An extension to RELOAD to support Direct Response and Relay Peer routing
draft-jiang-p2psip-relay-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 April 18, 2009.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

This document proposes an extension to RELOAD to support direct response and relay peer routing modes. The document starts with the problem statement and address concerns about these two routing modes mentioned in RELOAD. Then methods about how to discover NAT behavior of the client in P2PSIP situations are discussed. The last part proposes extensions to RELOAD for supporting these additional routing modes.
Table of Contents

1. Introduction .............................................. 3
2. Terminology .............................................. 3
3. Problem Statement ........................................ 4
   3.1. Symmetric Route Stability ............................ 4
   3.2. Application Scenarios ............................... 5
   3.3. Advantages from DRR and RPR ....................... 5
4. Concerns From RELOAD ..................................... 6
   4.1. Concerns About DRR ................................ 6
   4.2. Concerns About RPR ................................ 6
5. How To Identify The Role Of A Peer ....................... 7
   5.1. Discovery Of NAT Mapping Behavior .................. 7
   5.2. Discovery Of NAT Filtering Behavior ............... 8
6. Discovery Of Relay Peer .................................. 9
7. Relationship Between SRR And DRR/RPR .................... 10
8. Extensions To RELOAD ..................................... 10
   8.1. Steps For Running DRR/RPR .......................... 10
   8.2. Needed Modifications To RELOAD .................... 11
   8.3. New Message RM_TEST ............................... 11
      8.3.1. Detecting NAT Filtering Behavior ............... 12
      8.3.2. RPR Procedure ................................ 12
   8.4. Modification To RELOAD Message Structure .......... 13
   8.5. Request And Response Processing ................... 14
      8.5.1. Sending Request And Receiving Response ....... 14
      8.5.2. Receiving Request And Sending Response ....... 14
9. An Example ............................................... 15
10. Security Considerations ............................... 16
11. IANA Considerations ................................... 16
   11.1. New REALOAD message code ........................ 16
   11.2. A new REALOAD Forwarding Option .................. 17
12. Acknowledgements ..................................... 17
13. References ............................................. 17
   13.1. Normative References .............................. 17
   13.2. Informative References ............................ 17
Authors’ Addresses ......................................... 18
Intellectual Property and Copyright Statements ............ 19
1. Introduction

RELOAD recommends symmetric recursive routing for routing messages and describes the extensions that would be required to support additional routing algorithms. This document proposes an extension to RELOAD to support two additional routing options: direct response and relay peer which are also discussed in section 3.3.1.3 and 3.3.1.4 respectively in [I-D.ietf-p2psip-reload]. Symmetric recursive routing will not work well in some cases. For example, overlay churn will make the reverse path of the request unstable, therefore fail the transaction. In these cases, direct response or relay peer would be helpful to improve the reliability and efficiency of the P2PSIP transaction. So it is useful to make these three modes work together to provide better service to the upper layer applications on top of the P2PSIP overlay. We analyze the problem statement first and try to address the concerns in RELOAD [I-D.ietf-p2psip-reload] about these two routing options. Then, some methods are presented to show that it is feasible to discover NAT behavior in RELOAD architecture. In section 7 the document discusses the methods to make direct response and relay peer routing mode work together with symmetric routing mode. In the end, an extension to current RELOAD is proposed to make them work.

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.

- **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 addresses. 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 first who will further forward responses towards the sending peer. For simplicity, the abbreviation RPR is used instead in the following text.
o Symmetric Recursive Routing (SRR): refers to a routing mode in which responses follow the same request path in the reverse order to get back to the sending peer. For simplicity, the SRR is used instead in the following text.

o 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 RELOAD protocol may be used in a closed system.

o relay peer: a type of publicly reachable peers that can receive unsolicited messages from all other nodes in the overlay and forward the responses from destination peers towards the request sender. For example, if peer A is behind a p2p-unfriendly NAT, he can use relay peers to traverse the NAT and get the response from the destination peer.

3. Problem Statement

P2PSIP WG intends to make RELOAD work under a number of application scenarios. In reality, the situations where RELOAD is to be deployed differ greatly. For instance, some systems will run in an overlay composed of only ten to hundred stable nodes. On the other hand, overlays may experience heavy churn or comprise of nodes over 1 million. It is obvious that SRR currently adopted in RELOAD works well in the former case. But for the latter case, SRR may not work well.

