Internet Engineering Task Force (IETF)
Request for Comments: 7263
Category: Standards Track
ISSN: 2070-1721

N. Zong
X. Jiang
R. Even
Huawei Technologies
Y. Zhang
CoolPad / China Mobile

June 2014

An Extension to the REsource LOcation And Discovery (RELOAD) Protocol to Support Direct Response Routing

Abstract

This document defines an optional extension to the REsource LOcation And Discovery (RELOAD) protocol to support the direct response routing mode. RELOAD recommends symmetric recursive routing for routing messages. The new optional extension provides a shorter route for responses, thereby reducing overhead on intermediate peers. This document also describes potential cases where this extension can be used.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7263.

Table of Contents

1. Introduction

The REsource LOcation And Discovery (RELOAD) protocol [RFC6940] recommends symmetric recursive routing (SRR) for routing messages and describes the extensions that would be required to support additional routing algorithms. In addition to SRR, two other routing options -- direct response routing (DRR) and relay peer routing (RPR) -- are also discussed in Appendix A of [RFC6940]. As we show in Section 3, DRR is advantageous over SRR in some scenarios in that DRR can reduce 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 provide benefits than if SRR is used alone.

Note that in this document we focus on the DRR mode and its extensions to RELOAD to produce a standalone solution. Please refer to [RFC7264] for details on the RPR mode.

We first discuss the problem statement in Section 3. How to combine DRR and SRR is presented in Section 4. An extension to RELOAD to support DRR is defined in Section 5. Some optional methods to check peer connectivity are introduced in Appendix A. In Appendix B, we give a comparison of the cost of SRR and DRR in both managed and open networks.

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 terminology and definitions from the base RELOAD specification [RFC6940] extensively in this document. We also use terms defined in the NAT behavior discovery document [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 network.

Direct Response Routing (DRR): "DRR" refers to a routing mode in which responses to Peer-to-Peer SIP (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 in the rest of this document.

Symmetric Recursive Routing (SRR): "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 in the rest of this document.

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

3. Overview

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

3.1. SRR and DRR

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 illustrations that follow:

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

2) Peer X is the root peer responsible for Resource-ID k.

3) The intermediate peers for the path from A to X are peers B, C, and 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 [RFC6940]. As a result, the destination peer X will know the exact path that 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.

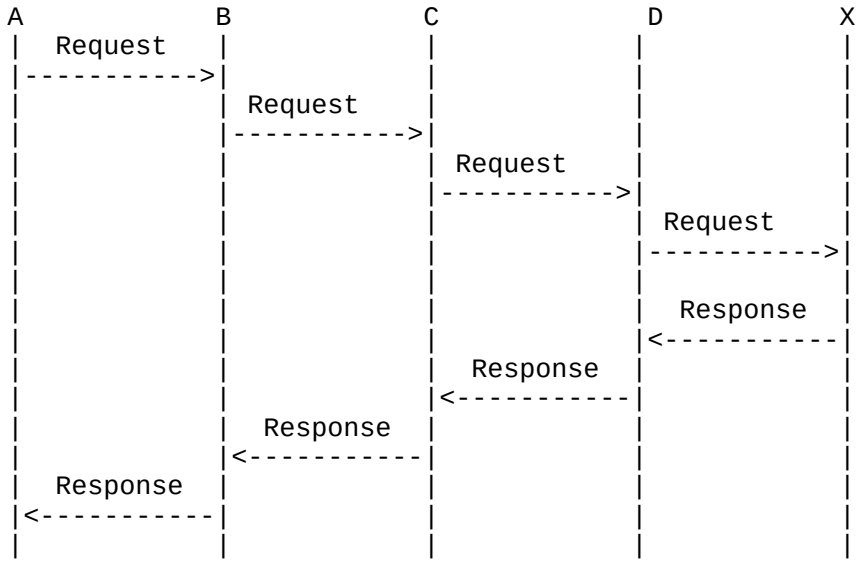

Figure 1: SRR Mode

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 peers 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 are limited. For example, in the absence of NATs, or if the NAT implements

endpoint-independent filtering, this is the optimal routing technique. Note that establishing a secure connection requires multiple round trips. Please refer to Appendix B for a cost comparison between SRR and DRR.

