A Hierarchical P2P-SIP Architecture
draft-shi-p2psip-hier-arch-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 February 23, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).
Abstract

This document outlines the hierarchical P2P-SIP architecture, which makes the combination of P2P and SIP practical. In this architecture, heterogeneous overlays are inter-connected via a decentralized manner provided by the hierarchical P2P-SIP architecture. The overhead of the session control process is reduced since the P2P overlay is used by SIP as the underlying peer locating approach. Some nodes in the overlay are proposed to be stateful to improve the reliability of the overlay. Finally, a session initial example is presented to illustrate this architecture. This work is being discussed on the p2psip@cs.columbia.edu mailing list.

Table of Contents

1. Introduction .................................................. 3
   1.1. Background .............................................. 3
   1.2. Motivation ............................................. 3
2. Requirements notation ......................................... 5
3. Hierarchical P2P-SIP Architecture ........................... 6
   3.1. Network architecture ................................... 6
   3.2. Protocol Stack Architecture ............................ 8
4. Generic P2P Overlay Service ................................ 10
   4.1. Bootstrap ............................................... 10
   4.2. Node Operations ........................................ 10
      4.2.1. Node Registration .................................. 10
      4.2.2. Node Leaving and Failure .......................... 11
   4.3. Resource Operations .................................... 11
5. P2P-SIP Naming ................................................. 13
6. P2P Proxy ..................................................... 14
   6.1. P2P Proxy Election ...................................... 14
   6.2. P2P Proxy Behavior on Routing the SIP Message ........ 14
   6.3. P2P Proxy Leaving and Failure .......................... 14
7. An example of Inter-Domain Communication Process ........... 15
8. Security Considerations ....................................... 18
9. Open Issues .................................................... 19
   9.1. Stateful and Stateless Problems ........................ 19
   9.2. Decentralized Bootstrap Process ......................... 19
10. Acknowledgements ............................................. 20
11. References .................................................... 21
    11.1. Normative References ................................. 21
    11.2. Informative References ............................... 21
Authors’ Addresses ............................................... 22
Intellectual Property and Copyright Statements ................ 23
1. Introduction

1.1. Background

Many benefits can be gained from the combination of SIP [2] and P2P [3]. From the SIP point of view, central servers such as the proxy and registrar can be removed to reduce the deployment and maintenance cost, and the robustness of SIP will also be increased. From the P2P point of view, the introducing of SIP will make the overlay controllable, and specific applications such as P2P Internet Phone and IM can be enabled.

1.2. Motivation

To combine P2P and SIP together, there are two approaches: P2P-over-SIP [4] [10] and SIP-over-P2P [11]. The former approach focuses on using SIP messages to maintain P2P overlay networks. The approach is motivated by specific multimedia session control requirements in P2P overlays. The upper layer SIP provides the P2P overlay with control functions and establishes multimedia sessions over unreliable P2P overlay networks. However, the primary disadvantage of the approach is that P2P-over-SIP needs to maintain many SIP dialog states during the overlay control course. For example, using P2P-over-SIP, we need to forward over 17 kinds of SIP messages from joining DHT to finishing registering and each SIP message has its own dialog states in user agents.

An alternative approach is to layer SIP over the P2P overlay network. The underlying P2P overlay networks such as Chord [5], CAN [6], Pastry [7] and etc provides SIP with the peer locating service. That is to say, the traditional centralized SIP trapezoid will be replaced by the decentralized P2P location service. Thus we can reduce the deploying cost of P2P-SIP and increase the robustness of the network. However, when we use P2P overlays as the SIP’s peer locating service, there are some problems need to be solved. First, one of the essential design targets of SIP is to find the peer anytime and anywhere using DNS, however, when we replace the DNS and SIP Proxy with P2P overlays, heterogeneous overlays will cause the connectivity problem. A peer in one P2P overlay (such as Chord) is difficult to route the SIP message to another P2P overlay (such as Pastry) based on their individual P2P peer locating service. Second, SIP is a stateful protocol which controls the session establishing, modifying and terminating process over stateless IP networks, however, when we use the P2P overlay to provide the peer locating service, how to inherit features of stateful proxy from the traditional SIP is unclear.

Here we summarize some crucial issues need to be considered when we
combine P2P and SIP together.

