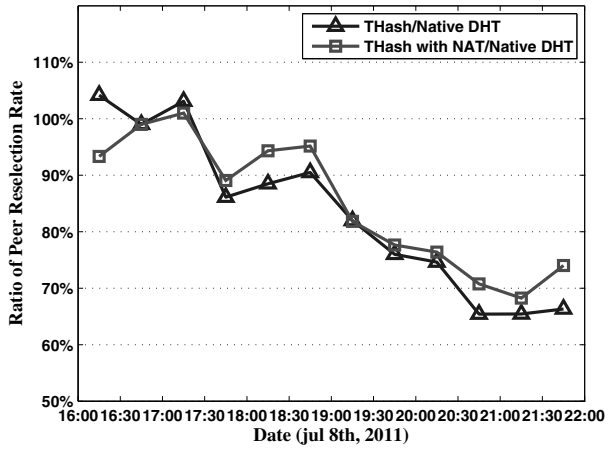


Fig. 7. Peer reselection rate (THash/Native DHT).

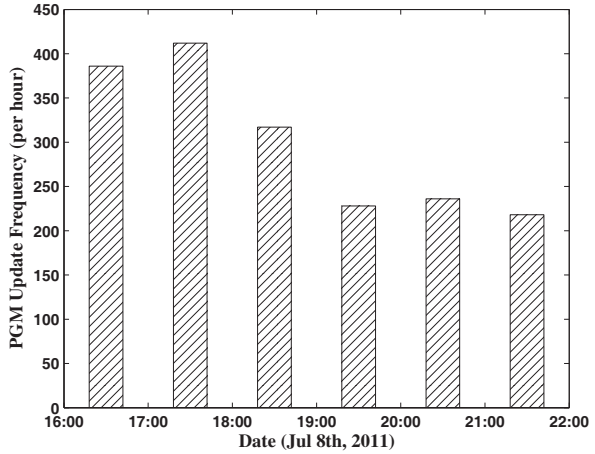


Fig. 8. PGM updating frequency (times in each hour).

network. Therefore, the PGMs for these resources are different and that results in diverse sharing peer behaviours in THash scheme. Fig. 4 showed that for Resource 1, 2 and 3 RPs are more likely to share resources with peers within the same domain, while for Resource 4 and 5 RPs connect to more peers outside. We see the effect of this fact in Fig. 6, where downloading performance enhanced by THash for resource 1, 2 and 3 are more obvious than that of 4 and 5. However, as seen from Fig. 6, even for Resource 4 and 5, the average downloading time with THash is reduced to at least 85.4% to that of Native DHT. Thereby, THash significantly improves the downloading performance of the applications. Including NATed peers into the DHT system efficiently increases the number of the candidate PPs in each domain for THash, thus the downloading performance can be improved. From Fig. 6, we have observed that the THash with NAT reduce 3.3% in downloading time by average over the five resources compared with the pure THash.

In order to guarantee the quality of service, PPLive client updates its peers periodically (peer reselection process); new peers replace bad performing peers with poor uploading capacities. We have compared in Fig. 7, the peer reselection in Native DHT to that in THash. It can be seen that for most of the time THash scheme need less peer reselection. By averaging the results over all intervals, we can observe that Native DHT generated 16.6% more peer reselections than pure THash and 14.9% more peer reselections than THash with NAT. This can be explained by the fact that peer selection in THash considers both link utilization in p4p-distance and load balancing in resource distribution information, thus the uploading performances of these peers are not badly impacted by the variations of the inter domain network conditions, resulting in higher stability and efficiency for systems.

