P2PSIP
Internet-Draft
Intended status: Standards Track
Expires: November 08, 2013

N. Zong
X. Jiang
R. Even
Huawei Technologies
Y. Zhang

May 07, 2013

# An extension to RELOAD to support Relay Peer Routing
# draft-ietf-p2psip-rpr-06

Abstract

This document proposes an optional extension to RELOAD to support relay peer routing mode. RELOAD recommends symmetric recursive routing for routing messages. The new optional extension provides a shorter route for responses reducing the overhead on intermediate peers and describes the potential use cases where this extension can be used.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on November 08, 2013.

Table of Contents

## 1. Introduction

RELOAD [I-D.ietf-p2psip-base] recommends symmetric recursive routing (SRR) for routing messages and describes the extensions that would be required to support additional routing algorithms. Other than SRR, two other routing options: direct response routing (DRR) and relay peer routing (RPR) are also discussed in Appendix A of [I-D.ietf-p2psip-base]. As we show in section 3, RPR is advantageous over SRR in some scenarios reducing load (CPU and link bandwidth) on intermediate peers. RPR works better in a network where relay peers are provisioned in advance so that relay peers are publicly reachable in the P2P system. In other scenarios, using a combination of RPR and SRR together is more likely to bring benefits than if SRR is used alone.

Note that in this document, we focus on RPR routing mode and its extensions to RELOAD to produce a standalone solution. Please refer to DRR document [I-D.ietf-p2psip-drr] for DRR routing mode.

We first discuss the problem statement in Section 3, then how to combine RPR and SRR is presented in Section 4. In Section 5, we give comparison on the cost of SRR and RPR in both managed and open networks. An extension to RELOAD to support RPR is proposed in Section 6. Discovery of relay peers is introduced in Section 7. Some optional methods to check peer connectivity are introduced in Appendix A.

## 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 [RFC2119].

We use the terminology and definitions from the Concepts and Terminology for Peer to Peer SIP [I-D.ietf-p2psip-concepts] draft extensively in this document. We also use terms defined in NAT behavior discovery [RFC5780]. Other terms used in this document are defined inline when used and are also defined below for reference.

Publicly Reachable: A peer is publicly reachable if it can receive unsolicited messages from any other peer in the same overlay. Note: "publicly" does not mean that the peers must be on the public Internet, because the RELOAD protocol may be used in a closed system.

Relay Peer: A type of publicly reachable peer that can receive unsolicited messages from all other peers in the overlay and forward the responses from destination peers towards the sender of the request.

Relay Peer Routing (RPR): refers to a routing mode in which responses to P2PSIP requests are sent by the destination peer to a relay peer transport address who will forward the responses towards the sending peer. For simplicity, the abbreviation RPR is used instead in the rest of the document.

Symmetric Recursive Routing (SRR): refers to a routing mode in which responses follow the reverse path of the request to get to the sending peer. For simplicity, the abbreviation SRR is used instead in the rest of the document.

## 3. Overview

RELOAD is expected to work under a great number of application scenarios. The situations where RELOAD is to be deployed differ greatly. For instance, some deployments are global, such as a Skype-like system intended to provide public service, while others run in closed networks of small scale. SRR works in any situation, but RPR may work better in some specific scenarios.

### 3.1. RPR

RELOAD is a simple request-response protocol. After sending a request, a peer waits for a response from a destination peer. There are several ways for the destination peer to send a response back to the source peer. In this section, we will provide detailed information on RPR.

Note that the same illustrative settings can be found in DRR document [I-D.ietf-p2psip-drr].

#### 3.1.1. Relay Peer Routing (RPR)

If peer A knows it is behind a NAT or NATs, and knows one or more relay peers with whom they have a prior connections, peer A can try RPR. Assume A is associated with relay peer R. When sending the request, peer A includes information describing peer R transport

address in the request. When peer X receives the request, peer X sends the response to peer R, which forwards it directly to Peer A on the existing connection. Figure 1 illustrates RPR. Note that RPR also allows a shorter route for responses compared to SRR, which means less overhead on intermediate peers. Establishing a connection to the relay with TLS requires multiple round trips. Please refer to Section 5 for cost comparison between SRR and RPR.