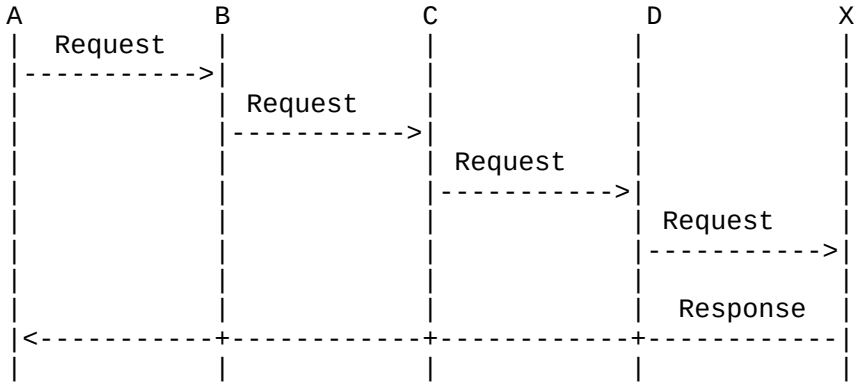

Figure 2: DRR Mode

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 servers, 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. During the process of establishing a direct connection with the sending peer, if the responding peer receives a request with SRR as the preferred routing mode (or it fails to establish the direct connection), the responding peer SHOULD NOT use DRR but instead switch to SRR. The simple policy is to try DRR and, if this fails, switch to SRR for all connections. In a finer-grained policy, a peer would keep a list of unreachable peers based on trying DRR and then would use only SRR for those peers. The advantage of using DRR is network stability, since it puts less overhead on the intermediate peers that will not route the responses. The intermediate peers will need to route fewer messages and will save CPU resources as well as link bandwidth usage.

3.2.2. Wireless Scenarios

In some mobile deployments, using DRR may help reduce 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 potential (publicly 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 for how to transition between the routing modes in RELOAD.

According to [RFC6940], SRR MUST be supported. An overlay MAY be configured to use alternative routing algorithms, and alternative routing algorithms MAY be selected on a per-message basis. That is, a node in an overlay that supports SRR and some other routing algorithm -- for example, DRR -- might use SRR some of the time and DRR some of the time. A node joining the overlay should get the preferred routing mode from the configuration file. 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. However, DRR SHOULD NOT be used in the open Internet or if the administrator does not feel he has enough information about the overlay network topology. A new overlay configuration element specifying the usage of DRR is defined in Section 6.

Alternatively, 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 of the success rate of different routing modes. In addition to data on the success rate, the peer can also get data of finer granularity -- for example, the number of retransmissions 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 configuration. Based on the success rate as indicated by statistics on lost messages or by responses that used DRR, the peer can either continue to offer DRR first or switch to

SRR. Note that a peer should use the DRR success statistics to decide whether to continue using DRR or fall back to SRR. Making such a decision per specific connection is not recommended; this should be an application decision.

5. DRR Extensions to RELOAD

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

5.1. Basic Requirements

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

5.2. 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.

5.2.1. 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 define a new flag, state-keeping flag, in the ForwardingOption structure 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 but who is not the destination of the message SHOULD forward the message with the full Via List and SHOULD NOT maintain any internal state.

5.2.2. Extensive Routing Mode

This document introduces a new forwarding option for an extensive routing mode. This option conforms to the description in Section 6.3.2.3 of [RFC6940].

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

The option value that defines the ExtensiveRoutingModeOption structure is illustrated below:

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

The above structure reuses the OverlayLinkType, Destination, and IpAddressPort structures as defined in Sections 6.5.1.1, 6.3.2.2, and 6.3.1.1 of [RFC6940], respectively.

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

OverlayLinkType: refers to the transport type that is used to deliver responses from the destination peer to the sending peer.

IpAddressPort: refers to the transport address that the destination peer will use for sending responses. 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.

5.3. Creating a Request

5.3.1. 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 ForwardingOption 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 ForwardingOption 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.

5.4. 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 [RFC6940] for RELOAD base procedures.

5.4.1. 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 cannot understand the extensive_routing_mode option in the request, it MUST attempt to use SRR to return an "Error_Unknown_Extension" response (defined in Sections 6.3.3.1 and 14.9 of [RFC6940]) to the sending peer.

If the routing mode is DRR, the destination peer MUST construct the Destination List for the response with only one entry, using the requesting peer's Node-ID from the Via List 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 Sections 6.3.3.1 and 14.9 of [RFC6940]) 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.

5.4.2. Sending Peer: Receiving a Response

Upon receiving a response, the peer follows the rules in [RFC6940]. The peer SHOULD note if DRR worked, in order to decide whether to offer DRR again. If the peer does not receive a response until the timeout, it SHOULD resend the request using SRR.

6. Overlay Configuration Extension

This document extends the RELOAD overlay configuration (see Section 11.1 of [RFC6940]) by adding one new element, "route-mode", inside each "configuration" element.

The Compact Regular Language for XML Next Generation (RELAX NG) grammar for this element is:

```
namespace route-mode = "urn:ietf:params:xml:ns:p2p:route-mode"

parameter &= element route-mode:mode { xsd:string }?
```

This namespace is added into the <mandatory-extension> element in the overlay configuration file. The defined routing modes include DRR and RPR.

The mode can be DRR or RPR and, if specified in the configuration, should be the preferred routing mode used by the application.

7. Security Considerations

The normative security recommendations of Section 13 of [RFC6940] are applicable to this document. As a routing alternative, the security part of DRR conforms to Section 13.6 of [RFC6940], which describes routing security. For example, the DRR routing option provides information about the route back to the source. According to

Section 13.6 of [RFC6940], the entire DRR routing message MUST be digitally signed and sent over via a protected channel to protect the DRR routing information.

8. IANA Considerations

8.1. A New RELOAD Forwarding Option

A new RELOAD Forwarding Option type has been added to the "RELOAD Forwarding Option" registry defined in [RFC6940].

