P2PSIP
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2011

X. Jiang
N. Zong
Huawei Technologies
R. Even
Gesher Erove
Y. Zhang
China Mobile

March 10, 2011

An extension to RELOAD to support Direct Response and Relay Peer routing
draft-jiang-p2psip-relay-05

Abstract

This document proposes an optional extension to RELOAD to support direct response and relay peer routing modes. RELOAD recommends symmetric recursive routing for routing messages. The new optional extensions provide a shorter route for responses reducing the overhead on intermediary peers and describe the potential cases where these extensions 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 September 11, 2011.

Table of Contents

1. Introduction

1.1. Backgrounds

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 D in [I-D.ietf-p2psip-base]. As we show in section 3, DRR and RPR are advantageous over RPR in some scenarios reducing load (CPU and link BW) on intermediary peers . For example, in a closed network where every node is in the same address realm, DRR performs better than SRR. On the other hand, 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 these three routing modes together is more likely to bring benefits than if SRR is used alone. Some discussion on connectivity is in Non-Transitive Connectivity and DHTs [http://srhea.net/papers/ntr-worlds05.pdf].

In this draft, we first discuss the problem statement, then the relationship between the three routing modes is presented. In Section 5, we give comparison on the cost of SRR, DRR and RPR in both managed and open networks. An extension to RELOAD to support DRR and RPR is proposed in Section 6.

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 [I-D.ietf-behave-nat-behavior-discovery]. Other terms used in this document are defined inline when used and are also defined below for reference.

There are two types of roles in the RELOAD architecture: peer and client. Node is used when describing both peer and client. In discussions specific to behavior of a peer or client, the term peer or client is used instead.

Publicly Reachable: A node is publicly reachable if it can receive unsolicited messages from any other node in the same overlay. Note:

"publicly" does not mean that the nodes 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 nodes in the overlay and forward the responses from destination peers towards the request sender.

Direct Response Routing (DRR): refers to a routing mode in which responses to P2PSIP requests are returned to the sending peer directly from the destination peer based on the sending peer's own local transport address(es). For simplicity, the abbreviation DRR is used instead in the following text.

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 following text.

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

3. Problem Statement

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. Some run in closed networks of small scale. SRR works in any situation, but DRR and RPR may work better in some specific scenarios.

3.1. Overview

RELOAD is a simple request-response protocol. After sending a request, a node waits for a response from a destination node. There are several ways for the destination node to send a response back to the source node. In this section, we will provide detailed information on three routing modes: SRR, DRR and RPR.

Some assumptions are made in the following illustrations.

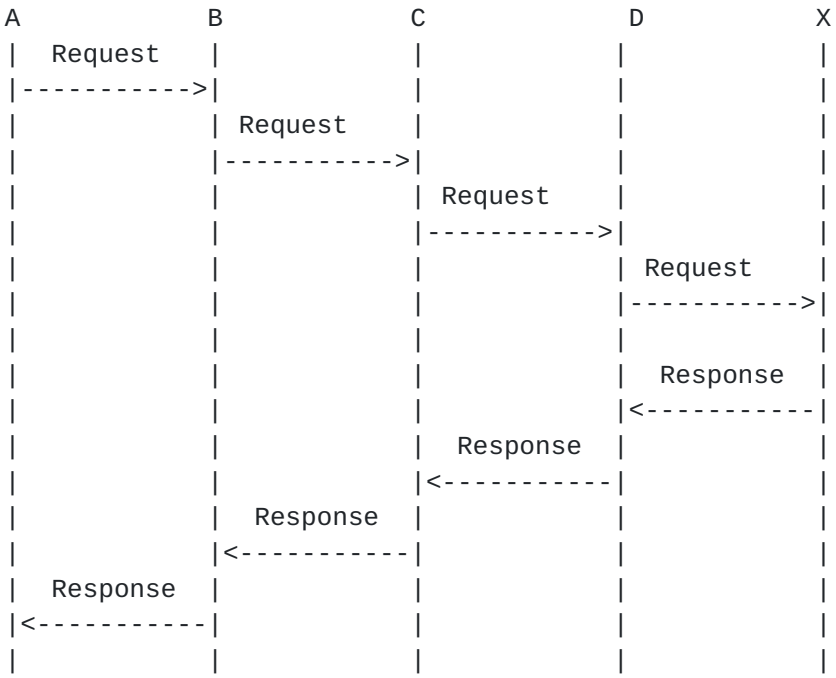
1) Peer A sends a request destined to a peer who is the responsible peer for Resource-ID k;

2) Peer X is the root peer being responsible for resource k;

3) The intermediate peers for the path from A to X are peer B, C, D.

