P2PSIP
Internet-Draft
Intended status: Standards Track
Expires: October 10, 2013

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

April 08, 2013

An extension to RELOAD to support Direct Response Routing
draft-ietf-p2psip-drr-05

Abstract

This document proposes an optional extension to RELOAD to support direct response 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 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 October 10, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction 3
  1.1. Backgrounds 3
2. Terminology 3
3. Introduction 4
  3.1. Overview 4
    3.1.1. Symmetric Recursive Routing (SRR) 5
    3.1.2. Direct Response Routing (DRR) 5
  3.2. Scenarios where DRR can be used 6
    3.2.1. Managed or closed P2P systems 6
    3.2.2. Wireless scenarios 7
4. Relationship between SRR and DRR 7
  4.1. How DRR works 7
  4.2. How SRR and DRR Work Together 7
5. Comparison on cost of SRR and DRR 7
  5.1. Closed or managed networks 8
  5.2. Open networks 9
6. DRR extensions to RELOAD 9
  6.1. Basic requirements 9
  6.2. Modification to RELOAD message structure 9
    6.2.1. State-keeping flag 10
    6.2.2. Extensive routing mode 10
  6.3. Creating a request 11
    6.3.1. Creating a request for DRR 11
  6.4. Request and response processing 11
    6.4.1. Destination peer: receiving a request and sending a

response 11
6.4.2. Sending peer: receiving a response 12
7. Security considerations 12
8. IANA considerations 12
8.1. A new RELOAD forwarding option 13

9. Acknowledgements 13
10. References 13
10.1. Normative references 13
10.2. Informative references 13
11. References 14
Appendix A. Optional methods to investigate peer connectivity . 14
A.1. Getting addresses to be used as candidates for DRR . . . 15
A.2. Public reachability test 16
Authors' Addresses 17

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 A of [I-D.ietf-p2psip-base]. As we show in section 3, DRR is advantageous over SRR in some scenarios by reducing load (CPU and link bandwidth) on intermediate peers. For example, in a closed network where every peer is in the same address realm, DRR performs better than SRR. In other scenarios, using a combination of DRR and SRR together is more likely to bring benefits than if SRR is used alone.

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

We first discuss the problem statement in Section 3, then how to combine DRR and SRR is presented in Section 4. In Section 5, we give comparison on the cost of SRR and DRR in both managed and open networks. An extension to RELOAD to support DRR is proposed in Section 6. 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.

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 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. Introduction

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 DRR may work better in some specific scenarios.

3.1. Overview

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 two routing modes: SRR and DRR.

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. Figure 1 illustrates SRR.

```
A            B            C             D           X
|  Request   |            |            |            |
|----------->|            |            |            |
|            | Request    |            |            |
|            |----------->|            |            |
|            |            | Request    |            |
|            |            |----------->|            |
|            |            |            | Request    |
|            |            |            |----------->|
|            |            |            |            |
|            |            |            |  Response  |
```

```
|             |             |             |<-----------|
|             |             |  Response   |            |
|             |             |<-----------|            |
|             |  Response   |             |            |
|             |<-----------|             |            |
|  Response   |             |             |            |
|<-----------|             |             |            |
|             |             |             |            |
```

Figure 1, SRR

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 peer, increasing the bandwidth usage and CPU/battery load of multiple peers.

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 is not routed through intermediate peers. Figure 2 illustrates DRR. Using a shorter route means less overhead on intermediate peers, especially in the case of wireless networks where the CPU and uplink bandwidth is limited. In

the absence of NATs or other connectivity issues, this is the optimal routing technique. Note that establishing a secure connection requires multiple round trips. Please refer to Section 5 for cost comparison between SRR and DRR.

```
A            B            C             D           X
|  Request   |            |            |            |
|----------->|            |            |            |
|            | Request    |            |            |
|            |----------->|            |            |
|            |            | Request    |            |
|            |            |----------->|            |
|            |            |            | Request    |
|            |            |            |----------->|
|            |            |            |            |
```

```
|            |            |            |  Response  |
|<-----------+------------+------------+------------|
|            |            |            |            |
```

Figure 2, DRR

3.2. Scenarios where DRR can be used

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

3.2.1. Managed or closed P2P systems

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 networks where peers 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 with the sending peer, the responding peer receives a response with SRR as the preferred routing mode (or it fails to establish the direct connection), the responding peer SHOULD NOT use DRR but switch to SRR. A peer can keep a list of unreachable peers based on trying DRR and use only SRR for these peers. The advantage in using DRR is on the network stability since it puts less overhead on the intermediate peers that will not route the responses. The intermediate peers will need to route less messages and save CPU resources as well as the link bandwidth usage.

3.2.2. Wireless scenarios

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

4. Relationship between SRR and DRR

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 SRR and DRR Work Together

DRR is not intended to replace SRR. It is better to use these two modes together to adapt to each peer's specific situation. In this section, we give some informative suggestions on how to transition between the routing modes in RELOAD.

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 this 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 finer granularity, for example, the number of retransmission the peer needs to achieve a desirable success rate.

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

The peer can decide whether to try DRR based on other information such as configuration file information. If an overlay runs within a private network and all peers in the system can reach each other directly, peers MAY send most of the transactions with DRR.