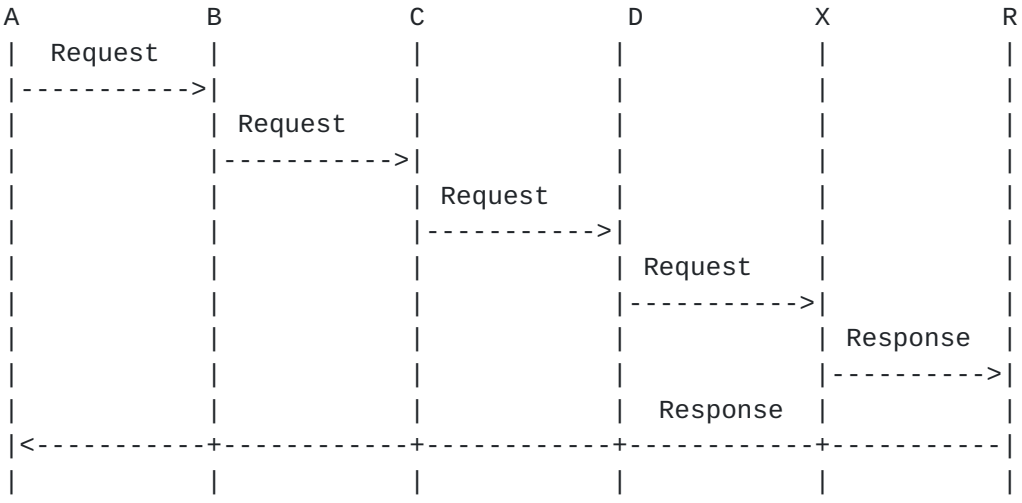


Figure 1. RPR routing mode

This technique relies on the relative population of peers such as A that require relay peers, and peers such as R that are capable of serving as a relay peers. It also requires a mechanism to enable peers to know which peers can be used as their relays. This mechanism may be based on configuration, for example as part of the overlay configuration an initial list of relay peers can be supplied. Another option is in a response message, the responding peer can announce that it can serve as a relay peer.

### 3.2. Scenarios where RPR can be beneficial

In this section, we will list several scenarios where using RPR would provide an improved performance.

#### 3.2.1. Managed or closed P2P systems

As described in Section 3.2.1 of DRR draft [I-D.ietf-p2psip-drr], many P2P systems run in a closed or managed environment so that network administrators can better manage their system. For example, the network administrator can deploy several relay peers which are publicly reachable in the system and indicate their presence in the configuration file. After learning where these relay peers are,

peers behind NATs can use RPR with the help from these relay peers. Peers MUST also support SRR in case RPR fails.

Another usage is to install relay peers on the managed network boundary allowing external peers to send responses to peers inside the managed network.

### 3.2.2. Using bootstrap nodes as relay peers

Bootstrap nodes are typically publicly reachable in a RELOAD architecture. As a result, one possible architecture would be to use the bootstrap nodes as relay peers for use with RPR. A relay peer SHOULD be publicly accessible and maintain a direct connection with its client. As such, bootstrap nodes are well suited to play the role of relay peers.

### 3.2.3. Wireless scenarios

In some mobile deployments, using RPR may help reducing radio battery usage and bandwidth by the intermediate peers. The service provider may recommend using RPR based on his knowledge of the topology.

## 4. Relationship between SRR and RPR

### 4.1. How RPR works

Peers using RPR MUST maintain a connection with their relay peer(s). This can be done in the same way as establishing a neighbor connection between peers by using the Attach method.

A requirement for RPR is for the source peer to convey their relay peer (or peers) transport address in the request, so the destination peer knows where the relay peer are and send the response to a relay peer first. The request SHOULD include also the requesting peer information enabling the relay peer to route the response back to the right peer.

Note that being a relay peer does not require that the relay peer has more functionality than an ordinary peer. As discussed later, relay peers comply with the same procedure as an ordinary peer to forward messages. The only difference is that there may be a larger traffic burden on relay peers. Relay peers can decide whether to accept a new connection based on their current burden.

