Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 10, 2015

A. Atlas, Ed.
Juniper Networks
T. Nadeau, Ed.
Brocade
D. Ward
Cisco Systems
January 6, 2015

# Interface to the Routing System Problem Statement
## draft-ietf-i2rs-problem-statement-05

## Abstract

As modern networks grow in scale and complexity, the need for rapid and dynamic control increases. With scale, the need to automate even the simplest operations is important, but even more critical is the ability to quickly interact with more complex operations such as policy-based controls.

In order to enable network applications to have access to and control over information in the Internet's routing system, we need a publicly documented interface specification. The interface needs to support real-time, asynchronous interactions using data models and encodings that are efficient and potentially different from those available today. Furthermore, the interface must be tailored to support a variety of use cases.

This document expands upon these statements of requirements to provide a detailed problem statement for an Interface to the Routing System (I2RS).

Table of Contents



## 1. Introduction

As modern networks grow in scale and complexity, the need for rapid, flexible and dynamic control increases. With scale, the need to automate even the simplest operation is important, but even more critical is the ability for network operators to quickly interact with these operations using mechanisms such as policy-based controls.

With complexity comes the need for more sophisticated automated network applications and orchestration software that can process large quantities of data, run complex algorithms, and adjust the routing state as required in order to support the network applications, their computations and their policies. Changes made to the routing state of a network by external applications must be

verifiable by those applications to ensure that the correct state has been installed in the correct places.

In the past, mechanisms to support the requirements outlined above have been developed piecemeal as proprietary solutions to specific





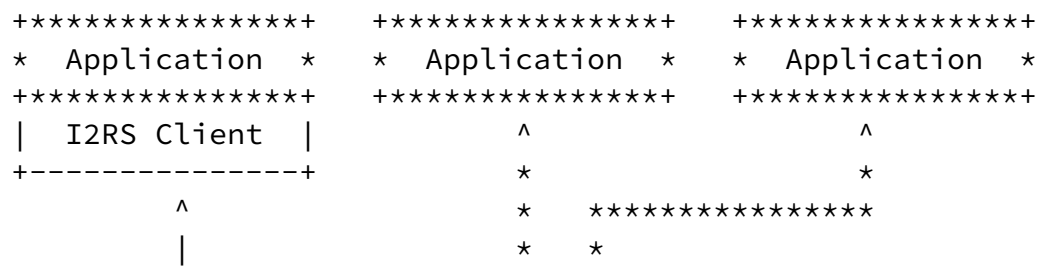
situations and needs. Many routing elements have an external interface to interact with routing - but since these vary between vendors, it is difficult to integrate use of those interfaces into a network. The existence of such proprietary interfaces demonstrates both that the need for such an interface is understood and that technology solutions are understood. What is needed are technological solutions with clearly defined operations that an application can initiate, and data-models to support such actions. These would facilitate wide-scale deployment of interoperable applications and routing systems. These solutions must be designed to facilitate rapid, isolated, secure, and dynamic changes to a device's routing system. In order to address these needs, the creation of an Interface to the Routing System (I2RS) is needed.

It should be noted that during the course of this document, the term "applications" is used. This is meant to refer to an executable program of some sort that has access to a network, such as an IP or MPLS network.