3.1.1. Symmetric Recursive Routing (SRR)

For SRR, when the request sent by peer A is received by an intermediate peer B, C or D, each intermediate peer will insert information on the peer from whom they got the request in the via-list as described in RELOAD. As a result, the destination peer X will know the exact path which the request has traversed. Peer X will then send back the response in the reverse path by constructing a destination list based on the via-list in the request.

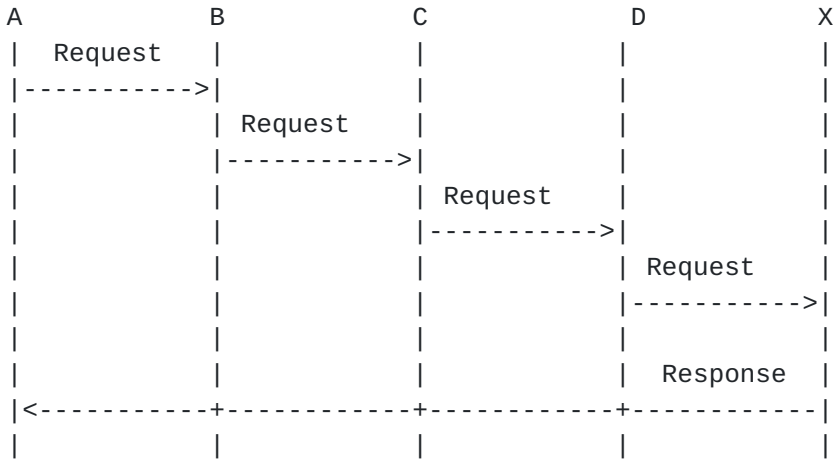


SRR works in any situation, especially when there are NATs or firewalls. A downside of this solution is that the message takes several hops to return to the client, increasing the bandwidth usage and CPU/battery load of multiple nodes.

3.1.2. Direct Response Routing (DRR)

In DRR, peer X receives the request sent by peer A through intermediate peer B, C and D, as in SRR. However, peer X sends the response back directly to peer A based on peer A's local transport address. In this case, the response won't be routed through intermediate peers. Shorter route means less overhead on intermediary peers, especially in the case of wireless network where the CPU and uplink BW is limited. In the absence of NATs or other connectivity issues, this is the optimal routing technique. Note that secure connection requires multiple round trips. Please refer

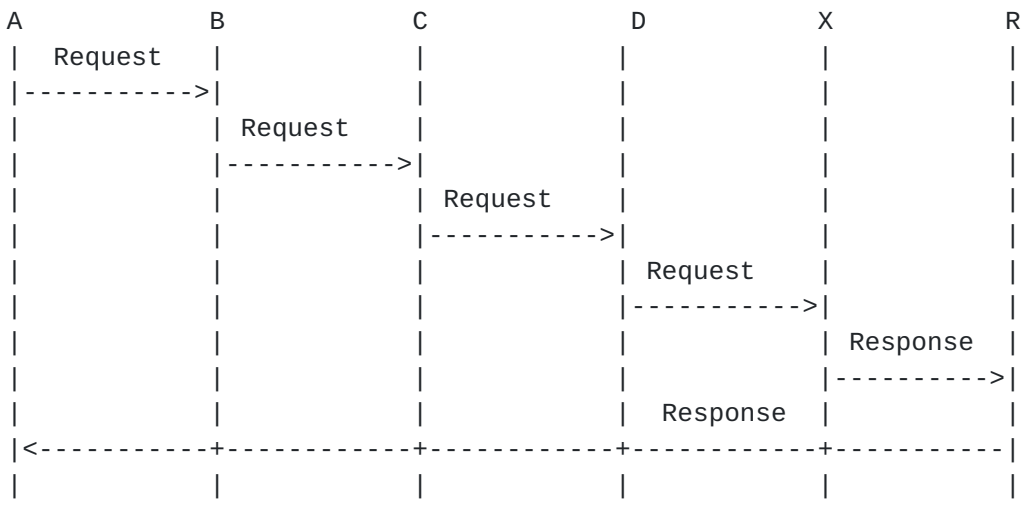
to Section 5 for cost comparison between SRR, DRR/RPR.



