P2PSIP WG
Internet-Draft
Intended status: Informational
Expires: August 27, 2008

V. Pascual
Pompeu Fabra University
M. Matuszewski
Nokia
E. Shim
Locus Telecommunications
H. Zheng
Y. Song
Huawei Technologies
February 24, 2008

P2PSIP Clients
draft-pascual-p2psip-clients-01

Status of this Memo

Abstract

This document describes why and when some devices would better be a Client rather than a Peer. The purpose of this document is to facilitate the discussion and understanding about the Client node type.

Table of Contents

1. Introduction

The P2PSIP Client node type was proposed and introduced quite a while ago. Many drafts mention the concept of Clients [3] [4] [5] [6] [7] [8]. Nevertheless, there is some confusion about its concept or definition or the benefit of having it yet in the group. The document elaborates on the concept of the role of Clients to facilitate the discussion and understanding about the Client node type.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The other concepts used in this document are compatible with RFC3261 [2] and the concept draft [3]

3. Definition of P2PSIP Client

Typical DHT-based P2P overlay networks (hereafter overlay) need the following fundamental functions.

- Bootstrapping (Letting a new node find some of the existing nodes in the overlay and facilitating the new node to join the overlay)
- Overlay maintenance (nodes in the overlay maintain information about other nodes and a routing table)
- Routing (nodes in the overlay route messages to other nodes in the overlay; the messages may be for joining the overlay, storing data(PUT), searching data (GET), and so on.)
- Storage (nodes store resource (user) records that contain information about resources and users, for example, the location information of a resource. Each user and resource has assigned an identifier that is used to locate user's or resource's records in the overlay. The user's and resource's records stored in a particular node have assigned identifiers picked from the same identifier space as the node identifier(s).)

Among these four functions, the Peers must support at least the overlay maintenance, routing, and storage functions. Whether a device provides any other function in addition to these three functions is irrelevant to being a Peer or not.

Then what is a Client? It is a device that uses the overlay services but does not perform overlay maintenance and routing. In this sense,

a possible analogy is that a Peer is a routing node and a Client a non-routing node. Again whether a device provides any other services is irrelevant to being a Client. A client uses the overlay services through its associated peer; it can be associated with more than one peer simultaneously to enhance redundancy.

Being a Client or a Peer is a matter in the overlay layer. It is independent of what a device does in a different layer or different context. For example, a device being a SIP UA or Proxy is completely independent of being a Client or a Peer. As a Peer may not be coupled with any SIP entity, a Client may not be coupled with any SIP entity.

Declaring a device as a Peer or a Client makes the role of the device significantly different from the overlay perspective. A device must define its role in the overlay layer, i.e., whether to be a Peer or a Client, explicitly and this should be recognized by other nodes clearly. Otherwise, the DHT or the overlay operation may be messed up.

3.1. Differences between a Client device and a device with SIP UA connected to a SIP Proxy on a Peer

First of all, please note that SIP UA and Client belong to completely different layers. SIP UA is an entity in the SIP 'layer' and Client is a node type in the overlay 'layer'. One device may work only as a SIP UA entity whereas another may work only as a Client device that do not support SIP protocol.

A device running a SIP UA entity may be associated with a SIP Proxy running on a Peer. This device is quite similar to a Client device in that sense that it uses the service of the overlay without being a Peer. But there are a number of differences.

First, the device with just SIP UA entity is not aware of the overlay or supports the Peer or Client protocol. The Client device is aware of the overlay and contains implementation of the Client Protocol. It may contain the implementation of the Peer Protocol as well (in particular, if the Client Protocol is a subset of the Peer Protocol). Then a Client device may turn into a Peer if its situation changes. For example, a device may choose to be a Client because its network connection is wireless or unstable. When its network connection becomes wired or stable, it can become a Peer. Even if the Client Protocol is something quite different from the Peer Protocol, it is anticipated that the two Protocols are going to be often packaged together so that a device may choose to be a Client or a Peer depending on its hardware resource or network condition, or uptime.