## 2. I2RS Model and Problem Area for The IETF

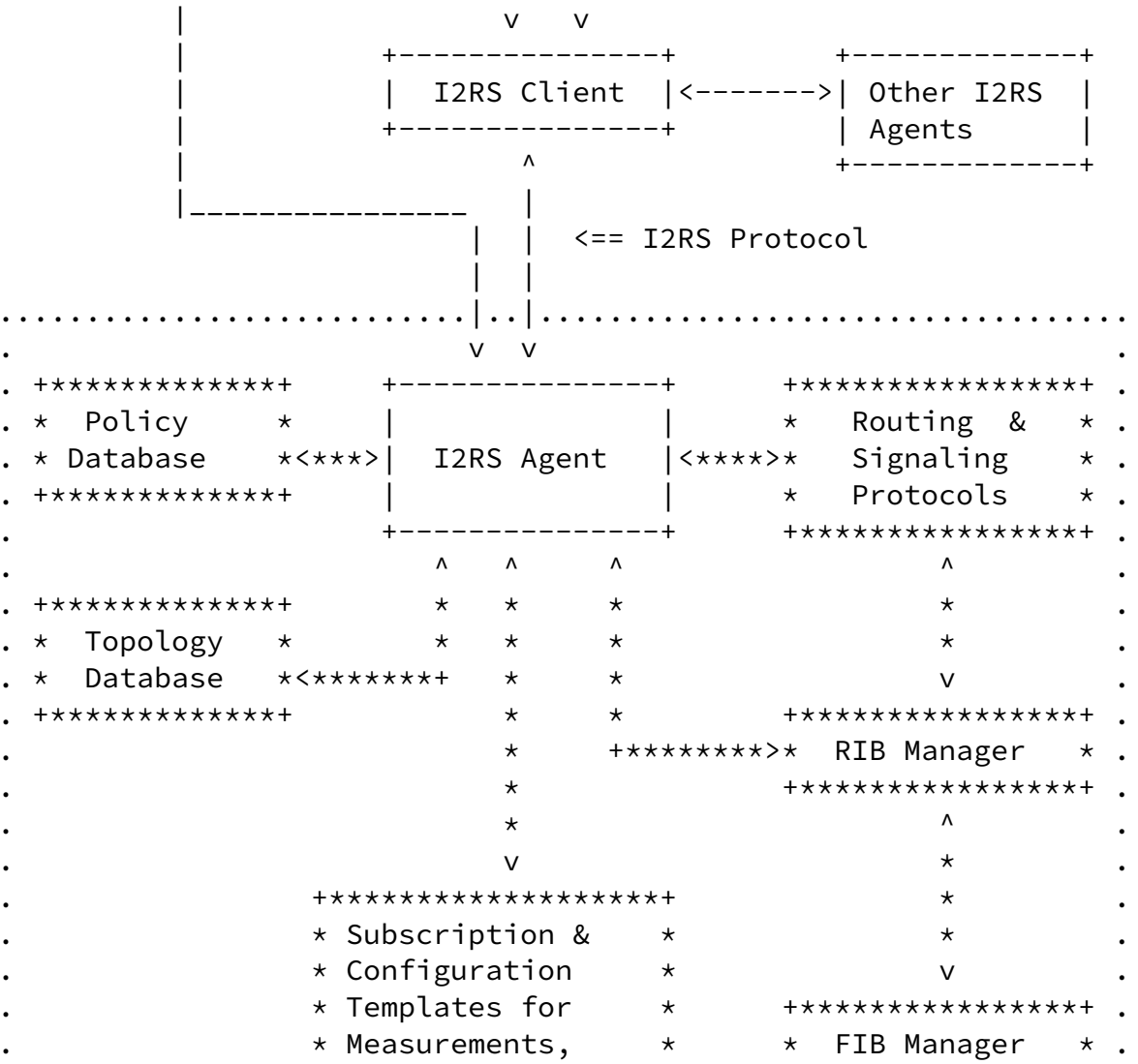
Managing a network of production devices running a variety of routing protocols involves interactions between multiple components within a device. Some of these components are virtual while some are physical; it may be desirable for many, or even all of these components to be made available to be managed and manipulated by applications, given that appropriate access, authentication, and policy hurdles have been crossed. The management of only some of these components require standardization, as others have already been standardized. The I2RS model is intended to incorporate existing mechanisms where appropriate, and to build extensions and new protocols where needed. The I2RS model and problem area for IETF work is illustrated in Figure 1. The I2RS Agent is associated with a routing element, which may or may not be co-located with a data-plane. The I2RS Client is used and controlled by one or more network applications; they may be co-located or the I2RS Client might be part

of a separate application, such as an orchestrator or controller. The scope of the data-models used by I2RS extends across the entire routing system and I2RS protocol.







