Different types of nodes in P2PSIP

draft-li-p2psip-node-types-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 25, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document is an attempt to classify different types of node in peer-to-peer Session Initiation Protocol (P2PSIP). Four possible types of nodes in P2PSIP are discussed based on two characters: whether it offers overlay-routing services and whether it offers storage functions. The behaviors of each type of nodes and their possible necessities are analyzed. This document is dedicated to be a reference for clarifying some controversial terms in P2PSIP working group, such as "client", "low-function peer", etc.
Table of Contents

1. Introduction ............................................... 3
  1.1. Requirements Language ................................. 3
2. Types of Nodes .............................................. 3
  2.1. Possible types of nodes .............................. 3
  2.2. Naming of different types ............................ 4
3. Type B Node ................................................ 4
4. Type C Node ................................................. 5
5. Type D Node (Client) ....................................... 6
  5.1. P2P Client ............................................. 6
  5.2. Non-P2P Client ........................................ 7
6. Summary .................................................... 8
  6.1. Reference Model ...................................... 9
7. IANA Considerations ....................................... 10
8. Security Considerations ................................... 10
9. Acknowledgements ......................................... 10
10. References ............................................... 11
  10.1. Normative References ............................... 11
  10.2. Informative References ............................. 11
Authors’ Addresses ........................................... 11
Intellectual Property and Copyright Statements ............... 12
1. Introduction

In IETF P2PSIP WG, the discussion on the term "client" and its possible behavior has lasted for a long time. But there is still no consensus on the roles and behaviors of non-peer nodes until now.

To clarify these concepts, based on two fundamental services provided by P2P overlay, which are overlay-routing and storage, we try to classify different node types within P2PSIP. For each type of nodes, their role and behavior are described firstly; then the necessity of such node is explored. Because not all the node types are necessary, for the node types that can be merged into other types, our arguments are presented.

This document is dedicated to be a reference for clarifying some controversial terms in P2PSIP WG, such as "client", "low-function peer", etc. It is hoped that this document could help the WG to reach a rough consensus on the types of node within P2PSIP system and the characters of each type’s behavior.

Many points and arguments in this document originate from the discussions in the P2PSIP maillist and related Internet-Drafts. The authors of this document appreciate the contributions that are made by the people who joined in the related discussions.

1.1. Requirements Language

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 RFC 2119 [RFC2119].

2. Types of Nodes

2.1. Possible types of nodes

According to [I-D.ietf-p2psip-concepts], P2PSIP overlay provides distributed database function and distributed transport function for SIP and other applications. In P2PSIP overlay, peers offer storage and transport services to allow the distributed database function and distributed transport function to be implemented. Other types of nodes MAY exist in P2PSIP.

Based on the functionalities provided by P2P overlay, say capability of storage and transport services, nodes in P2PSIP can be divided into four types as follow:
- Type A Nodes: nodes offering storage and transport services.
- Type B Nodes: nodes offering transport services but NOT offering storage services.
- Type C Nodes: nodes offering storage services but NOT offering transport services.
- Type D Nodes: nodes NEITHER offering storage NOR offering transport services, only using these services.

In all these types, Type A nodes have been accepted as "P2PSIP peer" or "peer" by the P2PSIP WG. Thus, in the following sections, we will mainly focus on the other three types of nodes.

### 2.2. Naming of different types

Given that the terms such as "client", "low-function peer" are too ambiguous, in this subsection the naming of each type of node are clarified.

In the client/server computing model, "client" refers to nodes only using services. In the peer-to-peer computing model, "peer" refers to the nodes both using services and providing services. Thus, type A node can be named as "peer", which has been accepted by the P2PSIP WG. Also type D node can be named "client". For type B and type C nodes, they will be assigned a suitable name based on the analysis of their main characters.

### 3. Type B Node

Type B nodes refer to the nodes which offer transport functionality but do not contribute their storage capacity.

In the current version of [I-D.ietf-p2psip-concepts], such type of nodes is referred as "peers with bad memory". Some want to call this type of nodes as "clients", and others view this type of nodes as one type of "low-function peers". The behavior of type B node is described as follow.

(Note: the term "client" in following description is corresponded to the "type B node" in this document.)

"A client has a ‘peer-ID’ and joins the overlay in the same way as a peer, perhaps establishing the same network of connections that a peer would. Clients participate in the distributed database algorithm, and can help in transporting messages to other peers and
clients. However, the distributed database algorithm does not assign resource records to clients."