Second, if we allow only SIP UA on a device that is not a Peer take advantage of the overlay, applications running over different application protocols in the same device will not be able to use the P2PSIP overlay. Let say conference announcements are made as data in the overlay. Since the interaction between SIP UA and Proxy is not for generic data search, a conferencing application running on a device with just SIP UA entity cannot get conference announcements. There is no such limitation with a Client device. Any application on a Client device can use the overlay. When someone says we don't need Clients because we can have Peers with SIP Proxy and SIP UA can connect to the SIP Proxies, she/he is thinking about only what a SIP UA can do via SIP Proxy. Clients can enable more than that and increase the value of the P2PSIP overlay significantly.

Third, when a device with SIP UA is connected to a Proxy on a Peer, the SIP messages from/to the SIP UA should go through the SIP Proxy on the Peer. This may not be something the user of the device likes since the SIP Proxy on a not-so-trusted Peer may modify the SIP messages. On the other hand, a SIP application running in a device that acts as a Client, may use the overlay just for search of data (eg. Location of the callee) and communicates with the SIP UA at the callee's device directly for session establishment. The callee location data is generated by the callee and may be digitally signed easily by the callee device. So getting just the location data through an associated Peer has much less security threat than passing INVITE messages through a SIP Proxy on a Peer.

Fourth, devices with just SIP UA need Peers with SIP Proxy entity. In another word, many if not most of the Peers must be coupled with a SIP Proxy entity if UA-Proxy relation is the only way non-Peer device can use the overlay for session establishment. If the Client type is allowed, there is no such need.

4. When a device should be a Client

In general, it is better for the overlay when more 'qualified' devices become Peers rather than Clients. So it is an issue whether a device should become a Client or a Peer.

If the main service of the particular P2P overlay network is to share the bandwidth and storage (eg. for file sharing) and thus the resource abundance is more important than search/lookup speed, definitely all of the devices should become peers whenever possible.

On the other hand, in real-time applications like P2PSIP, the search performance would be at least as important as the storage capacity. Concerning the search performance, the smaller number of Peers is the

faster the lookup can potentially be. Therefore a larger number of Peers is not necessarily beneficial. Of course the decision about what devices should become Peers must be made carefully. We have to remember that the smaller number of Peers is, the bigger impact of a particular peer on the P2P overlay network performance can be. There are situations where it is better for a device to be a Client rather than a Peer.

First, some devices have unstable network connection or they churn frequently. If a Peer churns frequently, it generates overhead to its neighbors for resynchronization of the routing tables and transfer of resource (user) records. If the device does abrupt churning, the data stored in the device become unreachable. To cover this, the overlay should increase the number of replicas.

Second, a device can be behind a very strict firewall/NAT that makes it almost impossible for the device to operate as a Peer. If a device is behind the symmetric NAT, then it is very hard for it to setup direct connections with its neighbors. In this situation the device would have to use relay servers when routing messages to its neighbors. If many devices of this kind become peers, then the overlay need many relays and the efficiency of the overlay becomes very low.

Third, a device may have insufficient resources to support Peer operation. Low-end mobile devices may have a little memory, a slow CPU and may not support fast packet radio interfaces.

Forth, a device does not support the routing algorithm used by the overlay. This is mentioned in [9]. The current direction of the P2PSIP protocol design is to support pluggable DHT and it is likely that overlays have choices of DHT algorithms to use. And a device may not support the particular DHT algorithm used by the overlay the device wants to join. Then such a device will have to be a Client even if it has hardware resources and network conditions good enough to be a Peer. This situation would not happen often with devices which can download and install software easily. But small devices with embedded software may be put into in this situation.