3.1. Symmetric Route Stability

As discussed in section 3.3.1.5 in RELAOD [I-D.ietf-p2psip-reload], one concern for the motivation to add DRR and RPR is, if there is a heavy churn in overlay, requests fail first. That is, transactions failed before the requests reach the destination peer. In this case, DRR and RPR won’t bring any benefit for the transactions in that these two routing modes are used to avoid the churn influence on the response path. RELOAD mentions that retries of the failed requests will reduce the influence of churn. By using optional DRR and RPR the probability of failure due to churn is also reduced.

In some literatures, for example [ChurnDHT], one way to handle churn in DHT is to get a more accurate timeout value between neighbors. When the timeout fires, the preferred way for the upstream peer to handle the failure is to try the request through another downstream peer, for there may be a large number of paths between a sending peer and a destination peer. Therefore, the request success rate will be improved greatly. This still leave open the issue of the response failure due to churn.
Through the above analysis, we can see that churn in request path could be solved to great extent by using existing mechanisms. The DRR and RPR routing mode are still helpful for the responses to be resilient to the churn.

3.2. Application Scenarios

Overlay churn causes nodes or connections between neighbor nodes to be unstable. If SRR is used for routing response, any failure of nodes or connections in the reverse path will fail the whole P2PSIP transaction. So a transaction with longer response latency will be vulnerable to churn. Below are two examples of overlays which produce slow response.

- P2P technology has been applied successfully in some global public VoIP services, such as Skype. The number of the simultaneous online users for this type of system is often over 1 million. For a typical overlay network, the lookup operation in P2P systems will end up with the hop counts within \([0, \log(N)]\). Here, \(N\) is the number of nodes in the system. For a system where \(N\) is over 1 million, the maximum hop counts would be about 20. For such a long path, the churn of the overlay will break the reverse path with high probability, for the failure of any node in the path would mean the failure of the reverse path, hence the failure of the transaction.

- If links of the overlay network are wireless link or some low speed links, the latency between links would be significantly long. So messages will take longer to traverse the path back and forth and the nodes in the path are more easily subject to failure during the message transportation duration because of churn.

3.3. Advantages from DRR and RPR

DRR and RPR try to shorten the response path to a fixed number of hops and thus reducing the response time. If DRR is used, the response path will be only 1 hop; if RPR is used, the response path will be 2 hops. According to the suggestions in this draft, before the peer start to use the DRR or RPR, a test that decides whether the DRR or RPR could work should be performed. In this way, it could mitigate the concerns about influence from the failure of the DRR and RPR. The main benefit from adding the option to use DRR and RPR with SRR fall-back is that it will improve the success rate and reliability of the P2PSIP transaction, especially for the scenarios described in section 3.2.
4. Concerns From RELOAD

For the DRR and RPR routing modes, the RELOAD authors expressed several concerns in section 3.3.1.3 and section 3.3.1.4 in [I-D.ietf-p2psip-reload] respectively. This section attempts to address these concerns.

4.1. Concerns About DRR

- Concern1: Due to NAT existence, DRR is subject to failure. So it requires the destination peer to detect the failure and then fallback to SRR.
  * Answer: The concern is on how the DRR works together with SRR. In this draft we suggest tests based on the NAT behavior discovery [I-D.ietf-behave-nat-behavior-discovery] that will have good probability to find a publicly reachable address or a relay peer. We can also offer flexibility in which routing mode will be the main one and which will be the fallback (SRR first or DRR/RPR first).
- Concern2: Due to NAT filtering behavior, the NAT would drop the response if it is not a p2p-friendly NAT.
  * Answer: This draft will introduce tests based on [I-D.ietf-behave-nat-behavior-discovery] that the peer needs to do before the DRR is offered in order to discover NAT filtering behavior, so that the DRR will succeed with high probability.
- Concern3: The response happens to be received by another peer who is not the sending peer and hence it adds to the transaction latency.
  * Answer: We think this is a rare case since this draft recommends a method to identify the NAT behavior [I-D.ietf-behave-nat-behavior-discovery]. Even if it would happen, the method proposed in this draft allows the sending peer to decide which routing mode will be used first. On the other hand, as we mentioned before in section 3.3, the main benefit from DRR is improving the transactions success rate in some cases, not only the transaction latency.

