I2RS working group
Internet-Draft
Intended status: Informational
Expires: May 20, 2017

S. Hares
Huawei
A. Dass
Ericsson

November 16, 2016

Yang for I2RS Protocol
draft-hares-netmod-i2rs-yang-02.txt

Abstract

This document requests one yang model addition that will support ephemeral state and provides notes for the implementers who wish to implement ephemeral state for the I2RS Protocol. The purpose of this document is to provide implementers of ephemeral state with background and open issues that they should consider when implementing ephemeral state that satifies the I2RS protocol.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This a proposal for yang additions to support the first version of the I2RS protocol.

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].

The first version of the I2RS protocol is comprised of extensions to existing features of NETCONF [RFC6241] and RESTCONF [I-D.ietf-netconf-restconf]. The data modeling language for the I2RS protocol will be Yang [RFC7950] with features and extensions proposed in this draft.

The structure of this document is:

Section 2 provides definitions for terms in this document.

Section 3 summarizes the changes to configuration data store, NETCONF, RESTCONF, and YANG.

[I-D.ietf-i2rs-ephemeral-state] specifies the I2RS requirements for the ephemeral state. Section 4 discusses how these requirements might be implemented in a control plane datastore.

Section 5 describes the one required Yang model addition for I2RS (ephemeral key word). This section also describes elements of information in the NETCONF/RESTCONF implementations that must be queryable by the I2RS protocol implementations.

2. Definitions Related to Ephemeral Configuration

This section reviews definitions from I2RS architecture [RFC7921] and NETCONF operational state definitions [I-D.nmdsdt-netmod-revised-datastores] before using these to construct a definition of the ephemeral data store.

2.1. 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.nmdsdt-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 is defined as the intended datastore [I-D.nmdsdt-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.

applied configuration Read only infromation regarding configuration state installed in the routing system.

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.nmdsdt-netmod-revised-datastores]). This operator policy consists of setting a priority for each of the following (per [I-D.ietf-i2rs-ephemeral-state]):

* intended configuration,

* any dynamic configuration protocols,

* any control plane datastores (one of which is ephemeral.)

An practical example with 3 priorities may help illustrated how this priority scheme works to install Assume high priority value wins and we have three inputs to configuration: intended configuration, dhcp dynamic configuration, and one ephemeral state control-plane datatstore loaded by all I2RS clients. Let us examine three use cases.

Monitoring Topology only: The purpose of the I2RS protocol is to monitor the topology of the network using a protocol independent data model. There should be no changes to interface configuration or learned addresses. DHCP can change the values of the configuration.

intended configuration priority = 2

dhcp dynamic configuration protocol = 3

ephemeral datastore = 1

I2RS Agent Changes BGP Config: BGP peer state is changed to add BGP Flow Specification data to the peer specification.

intended configuration priority = 2

dhcp dynamic configuration protocol = 3

ephemeral datastore = 4

DDoS Configuration change The ephemeral datastore is used to configure filters for DDoS attacks. After the DDoS attacks disappear the I2RS policy is removed. For simplicity of example, let us assume the following priority settings:

intended configuration priority = 2

dhcp dynamic configuration protocol = 1

ephemeral datastore = 4

The I2RS action is to install the ephemeral state during the DDoS attack. Since the dhcp dynamic configuration has a lower priority than the intended configuration this configuration is not tracked.

An important debugging aid that the applied configuration can provided is the indication of what process installed what type of configuration process installed things (E.g. intended configuration, dynamic configuration protocol, control-plane datastore), the identifier for that process (E.g. ephemeral datastore 1), and the priority the datastore has (E.g. priority 10).

3. Overview of Changes

This oveview reviews the following:

- o What NETCONF [RFC6241] protocol existing features required for I2RS protocol and what extension for these extension features that are needed for the I2RS protocol version 1,

- o What RESTCONF [I-D.ietf-netconf-restconf] protocol existing features are required for the I2RS protocol and what extensions are needed for I2RS protocol version 1.

- o An overview of the Yang 1.1 data modeling language[RFC7950] features are needed for I2RS protocol version 1.

- o An overview of the extensions to Yang 1.1 data modeling language [RFC7950] that are needed for the I2RS protocol version 1.

3.1. 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].

