The Peer Protocol for P2PSIP Networks
draft-hautakorpi-p2psip-peer-protocol-00.txt

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 August 26, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes the P2PSIP Peer Protocol that is used between the peers in P2PSIP networks. The described Peer Protocol is not an entirely new protocol, it is a combination of the new SIP method, LOCSEr, and the well-defined, XML-based body.

Note: At the current moment, the purpose of this document is just to present a relatively high level abstraction from the Peer Protocol, rather than give a very detailed description.
1. Introduction

P2PSIP is a mechanism which incorporates Peer-to-Peer (P2P) technologies and the Session Initiation Protocol (SIP) [2] signaling in a way which allows the transfer of proxy-registrar functionality to the end-hosts. In P2PSIP network, the storage and routing services are provided by Peers which participate to the overlay. These Peers need a protocol for mutual message exchange, and this document specifies just that, referred as Peer Protocol hereafter.

The Peer Protocol is not an entirely new protocol, it is a combination of the new SIP method, LOCSER, and the well defined, Extensible Markup Language (XML)-based body. The LOCSER method is defined in this document, and the substantial parts of the body are documented in [4]. Furthermore, this document defines how the Peer Protocol can be used to provide a location service for P2PSIP.

Most of the terminology and concepts presented in this document are aligned with the [3]. Other terms are defined when used.

The rest of this document is organized as follows. Section 3 introduces the Peer Protocol on a high level and Section 4 gives an example from LOCSER method. Section 5 compares LOCSER approach to other proposals. Section 6 present a call flow example and Section 7 outlines how LOCSER could be utilized in client/server SIP. Section 8 enumerates topics for further study. Section 9 and Section 10 present topics for further study and IANA considerations respectively.

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 RFC 2119 [1].

3. The Peer Protocol

From practical standpoint, this document defines only one new SIP method, LOCSER, and its use for providing a location service. In order to implement a location service, also the work by other needs to be used.

The described Peer Protocol has two distinct parts: The LOCSER SIP request and its well-defined XML body. Frank Dabek et al have studied how to create an Application Programming Interface (API) for structured P2P overlays. In [8] they have introduced a multi-tier
approach for the common P2P API. Tier 0 is a Key-Based Routing (KBR) layer, tier 1 is a higher layer abstraction, which is built on top of tier 0, and tier 2 represent the application on top of tier 1. The LOCSER SIP request corresponds to tier 0 and its XML body corresponds to tier 1. In other words, the LOCSER SIP request provides a routing based on a key, and the rest of the functionalities (like put and get) are handled in the XML body. Naturally, the actual P2PSIP application represents tier 2.

4. Example

An example from Peer Protocol message is presented in Figure 1. It has one new header field, Overlay-Key. The Overlay-Key contains the key that is used overlay routing. The content of the Overlay-Key header field is similar to the ResourceID and PeerID tokens, which are explained in [6] (see Sections 8.2.1 and 8.2.2). The difference is that there is no need to make a separation between the resource and peer id in Overlay-Key header field; the routing will be the same in both cases.

```
LOCSER sip:bob@biloxi.example.com SIP/2.0
Overlay-Key: C83B247AC4F5D91E23A9B6CD2F32B912CAD7209E
From: Alice <sip:alice@atlanta.example.com>;tag=2948351372
To: Bob <sip:bob@biloxi.example.com>
Via: SIP/2.0/UDP sip:alice@100.101.102.103;branch=y9gG4bL376csuhks
CSeq: 1 LOCSER
Call-ID: 12345601@atlanta.example.com
Max-Forwards: 70
Content-Type: text/xml
Content-length: 277

<?xml version="1.0" encoding="UTF-8">
<methodCall>
  <methodName>get</methodName>
  <params>
    <param>
      <dhtRecord>
        <key>bob@biloxi.example.com</key>
        <KeyInfo>
          <KeyValue>C83B247AC4F5D91E23A9B6CD2F32B912CAD7209E</KeyValue>
        </KeyInfo>
      </dhtRecord>
    </param>
  </params>
</methodCall>
```

Figure 1: Example Message
The XML body of the LOCSER message is re-using the XML data structures and interfaces specified in [4]. The example message in Figure 1 presents a case where XML body is used to resolve the location of bob@biloxi.example.com.

The interfaces presented in [4] does not specify how the XML body could be used for overlay maintenance. The XML body in LOCSER message has to support overlay maintenance, so the definitions in [4] needs to be extended. For example, the XML body should be able to accomodate the functions Chord [9] needs for injecting information about newly joined node to the finger tables of others.

There are two notable benefits from using SIP as a base for Peer Protocol: No need for yet-another protocol stack, and the possibility to re-use the SIP-compliant mechanisms. The first advantage is especially useful in mobile devices, which have relatively scarce memory resources. The second advantage is especially beneficial in a case of NAT traversal; existing mechanisms, like SIP Outbound [5], can be used with LOCSER method.