PGM is a new concept introduced in this paper. Thus, the cost of updating and distributing PGM is a special overhead for THash scheme. Since the size of the PGM is fixed, this overhead can be estimated by looking at the PGM updating frequency. We recorded the total number of the PGM updating for the five resources at our AOE server in each hour period. The PGM updating was triggered using the $L_1$ distance described above. In our experiments we set the distance threshold to 0.1. In addition, lifetime of the PGM was set to 10 min. From Fig. 8 we have observed that the number of PGM updating decreases sharply, from about 400 in the first two hours to about 220 in the last three hours. This is caused by the fact that initially there are not many PPs in each domain. The distribution of resources is therefore very sensitive to the dynamics in the P2P system. However, with the experiment going on, sufficient copies for the resources appear in each domain. Therefore, a single peer's coming/leaving has little impact on the entire distribution of the resources and the resulting PGM becomes stable. Moreover, as the average size of a PGM in our test is only 3.89KBytes, when the system enters into a stable state, the overhead of updating and distributing the PGM is negligible compared with the data traffic transmission in the system (several TBytes of data). Another interesting outcome of Fig. 8 is that the frequency of the PGM updating is not related to the intensity of traffic in the system. For example, the PGM updating frequency remains very low even during the period 20:00∼22:00 when the system is heavily loaded with the highest average resource publishing and searching delays.

## V. Discussion

In this section, we will present some noteworthy points and discussions about the THash scheme. First, one may ask, what are the incentives for ISPs, application providers and clients to utilize THash? Obviously, ISPs can benefit from THash due to its ability to control the inter-domain traffic and save the valuable peering network bandwidth. For P2P application providers, the utilization of THash significantly enhances the performance of their DHT systems, avoiding the cost of deploying centralized indexing servers and improving the robustness and reliability of their application systems. Moreover, THash can help in reducing the controversy between content providers and network operators by making them to cooperate with each other as the PGM in THash not only considers the p4p-distance from ISP perspective but also resource distribution from application perspective. Finally, for end users (P2P clients), a better traffic balance can greatly improve the resource downloading performance.

A major issue for operational deployment of distributed applications and in particular P2P distribution platforms is to ensure the robustness of the system to variations and changes. The robustness of a DHT system relies on two elements: the multiplicity of the InPs and the randomness of the peer selection. THash helps to improve the robustness, as: 1) The triple hash method in THash increases the number of the InPs storing the same resource, thus reducing the probability of the system collapse due to the failure of a small number of InPs. 2) As described in Section III-B, the peer selection in THash requires a RP to maintain a number of connections for different domains and even for different ISP networks. Therefore, the system collapse caused by the failure of a single domain or network is avoided. The THash scheme is dependent on two other services that can become point of failures: the iTracker that returns the network state and the AOE that calculates the PGMs. However if any one of these fail, the THash system have to use non-optimized PGM, but will still be operational and will have the same performance as a native DHT scheme. Indeed, one can make the system more resilient by duplicating the iTracker and the AOE.

The scalability is indeed an important issue for any distributed application and a fortiori for P2P applications. The scalability of THash is mandatory if one wishes to extend its usage to large-scale P2P networks. The scalability of THash is mainly based on the scalability of the underlying DHT system. This means that the underlying DHT system should ensure to have enough InPs to service the entire query load. Indeed THash assumes that the hash storage is tripled, as three hashes should be stored. However, the linear increase of storage and number of InPs does not fundamentally change the scalability properties as we can expect to have a large number of peers (potential InPs) in a P2P network and the storage requirement for content hashes is relatively low compared to the content itself. Another source of scalability problem is relative to the iTracker load. In our implementation, we used a single ALTO/P4P server at Yale. However, in a real operational deployment, we can expect that each ISP operator willing to optimize the P2P traffic in his network will implement or contribute to one or several ALTO/P4P server (based on the load resulting from the peers in their respective network) and as explained above, if an ISP decide not to deploy such a server or if its server become unavailable, the DHT system fall down to a non-optimized operation. The AOE can also have an impact on scalability. The P2P platform operator and ISP that jointly maintains AOEs should ensure that there are enough Application Optimization Engines to service all requests.

Another point to discuss is relative to our experimental setting where only three ISPs China Telecom, China Unicom and CERNET are used. Users in other ISPs have not optimized PGMs and still connect randomly to peers resulting in higher inter-domain traffic and lower downloading speed. Although we already had significant performance gains shown in Fig. 4 to Fig. 6, the improvement of system performance would be better if peers in other ISPs were also using PGM to guide their activities.