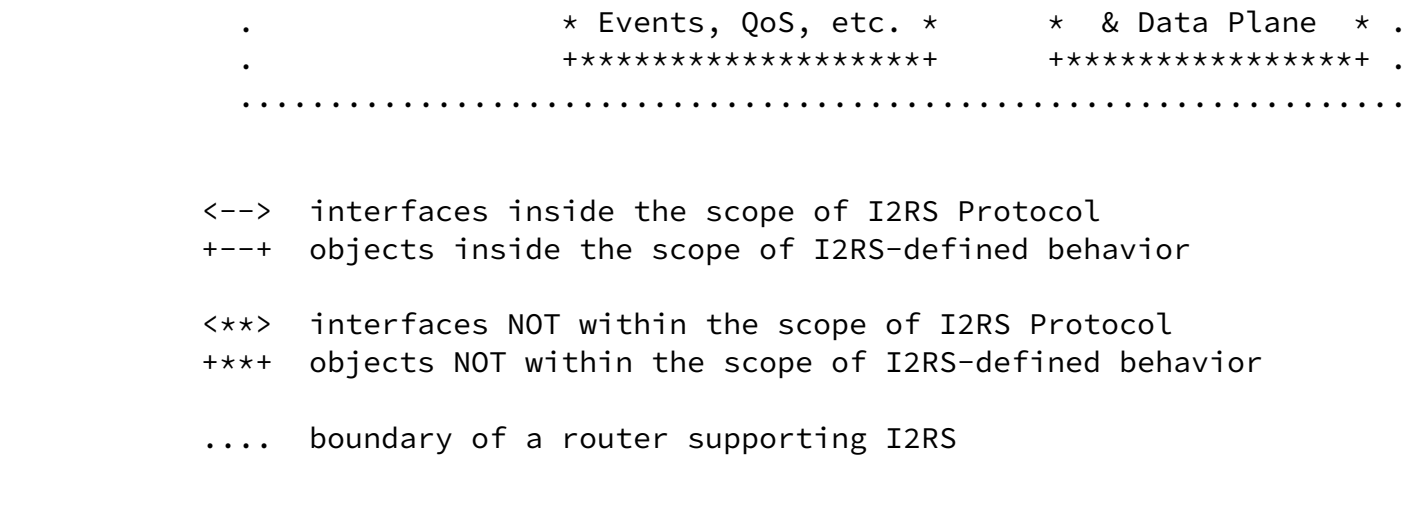



Figure 1: I2RS model and Problem Area

A critical aspect of I2RS is defining a suitable protocol or protocols to carry messages between the I2RS Clients and the I2RS Agent, and defining the data-models for use with those I2RS





protocol(s). The protocol should provide the key features specified in Section 5. The data models should translate into a concise transfer syntax, sent via the I2RS protocol, that is straightforward for applications to use (e.g., a Web Services design paradigm). The information transfer should use existing transport protocols to provide the reliability, security, and timeliness appropriate for the particular data.

The second critical aspect of I2RS is a set of meaningful data-models for information in the routing system and in a topology database. The data-model should describe the meaning and relationships of the modeled items. The data-models should be separable across different features of the managed components, versioned, and extendable. As shown in Figure 1, I2RS needs to interact with several logical components of the routing element: policy database, topology database, subscription and configuration for dynamic measurements/ events, routing signaling protocols, and its RIB manager. This interaction is both for writing (e.g. to policy databases or RIB manager) as well as for reading (e.g. dynamic measurement or topology database). An application should be able to combine data from individual routing elements to provide network-wide data-model(s).

## 3. Standard Data-Models of Routing State for Installation

There is a need to be able to precisely control routing and signaling state based upon policy or external measures. This can range from simple static routes to policy-based routing to static multicast replication and routing state. This means that, to usefully model next-hops, the data model employed needs to handle next-hop indirection and recursion (e.g. a prefix X is routed like prefix Y) as well as different types of tunneling and encapsulation. The relevant MIB modules (for example [RFC4292]) lack the necessary generality and flexibility. In addition, by having I2RS focus initially on interfaces to the RIB layer (e.g. RIB, LIB, multicast RIB, policy-based routing), the ability to use routing indirection allows flexibility and functionality that can't be as easily obtained at the forwarding layer.