There are also other reasons why a device should not become a peer:

Even if it is possible to traverse NATs and firewalls, every additional connection that has to be maintained with other peers in the overlay will cause the battery of battery powered devices to drain faster. If a device acts as a Client, it needs to maintain only one connection with the overlay (a connection with a peer). Besides the power consumption is impacted by maintenance of the routing state and by routing incoming messages. As measurements

shown these operations have huge impact on how long a battery-powered device can stay online without recharging. Nor the device user, nor the overlay network operator benefits by enforcing the mobile devices to be Peers and their battery to drain quickly. Such policy will push away mobile device users from the overlay.

Operating on battery does not mean necessarily that the device should be or wants to be a Client. For example, mobile devices in an emergency situation running in an ad-hoc fashion might have to be Peers since there is no other kind of devices to be Peers. Besides the battery consumption caused by maintenance of the routing state and by routing incoming messages can be reasonable in small overlays.

In mobile communication using P2PSIP, the consumptions of rare mobile bandwidth in the access networks due to P2PSIP traffic will be significantly reduced if mobile devices become more Clients and less P2PSIP peers as possible (assuming there are enough non-mobile devices being Peers). Besides we have to remember that bandwidth is expensive. If someone has to pay for every packet her/his device exchanges, probably having the device be a Peer makes no economical sense to the her/him especially that the overlay maintenance and routing of incoming messages are not directly linked to the service usage.

A device may not meet the requirements of Peers for the particular overlay desired by the overlay operator. For example, the OpenDHT is built with nodes on PlanetLab. Any devices that use the OpenDHT become a kind of Client. This was the choice of the people that designed and built the OpenDHT. Probably they needed access and control to the nodes to participate in the DHT that is not available in general devices. A similar thing may happen that an overlay operator wants only certain devices to be Peers due to various reasons such as high security requirements. For example, only devices that were authenticated using offline means may be allowed be a Peer or only devices that have been a Client for a long time without any bad behavior may be allowed to be a Peer or a devices that have enough fast network interface and uptime can become a peer.

5. What functions a client can contribute in P2P layer

Clients get services from the overlay network, however, clients can also contribute varialbe useful functions to the overlay. For example, a client can provide STUN server function to help establish connections between other peers, and also other useful functions. We focus on several functions that a client can contribute to the overlay in P2P layer here.

5.1. Storage function by a Client

In DHT, a Peer is responsible for storing data (resource (user) records) of a certain range of Resource IDs. For example, in Pastry/ Bamboo/OpenDHT, data (resource (user) records) are assigned to a Peer whose Node ID is closest to the Resource IDs than any other Peers.

It is possible for a Peer to delegate actual storage of the data to another node and just keep a pointer (location information) to the data in a different node. For example, a Peer may have a remote storage device and uses it as actual storage for the data the Peer is responsible for. Such a remote storage device may be another Peer, a Client, or a node not involved in the overlay at all. In this situation, any lookup message for the data will be routed to the Peer responsible for storing the data (Responsible Peer) and it is the responsibility of the Responsible Peer to reply to the lookup message with the data or a pointer to a node where the data is located. This is how a typical DHT operates. It should be transparent to other Peers whether a Peer uses its local memory or disk or remote memory or disk to store the data it is responsible for.

Whether to allow a Client to be a remote storage for a Peer (or multiple Peers) does not affect the overlay operation significantly because it is a local arrangement between the respective Peer(s) and the Client. However it affects the design of the protocol used between Peer and Client.

The reason for a device to become a Client instead of a Peer must not depend on the decision whether a Client may serve as a remote storage for a Peer or not.

Many of the reasons a device to become a Client rather than become a Peer make the device unsuitable to provide the storage service. For example, if a device is online only intermittently, storing data in such a device does not bring much benefit. However there is a case a Client may be able to store data for its associated Peers. It is when the device became a Client because it did not support the overlay's DHT algorithm while it met other requirements for a Peer.

In a sense, it is better for the overlay if some of its Clients can provide the storage service since the overlay's overall storage capacity increases. We need to look into two aspects:

- How much complexity is introduced to allow Clients to provide the storage service?

- Is the benefit of having Clients provide the storage service large enough?

More thought will be given to these questions.