Last but not least, the proposed Triple Hash scheme can be easily extended to other types of network information sources, as long as the NIS provides P4P cost information that are compatible with ALTO/P4P. As described in Section III-C, THash itself is not dependent on any particular derivation of the p4p-distance. This means that the ISP can use any type of method and policies to divide its ISP network and derive the p4p-distances between different domains. Moreover, the interfaces between the THash entities and the NIS can be standardized, independent with the specific form of the information source. Thereby, the THash scheme is a general design that is not only limited to the framework of ALTO/P4P.

## VI. Related Work

**ISP Approaches:** In [16]–[18] it is proposed to ISPs to deploy P2P caching devices to reduce the burden of P2P applications over inter-domain links. These devices, often deployed at the ISP edges, are combined with redirection techniques that transfer resource requests from peers inside the ISP to the caches in the same domain. P2P caching can significantly reduce inter-domain P2P traffic and enhance downloading performance of the peers. However, P2P caches need to be designed for specific P2P applications, which limit their generality. In addition, ISPs may not want to bear the costs of caches. Furthermore, caching contents may have legal liability implication that the ISP wants to avoid.

Another widely used ISP approach consists of reducing the bandwidth cost of P2P applications by applying traffic shaping [19], [20]. Traffic shaping devices identify P2P traffic through DPI (Deep Packet Inspection) or DFI (Deep Flow Inspection) technologies and then impose bandwidth constraint on them. The disadvantage of this approach is that it degrades highly the end-to-end performance of P2P [20]. This has led to major criticisms by users and has an important impact on the decision of customers to change their ISP. In addition, P2P traffics are becoming more and more difficult to identify because of the use of encryption and dynamic ports.

**P2P Approaches:** P2P application providers also devoted themselves to reduce the burden of P2P traffic on the ISP's network and improve the downloading performance of their applications. Most of these approaches have focused on centralized P2P applications. The basic idea of these approaches [3], [4]

is that a centralized server, the AppTracker, prioritize nearby nodes to share the resource with the requesting node. Indeed, this optimization improves network efficiency. However, the main disadvantage of this approach is that it relies on the application being able to probe the network and infer various types of network information such as topology, congestion status, cost, and policies. This inference is challenging and their accuracy is questionable. There also exist some decentralized solutions for network optimization. [21] proposes a distributed primal-dual algorithm to maximize the aggregate utility in P2P systems. The scalability of the approach is limited by the massive data exchanges between all peers. [22] introduces a fully distributed flow control scheme, Implicit-Primal-Dual, for peer-to-peer systems. However, it aims at P2P live streaming and our THash scheme mainly focuses on file downloading (VoD). Moreover, the Implicit-Primal-Dual scheme is designed to achieve the optimal traffic cost over the whole ISP network, whereas THash pays more attention only on reducing the traffic over the inter-domain links. Ono [3] and Vuze [23] are also using DHT components. Specifically, Ono uses CDN redirection information to approximate network information; Vuze DHT is based on a network coordinate system. But this information is mainly driven by latency or CDN server status, without consideration of other network information such as congestion status, cost and routing policy. In contrast, THash is a generic network optimization scheme for DHT systems that uses ALTO/P4P standard for network information source.

**Similar Experiments:** It is noteworthy that in 2010 China Telecom launched a large-scale experiment [24] similar to our experiment using ALTO/P4P but with Thunder P2P application. The main difference between their test and ours is that they estimated the tracker-based application performance and we focused on DHT system. In addition, the PGM introduced in this paper not only considers the p4p-distance from ISP perspective but also resource distribution from application perspective. Therefore, both the locality of the traffic and the user downloading speed has been improved by using the THash scheme.

More recently [25] confirmed that Bittorrent based P2P systems are moving from centralized tracker-based solutions to DHT-based solution and proposed a localization mechanism that intercepts all content announcements of peers and all requests for contents and answers to RPs a local peer set. This paper has similar goal to THash but does not account for ISP constraint. Moreover, the interception of all content announcements assumes a central position in the network that is not always guaranteed.