1. When SIP is made serverless, we should retain the connectivity of SIP especially when heterogeneous P2P overlays are served as the underlying peer locating service simultaneously.

2. When the P2P overlay is controlled by SIP, some approaches should be proposed to reduce the overhead caused by too many SIP dialogs.

3. When P2P and SIP are combined together, the emerging P2P-SIP protocol should also inherit the routing feature of the stateful proxy of traditional SIP.

The above issues motivate this document of the hierarchical architecture for P2P and SIP combination. P2P overlays are used by SIP as the underlying peer locating protocol thus the overhead will be significantly reduced compared with the P2P-over-SIP approach. Then, each overlay will elect one or more powerful peers to be P2P proxies which logically construct an upper level overlay and forward messages among heterogeneous overlays. Thus the connectivity problem is solved. Furthermore, it is also proposed to make several peers in an overlay to be starteful to inherit the controllable feature of stateful SIP proxies.
2. Requirements notation

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 [1].

Terminology defined in RFC 3261 [2] and in P2PSIP concepts draft [12] is used without definition.

Local P2P Overlay: A basic P2P overlay network that provides the location and routing functionalities.

Global P2P Overlay: An upper level P2P overlay that inter-connects hierarchical local P2P overlays.

P2P Proxy: A SIP proxy elected from P2P Proxy candidates. It runs both local P2P overlay and global P2P overlay’s stack, and it is responsive for inter-overlay SIP message exchange.

P2P Proxy Candidates: A node elected from the P2P-SIP Overlay Peer prepares to become the P2P Proxy when the proxy current leaves or fails.

P2P-SIP User URI: A SIP URI used for identifying the P2P-SIP user

P2P-SIP Overlay URI: A SIP URI used for identify the P2P-SIP overlay.
3. Hierarchical P2P-SIP Architecture

3.1. Network architecture

Figure 1 illustrates the proposed hierarchical P2P-SIP structure. Various P2P overlays (such as Pastry, CAN and etc, we call it the local overlay. Their real overlay structures may not be ring-based and here we only use ring for illustration) are respectively used as the underlying route discovery protocol for the decentralized SIP scheme, that is, when two peers of a call are in the same overlay network, the callee lookup course will be base on the overlay’s specific resource locating process. In this way, the communication overhead will be significantly reduced since relaying nodes need not maintain dialog-states.
To connect peers from heterogeneous overlays, it is proposed to organize an upper level overlay (defined as the global P2P overlay) which inter-connects the local overlays. That is, each overlay will elect one or more powerful peers to be the gateway-like nodes that
route messages among heterogeneous overlays. The elected gateway-like node is defined as the P2P proxy. In Figure 1, node 8 joins both the local Pastry overlay and global overlay to perform the gateway-like function.

Furthermore, several peers in each overlay are made to be stateful. Thus the controllable feature of the traditional stateful SIP proxy will be inherited. P2P proxies are good candidates for such kind of stateful nodes.

By introducing such a hierarchical architecture, peers in heterogeneous overlays can be inter-connected and the balance between low signaling overhead and easy management can be achieved.

3.2. Protocol Stack Architecture

Protocols are composed of two separate layers, the upper layer is the SIP layer and the lower one is the P2P overlay layer. The SIP layer is responsible for sending, receiving and processing SIP messages so as to establish or terminate multimedia sessions. The P2P overlay layer handles all the P2P overlay functions, including formation, maintenance, lookup, as well as offering peer locating service to the SIP layer.

For a peer node, the SIP layer can be a SIP User Agent; while for the proxy node, the SIP layer can be a B2BUA. Both SIP UA and B2BUA act the same as defined in the traditional SIP protocol except for the peer locating service. The upper layer SIP gets the location (i.e. IP address) of the destination user from the P2P overlay layer instead of location severs. That is, we implement the function of location related severs in a distributed manner, using P2P overlay networks.