3.1.3. 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. Note that RPR also allows a shorter route for responses compared to SRR, which means less overhead on intermediary peers. Establishing a connection to the relay with TLS requires multiple round trips. Please refer to Section 5 for cost comparison between SRR, DRR/RPR.

This technique relies on the relative population of nodes such as A that require relay peers and peers such as R that are capable of serving as a relay peers. It also requires mechanism to enable peers to know which nodes 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 to ATTACH request the peer can signal that it can be used as a relay peer.



3.2. Scenarios Where DRR can be Used

This section lists several scenarios where using DRR would work, and when the increased efficiency would be advantageous.

3.2.1. Managed or Closed P2P System

The properties that make P2P technology attractive, such as the lack of need for centralized severs, self-organization, etc. are attractive for managed systems as well as unmanaged systems. Many of these systems are deployed on private network where nodes are in the same address realm and/or can directly route to each other. In such a scenario, the network administrator can indicate preference for DRR in the peer's configuration file. Peers in such a system would always try DRR first, but peers must also support SRR in case DRR fails. If during the process of establishing a direct connection the responding peer receives a retransmit on a request with SRR as the preferred routing mode he should stop trying to establish a direct connection and use SRR. A node can keep a list of unreachable nodes based on trying DRR and use only SRR for these nodes. The advantage in using DRR is on the network stability since it puts less overhead on the intermediary peers that will not route the responses. The intermediary peers will need to route less messages and save CPU resources as well as the link bandwidth usage.

3.2.2. Wireless Scenarios

While some mobile deployments may use clients, in mobile networks with full peers, there is an advantage to using DRR in order to reduce the load on intermediary nodes. Using DRR helps with reducing radio battery usage and bandwidth by the intermediary peers. The service provider may recommend in the configuration using DRR based on his knowledge of the topology.

3.3. Scenarios Where RPR Benefits

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

3.3.1. Managed or Closed P2P System

As described in Section 3.2.1, 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. As with DRR, 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.

3.3.2. Using Bootstrap Peers as Relay Peers

Bootstrap peers must be publicly reachable in a RELOAD architecture. As a result, one possible architecture would be to use the bootstrap peers as relay peers for use with RPR. The requirements for being a relay peer are publicly accessible and maintaining a direct connection with its client. As such, bootstrap peers are well suited to play the role of relay peers.

3.3.3. Wireless Scenarios

While some mobile deployments may use clients, in mobile networks using peers, RPR, like DRR, may reduce radio battery usage and bandwidth usage by the intermediary peers. The service provider may recommend in the configuration using RPR based on his knowledge of the topology. Such relay peers may also help connectivity to external networks.

4. Relationship Between SRR and DRR/RPR

4.1. How DRR Works

DRR is very simple. The only requirement is for the source peers to provide their (publically reachable) transport address to the destination peers, so that the destination peer knows where to send the response. Responses are sent directly to the requesting peer.

4.2. How RPR Works

RPR is a bit more complicated than DRR. 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.

(Editor's Note: Being a relay peer does not require that the relay peer have 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.)

4.3. How These Three Routing Modes Work Together

DRR and RPR are not intended to replace SRR. As seen from Section 3, DRR or RPR have better performance in some scenarios, but have limitations as well, see for example section 4.3 in Non-Transitive Connectivity and DHTs [http://srhea.net/papers/ntr-worlds05.pdf]. As a result, it is better to use these three modes together to adapt to each peer's specific situation. In this section, we give some suggestions on how to transition between the routing modes in RELOAD.

Editor's Note: What this draft proposes are optional extensions to support DRR/RPR. There is no requirement for implementation to use the strategy described to choose the appropriate mode.

A peer can collect statistical data on the success of the different routing modes based on previous transactions and keep a list of non-reachable addresses. Based on the data, the peer will have a clearer view about the success rate of different routing modes. Other than the success rate, the peer can also get data of fine granularity, for example, the number of retransmission the peer needs to achieve a desirable success rate.