Despite the roles and behaviors of this type of nodes have been described above, the authors of this document could not find any convincing argument on the necessity of defining a separate concept for this type of nodes. If such type of node can "join the overlay in the same way as peer" and "help in transporting messages to other peers and clients", it should be recognized as "non-storage peer" rather than "client".

4. Type C Node

Type C nodes refer to the nodes which contribute their storage functionality but do not offer transport functions in P2P network.

There are two ways to view this type of nodes: one is "low-function peers"; the other is "P2P clients offering storage services". Compared with the former one, the latter view is accepted by more people.

With different views, this type of nodes might behave differently. "low-function peers" behave more like peers, while "P2P clients offering storage services" behave more like the P2P clients defined in the following section. As a "low-function peer", a type C node should join overlay, and provides storage service almost the same as peers do. As a "P2P client offering storage service", a type C node can only provides storage service to its associated peer(s), and the type C nodes might be transparent to other peers.

Because type C nodes are not overlay-routing nodes, therefore, it should not be recognized as some kind of peer. In addition, because "P2P client offering storage service" doesn't need to join overlay or need to know the distributed database algorithm using in overlay, designing and implementing "P2P client offering storage service" seems to be less complicated. Thus, in this document, the "P2P client offering storage service" is preferred and type C node is named as "storage P2P client", which is a special type of P2P client defined in section 5.1.

The necessity of storage P2P client contains two facets: the necessity of P2P client, the necessity of storage service. There is no reason to forbid P2P client from providing storage service. And these storage services can enlarge the capacity of storage of the overlay. The necessity of P2P client will be discussed in section 5.1.
5. Type D Node (Client)

Type D nodes refer to the nodes which neither offer transport functionality nor contribute their storage capacities. In this document, these nodes are named as "client".

Based on the way to access the distributed database function and distributed transport function provided by overlay, clients (Type D nodes) can be divided into two categories, P2P client and non-P2P client. A P2P client use one or more peers as its associated peer(s) to access these functions; they can directly execute operations such as PUT, GET, with the help of its associated peer. Non-P2P client access these functions indirectly through application entities residing in peers or P2P clients; for example, a SIP UA has to use SIP proxies coupled with peers to register itself into P2PSIP overlay.

5.1. P2P Client

The behavior of P2P clients mainly contains three aspects. Firstly, they comply with the common behavior of type D node; that is to say, they neither join the P2P overlay nor contribute the overlay-routing functionalities. Secondly, they are aware of P2P overlay and can execute operations such as PUT and GET via its associated peer to insert a resource into P2P overlay or retrieve it from P2P overlay. Finally, they can be coupled with any application entities. For example, a P2PSIP application can be coupled with P2P client, using the P2P operations as an alternative mechanism to [RFC3263].

Generally speaking, the necessity of P2P client comes from the heterogeneity among the nodes which want to use services provided by P2P overlay. The heterogeneity is explained in different aspects in the following paragraphs.

Firstly, the heterogeneity of P2P overlay algorithm makes P2P Client necessary. For example, a P2PSIP node which only supports Chord algorithm, needs to access services of P2PSIP overlay which uses Bamboo DHT. Due to the incompatible P2P algorithms, the Chord-based node can not join in the Bamboo-based DHT overlay. Thus the Chord-based node has to work as P2P Client and use the client protocol to access the services of DHT.

Secondly, the heterogeneity among P2PSIP nodes also makes P2P Client necessary. In order to make DHT overlay convergent and efficient, some kinds of nodes, such as nodes with short online time, nodes with limited computational resources, etc, are not suitable to join the overlay. These unqualified nodes can only act as client, using services provided by P2P overlay to avoid introducing badly churns.
into P2P overlay and (or) making itself overloaded in the maintenance of P2P overlay.

The third category of Client candidates comes from the nodes which want to provide services. For example, a STUN/TURN server wants to provide services to other nodes in a P2PSIP system. For two reasons, this server may choose not to join in the P2P overlay. One is for better performance, that is to say, the computational resources for P2P maintenance and routing can be saved for better performance of the service it provided, say STUN/TRUN in this case. The other one is for safety, i.e. if these nodes act as peer in addition to service provider, it may be under the extra threats which comes from its peer role.