4.2. Concerns About RPR

- Concern1: RPR requires the help from relay peers for the sending peer. It is not easy to identify which category a peer is in.
  * Answer: RPR does not intend to replace the SRR. This draft propose RPR and DRR as an enhancement to the SRR. As for the difficulty in identifying a peer’s role, the draft suggests tests for NAT behavior discovery techniques based on [I-D.ietf-behave-nat-behavior-discovery] under RELOAD architecture to achieve the goal. So it is possible for a node in the overlay to detect its location relative to NAT. Using
peer as relays is also common in p2p network where peers can declare themselves as supernodes if they have a publicly reachable address.

5. How To Identify The Role Of A Peer

For DRR and RPR to function correctly, a peer should be able to determine whether it is publicly reachable or not. If it is not, RPR may be used to route the response with the help from relay peers, otherwise, DRR may be offered. NATs and firewalls are two major contributors preventing DRR and RPR from functioning properly.

Draft [I-D.ietf-behave-nat-behavior-discovery] proposed a STUN usage to discover NAT behavior. The STUN usage needs a dedicated STUN server with multiple IP address or it must duplicate the behavior of such a dedicated STUN server with two nodes that share the role of providing the address-changing operations required by this usage. For DRR and RPR in P2P system, NAT address and port mapping and NAT filtering behaviors are the most important characteristics to be discovered. So, in order to make DRR and RPR work, the peer needs to determine if the NAT with respect to the peer has a endpoint-independent mapping property and a endpoint-independent filtering property.

At the same time, some conditions are unique in P2PSIP architecture which could be exploited to make the discovery easier. For example, there are some public servers, such as enrollment servers, bootstrap servers, who could be used as an anchor to perform NAT behavior discovery tests when the node joins the overlay. P2PSIP overlay has enough nodes which could provide help with the test when there is a need for receiving an unsolicited message from another address and port.

The following text, details suggestions on how to discover NAT behavior in P2PSIP architecture.

5.1. Discovery Of NAT Mapping Behavior

In RELOAD architecture there are a couple of servers that may have a publicly reachable address that can be used for discovering the NAT mapping behavior. The peer should look for endpoint independent mapping as a prerequisite to allow DDR or to be a relay peer.

To do that we can use the enrollment server used to get the required information, such as node-id, overlay specific configuration, etc. Usually, the enrollment server is publicly reachable and could be contacted directly by any nodes in the system. So it can act as an
anchor to help nodes to learn whether it is on the same network as the enrollment server. Besides enrollment server, there are other infrastructure servers in the system, depending on the real deployment. For example, bootstrap servers or diagnostic servers may exist in some P2P system for management purpose.

The methods a node uses to determine its location are various and depend on specific deployments. One way is a node can use a variant of STUN during exchanges with one of the servers. For example, when the node contacts with the enrollment server, the enrollment server could reflect the source transport address of the messages from the node, and convey it to the node in the response. The node compares its local transport address with the reflexive address in the response, and then makes the final determination. Another way is to use the STUN protocol as described in the NAT behavior discovery [I-D.ietf-behave-nat-behavior-discovery]. All nodes and server in the P2P system need to support STUN. In either way, the peer could learn through the above procedure whether the NAT has a endpoint-independent mapping behavior or not. Another method to get a reachable address can be based on NAT assisted methods like UPnP as suggested in [I-D.lowekamp-mmusic-ice-tcp-framework]

5.2. Discovery Of NAT Filtering Behavior

After discovering NAT mapping behavior is endpoint-independent, a node needs to continue with discovery of the NAT filtering behavior. The proposed test in [I-D.ietf-behave-nat-behavior-discovery] requires the STUN server has multiple addresses or if a few STUN servers with single address are involved, require coordination among them to complete the test. The test described in [I-D.ietf-behave-nat-behavior-discovery] is applicable to UDP while for RELAOD we need to use TCP address. In P2PSIP system, we can take advantage of the fact that a peer has a built-in mechanism to talk with other nodes with whom he has not contacted before, the routing function in the P2P algorithm can route the request to any peer in the overlay who is responsible for a key in a request. We can use this functionality to define a test for the NAT filtering behavior.

If a node could receive messages from other nodes with whom it does not have a connection, then the remote node could help the testing if the NAT has an endpoint-independent filtering behavior. In current P2PSIP architecture, a node can easily know whether it has existing connections with the unsolicited message sender, for any node maintains a connection table. The peer that wants to learn its NAT filtering behavior can send a request for a key in order to learn from the responder the filtering behavior of the NAT.