In [26] the authors proposed a relay selection mechanism for NAT traversal named Gradual Proximity Algorithm that uses a triple hash mechanism similar to our peer selection mechanism. However, the choice is not guided by a metric depending on operator and network constraints as well as content distribution as the PGM. Moreover, the proposed solution in [25], [26] is particular to specific purpose (relay selection for [26]) or application (Bittorrent for [25]), while our goal was to design a generic scheme applicable for all DHT-based P2P network.

## VII. Conclusion and Future Work

In this paper, we presented THash, a practical scheme based on standard DHT put/get semantics that implements an effective network optimization for DHT systems. THash uses a triple hash method to guide the peering decision of DHT clients under the instructions of a Peering Guidance Matrix (PGM). We also described a NAT traversal scheme that enables to increase the number of nodes participating in the P2P forwarding, thus further improving the performance of THash. We deployed our proposed schemes over the PPLive network and did a large-scale experiment with more than 28000 users. Our large-scale experiment results showed that THash can control the amount of inter-domain traffic while decreasing the resource downloading time. It therefore provides a solution for controlling the load of DHT-based P2P platform over ISP networks, opening the way for a larger deployment of these systems for content diffusion.

In future work, we shall investigate a light-weight method for collecting and aggregating in real time the upload bandwidth of peers in each domain and use this information in the derivation of the PGM as suggested in Section III-C-2. The main aim of this paper was to propose an architecture, to introduce the use of the PGM and to validate its interest by a particular choice. We introduced two types of PGM calculation; one including the upload capacity in the derivation. It will be interesting to compare different ways of deriving the PGM. We plan to do in the future a comparative analysis of different PGMs for different optimization objectives.

## Acknowledgements

An earlier version of this paper has been presented at the IEEE INFOCOM 2012 conference. The authors wish to thanks R. Alimi, Y. Wang, H. Liu from Yale University and G. Yang, H. Wu, P. Yang, Y. Ge from Institute of Computing Technology for their contributions to the design and implementation of THash.

## References

[1] Z. Shen, J. Luo, R. Zimmermann, and A. V. Vasilakos, “Peer-to-peer media streaming: Insights and new developments,” in *Proc. IEEE*, vol. 99, no. 12, pp. 2089–2109, 2011.

[2] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, and A. Krishnamurthy, “iPlane: An information plane for distributed services,” in *Proc. USENIX Symp. Operating Syst. Design Implementation (OSDI)*, Nov. 2006.

[3] D. R. Choffnes and F. E. Bustamante, “Taming the torrent: A practical approach to reducing cross-ISP traffic in peer-to-peer systems,” in *Proc. CCR*, 2008, vol. 38, no. 4.

[4] S. Tang, H. Wang, and P. V. Mieghem, ”The effect of peer selection with hopcount or delay constraint on peer-to-peer networking,” *Springer-Lecture Notes Computer Sci.*, vol. 4982, 2008.

[5] R. Alimi, R. Penno, and Y. Yang, ”ALTO protocol,” IETF Draft, June 2011.

[6] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, ”P4P: Provider portal for application,” in *Proc. ACM SIGCOMM*, Aug. 2008.

[7] P. Maymounkov and D. Mazieres, ”Kademlia: A peer-to-peer information system based on the XOR metric,” *Lecture Notes Comput. Sci.*, 2002, pp. 53–65.

[8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, ”Chord: A scalable peer-to-peer lookup service for Internet applications,” in *Proc. ACM SIGCOMM*, Aug. 2001.

[9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, ”A scalable content-addressable network,” in *Proc. ACM SIGCOMM*, Aug. 2001.