5.2. P2P Relay function by a client

In some scenarios like for real time applications, a larger number of Peers is not necessarily beneficial, therefore there is probability that some capable devices act as clients in the overlay. These devices have the ability to relay messages for other devices.

A node behind strict firewall/NAT may need a relay when it wants to communicate with others. Generally, it can route messages with the overlay routing, however, for more efficiency, it can use a "Relay" with public address to relay requests or responses when it knows the contact address of the other party. All capable peers or clients can act as Relays. The choice of Relays must be well designed for some criterions, e.g. proximity. The best choice of Relay for a peer or client may be a peer or a client. Not in all cases can peer Relays be more efficient than client Relays.

Clients that provide relay function do not need to understand the DHT algorithm of the overlay. It only provides relay service to a certain peer or client that needs its help for communication convenience.

We only focus on the relay function for P2P layer communication here, e.g., P2P requests or responses. All relay functions beyond those are out of scope.

6. Acknowledgments

Some of the idea described in the draft came from the discussion in the P2PSIP mailing list. Thanks to Henning Schulzrinne,Henry Sinreich, and Lichun Li

7. Security Considerations

Clients (providing no storage service) are free riders. If not many devices qualifed to be a Peer do not volunterr to be a Peer, the P2PSIP network may not work well. It is a concern for P2P networks in general.

8. IANA Considerations

There are no IANA considerations associated to this memo.

9. References

9.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

9.2. Informative References

[3] Bryan, D., Matthew, Shim, E., and D. Willis, "Concepts and Terminology for Peer to Peer SIP", Internet Draft draft-ietf=p2psip-concepts-00, June 2007.

[4] Bryan, D., Baset, S., Matuszewski, M., and Sinreich, "P2PSIP Protocol Framework and Requirements", Internet Draft draft-bryan-p2psip-requirements-00, June 2007.

[5] Jennings, C., Lowekamp, B., Rescorla, E., and Rosenberg, "REsource LOcation And Discovery (RELOAD)", Internet Draft draft-bryan-p2psip-reload-02, November 2007.

[6] Baset, S., Schulzrinne, H., and M. Matuszweski, "Peer-to-Peer Protocol (P2PP)", Internet Draft draft-baset-p2psip-p2pp-01, November 2007.

[7] Song, Y., Jiang, X., Zheng, H., and H. Deng, "P2PSIP Client Protocol", Internet Draft draft-zheng-p2psip-client-protocol-01, February 2008.

[8] Jiang, X., Zheng, H., Macian, C., and V. Pascual, "Service Extensible P2P Peer Protocol", Internet Draft draft-jiang-p2psip-sep-01, February 2008.

[9] Li, L Ch. and Y. Wang, "Different types of nodes in P2PSIP", Internet Draft draft-li-p2psip-node-types-00, November 2007.

Authors' Addresses

Victor Pascual
Pompeu Fabra University
Barcelona, Passeig de la Circumval.lacio 8 08003
Spain

Phone: +34-93-5421561
Fax: +34-93-5422517
Email: victor.pascuala@upf.edu

Marchin Matuszewski
Nokia
P.O.Box 407
NOKIA GROUP, FIN 00045
Finland

Phone: unlisted
Email: marcin.matsuszewski@nokia.com

Eunsoo Shim
Locus Telecommunications
111 Sylvan Avenue
Englewood Cliffs, New Jersey 07632
USA

Phone: unlisted
Email: eunsooshim@gmail.com

Hewen Zheng
Huawei Technologies
Baixia Road No. 91
Nanjing, Jiangsu Province 210001
PRC

Phone: +86-25-84565467
Fax: +86-25-84565354
Email: hwzheng@huawei.com

Song Yongchao
Huawei Technologies
Baixia Road No. 91
Nanjing, Jiangsu Province 210001
PRC

Phone: +86-25-84565081
Fax: +86-25-84565070
Email: melodysong@huawei.com

Full Copyright Statement

Intellectual Property