### 4.2. How SRR and RPR work together

RPR is not intended to replace SRR. It is better to use these two modes together to adapt to each peer's specific situation. Note that

the informative suggestions on how to transition between SRR and RPR (e.g. compute success rate of RPR, fall back to SRR, etc) are the same with that of DRR. Please refer to DRR document [I-D.ietf-p2psip-drr] for more details. Similarly, the peer can decide whether to try RPR based on other information such as configuration file information. If a relay peer is provided by the service provider, peers MAY prefer RPR over SRR.

## 5. Comparison on cost of SRR and RPR

The major advantage of the use of RPR is that it reduces the number of intermediate peers traversed by the response. By doing that, it reduces the load on those peers' resources like processing and communication bandwidth.

### 5.1. Closed or managed networks

As described in Section 3, many P2P systems run in a closed or managed environment (e.g. carrier networks) so that network administrators would know that they could safely use RPR.

The number of hops for a response in SRR and RPR are listed in the following table. Note that the same illustrative settings can be found in DRR document [I-D.ietf-p2psip-drr].

| Mode | Success | No. of Hops | No. of Msgs |
|---|---|---|---|
| SRR | Yes | log(N) | log(N) |
| RPR | Yes | 2 | 2 |
| RPR(DTLS) | Yes | 2 | 7+2 |

Table 1. Comparison of SRR and RPR in closed networks

From the above comparison, it is clear that:

1) In most cases when N > 4 (2^2), RPR uses fewer hops than SRR. Using a shorter route means less overhead and resource usage on intermediate peers, which is an important consideration for adopting RPR in the cases where the resources such as CPU and bandwidth are limited, e.g. the case of mobile, wireless networks.

2) In the cases when N > 512 (2^9), RPR also uses fewer messages than SRR.

3) In the cases when N < 512, RPR uses more messages than SRR (but still uses fewer hops than SRR). So the consideration on whether using RPR or SRR depends on other factors like using less resources

(bandwidth and processing) from the intermediate peers. Section 4 provides use cases where RPR has better chance to work or where the intermediary resources considerations are important.

### 5.2. Open networks

In open networks where RPR is not guaranteed to work, RPR can fall back to SRR if it fails after trial, as described in Section 4. Based on the same settings of Section 5.1, the number of hops, number of messages for a response in SRR and RPR are listed in the following table.

| Mode | Success | No. of Hops | No. of Msgs |
|---|---|---|---|
| SRR | Yes | log(N) | log(N) |
| RPR | Yes | 2 | 2 |
| | Fail&Fall back to SRR | 2+log(N) | 2+log(N) |
| RPR(DTLS) | Yes | 2 | 7+2 |
| | Fail&Fall back to SRR | 2+log(N) | 9+log(N) |

Table 2. Comparison of SRR and RPR in open networks

From the above comparison, it can be observed that trying to first use RPR could still provide an overall number of hops lower than directly using SRR. The detailed analysis is same as DRR case and can be found in DRR document [I-D.ietf-p2psip-drr].

## 6. RPR extensions to RELOAD

Adding support for RPR requires extensions to the current RELOAD protocol. In this section, we define the extensions required to the protocol, including extensions to message structure and to message processing.

### 6.1. Basic requirements

All peers MUST be able to process requests for routing in SRR, and MAY support RPR routing requests.

### 6.2. Modification to RELOAD message structure

RELOAD provides an extensible framework to accommodate future extensions. In this section, we define a ForwardingOption structure and present a state-keeping flag to support RPR mode.

#### 6.2.1. State-keeping flag

flag : 0x08 IGNORE-STATE-KEEPING

If IGNORE-STATE-KEEPING is set, any peer receiving this message and which is not the destination of the message SHOULD forward the message with the full via_list and SHOULD NOT maintain any internal state.

### 6.2.2. Extensive routing mode

We first define a new type to define the new option, extensive_routing_mode:

The option value is illustrated in the following figure, defining the ExtensiveRoutingModeOption structure:

```
enum {(0),DRR(1),RPR(2),(255)} RouteMode;
struct {
        RouteMode               routemode;
        OverlayLinkType         transport;
        IpAddressPort           ipaddressport;
        Destination             destinations<1..2^8-1>;
} ExtensiveRoutingModeOption;
```

Note that DRR value in RouteMode is defined in DRR document [I-D .ietf-p2psip-drr].

RouteMode: refers to which type of routing mode is indicated to the destination peer.