In the above description, we assume the destination peer has no
direct connections with the node in question before sending the response. But there are a few rare cases where the assumption does not hold. One case is the request only traverses one hop so that the destination peer is the direct neighbor with the node; another case is the destination peer happens to be the neighbor of the node even if the request traverses several hops. It often depends on the key set in the request. Two methods can address the above problem.

The first one is to have the sending node make the judgment because every node has a connection table. For example, when the sending node receives the response from a peer, it should check whether the response came over an existing connection or not. If yes, then the sending peer should try another key and perform another test.

The second one is to make a little modification to the way the test message is routed. In conventional P2P algorithms, the next hop is often determined by looking up the key in the routing table and the closest node will be chosen accordingly. But for the test purpose, the request is not really for STORE or FETCH values to/from the overlay. Its real purpose is to find a peer in the overlay with whom the peer has no connection. In a sense, intermediate peers need not route the message according to the conventional routing algorithm. Instead, they could do the random routing which is irrelevant to the key in a request. So the peer who has no direct connection with the sending peer would be considered as the destination peer and then sends the response directly to the sending peer. In the second method, the test message is resilient to overlay churn, for the routing is based on the random choice. Moreover it also means that we need to define a new message to perform the test and that all intermediate peers should know how to process the request.

6. Discovery Of Relay Peer

In order to support RPR, a node who is behind p2p-unfriendly NAT must have one or more relay peers to help him receive response from destination peers. The major requirement for relay peers is that they should be publicly reachable. By using the methods proposed in section 5, a peer could determine whether it is publicly reachable, i.e., with an endpoint-independent mapping and filtering behavior.

Generally, P2PSIP overlay needs a service discovery mechanism by means of which the node in the overlay could find which peers can provide which kind of service. The capability of the relay peers could be defined as a service and discovering relay peers means that peers use the general service mechanism to find which peer can provide relay peer capability.
7. Relationship Between SRR And DRR/RPR

DRR/RPR and SRR have their advantages and disadvantages. A major issue should be addressed before we design ways to combine these two kinds of routing modes. It is in which order the DRR/RPR or SRR is attempted. In order to make full use of advantages of every routing mode, the peer should decide which one is tried first. There are several strategies. If a peer wants to get quick response, it can try DRR/RPR first; if failed, then fallback to SRR. In this draft, DRR/RPR test SHOULD be passed before DRR/RPR is used for message routing so that DRR/RPR would work with high success rate. On the other hand, a peer can also try SRR first and then DRR/RPR would be used as a fallback. This draft does not mandate which order will be used. It depends on specific scenarios. This draft just provides available tools to give the peers different choices to deal with different situations. The decision if to use DRR/RPR first can also be based on past success percentage of this mode.

8. Extensions To RELOAD

Adding the support of DRR and RPR requires extensions to the current RELOAD. In this section, first we show the steps taken by a node in order to use DRR and RPR. Then based on the results, we could point out the difference from the current RELOAD and define the required extensions.

8.1. Steps For Running DRR/RPR

DRR and RPR may fail in the presence of NAT. So in order to improve the success rate of DRR and RPR, we need to do some necessary tests before a node can use it. Below are the required steps to make DRR and RPR work.

1. When a peer X joins the overlay, it uses proposed mechanism in section 5.1 to determine its NAT mapping behavior.
2. After X becomes a full-functional peer in the overlay, If the previous test shows that his NAT has a endpoint-independent mapping behavior, he will continue with the proposed mechanism in section 5.2 to determine its NAT filtering behavior.
3. Based on the above two tests, if peer X learns that it is behind a p2p-friendly NAT or non-NATed, i.e, endpoint-independent mapping and filtering behavior, the peer could try DRR in the future message. Moreover, he is also a candidate to be a relay peer. In order to adapt to the change of NAT mapping and filtering, peer X SHOULD periodically perform the tests.
4. If peer X is behind a p2p-unfirendly NAT, peer X tries to get relay peers with service discovery mechanism. Peer X should try to attach to one of the relay peers to verify that it can use it as a relay.

5. Peer X can offer DRR or RPR respectively depending on peer X location with regard to NAT. This offer is done by adding extensive_routing_mode structure in the forwardingHeader to each request and using the preference flag in the extensive_routing_mode structure to specify if SRR or DRR/RPR are to be used first. If the preference flag equals to 0 then DRR/RPR should be tried first otherwise SRR should be used first.

8.2. Needed Modifications To RELOAD

In order to support the DRR/RPR, we need to make the following modifications to RELOAD.

