Protocol Independent Use Cases for an Interface to the Routing System
draft-white-i2rs-use-case-06

Abstract

   Programmatic interfaces to provide control over individual forwarding
   devices in a network promise to reduce operational costs while
   improving scaling, control, and visibility into the operation of
   large scale networks. To this end, several programmatic interfaces
   have been proposed. OpenFlow, for instance, provides a mechanism to
   replace the dynamic control plane processes on individual forwarding
   devices throughout a network with off box processes that interact
   with the forwarding tables on each device. Another example is
   NETCONF, which provides a fast and flexible mechanism to interact
   with device configuration and policy.

   There is, however, no proposal which provides an interface to all
   aspects of the routing system as a system. Such a system would not
   interact with the forwarding system on individual devices, but rather
   with the control plane processes already used to discover the best
   path to any given destination through the network, as well as
   interact with the routing information base (RIB), which feeds the
   forwarding table the information needed to actually switch traffic at
   a local level.

   This document describes a set of use cases such a system could
   fulfill. It is designed to provide underlying support for the
   framework, policy, and other drafts describing the Interface to the
   Routing System (I2RS).

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 January 5, 2015.

Copyright Notice

Table of Contents

[1](1). Introduction

The Architecture for the Interface to the Routing System
[I-D.ietf-i2rs-architecture] allows for a mechanism where the
distributed control plane can be augmented by an outside control
plane through an open, accessible interface, including the Routing
Information Base (RIB), in individual devices to solve issues
described in [I-D.ietf-i2rs-problem-statement]. The RIB Info Model
[I-D.ietf-i2rs-rib-info-model] specifies the information elements
accessible by the I2RS system in the RIB.

This represents a "halfway point" between completely replacing the
traditional distributed control plane and directly configuring
devices to distribute policy or modifications to routing through off-
board processes. This draft proposes a set of use cases that explain
where the work described utilizing the RIB information model will be
useful. The goal is to inform not only the community's understanding
of where I2RS fits in the larger scheme of SDN proposals, but also to
inform the requirements, framework, and specification of I2RS to
provide the best fit for the purposes which make the most sense for
this type of programmatic interface.

Towards this end the authors have searched for a number of different
use cases representing not only complex modifications of the control
plane, including interaction with applications and network
conditions, but also simpler use cases. The array of use cases
presented here should provide the reader with a solid understanding
of the power of an SDN solution that will augment, rather than
replace, traditional distributed control planes.

1.1. 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. Summary of Requirements

Each use case is presented in its own section with a list of
requirements. The full list of requirements is below. Each
requirement in these lists has an identifier PI-REQnn where PI-REQ
standard for protocol independent requirements and nn is a
requirement number. Each unique requirement has been given a number
so in some use cases the numbers may skip numbers. For example, use
case 2 has requirements PI-REQ01, PI-REQ06, and PI-REQ07.

At the end of this document, these use case requirements are compared
against the current I2RS RIB informational model (RIB IM)
([I-D.ietf-i2rs-rib-info-model]). In this section, the RIB Inform
requirements are numbered PI-RIB_IM-nn where nn is the requirement
number.

o PI-REQ01: The ability to monitor the available routes installed in
   the RIB of each forwarding device, including near real time
   notification of route installation and removal. This information
   must include the destination prefix (NLRI), a table identifier (if
   the forwarding device has multiple forwarding instances), the
   metric of the installed route, and an identifier indicating the
   installing process.

o PI-REQ02: The ability to install source and destination based
   routes in the local RIB of each forwarding device. This must
   include the ability to supply the destination prefix (NLRI), the
   source prefix (NLRI), a table identifier (if the forwarding device
   has multiple forwarding instances), a route preference, a route
   metric, a next hop, an outbound interface, and a route process
   identifier.

o PI-REQ03: The ability to install a route to a null destination,
   effectively filtering traffic to this destination.