The P2P overlay layer of ordinary peer nodes differs from that of P2P proxy nodes. There is only one specific P2P technology in a normal peer node. Peers with the same P2P technology will logically form a P2P overlay. However, in a P2P Proxy node, the P2P overlay layer has double stacks. One is the P2P approach used in Local P2P Overlay, and the other is the Global P2P Overlay approach (i.e. Chord in Fig.1). On the one hand, a proxy node joins to form the Local P2P Overlay with other peer nodes in the specific overlay; on the other hand, all the elected P2P proxy nodes will join together to form a Global P2P Overlay network. Nodes located in different P2P domains can communicate with each other via this global overlay network.
Figure 2 illustrates the protocol stack architecture presented above. In this figure, CAN and Pastry are different P2P overlay technologies, and they are used in Local P2P Overlay. The Global overlay refers to the Inter-Domain P2P overlay technology (i.e., Chord). This protocol stack is an open architecture and any P2P overlay technology that satisfies the requirement in section 4 (Section 4) can be used.
4. Generic P2P Overlay Service

Generic P2P overlay service is used by SIP as the underlying peer locating service. In this section, we describe the generic DHT-based P2P overlay functions. Any non-DHT based approach is also acceptable for the P2P-SIP implementation.

4.1. Bootstrap

When a node intends to join an existing overlay, it must first locate some nodes that are already active in the overlay. Then it can join the overlay with the help of that node. We call this process bootstrap. There are several methods for bootstrap implementation.

1. Web Station
   We can maintain a web station, which provides the information of the nodes that are already active in the overlay. This kind of bootstrap can refer to the Gnutella Web Caching System[8].

2. Cached Address
   A node may cache some addresses during its prior connection. The cached list enables it to find the bootstrap node directly.

3. Pre-configured Methods
   Some nodes in the overlay may be persistent, and have well known addresses. These addresses could be pre-configured into nodes.

4. Broadcast
   The node that wants to join the overlay could also broadcast messages in the local network to locate a node that has already joined the overlay.

4.2. Node Operations

4.2.1. Node Registration

Node registration enables a peer node to join the structured overlay network with its Node-ID. The Node-ID is generated by the specific Hash algorithms the overlay adopts, that is:

Node-ID = Hash(Node-IP)

Then the joining peer will contact the bootstrap node to help it join the overlay. Specific DHT algorithms such as Chord, Pastry, CAN and etc have specified the node registration process. Note that in the hierarchical P2P-SIP overlay, the P2P proxy should register its Node-ID to both the local overlay and the global overlay, thus it can inter-connect various overlays.
4.2.2. Node Leaving and Failure

The maintenance of node leaving and failure is described in specific DHT algorithm and we do not discuss it in detail here. When a node leaves the overlay, another node should be notified to be responsible for the resources of the leaving node. When a node fails, the self-recovery approach should be adopted to re-organize the overlay network.

The P2P proxy node is particularly important in the hierarchical P2P-SIP overlay, and the leaving and failure of the P2P proxy is described in section 6.3.

4.3. Resource Operations

In the hierarchical P2P-SIP overlay network, there are primarily two kinds of resources need to be managed, that is, the P2P-SIP user’s identifier (P2P-SIP User URI) and the P2P overlay’s identifier (P2P-SIP Overlay URI). The former URI is used to uniformly identify P2P-SIP users and the latter URI is used to identify various overlays. The naming rule is described in section 5.

P2P overlays provide the approach which manages URI resources in a distributed manner. Thus the route discovery related elements such as DNS servers could be replaced by the distributed resource management functions provided by the P2P overlay.

1. In the local overlay, the P2P-SIP User URI is hashed as the User-Resource-ID, and then the User-Resource-ID is registered to a node in the local overlay.

   User-Resource-ID = Hash(P2PSIP-User-URI)

2. In the global overlay, the P2P-SIP Overlay URI is hashed as the Overlay-Resource-ID, and then the Overlay-Resource-ID is registered to a P2P proxy in the global overlay.

   Overlay-Resource-ID = Hash(P2PSIP-Overlay-URI)

In the hierarchical P2P-SIP protocol, two new SIP headers called Overlay-ID and Node-ID are added to identify the P2P-SIP scheme. For example, some headers of an INVITE Message might look like this:

   INVITE sip: bob@bupt.pastry SIP/2.0
   Via: SIP/2.0/UDP 59.64.186.33; branch=z9hG4bK776asdhds
   Max-Forwards: 70
From: Bob <sip:bob@bupt.pastry>;tag=1928301774
Call-ID: a84b4c76e66710@59.64.186.33
CSeq: 314159 INVITE
Contact: <sip:bob@59.64.186.33>
Node-ID: <sip:bob@bupt.pastry>
Overlay-ID: <sip:overlay@pastry>
Content-Type: application/sdp
Content-Length: 142
5. P2P-SIP Naming

There should be a district name that identifies the specific P2P-SIP user and overlay. The specific naming rule can refer to the SIP Address of Record (AOR).

