Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 31, 2013

A. Atlas, Ed.
T. Nadeau
Juniper Networks
D. Ward
Cisco Systems
July 30, 2012

Interface to the Routing System Problem Statement
draft-atlas-irs-problem-statement-00

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 applications to have access to and control over information in the Internet's routing system, we need a publically documented interface specification. The interface needs to support real-time, transaction-based interactions using efficient data models and encodings. Furthermore, the interface must support a variety of use cases including those where verified control feed-back loops are needed.

This document expands upon these statements of requirements to provide a problem statement for an interface to the Internet routing system.

Table of Contents

1. Introduction

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.

With complexity comes the need for more sophisticated automated 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 applications, their calculations 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 right places.

Mechanisms to support the requirements outlined above have been developed piecemeal as proprietary solutions to specific situations and needs. A standard protocol, clear operations that an application can initiate with that protocol, and data-models to support such actions would facilitate wide-scale deployment of interoperable applications and routing systems. That a protocol designed to facilitate rapid, isolated, secure, and dynamic routing changes is needed motivates the creation of an Interface to The Routing System (IRS).

2. IRS Model and Problem Area for The IETF

Managing a network of deployed devices running a variety of routing protocols involves interactions among multiple different functions and components that exist within the network. Some of these components are virtual while some are physical; all should 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 requires standardization, as others have already been standardized. The IRS model is intended to incorporate existing mechanisms where appropriate, and to build extensions and new protocols where needed. The IRS model and problem area proposed for IETF work is illustrated in Figure 1.

```
        +***************+   +***************+   +***************+
        *  Application  *   *  Application  *   *  Application  *
        +***************+   +***************+   +***************+
                 ^                   ^                  ^
                 *                   *                  *
                 *                   *   ****************
                 *                   *   *
                 v                   v   v
         +---------------+   +---------------+
         |  IRS Client   |   |  IRS Client   |
         +---------------+   +---------------+
                 ^                   ^
                 |________________   |
                                  |  |  <== IRS Protocol
                                  |  |
       ...........................|..|..................................
       .                          v  v                                 .
       . +*************+     +---------------+      +****************+ .
       . *    Policy   *     |               |      *   Routing  &   * .
       . *   Database  *<***>|  IRS Agent    |<****>*   Signaling    * .
       . +*************+     |               |      *   Protocols    * .
       .                     +---------------+      +****************+ .
       .                        ^   ^     ^                  ^         .
       . +*************+        *   *     *                  *         .
       . *  Topology   *        *   *     *                  *         .
       . *  Database   *<*******+   *     *                  v         .
       . +*************+            *     *         +****************+ .
       .                            *     +********>*  RIB Manager   * .
       .                            *               +****************+ .
       .                            *                        ^         .
       .                            v                        *         .
       .                 +*******************+               *         .
       .                 * Subscription &    *               *         .
       .                 * Configuration     *               v         .
       .                 * Templates for     *      +****************+ .
       .                 * Measurements,     *      *  FIB Manager   * .
       .                 * Events, QoS, etc. *      *  & Data Plane  * .
       .                 +*******************+      +****************+ .
       .................................................................

     <-->  interfaces inside the scope of IRS
     +--+  objects inside the scope of IRS

     <**>  interfaces NOT within the scope of IRS
     +**+  objects NOT within the scope of IRS

     ....  boundary of a router participating in the IRS
```

Figure 1: IRS model and Problem Area

A critical aspect of IRS is defining a suitable protocol or protocols to carry messages between the IRS Clients and the IRS Agent, and defining the encapsulation of data within those messages. This should provide a clear transfer syntax that is straightforward for applications to use (e.g., a Web Services design paradigm), and should provide the key features specified in Section 5.

The second critical aspect is semantic-aware data-models for information in the routing system and in a topology database. The data-models should be separable across different features of the managed components, versioned, and combine to provide a network data-model.

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 the data model employed needs to handle indirection 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 IRS focus initially on interfaces to the RIB layer (e.g. RIB, LFIB, 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]). IRS recognizes 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 MIBs at the present time.

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. IRS provides a framework for applications to register for asynchronous notifications and for them to 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.gredler-idr-ls-distribution]) still provides 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 is needed by applications; only the active paths or links are known versus those desired or unknown 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 is there the ability to set up the router to trigger different actions upon an event's occurrence.

