I2RS working group
Internet-Draft
Intended status: Informational
Expires: September 30, 2017

S. Hares
Huawei
A. Dass
Ericsson

March 29, 2017

RESTCONF Changes to Support I2RS Protocol
draft-hares-netconf-i2rs-restconf-02.txt

Abstract

This document describes two RESTCONF optional capabilities (i2rs-control plane capability, ephemeral state capabilities) that are needed to support the I2RS protocol needs.

The purpose of this draft is to kick-start the discussions with I2RS Working Group and NETCONF WG on these two capabilities.

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 30, 2017.

Copyright Notice

Copyright (c) 2017 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
1.1. Background on I2RS 3
1.2. Structure of draft 3
2. Definitions and Background on I2RS 3
2.1. IETF Requirements language 3
2.2. I2RS Definitions 3
2.3. I2RS protocol requirements 5
3. RESTCONF control plane datastore capability 5
3.1. Overview 5
3.2. Dependencies 6
3.3. New Operations 6
3.4. Modified Operations 6
4. RESTCONF protocol extensions for the ephemeral datastore . . 6
4.1. Overview 6
4.2. Dependencies 7
4.3. Capability identifier 8
4.4. New Operations 8
4.5. Modification to data resources 8
4.6. Modification to existing operations 8
5. IANA Considerations 9
6. Security Considerations 9
7. Acknowledgements 9
8. References 9
8.1. Normative References: 9
8.2. Informative References 11
Authors' Addresses 14

1. Introduction

This a proposal for the following two RESTCONF capabilities to augment RESTCONF [RFC8040] to support the first version of the I2RS protocol: Control plane datstore capability and ephemeral state capability. The yang that supports this proposal is described in [I-D.hares-netmod-i2rs-yang]. This work is based on the datastore definitions in [I-D.ietf-netmod-revised-datastores].

This draft parallels a similar proposal for NETCONF [RFC6241] is described in [I-D.hares-netconf-i2rs-protocol]. One difference between the proposed capabilities for i2rs control-plane capability additions to NETCONF and the proposed capabilities for i2rs control-plane for RESTCONF is write-collection. RESTCONF has edit-collision capability already which only needs a usage description.

1.1. Background on I2RS

The I2RS architecture [RFC7921] defines the I2RS interface "a programmatic interface for state transfer in and out of the Internet routing system". The I2RS protocol is a protocol designed to a higher level protocol comprised of a set of existing protocols which have been extended to work together to support a new interface to the routing system. The I2RS protocol is a "reuse" management protocol which creates new management protocols by reusing existing protocols and extending these protocols for new uses, and has been designed to be implemented in phases [RFC7921].

1.2. Structure of draft

The structure of this document is:

Section 2 provides definitions and background on I2RS work. (If you are familiar with the I2RS architecture and requirements, you can skip this section.)

Section 3 describes the RESTCONF control plane datastore capability.

Section 4 describes the RESTCONF ephemeral state capabiilty. .

2. Definitions and Background on I2RS

This section reviews definitions from I2RS architecture, and provides background on I2RS work for the reader.

2.1. IETF Requirements language

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].

2.2. I2RS Definitions

The I2RS architecture [RFC7921] defines the following terms:

ephemeral data: is data which does not persist across a reboot (software or hardware) or a power on/off condition. Ephemeral data can be configured data or data recorded from operations of the router. Ephemeral configuration data also has the property that a system cannot roll back to a previous ephemeral configuration state. (See [RFC7921] for an architectural overview, [I-D.ietf-i2rs-ephemeral-state] for requirements, and [I-D.ietf-netmod-revised-datastores] for discussion of how the

ephemeral datastore as a control plane datastore interacts with intended datstore and dynamic configuration protocols to form the applied datastore".

local configuration: is the data on a routing system which does persist across a reboot (software or hardware) and a power on/off condition. Local configuration has the ability to roll back to a pervious configuration state. Local configuration is defined as the intended datastore [I-D.ietf-netmod-revised-datastores] which is modified by dynamic configuration protocols (such as DHCP) and the I2RS ephemeral data store

dynamic configuration protocols datastore are configuration protocols such as DHCP that interact with the intended datastore (which does persist across a reboot (software or hardware) power on/off condition), and the I2RS ephemeral state control plane datastore.

control plane protocols datastore is a datastore which is loaded by control plane protocols (e.g. I2RS protocol) rather than system configuration protocols. (see [I-D.ietf-netmod-revised-datastores]).

operator-applied policy: is a policy that an operator sets that determines how the ephemeral datastore as a control plane data store interacts with applied datastore (as defined in [I-D.ietf-netmod-revised-datastores]). This operator policy consists of policy knobs that the operator sets to determine how the I2RS agent control plane ephemeral state datastore will interact with the intended configuration datastor and the dynamic configuration protocol datastore. Three policy knobs could be used to implement this policy:

* policy knob 1: I2RS Ephemeral control-plane datastore takes takes precedence over the intended datastore in the routing protocols.