A typical strategy for the node is as follows. A node chooses to start with DRR or RPR. Based on the success rate as seen from the lost message statistics or responses that used SRR, the node can either continue to offer DRR/RPR first or switch to SRR.

The node can decide whether to try DRR or RPR based on other information such as configuration file information. If an overlay runs within a private network and all nodes in the system can reach each other directly, nodes may send most of the transactions with DRR. If a relay peer is provided by the service provider, nodes may prefer RPR over SRR.

5. Comparison on cost of SRR and DRR/RPR

The major advantages in using DRR/RPR are in going through less intermediary peers on 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 DRR/RPR.

SRR brings out more routing hops than DRR and RPR. Assuming that there are N nodes in the P2P system and Chord is applied for routing, the number of hops for a response in SRR, DRR and RPR are listed in the following table. Establishing a secure connection between sending/relay peer and responding peer with (D)TLS requires multiple messages. Note that establishing (D)TLS secure connections for P2P overlay is not optimal in some cases, e.g. direct response routing where (D)TLS is heavy for temporary connections. Instead, some alternate security techniques, e.g. using public keys of the destination to encrypt the messages, signing timestamps to prevent reply attacks can be adopted. Therefore, in the following table, we show the cases of: 1) no (D)TLS in DRR/RPR; 2) still using DTLS in DRR/RPR as sub-optimal and, as the worst-cost case, 7 messages are used during the DTLS handshaking [DTLS]. (TLS Handshake is two round-trip negotiation protocol while DTLS handshake is three round-trip negotiation protocol.)

Mode	Success	No. of Hops	No. of Msgs
SRR	Yes	logN	logN
DRR	Yes	1	1
RPR	Yes	2	2
DRR(DTLS)	Yes	1	7+1
RPR(DTLS)	Yes	2	7+2

From the above comparison, it is clear that:

1) In most cases of N > 2^2=4, DRR/RPR has fewer hops than SRR. Shorter route means less overhead and resource usage on intermediary peers, which is an important consideration for adopting DRR/RPR in the cases where the resource such as CPU and BW is limited, e.g. the case of mobile, wireless network.

2) In the cases of N > 2^9=512, DRR/RPR also has fewer messages than SRR.

3) In the cases where 4 < N < 512, DRR/RPR has more messages than SRR (but still has fewer hops than SRR). So the consideration to use DRR/RPR or SRR depends on other factors like using less resources (bandwidth and processing) from the intermediaries peers. Section 4 provides use cases where DRR/RPR has better chance to work or where the intermediary resources considerations are important.

5.2. Open networks

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

Mode	Success	No. of Hops	No. of Msgs
SRR	Yes	logN	logN
DRR	Yes	1	1
	Fail&Fall back to SRR	1+logN	1+logN
RPR	Yes	2	2
	Fail&Fall back to SRR	2+logN	2+logN
DRR(DTLS)	Yes	1	7+1
	Fail&Fall back to SRR	1+logN	8+logN
RPR(DTLS)	Yes	2	7+2
	Fail&Fall back to SRR	2+logN	9+logN

From the above comparison, it can be observed that:

1) Trying DRR/RPR would still have a good chance of fewer hops than SRR. Suppose that P peers are publicly reachable, the number of hops in DRR and SRR is P*1+(N-P)*(1+logN), N*logN, respectively. The condition for fewer hops in DRR is P*1+(N-P)*(1+logN) < N*logN, which is P/N > 1/logN. This means that when the number of peers N grows, the required ratio of publicly reachable peers P/N for fewer hops in DRR decreases. Similar analysis can be easily applied to RPR. Therefore, the chance of trying DRR/RPR with fewer hops than SRR becomes better as the scale of the network increases.

2) In the cases of large network and the success rate of DRR/RPR is good, it is still possible that DRR/RPR has fewer messages than SRR. Otherwise, the consideration to use DRR/RPR or SRR depends on other factors like using less resources from the intermediaries peers.

6. Extensions to RELOAD

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

6.1. Basic Requirements

All peers implementing DRR or RPR MUST support SRR.

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

Peers that do not support or do not wish to provide DRR or RPR MAY reject these messages.

6.2. Modification To RELOAD Message Structure

RELOAD provides an extensible framework to accommodate future extensions. In this section, we define a ForwardingOption structure to support DRR and RPR modes. Additionally we present a state-keeping flag to inform intermediate peers if they are allowed to not maintain state for a transaction.