3.2. NETCONF Features and Extensions

The features the I2RS protocol requires are:

- o NETCONF [RFC6241] with its updates [RFC7803],
- o Network Access Control Model [RFC6536] with update (draft-bierman-netconf-rf6536bis)
- o Running NETCONF over TLS with mutually X.509 authentication [RFC7589]
- o Keystore Model [I-D.ietf-netconf-keystore],
- o Subscribing to Yang Datastore updates [I-D.ietf-netconf-yang-push],
- o NETCONF support for Event Notifications [I-D.ietf-netconf-netconf-event-notifications],

- o Subscribing to NETCONF Events (updated) [I-D.ietf-netconf-rfc5277bis]

- o Yang Patch Media type [I-D.ietf-netconf-yang-patch],

- o NETCONF/RESTCONF Zero Touch provisioning [I-D.ietf-netconf-zerotouch],

- o TLS Client and Server Models [I-D.ietf-netconf-restconf-client-server]

- o Call Home [I-D.ietf-netconf-call-home],

- o Module library [RFC7895],

- o NETCONF/RESTCONF Zero Touch provisioning [I-D.ietf-netconf-zerotouch],

3.3. RESTCONF features and Extensions

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

- o RESTCONF [I-D.ietf-netconf-restconf]

- o Module library [RFC7895],

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

- o Patch [I-D.ietf-netconf-yang-patch],

- o syslog yang module (both [RFC5424] and [I-D.ietf-netmod-syslog-model]

3.4. Assumptions on Data Store Model Melee

The NETMOD Working Group has been working to create new definitions of datastores based on feedback from operators on desiring a split between operational state and configuration state.

This document takes [I-D.nmdsdt-netmod-revised-datastores] as the current status of the datastore discussion on configuration state, operational state, ephemeral state changes (via I2RS), and routing protocol state. The following things need to be carefully defined in this work:

What protocol are classified as dynamic configuration protocols?

What is a control-plane datastore - (ephemeral state only or others?)

Error checking in ephemeral datastores installed by the I2RS protocol.

How does operational state allow for operational state to be defined by ephemeral-only data models, and mixed (ephemeral + intended configuration)

[I-D.nmdsdt-netmod-revised-datastores] is making good progress, but these additional details need to be tied down.

4. Ephemeral Data

This section provides an overview of the ephemeral data store as a control plane datastore and discusses several concepts that implementers need to consider and provide feedback on. The concepts include basic ephemeral datastore concepts, I2RS caching of ephemeral data, issues for massive data flow, error handling (normal and reduced), use of IPFIX or Binary for carrying I2RS ephemeral data, and ephemeral state.

This section augments [I-D.nmdsdt-netmod-revised-datastores] to begin to discuss how the ephemeral state control-plane datastore might be implemented.

The purpose of this section is to gather implementer wisdom on the ephemeral datastore into one place. This section discusses:

Ephemeral state as a control plane data store

Qualities of ephemeral datastores

Need to support Massive amounts of configuration data,

Two types of Error handling (regular, reduced)

Should we support link to IPFIX in I2RS protocol and ephemeral state?

Binary encoding for RESTCONF/NETCONF

Ephemeral state in DDoS environments.

[I-D.ietf-i2rs-ephemeral-state] describes the requirements for I2RS ephemeral state.

This section augments [I-D.nmdsdt-netmod-revised-datastores] to begin to discuss how the ephemeral state comtrol-plane datastore might be implemented. This initial draft refines the general description so that early I2RS ephemeral state implementations may progress.

4.1. Ephemeral Control Plane Datastore

[I-D.nmdsdt-netmod-revised-datastores] architecture suggests that the applied configuration is the combination of intended datastore, the dynamic configuration protocols, and the control-plane datastores. As described above, there are policy knobs which allow the I2RS Agent to handle deciding what specific configuration variables is installed in protocols (E.g BGP) or protocol independent functions (RIB or Filters). In addition, the control-plane datastore may store the parameters need to provide publication of events, statistics, telementry within the ephemeral control-plane datastore.

The ephemeral data-store may have models which learn operational state and augment it by configuration. For example [I-D.ietf-i2rs-yang-l3-topology] uploads ospf and isis topology information from the routing system and allows configuration of additional links or nodes.

This new architecture is a multiple panes-of-glass model where the decision on what value is chosen is based on policy. The extension of this model is that it is possible for two or more of the control-plane datastores to be ephemeral. If this occurs, then the policy knobs must define the how the 2+ ephemeral datastores interact with each other and the configuration state.

4.2. Qualities of Ephemeral Datastore

Note: The requirements for ephemeral state are in: [I-D.ietf-i2rs-ephemeral-state]).