* policy knob 2: Updated intended configuration datastore takes precedence over the I2RS ephemeral control-plane data store in the routing protocols

* policy knob 3: Ephemeral control plane datastore takes precedence over any other dynamic configuration protocols datastore.

2.3. I2RS protocol requirements

The requirements for the I2RS protocol are defined in the following documents:

- o I2RS Problem Statement [RFC7920],
- o I2RS Architecture [RFC7921],
- o I2RS Traceability [RFC7922],
- o Publication and Subscription [RFC7923],
- o I2RS Ephemeral State Requrements, , [I-D.ietf-i2rs-ephemeral-state]
- o I2RS Protocol Security Requirements, [I-D.ietf-i2rs-protocol-security-requirements]

The Interface to the routing System (I2RS) creates a new capability for the routing systems, and with greater capaiblities come a greater need for security. The requirements for a secure environment for I2RS is described in [I-D.ietf-i2rs-security-environment-reqs].

3. RESTCONF control plane datastore capability

capability-name: i2rs-control-plane

3.1. Overview

The i2rs-control-plane datastore capability enables the RESTCONF to support the following dynamic control plane datastore.

- o API resource that is {+restconf}/datastore/<datastore-name>/data/ and operational state specific to the control plane datastore ({+restconf/cp-data/opstate}).
- o It also includes the ability to have the applied datastore and the opstate datatstore (per [I-D.ietf-netmod-revised-datastores]) with the ability to return meta-data with the following information:
  - * Entity-Tag encoding of <client-id><priority> or any portion of the filter.
  - * "with defaults"
  - * "with validation" - Yang specified validation (Unclear if this is the best way for validation.)

Ability to provide read access for the configuration datstore

Ability to provide read access for other dynamic datastores

3.2. Dependencies

This protocol strawman utilizes the following existing proposed features for NETCONF and RESTCONF

- o RESTCONF [RFC8040].
- o Module library [RFC7895],
- o RESTCONF Patch Media Type [RFC8072],
- o NETCONF Support for event notifications [I-D.ietf-netconf-netconf-event-notifications],
- o Publication/Subscription via Push [I-D.ietf-netconf-yang-push],
- o NETCONF and HTTP Transport for Event Notivications [I-D.ietf-netconf-restconf-notif],
- o Publication/Subscription via Push [I-D.ietf-netconf-yang-push],
- o syslog yang module (both [RFC5424] and [I-D.ietf-netmod-syslog-model]

3.3. New Operations

none

3.4. Modified Operations

All RESTCONF methods (OPTIONS, HEAD, GET, POST, PT, PATCH, DELETE) need to work in the control plane datastores. config=TRUE data, and where appropriate config=FALSE data.

4. RESTCONF protocol extensions for the ephemeral datastore

capability-name: ephemeral-state

4.1. Overview

This capability defines the RESTCONF protocol extensions for control plane protocols that support control plane data stores with ephemeral data.

Ephemeral state is not unique to I2RS work.

The ephemeral capability is the ability to support a dynamic datastores which are entirely ephemeral or have ephemeral state modules, or ephemeral statements within objects in a modules. These objects can be configuration state (config=TRUE) or operational state (config=FALSE).

Ephemeral state in datastores, ephemeral modules or ephemeral objects within a module have one key characteristics: the data does not persist across reboots. The ephemeral configuration state must be restored by a client, and the operational state will need to be regenerated.

The entire requirements for ephemeral state for the I2RS control plane protocol are listed in [I-D.ietf-i2rs-ephemeral-state]. Compared to RESTCONF functionality there are 4 groups of additional changes:

Constraints The ability to enforce the constraints for get (aka read) references (to/from) the {+restconf/data} datastore, and {+restconf/cp-data} control plane datastore. ((see Ephemeral-REQ-02, Ephemeral-REQ-03, and Ephemeral-REQ-04 in [I-D.ietf-i2rs-ephemeral-state]). The "validation" yang statement in [I-D.hares-netmod-i2rs-yang] could encode specific validation for the ephemeral case per datatstore or per object. [Editor's note: Aid is needed to determine how validation occurs.]

Ephemeral in Data Modules Yang modules must identify Yang objects (modules, submodules or objects within yang modules which are ephemeral and augment other nodes) and allow an "ephemeral=TRUE" feature.