o PI-REQ04: The ability to interact with various policies configured
   on the forwarding devices, in order to inform the policies
   implemented by the dynamic routing processes. This interaction
   should be through existing configuration mechanisms, such as
   NETCONF, and should be recorded in the configuration of the local
   device so operators are aware of the full policy implemented in
   the network from the running configuration.

o PI-REQ05: The ability to interact with traffic flow and other
   network traffic level measurement protocols and systems, in order
   to determine path performance, top talkers, and other information
   required to make an informed path decision based on locally
   configured policy.

o PI-REQ06: The ability to install destination based routes in the
   local RIB of each forwarding device. This must include the
   ability to supply the destination prefix (NLRI), a table
   identifier (if the forwarding device has multiple forwarding
   instances), a route preference, a route metric, a next hop, an
   outbound interface, and a route process identifier.

o PI-REQ07: The ability to read the local RIB of each forwarding
   device, including the destination prefix (NLRI), a table
   identifier (if the forwarding device has multiple forwarding
   instances), the metric of each installed route, a route
   preference, and an identifier indicating the installing process.

o PI-REQ08: The ability to read the tables of other local protocol
   processes running on the device. This reading action should be
   supported through an import/export interface which can present the
   information in a consistent manner across all protocol
   implementations, rather than using a protocol specific model for
   each type of available process.

o PI-REQ09: The ability to inject information directly into the
   local tables of other protocol processes running on the forwarding
   device. This injection should be supported through an import/

export interface which can inject routing information in a
consistent manner across all protocol implementations, rather than
using a protocol specific model for each type of available
process.

o PI-REQ10: The ability to interact with policies and configurations
   on the forwarding devices using time based processing, either
   through timed auto-rollback or some other mechanism. This
   interaction should be through existing configuration mechanisms,
   such as NETCONF, and should be recorded in the configuration of
   the local device so operators are aware of the full policy
   implemented in the network from the running configuration.

3. Distributed Reaction to Network Based Attacks

Quickly modifying the control plane to reroute traffic for one
destination while leaving a standard configuration in place (filters,
metrics, and other policy mechanisms) is a challenge --but this is
precisely the challenge of a network engineer attempting to deal with
a network incursion. The ability to redirect specific flows of
information or specific classes of traffic into, through, and back
out of traffic analyzers on the fly is crucial in these situations.
The following network diagram provides an illustration of the
problem.

```
 Valid Source---\  /--R2--------------------\
                R1                           R3---Valid Destination
 Attack Source--/  \--Monitoring Device-----/
```

Modifying the cost of the link between R1 and R2 to draw the attack
traffic through the monitoring device in the distributed control
plane will, of necessity, also draw the valid traffic through the
monitoring device. Drawing valid traffic through a monitoring device
introduces delay, jitter, and other quality of service issues, as
well as posing a problem for the monitoring device itself in terms of
traffic load and management.

An I2RS controller could stand between the detection of the attack
and the control plane to facilitate the rapid modification of control
and forwarding planes to either block the traffic or redirect it to
analysis devices connected to the network.

```
                +----------+
                |i2rs-client|-------------------
                |          |-----------+      |
                -----------+           |      |
                    |              +------+   |
                    |              | ir2s |   |
                    |              | agent|   |
     Valid Source--\  |        /---|R2    |-------+\
                +-------+/    +------+   |       \
                |R1 i2rs|                |    R3--Valid
                | agent|                 |    Destination
                +-------+         i2rs agent
     Attack Source--/     \--Monitoring Device-----/
```
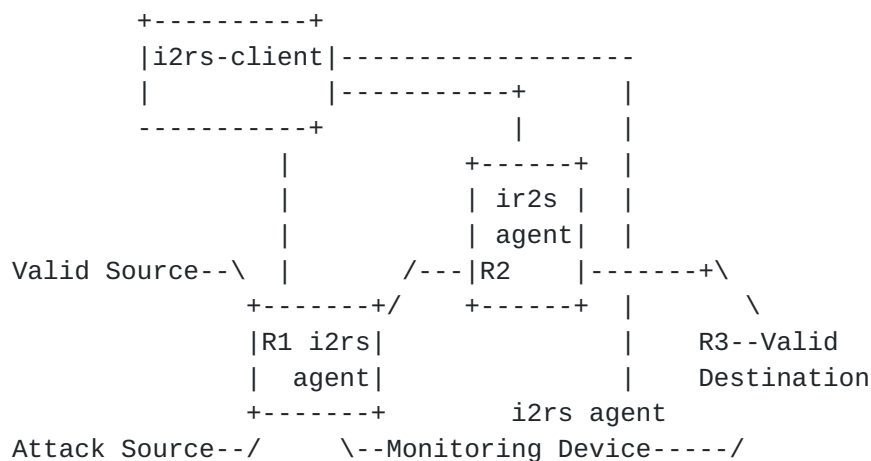
Summary of I2RS Capabilities and Interactions:

In the lists below, the REQnn in which nn is a number indicates a
unique requirement from the use cases.

o PI-REQ01: The ability to monitor the available routes installed in
   the RIB of each forwarding device, including near real time
   notification of route installation and removal. The information
   pulled from the RIB must include the destination prefix (NLRI),
   the table identifier (if the forwarding device has multiple
   forwarding instances), the metric of the installed route, and the
   identifier for the installing process.

o PI-REQ02: The ability to install source and destination based
   routes in the local RIB of each forwarding device. This must
   include the ability to supply the destination prefix (NLRI), the
   source prefix (NLRI), a table identifier (if the forwarding device
   has multiple forwarding instances), a route preference, a route
   metric, a next hop, an outbound interface, and a route process
   identifier.

o PI-REQ03: The ability to install a route to a null destination,
   effectively filtering traffic to this destination.

o PI-REQ04: The ability to interact with various policies configured
   on the forwarding devices, in order to inform the policies
   implemented by the dynamic routing processes. This interaction
   should be through existing configuration mechanisms, such as
   NETCONF, and should be recorded in the configuration of the local
   device so operators are aware of the full policy implemented in
   the network from the running configuration.

o PI-REQ5: The ability to interact with traffic flow and other
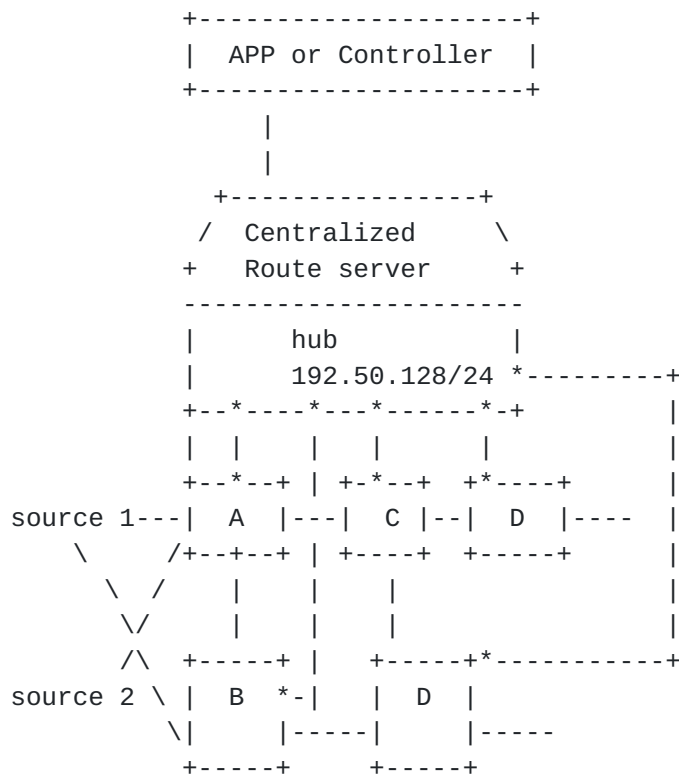   network traffic level measurement protocols and systems, in order

to determine path performance, top talkers, and other information
required to make an informed path decision based on locally
configured policy.