This section provides a discussion so that implementers writing code for these datastores can discuss what needs to be standardized and what does not need to be standardized.

The ephemeral data store has the following general qualities:

1. Ephemeral state is not unique to I2RS work.

2. The ephemeral datastore is never locked.

3. The ephemeral portion of the intended configuration, applied state, and derived state does not persist over a reboot,

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

5. Since ephemeral data store is just data that does not presist over a reboot, then in theory any node or group of nodes in a YANG data model could be ephemeral. The YANG data module must indicate what portion of the data model (if any) is ephemeral.

   * A YANG data module could be all ephemeral (e.g. [I-D.ietf-i2rs-rib-data-model]) with no directly associated configuration models,

   * A YANG model could be all ephemeral but associated with a configuration model

   * or a single data node or data tree could be made ephemeral.

6. The management protocol (NETCONF/RESTCONF) needs to signal which poritons of a data model(node, tree, or data model) are ephemeral in the module library [RFC7895].

4.3. I2RS Agent Caching of Ephemeral Data

The multiple control-plane datastore model [I-D.nmdsdt-netmod-revised-datastores] architecture allows multiple datastores which could allow an implementation of caching of ephemeral data in the I2RS Agent by having a main and a backup I2RS agent. Early implementations should at least support the single ephemeral data model, but MAY support the multiple datastore mode. It is important that these early implementations provide feedback for standardization on the following:

the policy knobs needed to make single ephemeral control planes datastores function,

the policy knobs neeed to make multiple ephemeral control plane datastores which support caching work.

4.4. Massive Amounts of Configuration Data

Large amounts of data can flow from the I2RS agent to the I2RS client, or from the I2RS client to the I2RS Agent. The I2RS client may set or query ephemeral configuration in the routing system via the I2RS agent and receive operational state, notifications, or logging from the I2RS Agent on behalf of the I2RS routing system. I2RS Clients can send large amount of ephemeral configuration data to the I2RS Agent. The writes may be done via NETCONF (<edit-config> or an rpc function), or via RESTCONF (PUT, PATCH, POST). Reads can be done via NETCONF <get-config> or RESTCONF GET or query.

The I2RS RIB Data Model [I-D.ietf-i2rs-rib-data-model] also supports the use of rpc to add/delete RIBs, add/delete/update routes, and add/delete nexthops. If the I2RS client does a small to medium number of writes to the I2RS ephemeral state in the I2RS Agent in a routing system, the full validation that NETCONF or RESTCONF does will be able to be done without any reduction in speed to the I2RS high-performance system. For example, if the I2RS RIB Data Model has adds a 1000 routes, the I2RS RIB use of rpc to add/delete/update routes should be able to provide a high-performance system. Alternatively the NETCONF <edit-config> could update these 1000 routes with a write, or the RESTCONF POST, PUT or PATCH should be able to add the 1000 routes.

If a large number of ephemeral routes or filters are written (updates or new) by the I2RS Client to the ephemeral state in the I2RS agent, one of the key issues for a high performance interface is the time it takes to validate routes. Due to this concern, the I2RS architecture was design to allow less than the full NETCONF or RESTCONF validation. The concept is that the I2RS routes would be validated within the I2RS client and sent via a 99.999% reliable connection. In this scenario, the I2RS Agent would trust the validation that the I2RS Client did, and the communication of the route additions via the network connection.

An experiment regarding this has been done with the ODL code base update of ephemeral routes, but additional experimentation needs to be done prior to finalizing this design. Section 3.4.2 reviews how this process might be done, but many open issues exist in implementing this "low-validation" interface. Without additional experimentation and prototype code, this type of "low-validation",

4.5. Write Error handling

This section reviews I2RS normal error handling and error handling for rpc with no validation checks.

4.5.1. Normal validation checks

An I2RS agent validates an I2RS client’s information by examining the following:

- o message syntax validation,
- o syntax validation for nodes of data model,
- o referential checks (leafref checks MUST clauses, and instance indentifier),

- o checks groups of data within a data model or groups of data across data models,
- o write access to data,
- o if write access and values already exist, if I2RS client write access is higher than existing priority.

4.5.1.1. Reduced Validation (Experimental)

Can the I2RS protocol allow for reduced error checking? The need for speed in the I2RS protocol insertions in to the I2RS RIB suggest that it is worth experimenting for reduced validation in order to obtain high levels of throughput. If NETCONF or RESTCONf streams pre-checked routes to the datastore, what happens? Implementation experience is needed to determine the feasibility of this approach.