Although the P2P-SIP paradigms intend to remove the dependence on central servers such as the DNS server, P2P-SIP User ID can also inherit the hierarchical organization feature of DNS. In this way, top level labels identify heterogeneous overlays such as CAN and Pastry in Figure 1, while the second level labels identify the specific users or groups within each overlay network. Furthermore, since there’re two kinds of URI in the P2P-SIP overlay, the overlay URI should be automatically generated by the P2P proxy using the user URI.

For instance, the URI sip:bob@bupt.pastry can be one of the implementation of the P2P-SIP user identifier. Bob at Beijing University of Posts and Telecommunications (BUPT) should register the P2P-SIP User URI sip:bob@bupt.pastry to the Pastry overlay based on the Pastry algorithm and one or more P2P proxies of BUPT in the Pastry overlay should register the P2P-SIP Overlay URI sip:overlay@pastry to the top level overlay. Then when Bob wants to call Alice (sip:alice@pku.can) at Peking University in the CAN overlay, the request message will first be routed to the P2P proxy at BUPT since the callee’s URI does not belong to the Pastry overlay. Then the proxy will forward the message to the P2P proxy at Peking University in the CAN overlay based on the overlay’s URI (sip:overlay@can). Finally the P2P proxy at Peking University will route the message to Alice based on the user ID (sip:alice@pku.can).

Note that this kind of naming and addressing method significantly differs from that of DNS although both of them are hierarchically organized. The addressing process in DNS is based on centralized DNS servers while that in P2P paradigms is based on self-organized overlay hosts. Thus we can call it the P2P Domain System.
6. P2P Proxy

6.1. P2P Proxy Election

The P2P proxy election approach is proposed to make the proxies to be more stable and powerful. The election metrics should be online time, bandwidth, CPU capacity and etc. Then specific utility functions could be formulated to evaluate the peer’s capacity.

\[ U = f(\text{online, bandwidth, cpu, others}) \]

Based on the utility function, each node is evaluated and assigned a score. Nodes with higher score will become P2P proxy candidates. Then the P2P proxy will be elected from the P2P proxy candidates.

The elected proxy candidates do not perform as P2P proxies, but only monitor the P2P proxy and prepare to replace it when it leaves or fails. Thus we call the P2P proxy candidate inactivated proxy. Each overlay should maintain its inactivated proxy list. Thus when the P2P proxy fails, the connectivity can be fast recovered.

6.2. P2P Proxy Behavior on Routing the SIP Message

The P2P proxy serves as a gateway-like element among heterogeneous overlay networks, and it runs double stacks (i.e. the local overlay protocol and global overlay protocol). The P2P proxy registers to both the local overlay and the global overlay using respective protocol stacks.

When peers in the local overlay detect that the callee’s domain name doesn’t belong to the overlay, they will send the SIP message to the P2P proxy. Then the P2P proxy is responsible for routing the message to another P2P domain using the upper level overlay’s peer locating service. When the P2P proxy in the callee’s overlay receives the inter-domain routed SIP message, it will send it to the callee’s UA using the local overlay’s peer locating service. Examples in section 7 (Section 7) illustrate the routing behavior of the P2P proxy.

6.3. P2P Proxy Leaving and Failure

The local overlay will maintain an inactivated P2P proxy list, and members in the list are considered as the P2P proxy candidates. P2P proxy candidate will periodically send the HELLO message to the P2P proxy so as to determine whether the proxy is alive. When the P2P proxy fails, a P2P proxy candidate in the list will be elected to become the active P2P proxy which joins the upper level overlay. If the P2P proxy leaves, it will elect a candidate to make it the proxy.
7. An example of Inter-Domain Communication Process

<table>
<thead>
<tr>
<th>Bob's UA</th>
<th>Pastry Overlay</th>
<th>P1 Proxy</th>
<th>chord Overlay</th>
<th>P2 Proxy</th>
<th>CAN Overlay</th>
<th>Alice's UA</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 Get Local Proxy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2 Addr Of LocalProxy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F3 INVITE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F4 100 TRYING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F5 Get Addr of P2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F6 Addr of P2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F7 INVITE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F8 100 TRYING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F9 Get Alice's UA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F10 Addr of Alice's UA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F11 INVITE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F12 180 RINGING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F13 180 RINGING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F14 180 RINGING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F15 200OK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
When Bob (identified by sip:bob@bupt.pastry) at BUPT wants to call Alice identified by sip:alice@pku.can) at Peking University, the session initial process is as follows.