4. Remote Service Routing

In hub and spoke overlay networks, there is always an issue with
balancing between the information held in the spoke routing table,
optimal routing through the network underlying the overlay, and
mobility. Most solutions in this space use some form of centralized
route server that acts as a directory of all reachable destinations
and next hops, a protocol by which spoke devices and this route
server communicate, and caches at the remote sites.

An I2RS solution would use the same elements, but with a different
control plane. Remote sites would register (or advertise through
some standard routing protocol, such as BGP), the reachable
destinations at each site, along with the address of the router (or
other device) used to reach that destination. These would, as
always, be stored in a route server (or several redundant route
servers) at a central location.

When a remote site sends a set of packets to the central location
that are eventually destined to some other remote site, the central
location can forward this traffic, but at the same time simply
directly insert the correct routing information into the remote
site's routing table. If the location of the destination changes,
the route server can directly modify the routing information at the
remote site as needed.

```
                +---------------------+
                |  APP or Controller  |
                +---------------------+
                     |
                     |
                  +----------------+
                 /  Centralized     \
                 +   Route server      +
                 ----------------------
                 |       hub         |
                 |       192.50.128/24 *---------+
                 +--*----*---*------*-+           |
                 |  |    |   |      |             |
                 +--*--+ | +-*--+  +*----+        |
    source 1---|  A  |---|  C |--|  D  |----    |
         \      /+--+--+ | +----+   +-----+        |
          \  /    |     |     |                    |
           \/     |     |     |                    |
          /\   +-----+ |    +-----+*----------+
    source 2 \ |  B  *-|    |  D  |
            \|      |-----|      |-----
            +-----+      +-----+
```

```
    *- are RS connections
    |- are hub/spoke connects
```

An interesting aspect of this solution is that no new and specialized
protocols are needed between the remote sites and the centralized
route server(s). Normal routing protocols can be used to notify the
centralized route server(s) of modifications in reachability
information, and the route server(s) can respond as needed, based on
local algorithms optimized for a particular application or network.
For instance, short lived flows might be allowed to simply pass
through the hub site with no reaction, while longer lived flows might
warrant a specific route to be installed in the remote router.
Algorithms can also be developed that would optimize traffic flow
through the overlay, and also to remove routing entries from remote
devices when they are no longer needed based on far greater
intelligence than simple non-use for some period of time.

Summary of I2RS Capabilities and Interactions:

o PI-REQ01: The ability to monitor the available routes installed in
   the RIB of each forwarding device, including near real time
   notification of route installation and removal. This information
   must include the destination prefix (NLRI), a table identifier (if

the forwarding device has multiple forwarding instances), the
metric of the installed route, and an identifier indicating the
installing process.

o PI-REQ06: The ability to install destination based routes in the
   local RIB of each forwarding device. This must include the
   ability to supply the destination prefix (NLRI), a table
   identifier (if the forwarding device has multiple forwarding
   instances), a route preference, a route metric, a next hop, an
   outbound interface, and a route process identifier.

o PI-REQ07: The ability to read the local RIB of each forwarding
   device, including the destination prefix (NLRI), a table
   identifier (if the forwarding device has multiple forwarding
   instances), the metric of each installed route, a route
   preference, and an identifier indicating the installing process.

5. Within Data Center Routing

Data Centers have evolved into massive topologies with thousands of
server racks and millions of hosts. Data Centers use BGP with ECMP,
ISIS (with multiple LAGs), or other protocols to tie the data center
together. Data centers are currently designed around a three or four
tier structure with: server, top-of-rack switches, aggregation
switches, and router interfacing the data center to the Internet.
[I-D.lapukhov-bgp-routing-large-dc] examines many of these elements
of data center design.