6.2.1. State-keeping Flag

RELOAD allows intermediate peers to maintain state in order to implement SRR, for example for implementing hop-by-hop retransmission. If DRR or RPR is used, the response will not follow the reverse path, and the state in the intermediate peers won't be cleared until such state expires. In order to address this issue, we propose a new flag, state-keeping flag, in the message header to indicate whether the state should be maintained in the intermediate peers.

flag : 0x3 IGNORE-STATE-KEEPING

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

6.2.2. Extensive Routing Mode

This draft introduces a new forwarding option for an extensive routing mode. This option conforms to the description in section 5.3.2.3 in [I-D.ietf-p2psip-base].

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

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

```
enum { 0x0, 0x01 (DRR), 0x02(RPR), 255} RouteMode;
struct {
        RouteMode               routemode;
        OverlayLink             transport;
        IpAddressPort          ipaddressport;
        Destination             destination<1..2>;
} ExtensiveRoutingModeOption;
```

The above structure reuses: Transport, Destination and IpAddressPort structure defined in section 5.3.1.1 and 5.3.2.2 in [I-D.ietf-p2psip-base].

Route mode: refers to which type of routing mode is indicated to the destination peer. Currently, only DRR and RPR are defined.

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

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

Destination: refers to the relay peer or the sending node itself. if the routing mode is DRR, then the destination only contains the sending node's node-id; If the routing mode is RPR, then the destination contains two destinations, which are the relay peer's node-id and the sending node's node-id.

6.3. Creating a Request

6.3.1. Creating a Request for DRR

When using DRR 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_TYPE.

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 0x01 (DRR).

2.2) Transport set as appropriate for the sender.

2.3) IPAddressPort set to the peer's associated transport address.

2.4) The destination structure MUST contain one vaule, defined as type peer and set with the sending peer's own values.

6.3.2. 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_TYPE.

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 peer and set with the values for the relay peer. The second MUST be defined as type peer and set with the sending peer's own values.

6.4. Request And Response Processing

This section gives normative text for message processing after DRR and RPR are introduced. Here, we only describe the additional

procedures for supporting DRR and 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 to use SRR to return a error response to the sending peer.

If the routing mode is DRR, the peer MUST construct the Destination list for the response with only one entry, using the sending peer's node-id from the option in the request as the value.

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 node node-id from the option of the request.

In the event that the routing mode is set to DRR and there is not exactly one destination, or the routing mode is set to RPR and there are not exactly two destinations the destination peer MUST try to send a error response 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 he is trying to response to now, the responding peer should abort the DRR/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]. The peer should note if DRR worked in order to decide if to offer DRR again. If the peer does not receive a response until the timeout it SHOULD resend the request using SRR.

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 nodes who are not publicly reachable. For the routing of the response, this draft

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 Peer

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.

Editor note: This section will be extended if we adopt RPR, but like other configuration information, there may be many ways to obtain this.

8. Optional Methods to Investigate Node Connectivity

This section is for informational purposes only for providing some mecahnsism that can be used when the configuration information does not specify if DRR or RPR can be used. It summarizes some methods which can be used for a node to determine its own network location compared with NAT. These methods may help a node to decide which routing mode it may wish to try. Note that there is no foolproof way to determine if a node 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 DRR and RPR to function correctly, a node 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. If the peer believes it is publically reachable, DRR may be attempted. NATs and firewalls are two major contributors preventing DRR and RPR from functioning properly. There

are a number of techniques by which a node 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 nodes to which the address belongs can use DRR for responses and can also be a candidate to serve as a relay peer. Nodes 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 node only has partial a view of the whole network, and knows of a few nodes in the overlay. P2P routing algorithms can easily deliver a request from a sending node to a peer with whom the sending node has no direct connection. This makes it easy for a node to ask other nodes to send unsolicited messages back to the requester.

In the following sections, we first introduce several ways for a node to get the addresses needed for the further tests. Then a test for learning whether a peer may be publicly reachable is proposed.

8.1. Getting Addresses To Be Used As Candidates for DRR

In order to test whether a peer may be publicly reachable, the node should first get one or more addresses which will be used by other nodes to send him messages directly. This address is either a local address of a node or a translated address which is assigned by a NAT to the node.