Efforts to provide this level of control have focused on standardizing data models that describe the forwarding plane (e.g. ForCES [RFC3746]). I2RS posits that the routing system and a router's OS provide useful mechanisms that applications could usefully harness to accomplish application-level goals.

In addition to interfaces to the RIB layer, there is a need to configure the various routing and signaling protocols with differing dynamic state based upon application-level policy decisions. The range desired is not available via MIB modules at the present time.





Additionally, on March 2, 2014, the IESG issued a statement about Writeable MIB Modules [IESG-Statement] which is expected to limit creation of future writeable MIB modules.

## 4. Learning Router Information

A router has information that applications may require so that they can understand the network, verify that programmed state is installed in the forwarding plane, measure the behavior of various flows, and understand the existing configuration and state of the router. I2RS provides a framework so that applications can register for asynchronous notifications and can make specific requests for information.

Although there are efforts to extend the topological information

available, even the best of these (e.g., BGP-LS [I-D.ietf-idr-ls-distribution]) still provide only the current active state as seen at the IGP layer and above. Detailed topological state that provides more information than the current functional status (e.g. active paths and links) is needed by applications. Examples of missing information include paths or link that are potentially available (e.g. administratively down) or unknown (e.g. to peers or customers) to the routing topology.

For applications to have a feedback loop that includes awareness of the relevant traffic, an application must be able to request the measurement and timely, scalable reporting of data. While a mechanism such as IPFIX [RFC5470] may be the facilitator for delivering the data, the need for an application to be able to dynamically request that measurements be taken and data delivered is critical.

There are a wide range of events that applications could use for either verification of router state before other network state is changed (e.g. that a route has been installed), to act upon changes to relevant routes by others, or upon router events (e.g. link up/down). While a few of these (e.g. link up/down) may be available via MIB notifications today, the full range is not - nor has there been successfully deployed the standardized ability to set up the router to trigger different actions upon an event's occurrence so that a rapid reaction can be accomplished.

## 5. Aspects to be Considered for an I2RS Protocol

This section describes required aspects of a protocol that could support I2RS. Whether such a protocol is built upon extending existing mechanisms or requires a new mechanism requires further investigation.





The key aspects needed in an interface to the routing system are:

Multiple Simultaneous Asynchronous Operations: A single application should be able to send multiple independent atomic operations via I2RS without being required to wait for each to complete before sending the next.

Very Fine Granularity of Data Locking for Writing: When an I2RS

operation is processed, it is required that the data locked for writing is very granular (e.g. a particular prefix and route) rather than extremely coarse, as is done for writing configuration. This should improve the number of concurrent I2RS operations that are feasible and reduce blocking delays.

Multi-Headed Control: Multiple applications may communicate to the same I2RS agent in a minimally coordinated fashion. It is necessary that the I2RS agent can handle multiple requests in a well-known policy-based fashion. Data written can be owned by different I2RS clients at different times; data may even be overwritten by a different I2RS client. The details of how this should be handled are described in [I-D.ietf-i2rs-architecture].

Duplex: Communications can be established by either the I2RS client (i.e.: that resides within the application or is used by it to communicate with the I2RS agent), or the I2RS agent. Similarly, events, acknowledgements, failures, operations, etc. can be sent at any time by both the router and the application. The I2RS is not a pure pull-model where only the application queries to pull responses.

High-Throughput: At a minimum, the I2RS Agent and associated router should be able to handle a considerable number of operations per second (for example 10,000 per second to handle many individual subscriber routes changing simultaneously).

Low-Latency: Within a sub-second time-scale, it should be possible to complete simple operations (e.g. reading or writing a single prefix route).

Multi-Channel: It should be possible for information to be communicated via the interface from different components in the router without requiring going through a single channel. For example, for scaling, some exported data or events may be better sent directly from the forwarding plane, while other interactions may come from the control-plane. Thus a single TCP session would not be a good match.