Code: 2
Forwarding Option: extensive_routing_mode

8.2. A New IETF XML Registry

IANA has registered the following URN in the "XML Namespaces" class of the "IETF XML Registry" in accordance with [RFC3688].

URI: urn:ietf:params:xml:ns:p2p:route-mode

Registrant Contact: The IESG

XML: This specification

9. Acknowledgments

David Bryan 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.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC6940] Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", RFC 6940, January 2014.

10.2. Informative References

[Chord] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M., Dabek, F., and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications", IEEE/ACM Transactions on Networking Volume 11, Issue 1, 17-32, February 2003.

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

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

[RFC3424] Daigle, L. and IAB, "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation", RFC 3424, November 2002.

[RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, May 2010.

[RFC6886] Cheshire, S. and M. Krochmal, "NAT Port Mapping Protocol (NAT-PMP)", RFC 6886, April 2013.

[RFC7264] Zong, N., Jiang, X., Even, R., and Y. Zhang, "An Extension to the REsource LOcation And Discovery (RELOAD) Protocol to Support Relay Peer Routing", RFC 7264, June 2014.

[wikiChord] Wikipedia, "Chord (peer-to-peer)", 2013, <http://en.wikipedia.org/w/index.php?title=Chord_%28peer-to-peer%29&oldid=549516287>.

Appendix A. Optional Methods to Investigate Peer Connectivity

This section is for informational purposes only and provides some mechanisms that can be used when the configuration information does not specify if DRR can be used. It summarizes some methods that can be used by 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 whether a peer is publicly reachable, other than via out-of-band mechanisms. This document addresses UNilateral Self-Address Fixing (UNSAF) [RFC3424] considerations by specifying a fallback plan to SRR [RFC6940]. SRR is not an UNSAF mechanism. This document does not define any new UNSAF mechanisms.

For DRR to function correctly, a peer may attempt to determine whether it is publicly reachable. If it is not, the peer should fall back to SRR. If the peer believes it is publicly reachable, DRR may be attempted. NATs and firewalls are two major contributors to 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 peer to which the address belongs can use DRR for responses.

Some conditions that are unique in P2PSIP architecture could be leveraged to facilitate the tests. In a P2P overlay network, each peer has only a partial 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 that 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 that is assigned by a NAT to the peer.

Session Traversal Utilities for NAT (STUN) is used to get a reflexive address on the public side of a NAT with the help of STUN servers. NAT behavior discovery using STUN is specified in [RFC5780]. Under the RELOAD architecture, a few infrastructure servers can be leveraged for discovering NAT behavior, such as enrollment servers, diagnostic servers, bootstrap servers, etc.

The peer can use a STUN Binding request to one of the 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 is likely 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 servers are publicly reachable in the overlay, so the peer can be considered publicly reachable. On the other hand, using 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.

The Universal Plug and Play Internet Gateway Device (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 learn its assigned address by NAT is via the NAT Port Mapping Protocol (NAT-PMP) [RFC6886]. As with 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 of the test is that a peer sends a request and expects another peer in the overlay to send back a response. If the response is successfully received by the sending peer and 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. Every peer maintains a connection table, particularly in the RELOAD architecture, so it is easier for a peer to see 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 or fetching something to or from the overlay, or for locating a specific peer; instead, it is to get a peer who can deliver to the sending peer an unsolicited response and who has no direct connection with him. Each intermediate peer receiving the request first checks to see whether it has a direct connection 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.

Appendix B. Comparison of Cost of SRR and DRR

The major advantage of using DRR is that it reduces the number of intermediate peers traversed by the response. This reduces the load, such as processing and communication bandwidth, on those peers' resources.

B.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 network administrators would know that they could safely use DRR.

SRR uses more routing hops than DRR. Assuming that there are N peers in the P2P system and Chord [Chord] [wikiChord] is applied for routing, the number of hops for a response in SRR and in DRR are listed in the following table. Establishing a secure connection between the sending peer and the responding peer with Transport Layer Security (TLS) or Datagram TLS (DTLS) requires multiple messages. Note that establishing (D)TLS secure connections for a P2P overlay is

not optimal in some cases, e.g., DRR where (D)TLS is heavy for temporary connections. Therefore, in the following table we show the cases of 1) no (D)TLS in DRR and 2) still using DTLS in DRR as sub-optimal. As the worst-cost case, seven (7) messages are used during DTLS handshaking [DTLS]. (The TLS handshake is a negotiation protocol that requires two (2) round trips, while the DTLS handshake is a negotiation protocol that requires three (3) round trips.)

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 the number of peers (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 such resources 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 of whether to use DRR or SRR depends on other factors such as using less resources (bandwidth and processing) from the intermediate peers. Section 4 provides use cases where DRR has a better chance of working or where the considerations of intermediary resources are important.

B.2. Open Networks

In open networks (e.g., the Internet) 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 as those listed in Appendix B.1, the number of hops, as well as the 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) and 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 improves as the scale of the network increases.

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
CoolPad / China Mobile

EMail: hishigh@gmail.com