OverlayLinkType: refers to the transport type which is used to deliver responses from the destination peer to the relay peer.

IpAddressPort: refers to the transport address that the destination peer should use to send the response to. This will be a relay peer address for RPR.

Destination: refers to the relay peer itself. If the routing mode is RPR, then the destination contains two destinations, which are the relay peer's Node-ID and the sending peer's Node-ID.

## 6.3. Creating a request

### 6.3.1. Creating a request for RPR

When using RPR for a transaction, the sending peer MUST set the IGNORE-STATE-KEEPING flag in the ForwardingHeader. Additionally, the peer MUST construct and include a ForwardingOptions structure in the ForwardingHeader. When constructing the ForwardingOption structure, the fields MUST be set as follows:

1) The type MUST be set to extensive_routing_mode.

2) The ExtensiveRoutingModeOption structure MUST be used for the option field within the ForwardingOptions structure. The fields MUST be defined as follows:

2.1) routemode set to 0x02 (RPR).

2.2) transport set as appropriate for the relay peer.

2.3) ipaddressport set to the transport address of the relay peer that the sender wishes the message to be relayed through.

2.4) destination structure MUST contain two values. The first MUST be defined as type node and set with the values for the relay peer. The second MUST be defined as type node and set with the sending peer's own values.

## 6.4. Request and response processing

This section gives normative text for message processing after RPR is introduced. Here, we only describe the additional procedures for supporting RPR. Please refer to [I-D.ietf-p2psip-base] for RELOAD base procedures.

### 6.4.1. Destination peer: receiving a request and sending a response

When the destination peer receives a request, it will check the options in the forwarding header. If the destination peer can not understand extensive_routing_mode option in the request, it MUST attempt using SRR to return an "Error_Unknown_Extension" response (defined in Section 6.3.3.1 and Section 14.9 of [I-D.ietf-p2psip-base]) to the sending peer.

If the routing mode is RPR, the destination peer MUST construct a destination_list for the response with two entries. The first MUST be set to the relay peer Node-ID from the option in the request and the second MUST be the sending peer Node-ID from the option of the request.

In the event that the routing mode is set to RPR and there are not exactly two destinations, the destination peer MUST try to send an "Error_Unknown_Extension" response (defined in Section 6.3.3.1 and Section 14.9 of [I-D.ietf-p2psip-base]) to the sending peer using SRR.

After the peer constructs the destination_list for the response, it sends the response to the transport address which is indicated in the ipaddressport field in the option using the specific transport mode in the Forwardingoption. If the destination peer receives a retransmit with SRR preference on the message it is trying to response to now, the responding peer SHOULD abort the RPR response and use SRR.

### 6.4.2. Sending peer: receiving a response

Upon receiving a response, the peer follows the rules in [I-D.ietf-p2psip-base]. If the sender used RPR and does not get a response until the timeout, it MAY either resend the message using RPR but with a different relay peer (if available), or resend the message using SRR.

### 6.4.3. Relay peer processing

Relay peers are designed to forward responses to peers who are not publicly reachable. For the routing of the response, this document still uses the destination_list. The only difference from SRR is that the destination_list is not the reverse of the via_list. Instead, it is constructed from the forwarding option as described below.

When a relay peer receives a response, it MUST follow the rules in [I-D.ietf-p2psip-base]. It receives the response, validates the message, re-adjust the destination_list and forward the response to the next hop in the destination_list based on the connection table. There is no added requirement for relay peer.

## 7. Discovery of relay peers

There are several ways to distribute the information about relay peers throughout the overlay. P2P network providers can deploy some relay peers and advertise them in the configuration file. With the configuration file at hand, peers can get relay peers to try RPR. Another way is to consider relay peer as a service and then some service advertisement and discovery mechanism can also be used for discovering relay peers, for example, using the same mechanism as used in TURN server discovery in base RELOAD [I-D.ietf-p2psip-base]. Another option is to let a peer advertise his capability to be a relay in the response to ATTACH or JOIN.

## 8. Security Considerations

As a routing alternative, the security part of RPR conforms to section 13.6 of the base draft [I-D.ietf-p2psip-base] which describes routing security. RPR behave like a DRR requesting node towards the destination node. The RPR relay node is not an arbitrary node but SHOULD be a trusted one (managed network, bootstrap nodes or configured relay) which will make it less of a risk as outlined in section13 of the based draft.