Scalable, Filterable Information Access: To extract information in a

scalable fashion that is more easily used by applications, the ability to specify filtering constructs in an operation requesting data or requesting an asynchronous notification is very valuable.

Secure Control and Access: Any ability to manipulate routing state must be subject to authentication and authorization. Sensitive routing information may also need to be provided via secure access back to the I2RS client. Such communications must be integrity protected. Some communications will also require confidentiality.

Extensible and Interoperability: Both the I2RS protocol and models must be extensible and interoperate between different versions of protocols and models.

## 6. Acknowledgements

The authors would like to thank Ken Gray, Ed Crabbe, Nic Leymann, Carlos Pignataro, Kwang-koog Lee, Linda Dunbar, Sue Hares, Russ Housley, Eric Grey, Qin Wu, and Stephen Kent for their suggestions and review.

## 7. IANA Considerations

This document includes no request to IANA.

## 8. Security Considerations

Security is a key aspect of any protocol that allows state installation and extracting of detailed router state. The need for secure control and access is mentioned in Section 5 More architectural security considerations are discussed in [I-D.ietf-i2rs-architecture]. Briefly, the I2RS Agent is assumed to have a separate authentication and authorization channel by which it can validate both the identity and the permissions associated with an I2RS Client. Mutual authentication between the I2RS Agent and I2RS Client is required. Different levels of integrity, confidentiality, and replay protection are relevant for different aspects of I2RS.

## 9. 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-07 (work in progress), December 2014.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-07 (work in progress), November 2014.

[IESG-Statement]
IESG, "Writable MIB Module IESG Statement", March 2014, <https://www.ietf.org/iesg/statement/writable-mib-module.html>.

[RFC3746] Yang, L., Dantu, R., Anderson, T., and R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework", RFC 3746, April 2004.

[RFC4292] Haberman, B., "IP Forwarding Table MIB", RFC 4292, April 2006.

[RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, March 2009.

## Appendix A. Existing Management Interfaces

This section discusses as a single entity the combination of the abstract data models, their representation in a data language, and the transfer protocol commonly used with them. While other combinations of these existing standard technologies are possible, the ways described are those that have significant deployment.

There are three basic ways that routers are managed. The most popular is the command line interface (CLI), which allows both configuration and learning of device state. This is a proprietary interface resembling a UNIX shell that allows for very customized control and observation of a device, and, specifically of interest in this case, its routing system. Some form of this interface exists on almost every device (virtual or otherwise). Processing of information returned to the CLI (called "screen scraping") is a burdensome activity because the data is normally formatted for use by a human operator, and because the layout of the data can vary from device to device, and between different software versions. Despite its ubiquity, this interface has never been standardized and is unlikely to ever be standardized. CLI standardization is not considered as a candidate solution for the problems motivating I2RS.

The second most popular interface for interrogation of a device's

state, statistics, and configuration is The Simple Network Management Protocol (SNMP) and a set of relevant standards-based and proprietary





Management Information Base (MIB) modules. SNMP has a strong history of being used by network managers to gather statistical and state information about devices, including their routing systems. However, SNMP is very rarely used to configure a device or any of its systems for reasons that vary depending upon the network operator. Some example reasons include complexity, the lack of desired configuration semantics (e.g., configuration "roll-back", "sandboxing" or configuration versioning), and the difficulty of using the semantics (or lack thereof) as defined in the MIB modules to configure device features. Therefore, SNMP is not considered as a candidate solution for the problems motivating I2RS.

Finally, the IETF's Network Configuration (or NETCONF) protocol has made many strides at overcoming most of the limitations around configuration that were just described. However, the initial lack of standard data models have hampered the adoption of NETCONF. Naturally, I2RS may help define needed information and data models. Additional extensions to handle multi-headed control may need to be added to NETCONF and/or appropriate data models.

## Authors' Addresses

Alia Atlas (editor)
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

Email: akatlas@juniper.net

Thomas D. Nadeau (editor)
Brocade

Email: tnadeau@lucidvision.com

Dave Ward
Cisco Systems

Tasman Drive
San Jose, CA  95134
USA

Email: wardd@cisco.com