1. Define a forwarding option in the forward header to indicate which routing mode can be used. Depending on different routing mode, the information in the option value field is different. For example, if DRR is used, the information about sending nodes MUST be included; If RPR is used, the information about the send node’s relay peers MUST be included.

2. Define a new message to perform tests that will be used to detect NAT filtering behavior and to test whether DRR or RPR works. In this draft, the new message is named RM_TEST( Routing Mode Test ).

3. Add a new flag bit (hop-by-hop flag: 0x04) in the flags field in the forward header. This flag is used to indicate to the intermediate peers that they need to know the semantics of the message and take some actions accordingly. As mentioned in section 5.2, in order to make sure the destination peer has no direct connection with the sending peer, every intermediate peer determines whether it has existing connections with the sending peer. So it requires the intermediate peers involvement to achieve the test. Of course, it could use other methods from section 5.2 to avoid define a new flag. At this stage, we think the flag is useful, but it is subject to change depending on working group’s consensus.

8.3. New Message RM_TEST

RM_TEST serves two purposes: detecting NAT filtering behavior and checking whether DRR or RPR works. The following is the normative procedure for processing the RM_TEST.
8.3.1. Detecting NAT Filtering Behavior

In order to detect NAT filtering behavior, the sending peer MUST determine whether his NAT has an endpoint-independent mapping behavior first as specified in [I-D.ietf-behave-nat-behavior-discovery]. Then sending peer sets the routing mode as DRR in the forwarding header, in which one of the sending peer’s reflexive transport address is also carried. **It is REQUIRED that in each test message, only one reflexive transport address is included.** At the same time, set the hop-by-hop flag 0x04 in the forwarding header. The key in the destination list SHOULD be a random one.

[Open issue: because the RM_TEST message is not for real data or management operations, just for a test, do we need a specialized ID which indicates to the intermediate peers that the destination peer for the message depends on the semantics of the message type?]

When an intermediate peer receives the message, it checks the hop-by-hop flags first. If the flag is set, then get the message type. If the message type is RM_TEST, get the routing mode and its associated information, i.e., transport address information. If the intermediate peer has no existing connections with the sending peer after checking its connection table based on the node-id, then he can act as the destination peer for the RM_TEST message, constructs a response and sends directly to the sending peer. If the intermediate peer has existing connections, it forwards the request further.

The sending peer waits for the RM_TEST response by setting a retransmission timer. If the response succeeds in returning to the sending peer, the sending peer can determine that his NAT has endpoint-independent filtering behavior. To avoid the unreliability of the underlying network, the sending peer will try the RM_TEST several times before he gives up the test.

8.3.2. RPR Procedure

If a peer is behind a p2p-unfriendly NAT after doing the NAT behavior tests, and would like to offer RPR it MUST get relay peers first. A peer that declared itself as a relay peer have already done the NAT behavior tests and have at least a publicly reachable address. Furthermore a peer may learn about more than one available relay peers.

After getting the information of relay peers, the peer uses Attach method to establish direct connections with one of the possible relay peers. A relay peer may reject the Attach request if it is currently loaded, in this case the peer may try another relay peer. When a peer attached to a relay peer it can now offer RPR.
8.4. Modification To RELOAD Message Structure

Some necessary modifications to RELOAD message structure are explained in the following text.

This draft defines a new forwarding option in the ForwardingHeader structure. The new forwarding option must conform to the structure defined in section 6.2.2.3 in [I-D.ietf-p2psip-reload].

type : 0x1 extensive_routing_mode

flag: 0x02 DESTINATION_CRITICAL

The option value will be illustrated in the following figure.

```
enum { 0x0, 0x01 (DRR), 0x02(RPR), 255} RouteMode;
```

```
struct {
    RouteMode routemode;
    uint8 preference;
    Transport transport;
    Destination destination<1..2>;
    IpAddreessPort ipaddressport;
} Extensive_Routing_Mode;
```

The above structure reuses: Transport, Destination and IpAddressPort structure defined in section 6.2.1.1 and 6.2.2.1 in [I-D.ietf-p2psip-reload] respectively. The node-id in Destination structure is useful to match the existing connections.

routemode: refers to which kind of routing mode is preferred to be used among extensive routing modes. Currently, only DRR and RPR are defined.

preference: refers to which kind of routing mode the destination peer will try first. 0x1 means SRR first, 0x0 means the extensive routing mode will be tried first;

Transport: refers to the transport type which will be used to deliver the response from the destination peer to the sending peer or the relay peer;

Destination: refers to the relay peer or the sending peer itself. If
routing type is DRR, then the destination will contain the sending peer’s node-id; If routing type is RPR, then the destination will contain two destinations, which are the relay peer’s node-id and the sending peer’s node-id;

IpAddressPort: refers to the transport address the destination has to use to send the response to.

