Networking Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 21, 2015

R. Chen
lin. Jiao
Frank. Feng
ZTE Corporation

March 20, 2015

Identifier Management for I2RS Protocol
draft-chen-i2rs-identifier-management-00

Abstract

An I2RS Agent may communicate with multiple clients. Two (or more) client may attempt to manipulate the same piece of data on the I2RS Agent. In order to solve this collision, the proposal is to have a simple priority associated with each I2RS Client.

In addition, if the hold timer for an I2RS session expires, the I2RS Agent should delete the I2RS data associated with the I2RS Client. In order to delete the related I2RS data correctly, it is important to identify the client on the I2RS Agent.

This document describes how to deploy identifier and priority associated with each I2RS Client based on [I-D.ietf-i2rs-architecture].

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 21, 2015.

Copyright Notice

Copyright (c) 2015 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.

Table of Contents

1. Introduction 2
2. Conventions used in this document 2
3. Terminology 3
4. Approach 3
5. NETCONF Extensions 4
6. RESTCONF Extensions 5
7. I2RS-IDM YANG Data model 6
8. Security Considerations 8
9. IANA Considerations 8
10. References 8
10.1. Normative references 8
10.2. Informative references 8
Authors' Addresses 9

1. Introduction

An I2RS Agent may communicate with multiple clients. Two (or more) client may attempt to manipulate the same piece of data on the I2RS Agent. In order to solve this collision, the proposal is to have a simple priority associated with each I2RS Client.

In addition, if the hold timer for an I2RS session expires, the I2RS Agent should delete the I2RS data associated with the I2RS Client. In order to delete the related I2RS data correctly, it is important to identify the client on the I2RS Agent.

This document describes how to deploy identifier and priority associated with each I2RS Client based on [I-D.ietf-i2rs-architecture].

2. Conventions used in this document

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 RFC2119.

3. Terminology

The following terminology is used in this document.

Client identifier: An I2RS Client's identifier is a string of 1 to 31 characters, used to uniquely identify a client. The first character of a client identifier should be a letter or "_", and the rest of it can only be numbers, letters, or "_". The data of an I2RS session will be maintained within a hold time after the I2RS session is terminated, and it will be restored if the client uses the same client identifier to reconnect to the agent during the hold time.

Priority: Priority is defined as an integer between 0 and 255, in which, number 0 is with the highest priority and 255 is with the lowest priority. A client is associated with a priority, which determines whether the client has the authority to modify data of a specific I2RS session. A client with higher priority can modify I2RS data distributed by a client with lower priority. A client can always modify I2RS data distributed by itself. Once the I2RS data is modified by a client, the client identifier associated with the I2RS data will be changed to the client identifier of the client that modified the I2RS data.

4. Approach

When the I2RS session is opened, An I2RS Client should advertise to an I2RS Agent that the I2RS Client's identifier and priority. The I2RS Client's identifier and priority should be managed by the I2RS Agent, and associated with its I2RS session.

If the hold timer for an I2RS session expires, the I2RS Agent should delete the I2RS Client's identifier and priority associated with the I2RS session.

When the I2RS session is active, the I2RS Client can only update its own priority, and it must not update other client's priority.

In addition, I2RS Client's identifier management data model MUST be supported by the I2RS Agent.

YANG Tree Diagram for "ietf-i2rs-idm" module:

```
module: ietf-i2rs-idm
+--ro identifiers
   +--ro identifier [name]
      +--ro name           identifier
      +--ro priority?      uint8
      +--ro session-id?    uint32
rpcs:
+---x set-priority
|  +--ro input
|     +--ro priority?       uint8
+---x advertise-identifier
   +--ro input
      +--ro identifier?     identifier
      +--ro priority?       uint8
```

5. NETCONF Extensions

This approach extends hello message to the Network Configuration Protocol (NETCONF) defined in [RFC6241]. More precisely, it defines capability-based extension that an I2RS Client can use to advertise to an I2RS Agent that the I2RS Client's identifier and priority during I2RS session establishment.I2RS Client's identifier and priority are only advertised in hello messages sent by Clients during I2RS session establishment. The following will describe the procedure in detail:

- o When the I2RS session is opened, An I2RS Client sends a <hello> element containing I2RS Client's identifier and priority.

- o When the I2RS session is successfully established, An I2RS Agent receiving a <hello> message with I2RS Client's identifier and priority stores it and associated with its I2RS session.

- o The client can modify the client's priority by <set-priority>RPC operation.

- o The I2RS Agent will allocate an initial priority to the I2RS Client which does not send the client priority to the Agent.