## 9. IANA Considerations

### 9.1. A new RELOAD Forwarding Option

A new RELOAD Forwarding Option type is added to the Forwarding Option Registry defined in [I-D.ietf-p2psip-base].

Type: 0x02 - extensive_routing_mode

## 10. Acknowledgements

David Bryan has helped extensively with this document, and helped provide some of the text, analysis, and ideas contained here. The authors would like to thank Ted Hardie, Narayanan Vidya, Dondeti Lakshminath, Bruce Lowekamp, Stephane Bryant, Marc Petit-Huguenin and Carlos Jesus Bernardos Cano for their constructive comments.

## 11. References

### 11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[I-D.ietf-p2psip-base] Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", draft-ietf-p2psip-base-26 (work in progress), February 2013.

### 11.2. Informative References

[ChurnDHT] Rhea, S., "Handling Churn in a DHT", Proceedings of the USENIX Annual Technical Conference. Handling Churn in a DHT, June 2004.

[DTLS] Modadugu, N., Rescorla, E., "The Design and Implementation of Datagram TLS", 11th Network and Distributed System Security Symposium (NDSS), 2004.

[RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using STUN", RFC5780, May 2010.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", RFC5382, October 2008.

[I-D.lowekamp-mmusic-ice-tcp-framework] Lowekamp, B. and A. Roach, "A Proposal to Define Interactive Connectivity Establishment for the Transport Control Protocol (ICE-TCP) as an Extensible Framework", draft-lowekamp-mmusic-ice-tcp-framework-00 (work in progress), October 2008.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[I-D.ietf-p2psip-drr] Zong, N., Jiang, X., Even, R. and Zhang, Y., "An extension to RELOAD to support Direct Response Routing", draft-ietf-p2psip-drr-05, April 2013.

[I-D.ietf-p2psip-concepts] Bryan, D., Matthews, P., Shim, E., Willis, D., and S. Dawkins, "Concepts and Terminology for Peer to Peer SIP", draft-ietf-p2psip-concepts-04 (work in progress), October 2011.

## 12. References

## Appendix A. Optional methods to investigate peer connectivity

This section is for informational purposes only for providing some mechanisms that can be used when the configuration information does not specify if RPR can be used. It summarizes some methods which can be used for a peer to determine its own network location compared

with NAT. These methods may help a peer to decide which routing mode it may wish to try. Note that there is no foolproof way to determine if a peer is publically reachable, other than via out-of-band mechanisms. As such, peers using these mechanisms may be able to optimize traffic, but must be able to fall back to SRR routing if the other routing mechanisms fail.

For RPR to function correctly, a peer may attempt to determine whether it is publicly reachable. If it is not, RPR may be chosen to route the response with the help from relay peers, or the peers should fall back to SRR. NATs and firewalls are two major contributors preventing RPR from functioning properly. There are a number of techniques by which a peer can get its reflexive address on the public side of the NAT. After obtaining the reflexive address, a peer can perform further tests to learn whether the reflexive address is publicly reachable. If the address appears to be publicly reachable, the peers to which the address belongs can be a candidate to serve as a relay peer. Peers which are not publicly reachable may still use RPR to shorten the response path with the help from relay peers.

Some conditions are unique in P2PSIP architecture which could be leveraged to facilitate the tests. In P2P overlay network, each peer only has partial a view of the whole network, and knows of a few peers in the overlay. P2P routing algorithms can easily deliver a request from a sending peer to a peer with whom the sending peer has no direct connection. This makes it easy for a peer to ask other peers to send unsolicited messages back to the requester.

The approaches for a peer to get the addresses needed for the further tests, as well as the test for learning whether a peer may be publicly reachable is same as the DRR case. Please refer to DRR document [I-D.ietf-p2psip-drr] for more details.

## Authors' Addresses

Ning Zong
Huawei Technologies

Email: zongning@huawei.com

Xingfeng Jiang
Huawei Technologies

Email: jiang.x.f@huawei.com

Roni Even
Huawei Technologies

Email: roni.even@mail01.huawei.com

Yunfei Zhang

Email: hishigh@gmail.com