One element of these Data Center routing infrastructures is the
ability to quickly read topology information and execute
configuration from a centralized location. Key to this environment
is the tight feedback loop between learning about topology changes or
loading changes, and instantiating new routing policy. Without I2RS,
may Data Centers are using extra physical topologies or logical
topologies to work around the features.

An I2RS solution would use the same elements, but with a different
control plane. The I2RS enabled control plane could provide the Data
Center 4 tier infrastructure the quick access to topology and data
flow information needed for traffic flow optimization. Changes to
the Data Center infrastructure done via I2RS could have a tight
feedback loop.

Again, this solution would reduce the need for new and specialized
protocols while giving the Data Center the control it desire. The
I2RS routing interface could be extended to virtual routers.

Summary of I2RS Capabilities and Interactions:

o PI-REQ01: The ability to monitor the available routes installed in
   the RIB of each forwarding device, including near real time
   notification of route installation and removal. This information
   must include the destination prefix (NLRI), a table identifier (if
   the forwarding device has multiple forwarding instances), the
   metric of the installed route, and an identifier indicating the
   installing process.

o PI-REQ04: The ability to interact with various policies configured
   on the forwarding devices, in order to inform the policies
   implemented by the dynamic routing processes. This interaction
   should be through existing configuration mechanisms, such as
   NETCONF, and should be recorded in the configuration of the local
   device so operators are aware of the full policy implemented in
   the network from the running configuration.

o PI-REQ05: The ability to interact with traffic flow and other
   network traffic level measurement protocols and systems, in order
   to determine path performance, top talkers, and other information
   required to make an informed path decision based on locally
   configured policy.

o PI-REQ06: The ability to install destination based routes in the
   local RIB of each forwarding device. This must include the
   ability to supply the destination prefix (NLRI), a table
   identifier (if the forwarding device has multiple forwarding
   instances), a route preference, a route metric, a next hop, an
   outbound interface, and a route process identifier.

o PI-REQ07: The ability to read the local RIB of each forwarding
   device, including the destination prefix (NLRI), a table
   identifier (if the forwarding device has multiple forwarding
   instances), the metric of each installed route, a route
   preference, and an identifier indicating the installing process.

o PI-REQ08: The ability to read the tables of other local protocol
   processes running on the device. This reading action should be
   supported through an import/export interface which can present the
   information in a consistent manner across all protocol
   implementations, rather than using a protocol specific model for
   each type of available process.

o PI-REQ09: The ability to inject information directly into the
   local tables of other protocol processes running on the forwarding
   device. This injection should be supported through an import/
   export interface which can inject routing information in a
   consistent manner across all protocol implementations, rather than

using a protocol specific model for each type of available
process.

6. Temporary Overlays between Data Centers

Data Centers within one organization may operate as one single entity
even though they may be geographically distributed. Applications are
load balanced within Data Centers and between data centers to take
advantage of cost economics in power, storage, and server
availability for compute resources. Applications are also transfer
to alternate data centers in case of failures within a data center.
To reduce time during failure, Data Centers often replicate user
storage between two or more data centers. During the transfer of
stored information prior to a Data Center to Data Center move, the
Data Center controllers need to dynamically acquire a large amount of
inter-data center bandwidth through an overlay network, often during
off hours.

I2RS could provide the connection between the overlay network
configuration, local policies, and the control plane to dynamically
bring a large bandwidth inter-data center overlay or channel into
use, and then to remove it from use when the data transfer is
completed.

Similarly, during a fail-over, a control process within data centers
interacts with a group host process and the network to seamless move
the processing to another data center. During the fail-over case,
additional process state may need to be moved as well to restart the
system. The difference between these data-to-data center moves is
immediate and urgent need to move systems. If an application (such
as medical or banking services) pays to have this type of fail-over,
it is likely the network service this application uses will pay for
the preemption on network bandwidth and a Quality of Service (QoS) on
the data transfer to allow the failover traffic to flow quickly to
the remote datacenter. I2RS can allow the Data Center network and
the Network connecting the data center to preempt other best-effort
traffic to send this priority data flow and to set a QoS that will
allow quick data transfer. After the high priority data flow has
finished, networks can return to their previous condition.