5. Comparison on cost of SRR and DRR

The major advantages in using DRR are in going through less intermediate 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.

SRR brings out more routing hops than DRR. Assuming that there are N peers in the P2P system and Chord is applied for routing, the number of hops for a response in SRR and DRR are listed in the following table. Establishing a secure connection between the sending peer and the 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, and 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; 2) still using DTLS in DRR as sub-optimal. As the worst-cost case, 7 messages are used during the DTLS handshaking [DTLS]. (TLS Handshake is a two round-trip negotiation protocol while DTLS handshake is a three round-trip negotiation protocol.)

Mode	Success	No. of Hops	No. of Msgs
SRR	Yes	log(N)	log(N)
DRR	Yes	1	1
DRR(DTLS)	Yes	1	7+1

Table 1, comparison of SRR and DRR in closed networks

From the above comparison, it is clear that:

1) In most cases when N > 2 (2^1), DRR 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 DRR 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 > 256 (2^8), DRR also uses fewer messages than SRR.

3) In the cases when N < 256, DRR uses more messages than SRR (but still uses fewer hops than SRR). So the consideration on whether using DRR or SRR depends on other factors like using less resources (bandwidth and processing) from the intermediate peers. Section 4 provides use cases where DRR has better chance to work or where the intermediary resources considerations are important.

5.2. Open networks

In open networks where DRR is not guaranteed to work, DRR 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 and DRR are listed in the following table.

Mode	Success	No. of Hops	No. of Msgs
SRR	Yes	log(N)	log(N)
DRR	Yes	1	1
	Fail&Fall back to SRR	1+log(N)	1+log(N)
DRR(DTLS)	Yes	1	7+1
	Fail&Fall back to SRR	1+log(N)	8+log(N)

Table 2, comparison of SRR and DRR in open networks

From the above comparison, it can be observed that trying to first use DRR could still provide an overall number of hops lower than directly using 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. Therefore, the chance of trying DRR with fewer hops than SRR becomes better as the scale of the network increases.

6. DRR extensions to RELOAD

Adding support for DRR 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 DRR 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 to support DRR mode. 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 is used, the response will not follow the reverse path, and the state in the intermediate peers will not 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 keeping is required in the intermediate peers.

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

This draft introduces a new forwarding option for an extensive routing mode. This option conforms to the description in section 6.3.2.3 of [I-D.ietf-p2psip-base].

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),(255)} RouteMode;
struct {
        RouteMode               routemode;
```

```
        OverlayLinkType         transport;
        IpAddressPort           ipaddressport;
        Destination             destinations<1..2^8-1>;
} ExtensiveRoutingModeOption;
```

The above structure reuses OverlayLinkType, Destination and IpAddressPort structure defined in section 6.5.1.1, 6.3.2.2 and 6.3.1.1 of [I-D.ietf-p2psip-base].

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 sending peer.

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

Destination: refers to the sending peer itself. If the routing mode is DRR, then the destination only contains the sending peer'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.

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 value, 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 DRR is introduced. Here, we only describe the additional procedures for supporting DRR. 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 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 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.

In the event that the routing mode is set to DRR and there is not exactly one destination, the destination peer MUST try 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 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 respond to now, the responding peer SHOULD abort the DRR 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.

7. Security considerations

As a routing alternative, the security part of DRR conforms to section 13.6 of the base draft [I-D.ietf-p2psip-base] which describes routing security. The DRR routing option provide the information about the route back to the source. According to section 13 of the base drat the forwarding header MUST be digitally signed protecting the DRR routing information.

8. IANA considerations

8.1. 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

9. 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.

10. References

10.1. 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.

10.2. 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-rpr] Zong, N., Jiang, X., Even, R. and Zhang, Y.,

"An extension to RELOAD to support Relay Peer Routing", draft-ietf-p2psip-rpr-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.

[IGD2] UPnP Forum, "WANIPConnection:2 Service (http://upnp.org/specs/gw/UPnP-gw-WANIPConnection-v2-Service.pdf)", September 2010.

[PMP] Cheshire, S., Krochmal M., and K. Sekar, "NAT Port Mapping Protocol (NAT-PMP)", draft-cheshire-nat-pmp-03 (work in progress), April 2008.

11. 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 DRR 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 DRR to function correctly, a peer may attempt to determine whether it is publicly reachable. If it is not, 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 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 use DRR for responses.

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.

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

A.1. Getting addresses to be used as candidates for DRR

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

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 [RFC5780] 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 peer 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 peer can determine that there likely is no NAT between it 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 peer can be considered publicly reachable. On the other hand, with the techniques in [RFC5780], a peer can also decide whether it is behind a NAT with endpoint-independent mapping behavior. If the peer is behind a NAT with endpoint- independent mapping behavior, the reflexive address should also be a candidate for further tests.

UPnP-IGD [IGD2] is a mechanism that a peer can use to get the assigned address from its residential gateway and after obtaining this address to communicate it with other peers, the peer 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 peer behind NAT can use to learn its assigned address by NAT is NAT-PMP [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.

A.2. Public reachability test

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

In a 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 peer. Especially in RELOAD architecture, every peer maintains a connection table. So it is easier for a peer to check whether it has direct connection with another peer.

If a peer 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 peer, instead it is to get a peer who can deliver the sending peer 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 peer 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 transactions.

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