8.5. Request And Response Processing

This section gives normative text on the normal message processing after DRR and RPR are introduced. Here, we only describe the additional procedures for supporting DRR and RPR and please refer to [I-D.ietf-p2psip-reload] for RELOAD base procedures.

8.5.1. Sending Request And Receiving Response

If a peer can offer DRR, it MUST fill the extensive_routing_mode option first defined in section 8.4 before sending the request out. The routing mode MUST be set to 0x1(DRR); the preference MUST be 0 or 1 which depends on the choice of the sending peer. If the peer wants to try DRR first, then the field MUST be set to 0; otherwise 1. The destination carries the sending peer node-id and the ipaddressport MUST carry only one transport address which has been tested.

If a peer can offer RPR, it MUST fill extensive_routing_mode option too. Routing mode MUST be set to 0x2(RPR). The preference MUST be set to 0 or 1 depending on the choice of the sending peer. The destination MUST hold one relay peer’s node-id and the sending peer’s node-id in a sequential order. The ipaddressport MUST carry one of the relay peer’s transport address which has been tested.

For receiving response, the peer follows the rule in [I-D.ietf-p2psip-reload].

8.5.2. Receiving Request And Sending Response

When the destination peer receives a request, it will check the options in the forwarding header. If the destination peer can understand extensive_routing_mode option, it will check the preference first, if the preference shows SRR first, then the peer will use the SRR to return the response.

If the preference shows DRR/RPR first, then the peer check the routing mode. If the routing mode is DRR, then the peer constructs the Destination lists with only one entry, i.e, the sending peer node-id in the option; if the routing mode is RPR, then the peer construct Destination list with two entries, one is the relay peer
node-id got from the option, the other one is the sending peer node-id which is also carried in the option in RPR case.

After the peer construct the destination list, if the peer will use DRR or RPR, then they will send the response to the transport address which is indicated in the ipaddressport field in the option using the specific transport mode in the option.

9. An Example

In this section, we will illustrate how a peer uses the DRR or RPR to route messages. In the following illustration, we use SP for sending peer, IP for intermediate peer, DP for destination peer, RP for relay peer; The letter H in the figure refers to hop-by-hop flag in the forwarding header.

1. Detect the NAT filtering behavior

```
SP                     IP                     DP
|   RM_TEST(DRR | H)   |                      |
|--------------------->|                      |
|                      |     RM_TEST(DRR | H) |
|                      |--------------------->|
|                      |                      |
|     RM_TEST Resp     |                      |
|<--------------------------------------------|
```

SP sends a RM_TEST request and set the DRR routing mode and hop-by-hop flag in the request. IP will forward the request until the intermediate peer checks that it has no existing connections with SP, then it changes its role to DP. Finally, DP sends a RM_TEST response back to the SP.

2. Sending other messages than RM_TEST with DRR or RPR
The above figures show how DRR and RPR are used to route normal message. The major difference from RM_TEST is that hop-by-hop flag is not required. It may depend on the message in use.

10. Security Considerations

TBD

11. IANA Considerations

11.1. New REALOAD message code

A new RELOAD message code RM_TEST and RM_TEST_ANS are added in this contribution. The new REALOAD message codes should be assigned by IANA.

RM_TEST: 23

RM_TEST_ANS: 24
11.2. A new RELOAD Forwarding Option

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

Type: 0x1 - extensive_routing_mode

12. Acknowledgements

Authors would like to thank Ted Hardie, Narayanan Vidya and Dondeti Lakshminath for their comments.

13. References

13.1. Normative References

[ChurnDHT]

[I-D.ietf-p2psip-concepts]

[I-D.ietf-p2psip-reload]


13.2. Informative References

[I-D.ietf-behave-nat-behavior-discovery]

[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]


Authors’ Addresses

XingFeng Jiang
Huawei Tech.
Huihong Mansion,No.91 Baixia Rd.
Nanjing, Jiangsu  210001
P.R.China

Phone: +86(25)84565867
Email: jiang.x.f@huawei.com

Roni Even
Gesher Erove
14 David Hamelech
Tel Aviv  64953
Israel

Email: ron.even.tlv@gmail.com