Summary of I2RS Capabilities and Interactions:

o PI-REQ01: The ability to monitor the available routes installed in
   the RIB of each forwarding device, including near real time
   notification of route installation and removal. This information
   must include the destination prefix (NLRI), a table identifier (if
   the forwarding device has multiple forwarding instances), the

metric of the installed route, and an identifier indicating the
installing process.

o PI-REQ02: The ability to install source and destination based
   routes in the local RIB of each forwarding device. This must
   include the ability to supply the destination prefix (NLRI), the
   source prefix (NLRI), a table identifier (if the forwarding device
   has multiple forwarding instances), a route preference, a route
   metric, a next hop, an outbound interface, and a route process
   identifier.

o PI-REQ05: The ability to interact with traffic flow and other
   network traffic level measurement protocols and systems, in order
   to determine path performance, top talkers, and other information
   required to make an informed path decision based on locally
   configured policy.

o PI-REQ06: The ability to install destination based routes in the
   local RIB of each forwarding device. This must include the
   ability to supply the destination prefix (NLRI), a table
   identifier (if the forwarding device has multiple forwarding
   instances), a route preference, a route metric, a next hop, an
   outbound interface, and a route process identifier.

o PI-REQ07: The ability to read the local RIB of each forwarding
   device, including the destination prefix (NLRI), a table
   identifier (if the forwarding device has multiple forwarding
   instances), the metric of each installed route, a route
   preference, and an identifier indicating the installing process.

o PI-REQ10: The ability to interact with policies and configurations
   on the forwarding devices using time based processing, either
   through timed auto-rollback or some other mechanism. This
   interaction should be through existing configuration mechanisms,
   such as NETCONF, and should be recorded in the configuration of
   the local device so operators are aware of the full policy
   implemented in the network from the running configuration.

7. Comparison of I2RS requirements with I2RS RIB Information Model

Comparison of I2RS Capabilities versus the I2RS RIB

In summary, the I2RS client/agent architecture and protocol SHOULD
support the following requirements:

   PI-RIB_IM-REQ01: The RIB Info Model [I-D.ietf-i2rs-rib-info-model]
   specifies the routes as associated with routing-instance,
   interface-list and RIBs within a Router (virtual or physical)

identified by Router_ID. Each route has a prefixes, route
attributes, family attributes (IPv4, Ipv6, MPLS, MAC, interface),
and next-hop list. The RIB info model does not keep information
on the FIB the route was installed in, the metric of the installed
route, or the identifier of the installing process. The RIB Info
Model [I-D.ietf-i2rs-rib-info-model] does provide a notification
for route change with an installed in FIB flag, active flag, and
reason (for non-installation).

PI-RIB_IM-REQ02: The RIB Info Model [I-D.ietf-i2rs-rib-info-model]
supports installing source-destination routes for IPv4 and IPv6
for a RIB associated with set of RIBs within a route instance of a
Router (virtual or physical). If there is policy that links RIBs
within a route instance of router to a specific FIB, it is not
clearly stated in the RIB Info Model
[I-D.ietf-i2rs-rib-info-model]. The source-destination is only
for IPv4 and IPv6 NLRI. PI-RIB_IM-REQ02: The RIB Info Model
[I-D.ietf-i2rs-rib-info-model]. does has a route_preference
(ROUTE_PREFERENCE), but not a route metric. The nexthop field
does have a primary/back preference, a load balancing weight, and
an egress (outbound) interface per nexthop, and a RIB_ID. It is
unclear if the RIB_ID serves the same purpose as the multiple
forwarding instances.

PI-RIB_IM-REQ03: The RIB Info Model does not provide a specific
indication that the default (zero length prefix) route can be
installed, but this can be implied from the different match
lengths.