5.2. Non-P2P Client

Similar to that of P2P client, the behavior of non-P2P client also contains three parts. They not only comply with the general behavior of Type D node, but also can be coupled with any application entities. The main difference is the way they use storage and routing services provided by P2P overlay. The non-P2P clients are using these services via SIP entities or other entities coupled with peers or P2P clients. Taking the SIP UA for example, because they are unaware of P2P overlay, they can only use peers or P2P clients coupled with SIP proxy/registrar to register into P2PSIP system and establish SIP sessions.

Although people do not believe this kind of nodes and their protocol should be within the scope of P2PSIP WG, we argue that these non-P2P clients are necessary to P2PSIP system. The reasons mainly come from two aspects.

Firstly, in some application scenarios, for security or other considerations, some nodes are not entitled to join overlay and also are forbidden to access P2P overlay directly. Here is an example. According to [I-D.bryan-p2psip-app-scenarios], one possible application scenario is "P2P for Redundant SIP Proxies". Under such circumstances, to ensure the security of P2P network formed by SIP proxies, the operator who deployed these SIP proxies will not allow user nodes join the overlay and act as peers. In addition, for the safety of data stored in the overlay, the user nodes would not be allowed to directly use the PUT, GET or other functions of the P2P overlay either. Therefore, these user nodes have to act as non-P2P client.

Secondly, for the compatibility with the existing SIP devices, the non-P2P client should exist in P2PSIP system. These non-P2P aware SIP clients can neither join in P2PSIP overlay nor access the P2P
services by using client protocol. However, for smooth evolution of P2PSIP, these SIP clients should be taken into consideration at the beginning of P2PSIP’s commercial deployment. Therefore, the compatibility with traditional SIP [RFC3261] is indispensable. Non-P2P client may be the only suitable role for these non-P2P aware SIP clients.

6. Summary

Based on the description of behaviors of different node types and analysis on the necessity of each type, we propose the concept of P2PSIP peer (Type A node, MAY include Type B node), storage P2P client (Type C node), P2P client (Type D nodes) and non-P2P client (Type D nodes) should be integrated into the concept of P2PSIP system and their respective protocols should be taken into consideration in the process of the standardization of P2PSIP if necessary.

In these proposed concepts, except P2PSIP peer, all the other types can be merged into a more general term which is "P2PSIP client". Currently, the concept of "P2PSIP client" has not been clearly defined. It is hoped that the WG can reach a rough consensus on "P2PSIP client" based on the analysis of different roles and behaviors in P2PSIP system.
6.1. Reference Model

P2PSIP Network Reference Model

Figure 1
Fig 1 illustrates the architecture of P2PSIP networks. This figure is mainly based on the "Figure: P2PSIP Overlay Reference Model" in [I-D.ietf-p2psip-concepts]. In addition, some modifications are made in order to highlight different types of nodes.

In Fig 1, each node is separated into two parts. The upper part describes the application entity/entities coupled with the node; while the lower part tells the node’s type.

Several peers construct a P2PSIP overlay defined in [I-D.ietf-p2psip-concepts] using peer protocol. Some peers may provide extra services, such as STUN/TURN services.

P2P Clients and storage P2P clients do not join the P2PSIP overlay but use the distributed database function and distributed transport function provided by overlay. To access these functions, P2P Clients and storage P2P clients used P2P client protocol to communicate with peers. Through P2P client protocol, storage P2P clients can also provide their storage services.

Some peers and P2P clients/storage P2P clients are coupled with SIP proxy, thus they can be used as SIP outbound proxies by SIP UAs residing in non-P2P clients. These SIP UAs might be conventional SIP UAs or might support some SIP extensions for P2PSIP.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

No security consideration in current version

9. Acknowledgements

This document is mainly inspired by the related discussion in P2PSIP maillist. The authors are grateful to those who have actively participated in the debate on the necessity, behaviors and characters of P2PSIP client.

10. References
10.1. Normative References


10.2. Informative References


Authors' Addresses

LiChun Li
Beijing University of Posts and Telecommunications.
P.O. Box 92, No.10, Xitucheng Road
Beijing, Haidian District 100876
P.R.China

Email: lilichun@gmail.com

Yao Wang
Beijing University of Posts and Telecommunications.
P.O. Box 92, No.10, Xitucheng Road
Beijing, Haidian District 100876
P.R.China

Email: yaowang.bupt@gmail.com
Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).