1. Caller Bob’s UA first determines whether the callee’s domain pku.can is in its domain .pastry, and it finds that it is not. Then it finds the P2P proxy in its overlay which joins the upper level overlay based on the specific global P2P approach.

2. Caller Bob’s UA generates an INVITE message based on RFC 3261, and sets the Request-URI to be the Address-of-Record of the callee Alice.

3. Bob’s UA sends the INVITE message to the P2P proxy (P1) in the Pastry overlay.

4. P1 looks up Alice’s overlay via the top level overlay Chord, and gets the P2P proxy’s IP address of the CAN overlay.

5. P1 adds its address in the VIA header of the INVITE message and forwards that message to P2; P1 also sends a 100 Trying to Bob’s UA.

6. When P2 receives the INVITE message, it looks up the address of Alice’s UA via the CAN overlay.

7. P2 adds its address in the VIA header of the INVITE message and forwards that message to Alice’s UA; P2 also sends a 100 Trying to P1.

8. When Alice’s UA receives the INVITE message, it deals with the message according to the specification in RFC 3261.
9. When the INVITE message arrives at Alice’s UA, the 180 Ring message is sent back via the information in its Via header.

10. When Alice picks up the phone, the 200 OK message is sent back to Bob’s UA

11. When Bob’s UA receives the 200 OK message, it sends the ACK message to Alice via the information in the Contact header.

12. Finally, the peer to peer multimedia session between Bob’s UA and Alice’s UA is established.
8. Security Considerations

Security related problems such as DoS attack are not discussed in this document. However, the attacker may have powerful machine to DoS the proxy node and these problems need to be considered in the future draft.
9. Open Issues

9.1. Stateful and Stateless Problems

SIP can be based on the control of stateful proxy which makes the session management reliable. However, when we use the P2P overlay as SIP’s route discovery protocol, the state in peers is unclear. The essence is the tradeoff between controllability and overhead when we consider whether the peers should be stateful or stateless.

One of the design choices is to make some peers such as the P2P proxy to be stateful, and both P2P-SIP UA and the stateful node will be aware of the state of the underlying overlay. For example, the P2P proxy mentioned in this draft should be stateful.

9.2. Decentralized Bootstrap Process

Although several approaches are proposed to implement the bootstrap process, the decentralized zero-configure bootstrap is still hard to achieve. Thus the session initial process cannot be absolutely decentralized.

It is practical to deploy a centralized bootstrap server, however, if we want the SIP to be absolutely decentralized, we should seek for some novel decentralized bootstrap methods.
10. Acknowledgements

The authors wish to thank Yao Wang, Hua Xu, Xiaosheng Tang, Xu Wang and Zhiyong Feng for their comments. The authors would like to thank Hui Wang and Ning Zhang for the help of review.
11. References

11.1. Normative References


11.2. Informative References


Authors’ Addresses

JuWei
Wireless Technology Innovation (WTI) Institute of BUPT.
P.O. Box 92, No.10, Xitucheng Road
BeiJing, Haidian District 100876
CN
Email: jwshi@bupt.edu.cn

Yang
Wireless Technology Innovation (WTI) Institute of BUPT.
P.O. Box 92, No.10, Xitucheng Road
BeiJing, Haidian District 100876
CN
Email: jiyang@bupt.edu.cn

Hua
Wireless Technology Innovation (WTI) Institute of BUPT.
P.O. Box 92, No.10, Xitucheng Road
BeiJing, Haidian District 100876
CN
Email: hbzhzw@gmail.com

YiNong
Wireless Technology Innovation (WTI) Institute of BUPT.
P.O. Box 92, No.10, Xitucheng Road
BeiJing, Haidian District 100876
CN
Email: hoplee@bupt.edu.cn