PI-RIB_IM-REQ04: The ability to interact with various policies via
existing configuration functions such as NETCONF has not be
specified directly in RIB Informational Model
[I-D.ietf-i2rs-rib-info-model] or any other informational model.

PI-TED-REQ01/PI-RIB_IM_REQ05: The PI-REQ05 use case requirement
requires the ability to interact with traffic flow and other
network traffic level measurement protocols and systems is not
included the I2RS RIB information model. The traffic flow and
traffic level measurement does not need to be in the I2RS RIB, but
in some Traffic Engineering Information model.

PI-RIB_IM-REQ06: The RIB Info Model [I-D.ietf-i2rs-rib-info-model]
supports installing destination routes in a RIB for all RIB
families from a route-instance. The route_instance is identified
by the tuple of INSTANCE_NAME and ROUTER_ID. Each route may
contain a route preference and multiple next hops, but it does not
contain a metric per route. Each nexthop may contains a RIB_ID to
look-up nexthop, an egress-interface, tunnel nexthop (logical or

encapsulated). This must include the ability to supply the destination prefix (NLRI), a table identifier (if the forwarding device has multiple forwarding instances), a route preference, a route metric, a next hop, an outbound interface, and a route process identifier.

PI-RIB_IM-REQ07: The RIB Info Model [I-D.ietf-i2rs-rib-info-model] supports the ability to read the RIBs associated with each routing-instance. The routing instance is identified by tuple of ROUTE_INSTANCE plus optionally interface_list and router ID. The RIBS are identified by NAME and RIB_FAMILY. The routes within the RIB Can be identified by route-match that identifies destination (IPv4, IPv6, IEEE MAC, MPLS, interface). The information read per route is the destination prefix (NLRI), route preference and multiple nexthops (each with associated metric and RIB). However, it does not indicate the metric of an installed route and the identifier of the installing process.

PI-RIB_IM-REQ08: The ability to read the tables of other local protocol processes running on the device. This reading action should be supported through an import/export interface which can present the information in a consistent manner across all protocol implementations, rather than using a protocol specific model for each type of available process.

PI-RIB_IM-REQ09: No informational model supports the ability to inject information of other protocol processing running on the forwarding device. The RIB Info Model [I-D.ietf-i2rs-rib-info-model] inserts the routes with RIB with a The ability to inject information directly into the local tables of other protocol processes running on the forwarding device. This injection should be supported through an import/export interface which can inject routing information in a consistent manner across all protocol implementations, rather than using a protocol specific model for each type of available process.

PI-RIB_IM-REQ10: The ability to interact with policies and configurations on the forwarding devices using time based processing, either through timed auto-rollback or some other mechanism. This interaction should be through existing configuration mechanisms, such as NETCONF, and should be recorded in the configuration of the local device so operators are aware of the full policy implemented in the network from the running configuration.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. 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-04 (work in
           progress), June 2014.

[I-D.ietf-i2rs-problem-statement]
           Atlas, A., Nadeau, T., and D. Ward, "Interface to the
           Routing System Problem Statement", draft-ietf-i2rs-
           problem-statement-04 (work in progress), June 2014.

[I-D.ietf-i2rs-rib-info-model]
           Bahadur, N., Folkes, R., Kini, S., and J. Medved, "Routing
           Information Base Info Model", draft-ietf-i2rs-rib-info-
           model-03 (work in progress), May 2014.

[I-D.lapukhov-bgp-routing-large-dc]
           Lapukhov, P., Premji, A., and J. Mitchell, "Use of BGP for
           routing in large-scale data centers", draft-lapukhov-bgp-
           routing-large-dc-06 (work in progress), August 2013.

Authors' Addresses

   Russ White
   Ericsson

   Email: russw@riw.us

   Susan Hares
   Huawei

   Email: shares@ndzh.com

Alvaro Retana
Cisco Systems, Inc.
7025 Kit Creek Road
Research Triangle Park, NC 27617
USA

Email: aretana@cisco.com