STUN is used to get a reflexive address on the public side of a NAT with the help of STUN servers. There is also a STUN usage [I-D.ietf-behave-nat-behavior-discovery] to discover NAT behavior. Under RELOAD architecture, a few infrastructure servers can be leveraged for this usage, such as enrollment servers, diagnostic servers, bootstrap servers, etc.

The node can use a STUN Binding request to one of STUN servers to trigger a STUN Binding response which returns the reflexive address from the server's perspective. If the reflexive transport address is the same as the source address of the Binding request, the node can determine that there likely is no NAT between him and the chosen infrastructure server. (Certainly, in some rare cases, the allocated address happens to be the same as the source address. Further tests will detect this case and rule it out in the end.). Usually, these infrastructure severs are publicly reachable in the overlay, so the node can be considered publicly reachable. On the other hand, with the techniques in [I-D.ietf-behave-nat-behavior-discovery], a node can also decide whether it is behind NAT with endpoint-independent

mapping behavior. If the node is behind a NAT with endpoint-independent mapping behavior, the reflexive address should also be a candidate for further tests.

UPnP-IGD is a mechanism that a node can use to get the assigned address from its residential gateway and after obtaining this address to communicate it with other nodes, the node can receive unsolicited messages from outside, even though it is behind a NAT. So the address obtained through the UPnP mechanism should also be used for further tests.

Another way that a node behind NAT can use to learn its assigned address by NAT is NAT-PMP. Like in UPnP-IGD, the address obtained using this mechanism should also be tested further.

The above techniques are not exhaustive. These techniques can be used to get candidate transport addresses for further tests.

8.2. Public Reachability Test

Using the transport addresses obtained by the above techniques, a node can start a test to learn whether the candidate transport address is publicly reachable. The basic idea for the test is for a node to send a request and expect another node in the overlay to send back a response. If the response is received by the sending node successfully and also the node giving the response has no direct connection with the sending node, the sending node can determine that the address is probably publicly reachable and hence the node may be publicly reachable at the tested transport address.

In P2P overlay, a request is routed through the overlay and finally a destination peer will terminate the request and give the response. In a large system, there is a high probability that the destination peer has no direct connection with the sending node. Especially in RELOAD architecture, every node maintains a connection table. So it is easier for a node to check whether it has direct connection with another node.

Note: Currently, no existing message in base RELOAD can achieve the test. In our opinion, this kind of test is within diagnostic scope, so authors hope WG can define a new diagnostic message to do that. We don't plan to define the message in this document, for the objective of this draft is to propose an extension to support DRR and RPR. The following text is informative.

If a node wants to test whether its transport address is publicly reachable, it can send a request to the overlay. The routing for the test message would be different from other kinds of requests because

it is not for storing/fetching something to/from the overlay or locating a specific node, instead it is to get a peer who can deliver the sending node an unsolicited response and which has no direct connection with him. Each intermediate peer receiving the request first checks whether it has a direct connections with the sending peer. If there is a direct connection, the request is routed to the next peer. If there is no direct connection, the intermediate peer terminates the request and sends the response back directly to the sending node with the transport address under test.

After performing the test, if the peer determines that it may be publicly reachable, it can try DRR in subsequent transaction, and may advertise that it is a candidate to serve as a relay peer.

9. Security Considerations

TBD

10. IANA Considerations

10.1. A new RELOAD Forwarding Option

A new RELOAD Forwarding Option type is add to the Registry.

Type: 0x1 - extensive_routing_mode

11. 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 and Bruce Lowekamp for their constructive comments.

12. References

12.1. Normative References

[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-12 (work in progress), March 2010.

[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-03 (work in progress), October 2010.

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

12.2. 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.

[I-D.ietf-behave-nat-behavior-discovery] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using STUN", draft-ietf-behave-nat-behavior-discovery-04 (work in progress), July 2008.

[I-D.ietf-behave-tcp] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", draft-ietf-behave-tcp-08 (work in progress), September 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.

Authors' Addresses

Xingfeng Jiang
Huawei Technologies

Email: jiang.x.f@huawei.com

Ning Zong
Huawei Technologies

Email: zongning@huawei.com

Roni Even
Gesher Erove

Email: ron.even.tlv@gmail.com

Yunfei Zhang
China Mobile

Email: zhangyunfei@chinamobile.com