In the following example, An I2RS Client advertises <hello> element containing I2RS Client's identifier and priority.

```
  <hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <capabilities>
       <capability>
         urn:ietf:params:netconf:base:1.1
       </capability>
       <capability>
        urn:ietf:params:xml:ns:yang:ietf-i2rs:idm?client
identifier=test&priority=10
       </capability>
     </capabilities>
</hello>
```

In order to modify the client's priority, the client might send the following <set-priority>RPC message:

```
    <rpc message-id="101"
       xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <set-priority xmlns=urn:ietf:params:xml:ns:yang:ietf-i2rs:idm>
       <priority>10</priority>
     </set-priority>
   </rpc>
```

6. RESTCONF Extensions

This approach extends RESTCONF protocol. When a new I2RS session is established, the I2RS Client's identifier and priority must be advertised to the I2RS Agent. To handle the Client's update, the I2RS Agent must associate the I2RS session with the I2RS Client's identifier and priority. In addition, if the hold timer for an I2RS session expires, the I2RS Agent should delete the I2RS Client's identifier and priority associated with the I2RS session.

In order to advertise the I2RS client's identifier and priority to the I2RS Agent, the client might send the following <post> message:

```
 POST http://localaddress:8080/restconf/operations/i2rs-idm:advertise-
identifier
 name: Content-Type value:  application/yang.data+json
 name: Accept       value: application/xml
{
  "input" :
  {
                "identifier":"test",
                "priority":"50"
  }
}
```

In order to update the I2RS client's identifier and priority to the I2RS Agent, the client might send the following <POST>message:

```
  POST http://localaddress:8080/restconf/operations/i2rs-idm:set-priority
  name: Content-Type value:  application/yang.data+json
  name: Accept       value: application/xml
{
  "input" :
  {
                "priority":"10"
  }
}
```

7. I2RS-IDM YANG Data model

```
  module ietf-i2rs-idm {
    namespace "urn:ietf:params:xml:ns:yang:ietf-i2rs:idm";
    prefix "i2rs-idm";

    revision 2015-03-06 {
        description "Initial revision";
    }

    typedef identifier {
        description "The typedef of identifier.";
        type string {
                length 1..31;
                pattern "([a-zA-Z_])([0-9a-zA-Z_]{0,30})";
        }
    }

    rpc set-priority {
        description "This rpc can be used to update priority when
                session is active. A client can only update its own
                priority.";
        input {
                leaf priority {
                        description "The priority of the identifier associated
                                with this session. 0 indicates the highest
priority,
                                and 255 indicates the lowest priority.";
                        type uint8;
                }
        }
    }

    rpc advertise-identifier {
        description "This rpc can be used to advertise identifier associated
```

```
                with this session. If the identifier has been exist and the
                session-id associated with it is not zero, agent MUST report an
                error and SHOULD terminate the session immediately. Otherwise,
                Agent MUST store this identifier and assocoate it with its
session-id.";
        input {
                leaf identifier {
                        description "The identifer advertised by this session.
Identifier
                                can be used to indicate data, when a client
create or update
                                data of i2rs datastore, these data will be
associated with
                                the identifier of this client.";
                        type identifier;
                }

                leaf priority {
                        description "The priority of the identifier associated
                                with this session. 0 indicates the highest
priority,
                                and 255 indicates the lowest priority.";
                        type uint8;
                }
        }
    }
    container identifiers {
        description "This tree contains all detail information of identifiers
                supported by agent. This is a read-only tree";
        config false;
        list identifier {
                description "The detail information of an identifier. When the
                        session is terminated, the session-id will be set by
zero,
                        and other identifier information associated with this
session
                        will be held until the hold time is out.";
                key name;
                unique session-id;
                leaf name {
                        type identifier;
                }

                leaf priority {
                        type uint8;
                }

                leaf session-id {
```

```
                        description "The session id associated with identifier.
Zero indicates
                        none.";
                        type uint32;
                }
        }

    }
```

}

8. Security Considerations

TBD.

9. IANA Considerations

TBD.

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.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6243] Bierman, A. and B. Lengyel, "With-defaults Capability for NETCONF", RFC 6243, June 2011.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012.

10.2. Informative references

[I-D.ietf-i2rs-architecture]
Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-09 (work in progress), March 2015.

[I-D.ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-04 (work in progress), January 2015.

Authors' Addresses

Ran Chen
ZTE Corporation
No.50 Software Avenue
Yuhuatai District,Nanjing 210012
P.R.China

Phone: +86 025 88014636
Email: chen.ran@zte.com.cn

lin Jiao
ZTE Corporation
No.86 Zijinghua Rd
Yuhuatai District,Nanjing 210012
P.R.China

Email: Jiao.lin@zte.com.cn

Frank Feng
ZTE Corporation
No.86 Zijinghua Rd
Yuhuatai District,Nanjing 210012
P.R.China

Phone: +86 021 68896273
Email: feng.chong33@zte.com.cn