Roll-back an ephemeral node cannot roll-back to its previous value,

4.2. Dependencies

The ephemeral capabilities have the following dependencies:

- o Yang modules must support the following:
  - * identifying datastores, modules, and objects as ephemeral. (ephemeral=True)
  - * Ability to have control plane datastores which are ephemeral.
- o The following features must be supported by RESTCONF

* Module library [RFC7895],

* RESTCONF Protocol [RFC8040],

* RESTCONF Patch Media Type [RFC8072],

* NETCONF Support for event notifications [I-D.ietf-netconf-netconf-event-notifications],

* Publication/Subscription via Push [I-D.ietf-netconf-yang-push],

* NETCONF and HTTP Transport for Event Notivications [I-D.ietf-netconf-restconf-notif],

* Subsribing to Yang datastore push updates [I-D.ietf-netconf-yang-push],

4.3. Capability identifier

The ephemeral-datastore capability is identified by the following capability string: ephemeral (TBD URI)

4.4. New Operations

none

4.5. Modification to data resources

RESTCONF must be able to support the ephemeral data in an an control-plane dynamic datastore. This is any API resource that is {+restconf}/datastore/<datastore-name>/data/ and operational state specific to the control plane datastore ({+restconf/cp-data/opstate}).

RESTCONF library functions must be able to store an indication that a data module has ephemeral state as meta-data.

4.6. Modification to existing operations

RESTCONF operations of GET, POST, PUT, PATCH, and DELETE must be able to filter on meta-data with "ephemeral" flag. (Should this be only read).

The operations must support the following things about ephemeral.

1. The ephemeral does not persist over a reboot,

2. an ephemeral node cannot roll-back to its previous value,

5. IANA Considerations

TBD -

6. Security Considerations

The security requirements for the I2RS protocol are covered in [I-D.ietf-i2rs-protocol-security-requirements]. The security environment the I2RS protocol is covered in [I-D.ietf-i2rs-security-environment-reqs]. Any person implementing or deploying the I2RS protocol should consider both security requirements.

7. Acknowledgements

TBD

8. References

8.1. Normative References:

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107, June 2005, <http://www.rfc-editor.org/info/rfc4107>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <http://www.rfc-editor.org/info/rfc4960>.

[RFC5339] Le Roux, JL., Ed. and D. Papadimitriou, Ed., "Evaluation of Existing GMPLS Protocols against Multi-Layer and Multi-Region Networks (MLN/MRN)", RFC 5339, DOI 10.17487/RFC5339, September 2008, <http://www.rfc-editor.org/info/rfc5339>.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <http://www.rfc-editor.org/info/rfc5424>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6244] Shafer, P., "An Architecture for Network Management Using NETCONF and YANG", RFC 6244, DOI 10.17487/RFC6244, June 2011, <http://www.rfc-editor.org/info/rfc6244>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC7158] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7158, DOI 10.17487/RFC7158, March 2014, <http://www.rfc-editor.org/info/rfc7158>.

[RFC7589] Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication", RFC 7589, DOI 10.17487/RFC7589, June 2015, <http://www.rfc-editor.org/info/rfc7589>.

[RFC7803] Leiba, B., "Changing the Registration Policy for the NETCONF Capability URNs Registry", BCP 203, RFC 7803, DOI 10.17487/RFC7803, February 2016, <http://www.rfc-editor.org/info/rfc7803>.

[RFC7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", RFC 7895, DOI 10.17487/RFC7895, June 2016, <http://www.rfc-editor.org/info/rfc7895>.

[RFC7920] Atlas, A., Ed., Nadeau, T., Ed., and D. Ward, "Problem Statement for the Interface to the Routing System", RFC 7920, DOI 10.17487/RFC7920, June 2016, <http://www.rfc-editor.org/info/rfc7920>.

[RFC7921] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", RFC 7921, DOI 10.17487/RFC7921, June 2016, <http://www.rfc-editor.org/info/rfc7921>.

[RFC7922] Clarke, J., Salgueiro, G., and C. Pignataro, "Interface to the Routing System (I2RS) Traceability: Framework and Information Model", RFC 7922, DOI 10.17487/RFC7922, June 2016, <http://www.rfc-editor.org/info/rfc7922>.

[RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <http://www.rfc-editor.org/info/rfc7923>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <http://www.rfc-editor.org/info/rfc7950>.

[RFC7952] Lhotka, L., "Defining and Using Metadata with YANG", RFC 7952, DOI 10.17487/RFC7952, August 2016, <http://www.rfc-editor.org/info/rfc7952>.

[RFC7958] Abley, J., Schlyter, J., Bailey, G., and P. Hoffman, "DNSSEC Trust Anchor Publication for the Root Zone", RFC 7958, DOI 10.17487/RFC7958, August 2016, <http://www.rfc-editor.org/info/rfc7958>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <http://www.rfc-editor.org/info/rfc8040>.

