# Analysis of the Service Discovery in DHT network

Jiang XingFeng
jiang.x.f@huawei.com

# 1.Introduction

The essence of the P2P technology is peers serving each other. Peers in the overlay may be heterogenous in their capabilities which could be used to serve other peers in the same overlay. For example, some peers may provide TURN service for other peers which are behind a NAT or a firewall. So in order to make the overlay function properly or provide upper layer applications with abundant services, it is important to discover which peers provide which kind of service. Upper layer applications could get needed services and make up more advanced functions by assembling the service provided by peers in the overlay.

The article will identify the requirements for the service discovery in DHT network at first. Then it will compare a few existing solutions and in the end gives an improved solution. Here, "service" refers to the capabilities of the peer. After the peers who could provide some service are discovered, the service requesters often send service requests directly to the peers.

This article also focuses on using the overlay to discover the service providers. Service discovery using rendezvous mechanisms is out of scope.

# 2.Requirements for the Service Discovery

Firstly, in P2P network, every peer may provide a specific service. So for some service, the service peers might account for 90% of a overlay, but for others, the ratio might be under 1%. DHT algorithms store the value on the root peer based on the key. So if there is only a single key for the specific service, all values will be stored on the single root peer. Moreover, all the service requests will go to the single root peer. It may overburden the root peer with much data and much traffic. Even with special techniques for load-sharing, the burden will fall on a small number of peers.

Secondly, a service requester often gets several candidate service peers after applying service discovery methods. The service requester often wants to choose the best one in terms of the service. It is better for the service peers to distribute their associated status with their location information to facilitate the service requester's choice. The possible status might be upstream bandwidth, processing power, etc.

Finally, the information about service status also changes because the peer may leave the overlay, the peer may not provide that specific service any more or the service may be down. So the service peers had better update the state changes as soon as possible, since the information may be inaccurate for the moment until this update happens.

So through the above analysis, a few requirements are proposed as follows:
o The service discovery mechanism MUST succeed in discovering service peers within reasonable time period.

o The service discovery mechanism MUST balance traffic to the root peer and MUST NOT store too much data on the root peer( or on a small number of replicas );
o The service discovery mechanism SHOULD allow the service peers to advertise status information other than their location information;
o The service discovery mechanism SHOULD provide the service advertisement maintenance with low cost.

# 3. Comparison of Candidate Solutions

There are a few solutions to service discovery in P2P literature and in IETF P2PSIP WG. In this section, some candidate solutions from IETF drafts ([P2PP] [RELOAD-3] [SEP]) are to be discussed here.

3.1 Standard Service Name (P2PP)

The method standardizes a service name for the specific service. For example, turn.example.overlay is chosen for describing TURN service in example.overlay. Every service peer providing the service will get the key by applying hash function to the service name and put its location and associated status information in the overlay. So when a peer wants to discover service peers, it derives the key first and sends the GET request to the overlay for the key.

The method is very simple and works well in the situation where the number of the service peers is small. But under the case where there are many service peers, the root peer will receive too much service request traffic and have to store too much data about service information.

3.2 Service Density (RELOAD-3)

This method will estimate the percentage of the service peer in the all peers of the overlay at first, then compute the how many times the service peers will store itself in the overlay. We call the computation result as service density parameter. Often the density parameter is the reciprocal of the percentage. The service peer will get a set of random IDs the number of which is service density and put the service information in the overlay under the set of random IDs. The service requesters will choose a random ID and sends a GET request for the value under the random ID. The service requesters will get the above GET operations repeated until the service peers is found.

The method could balance the service request traffic very well. But the success rate of the method depends highly on the accuracy of the estimate of the service density. Accurate estimates are difficult if no peer has full knowledge about the overlay and the overlay status is changing all the time. On the other hand, if the service density is high, it means a peer should keep a high number of connections with the overlay to keep its information alive. If something changes, the peer should update data stored under all associated random IDs.

3.3 Random Walk(SEP)

The method tries to extend the DHT algorithm to advertise peers' service capability. For example, if a peer could provide TURN service, it will tell its service capability through the overlay maintenance mechanism. When a service requester wants to discover service peers, it will choose a random ID and send a request to ask the intermediate peers and root peer to check their routing states to discover service peers. The more peers the service request traverses, the higher probability the service peers is got successfully.

This method advertises peers' service capability and also updates the service information at a very low cost. It also balances the service request traffic because every intermediate peer may give the response if it could meet the service request. But it may not work well in the case where the percentage of service peers is too small.

# 4. Considerations for Improved Solution

The solutions in section 3 have their respective advantages and disadvantages. We need a general method to incorporate advantages from every solution. So here, we consider whether could combine ideas discussed in the section 3 to work a better method out.

Discovery method making use of every peer's routing states could balance the traffic, but the big problem is it won't work well where the number of service peers is few. The standard service name method determines the storage location of the service peers information. For the service requester, it will get the service peers for certain in a GET operations. Using a set of well known random ID could also balance traffic and storage.

So if we combine the above advantages, we could get a new solution. First, the overlay will choose a set of random ID for a service. The number of the set should not be specific to the service. It is at the overlay level. So, every service peer not only distributes its service capabilities by extending overlay maintenance mechanism, but also put the information in the overlay under the set of random ID. Second, the service requesters will send a service request to ask the intermediate peers and root peer to discover service peers. Here, the key for the request will be chosen from the set of the well known random ID. So even if the request could not be met by the intermediate peers, the root peer could check its resource table to get the service peers.

# 5. Summary

There are a few existing solutions to the service discovery issues and they have their respective advantages and disadvantages. A new solution trying to borrow advantages from existing solutions is also discussed. The future work should be done to test different solutions and compare them.

# Reference

[RELOAD-3] C. Jennings, B. Lowekamp, E. Rescorla, J. Rosenberg, S. Baset, H. Schulzrinne, "REsource LOcation And Discovery
(RELOAD)" draft-bryan-p2psip-reload-03 (work in progress ) February 24, 2008
[P2PP]S. Baset, H. Schulzrinne, M. Matuszewski, "Peer-to-Peer Protocol (P2PP)
"draft-baset-p2psip-p2pp-01(work in progress)
November 19, 2007
[SEP] X. Jiang, H. Zheng, C. Macian, V. Pascual, "Service Extensible P2P Peer Protocol"
draft-jiang-p2psip-sep-01(work in progress)