[10] Y. Huang, T. Fu, D. Chiu, J. Liu, and C. Huang, "Challenges, design and analysis of a large-scale P2P-VoD system," in *Proc. ACM SIGCOMM*, Aug. 2008.
[11] P. Srisuresh and K. Egevang, "Traditional IP network address translator (traditional NAT)," *IETF RFC 3022*, Jan. 2001.
[12] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session traversal utilities for NAT (STUN), *IETF RFC 5389*, Oct. 2008.
[13] R. Mahy, P. Matthews, and J. Rosenherg, "Trasversal using relays around NAT (TURN): Relay extensions to session trasversal utilities for NAT (STUN)," *IETF RFC 5766*, Apr. 2010
[14] "Development of Chinese Internet," [Online]. Available: http://www.c114.net/news/46/a636267.html
[15] "P4P maps," [Online]. Available: http://p4p.cs.yale.edu/files/doc/latest/p4p-common-cpp/html
[16] M. Hefeeda and B. Noorizadeh, "On the benefits of cooperative proxy caching for peer-to-peer traffic, *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no.7, pp. 988–1010, 2010.
[17] J. Dai, B. Li, F. Liu, B. Li, and H. Jin, "On the efficiency of collaborative caching in ISP-aware P2P networks," in *Proc. IEEE INFOCOM*, Apr. 2011.
[18] E. Rosensweig, J. Kurose, and D. Towsley, "Approximate models for general cache networks, in *Proc. IEEE INFOCOM*, Mar. 2010.
[19] C. Wang, N. Wang, M. Howarth, and G. Pavlou, "A dynamic peer-to-peer traffic limiting policy for ISP networks," in *Proc. IEEE NOMS*, Apr. 2010.
[20] M. Marcon, M. Dischinger, K. P. Gummadi, and A. Vahdat, "The local and global effects of traffic shaping in the Internet," in *Proc. ACM SIGCOMM*, Aug. 2008.
[21] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer systems," in *Proc. ACM SIGMETRICS*, June 2008.
[22] D. Tomozei and L. Massoulie, "Flow control for cost-efficient peer-to-peer streaming," in *Proc. IEEE INFOCOM*, Mar. 2010
[23] "Vuze," [Online]. Available: http://www.vuze.com
[24] K. Lee and G. Jian, "ALTO and DECADE service trial within China telecom," IETF Draft, Apr. 2011.
[25] M. Varvello and M. Steiner, "Traffic localization for DHT-based bittorrent networks," *IFIP Netw.* 2011.
[26] R. Cuevas, A. Cuevas, A. Cabellos-Aparicio, L. Jakab, and C. Guerrero, "A collaborative P2P scheme for NAT traversal server discovery based on topological information," *Comput. Netw.*, Aug. 2010.

**Yi Sun** has been an associate professor at the Institute of Computing Technology (ICT) since 2009, and is also an adjunct associate professor at Macquarie University, Australia. His research interests include network resource management, mobile computing, and flow distribution in heterogeneous wireless networks. Since 2007, his research has focused on future network design, including service oriented routing and traffic optimization. He has published more than 50 academic papers.

**Y. Richard Yang** is an associate professor at Yale University. His general research interests include computer networks, wireless networks, sensor networks, mobile computing, and network security. He leads the Laboratory of Networked Systems (LANS) at Yale University. Currently, his primary research interest is on designing robust, efficient, and fair computer networks.

**Xiaobing Zhang** is a co-founder of PPLive/PPTV. Before joining PPLive, he was a lecturer at Huazhong University of Science and Technology. Since 2005, he has worked for PPLive and focused on P2P architecture design and protocol implementation. He has a considerable amount of experience in live streaming, video on demand, CDN, DRM, etc. Previously, he was a member of the IETF ALTO Working Group.

**Yang Guo** is a Ph.D. candidate at the Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS). His research interests include Internet architecture and mobile Internet.

**Jun Li** is an associate professor of the Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS). His recent research interests include service-oriented networking, content service communication, and content right permission. He received the Ph.D. degree in 2006. From 2007 to 2009, he was a Post Doctor of Beijing University of Posts and Telecommunications.

**Kave Salamatian** is a professor at the University of Savoie. His main areas of research are Internet measurement, and modeling and networking information theory. He was previously a reader at Lancaster University, UK, and an associate professor at the University Pierre et Marie Curie. In 2011, he was a visiting professor at the Chinese Academy of Sciences.