[RFC8072] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Patch Media Type", RFC 8072, DOI 10.17487/RFC8072, February 2017, <http://www.rfc-editor.org/info/rfc8072>.

8.2. Informative References

[I-D.hares-netconf-i2rs-protocol]
Hares, S. and a. amit.dass@ericsson.com, "NETCONF Changes to Support I2RS Protocol", draft-hares-netconf-i2rs-protocol-00 (work in progress), November 2016.

[I-D.hares-netmod-i2rs-yang]
Hares, S. and a. amit.dass@ericsson.com, "Yang for I2RS Protocol", draft-hares-netmod-i2rs-yang-04 (work in progress), March 2017.

[I-D.ietf-i2rs-ephemeral-state]
Haas, J. and S. Hares, "I2RS Ephemeral State Requirements", draft-ietf-i2rs-ephemeral-state-23 (work in progress), November 2016.

[I-D.ietf-i2rs-protocol-security-requirements]
Hares, S., Migault, D., and J. Halpern, "I2RS Security Related Requirements", draft-ietf-i2rs-protocol-security-requirements-17 (work in progress), September 2016.

[I-D.ietf-i2rs-rib-data-model]
Wang, L., Ananthakrishnan, H., Chen, M., amit.dass@ericsson.com, a., Kini, S., and N. Bahadur, "A YANG Data Model for Routing Information Base (RIB)", draft-ietf-i2rs-rib-data-model-07 (work in progress), January 2017.

[I-D.ietf-i2rs-rib-info-model]
Bahadur, N., Kini, S., and J. Medved, "Routing Information Base Info Model", draft-ietf-i2rs-rib-info-model-10 (work in progress), December 2016.

[I-D.ietf-i2rs-security-environment-reqs]
Migault, D., Halpern, J., and S. Hares, "I2RS Environment Security Requirements", draft-ietf-i2rs-security-environment-reqs-05 (work in progress), March 2017.

[I-D.ietf-i2rs-yang-l3-topology]
Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", draft-ietf-i2rs-yang-l3-topology-08 (work in progress), January 2017.

[I-D.ietf-netconf-call-home]
Watsen, K., "NETCONF Call Home and RESTCONF Call Home", draft-ietf-netconf-call-home-17 (work in progress), December 2015.

[I-D.ietf-netconf-keystore]
Watsen, K., "Keystore Model", draft-ietf-netconf-keystore-01 (work in progress), March 2017.

[I-D.ietf-netconf-netconf-event-notifications]
Prieto, A., Clemm, A., Voit, E., Nilsen-Nygaard, E., Tripathy, A., Chisholm, S., and H. Trevino, "NETCONF Support for Event Notifications", draft-ietf-netconf-netconf-event-notifications-01 (work in progress), October 2016.

[I-D.ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-18 (work in progress), October 2016.

[I-D.ietf-netconf-restconf-notif]
Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Clemm, A., and A. Bierman, "Restconf and HTTP Transport for Event Notifications", draft-ietf-netconf-restconf-notif-02 (work in progress), March 2017.

[I-D.ietf-netconf-rfc5277bis]
Clemm, A., Prieto, A., Voit, E., Nilsen-Nygaard, E., Tripathy, A., Chisholm, S., and H. Trevino, "Subscribing to Event Notifications", draft-ietf-netconf-rfc5277bis-01 (work in progress), October 2016.

[I-D.ietf-netconf-tls-client-server]
Watsen, K. and G. Wu, "TLS Client and Server Models", draft-ietf-netconf-tls-client-server-02 (work in progress), March 2017.

[I-D.ietf-netconf-yang-patch]
Bierman, A., Bjorklund, M., and K. Watsen, "YANG Patch Media Type", draft-ietf-netconf-yang-patch-14 (work in progress), November 2016.

[I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", draft-ietf-netconf-yang-push-05 (work in progress), March 2017.

[I-D.ietf-netconf-zerotouch]
Watsen, K. and M. Abrahamsson, "Zero Touch Provisioning for NETCONF or RESTCONF based Management", draft-ietf-netconf-zerotouch-13 (work in progress), March 2017.

[I-D.ietf-netmod-revised-datastores]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-01 (work in progress), March 2017.

[I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-04 (work in progress), March 2017.

[I-D.ietf-netmod-syslog-model]
Wildes, C. and K. Koushik, "A YANG Data Model for Syslog Configuration", draft-ietf-netmod-syslog-model-14 (work in progress), March 2017.

Authors' Addresses

Susan Hares
Huawei
Saline
US

Email: shares@ndzh.com

Amit Daas
Ericsson

Email: amit.dass@ericsson.com