This feature may require a operator-applied policy knob swith a "no validation" feature

- o operator-applied policy knob enabling this feature;
- o rpc in a data model with the yang "ephemeral-validation no-check;"

4.6. IPFIX for traffic monitoring

Due to the potentially large data flow the traffic measurment statistics generate, these statistics are best handled by publication techniques within NETCONF or a separate protocol such as IPFIX. In the future version of the I2RS protocol may desire to support a data stream outbound from the I2RS Agent to an I2RS client via the IPFIX protocol.

4.7. Binary encoding of RESTCONF/NETCONF

The binary encoding of JSON or XML encodnig in RESTCONF or NETCONF may provide a better throughput. Research needs to be done on what is the appropriate binary encoding.

4.8. Ephemeral state in DDoS environments

I2RS ephemeral state may operate in places where there is a DDoS attacks where the network devices are attacked. Is one attack plane the ability to remove all tracing if the I2RS reboots an attack vector?

5. Yang Changes

The data modules supporting the ephemeral datastore can use the Yang module library to describe their datastore.

The following key word must be able to specify ephemeral

    ephemeral true;

Nice to have features:

It would be helpful for implementation of I2RS ephemeral data models to determine if the I2RS protocol feature set can support the I2RS data model needs. For this reason, it is helpful to group protocol features into "versions" and to put flags in the data model. At this point, the best place to put the summary of features is in an data model which defines these features. The discussion between implementers should be whether it is useful to have this features in some general yang location. An example of features that might be needed are:

- o i2rs version indicator;
- o i2rs transport-nonsecure "ok-to-use";
- o i2rs ephemeral-validation nocheck;
- o I2rs caching

6. IANA Considerations

This is a protocol strawman - nothing is going to IANA.

7. 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.

8. Acknowledgements

This document good work arises out of discussions with many experts in NETCONF, NETMOD, and I2RS WGs including:

- o Alia Atlas,

- o Ignas Bagdonas,
- o Andy Bierman,
- o Alex Clemm,
- o Eric Voit,
- o Kent Watsen,
- o Jeff Haas,
- o Russ White,
- o Keyur Patel,
- o Hariharan Ananthakrishnan,
- o Dean Bogdanavich,
- o Anu Nair,
- o Juergen Schoenwaelder, and
- o Kent Watsen.

Any errors or assumptions should be blamed on the authors, and not these experts.

9. References

9.1. 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>.

9.2. Informative References

[I-D.ietf-i2rs-ephemeral-state]
Haas, J. and S. Hares, "I2RS Ephemeral State Requirements", draft-ietf-i2rs-ephemeral-state-22 (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-06 (work in progress), July 2016.

[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-09 (work in progress), July 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-02 (work in progress), November 2016.

[I-D.ietf-i2rs-yang-l3-topology]
Clemm, A., Medved, J., Varga, R., Liu, X., Bryskin, I., Guo, A., Ananthakrishnan, H., Bahadur, N., and V. Beeram, "A YANG Data Model for Layer 3 Topologies", draft-ietf-i2rs-yang-l3-topology-05 (work in progress), November 2016.

[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. and G. Wu, "Keystore Model", draft-ietf-netconf-keystore-00 (work in progress), October 2016.

[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-client-server]
Watsen, K. and J. Schoenwaelder, "RESTCONF Client and Server Models", draft-ietf-netconf-restconf-client-server-01 (work in progress), November 2016.

[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-yang-patch]
Bierman, A., Bjorklund, M., and K. Watsen, "YANG Patch Media Type", draft-ietf-netconf-yang-patch-13 (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-04 (work in progress), October 2016.

[I-D.ietf-netconf-zerotouch]
Watsen, K. and M. Abrahamsson, "Zero Touch Provisioning for NETCONF or RESTCONF based Management", draft-ietf-netconf-zerotouch-11 (work in progress), October 2016.

[I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-03 (work in progress), October 2016.

[I-D.ietf-netmod-syslog-model]
Wildes, C. and K. Koushik, "A YANG Data Model for Syslog Configuration", draft-ietf-netmod-syslog-model-11 (work in progress), November 2016.

[I-D.nmdsdt-netmod-revised-datastores]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "A Revised Conceptual Model for YANG Datastores", draft-nmdsdt-netmod-revised-datastores-00 (work in progress), October 2016.

Authors' Addresses

Susan Hares
Huawei
Saline
US

Email: shares@ndzh.com

Amit Daas
Ericsson

Email: amit.dass@ericsson.com