5. Commonalities and Differences to Other Proposals

5.1. A P2P Approach to SIP Registration and Resource Location

P2P based approach for SIP registration and resource discovery has been presented in [6]. The current document and [6] are similar in a sense that they both define a Peer Protocol for P2PSIP and they both use SIP. The current document also re-uses some of work done in [6]. For example, the Overlay-Key header field is based on the ResourceID and PeerID tokens.

The biggest difference between the current document and [6] is that the former utilizes the LOCSER method with well-defined XML body, and the latter utilized modified REGISTER requests.

The main reason why we chose not to use REGISTER request in this document is to avoid its semantic overloading. The RF3261 [2], Section 10.2, says the following about REGISTER request:

REGISTER requests add, remove, and query bindings.

In a P2P scenario it would be desirable to be able to use the Peer Protocol for all overlay management functions, for example for injecting information about newly joined node to the finger tables of others. This kind of functionality is out of scope for REGISTER request.
Another reason why REGISTER was not chosen is that the REGISTER method is already strictly defined. Some of the existing definitions are not well-suited for a P2P scenario. For example, strict processing rules, defined in Section 10.3 on RFC3261 [2], are suboptimal for P2P scenario.

The LOCSER SIP method and its well-defined XML body have been designed to be compatible with existing specifications. The RFC3621 [2], Section 6, says the following about location service:

The bindings can be created and removed in many ways; this specification defines a REGISTER method that updates the bindings.

Given this, the LOCSER method and its XML body can be seen as an alternative method, which is well suited of P2P scenario, for binding management.

5.2. Data format and Interface to P2P network for SIP Location Service

The data format and interface for decentralized location service are defined in [4]. The current document and the [4] has a lot in common. The main link between the documents is the fact that the Peer Protocol defined in this document re-uses the data format and the interface specified in [4].

There are also some differences. This document uses SIP as a underlying transport while [4] indirectly proposes the use of Hypertext Transfer Protocol (HTTP). Another difference is that [4] suggest to use two separate protocols for DHT maintenance and location service, while this document uses the same protocol for both.

6. Call Flow Example

The following example call flow, in Figure 2, illustrates a case where the caller, Alice, initiates a phone call to callee, Bob. Prior to this event, Bob has placed his Resource Record to an overlay network. In this example, all the nodes are Peers for the sake of simplicity.
First Alice generates a LOCSER request containing an Overlay-Key, which is a hash from Bob's URI. The LOCSER request contains also a well-defined XML body, which has a 'get' method. After generating the request, Alice send it to the most suitable peer candidate in the overlay. The mechanisms for determining which peer is the best candidate is up to the used overlay routing mechanism.

When the next peer, second from the left, receives the LOCSER request, it check whether the request is targeted to itself or not. If this case it is not and the peer forwards it to the most suitable peer candidate it knows. Then the next peer, third from the left, receives it and performs the same check. Now the result is that the current peer is the target peer, because it has Bob's resource record, and therefore the peer generates 200 OK response. The 200 OK response contains Bob's contact information, typically IP and port, and it is returned to Alice.

When Alice receives the 200 OK, it learns the Bob's contact information and is able to initiate a call to him. After this point the signaling takes place directly between Alice and Bob. First Alice send a typical INVITE request to Bob and Bob accepts the connection by sending back the 200 OK response. Finally, the media can start flowing between Alice and Bob.
7. Usage with Client/Server SIP

The idea from Location Service Protocol (LSP) is introduces in [7]. According to [7], the LSP could be used in client/server SIP and P2P scenarios. The LSP in client/server SIP would be the protocol used between a registrar and a location service or a proxy and a location service.

The LOCSER method can also be used in the client/server scenario. In that case, LOCSER method does not need to carry Overlay-Key header field, because key-based routing is not required. The semantics of the LOCSER are the same in client/server and P2P scenarios.

8. For Further Study

The purpose of this document, at the current moment, is to present a relatively high level abstraction from the described Peer Protocol, rather that give very detailed information. Naturally, all the details needs to be worked on and the following is a list from topics that require further attention:

1. Specify the interfaces and data structures used for overlay maintenance in XML. These should be developed in the spirit of [4].
2. Specify the exact syntax of the LOCSER message. The Overlay-Key header field needs exact description as well.
3. Specify how peers should handle incoming LOCSER messages.
5. Specify how the presence and offline messaging can be implemented in a decentralized fashion by using the XML body.

9. Security Considerations

For further study.

10. IANA Considerations

For further study. At least the LOCSER method and the Overlay-Key header field need to be registered to the IANA.

11. References
11.1. Normative References


11.2. Informational References


Authors' Addresses

Jani Hautakorpi
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Jani.Hautakorpi@ericsson.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.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).