5. Desired Aspects of a Protocol for IRS

This section describes required aspects of a protocol that could support IRS. 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 operations to IRS without needing to wait for each to complete before sending the next.

Configuration Not Re-Processed: When an IRS operation is processed, it does not require that any of the configuration be processed. I.e., the desired behavior is orthogonal to the static configuration.

Duplex: Communications can be established by either the router or the application. Similarly, events, acknowledgements, failures, operations, etc. can be sent at any time by both the router and the application. The IRS is not a pure pull-model where only the application queries to pull responses.

High-Throughput: At a minimum, the IRS Agent and associated router should be able to handle hundreds of simple operations per second.

Responsive: It should be possible to complete simple operations within a sub-second time-scale.

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.

Timing of State Installation and Expiration: The ability to have state installed with different lifetimes and different start-times is very valuable. In particular, the ability of an IRS client to request that a pre-sent operation be started based upon a dynamic event would provide a powerful functionality.

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.

6. Existing Management Interfaces

This section discusses the combination of the abstract data models, their representation in a data language, and the transfer protocol commonly used with them as a single entity. While other combinations are possible, the combinations 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. IRS does not involve CLI standardization.

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 IRS.

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 lack of standard data models have hampered the adoption of NetConf.

7. Acknowledgements

The authors would like to thank Ken Gray for their suggestions and review.

8. IANA Considerations

This document includes no request to IANA.

9. Security Considerations

Security is a key aspect of any protocol that allows state installation and extracting of detailed router state. More investigation remains to fully define the security requirements, such as authorization and authentication levels.

10. Informative References

[I-D.gredler-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., and A. Farrel, "North-Bound Distribution of Link-State and TE Information using BGP", draft-gredler-idr-ls-distribution-02 (work in progress), July 2012.

[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. Gaps and Concerns for SNMP

Though SNMP can allow state to be written, the overhead of the required infrastructure is quite high. Clients and servers that wish to use SNMP must build in and understand a large number of MIB modules, including many proprietary modules. Even when ignoring the overhead in building the SNMP processing and handling functions into an application, these properties lend themselves well to read-only operations. A critical lack in MIB modules for read-write (i.e., for configuration) operations is that there is no semantic understanding of the objects defined in the modules encoded in those modules. Any required semantic knowledge must be specifically hand-coded into applications or ignored. Further, many MIB modules do not allow the writing of state, and this limits coverage; owing to the cumbersome nature, there has not been interest in increasing coverage.

An additional deficiency in using SNMP MIB modules either for reading or writing comes in the inherent co-mingling of semantics and syntax in the form of indexing requirements. SNMP MIB modules contain tables that also define an index format. This format is then translated - often mapped onto - a device's actual implementation. Such a mapping means an implementation either matches the module's indexing during searches or not; if not, then an implementation is slowed down when it searches for objects. Even in not-so-extreme cases, such slow performance can result in the SNMP manager's request timing-out owing to the delay of the SNMP agent's response.

For example, if a search of N*M objects is stipulated as (N, M) in

the standard MIB module, but the implementation happens to choose to index its tables internally as (M, N), this will result in search times of O(N^2). When N or M become large, as they do in routing tables, this results in wasted processing cycles for the device, and either extremely long wait times for queries, or simply a lack of answers to queries. It is a clear requirement for the IRS to not suffer from this issue.

In addition to these difficulties, SNMP matches up to the key needed aspects as follows:

Multiple Simultaneous Asynchronous Operations: Supported, but difficult for configuration.

Configuration Not Re-Processed: Supported

Duplex: The manager must initiate communications with the SNMP agent on the router. With the limited exception of Notifications and InformRequests defined in a MIB module, this is a pull model.

High-Throughput: Possible

Responsive: Possible

Multi-Channel: Possible

The key gaps identified for SNMP are:

a. Infrastructure Overhead

b. Lack of Semantic Information in the Data-Model

c. Required Indexing, from mingling of semantics and syntax, badly impacting performance or driving implementation decisions.

d. Limited interest and use for configuration

e. Communication model isn't naturally duplex.

Appendix B. Gaps and Concerns with NetConf

While NetConf has solved many of the deficiencies present in SNMP in terms of configuration, it still does not satisfy a number of requirements needed to manage today's routing information. First, the lack of standard data models have hampered the adoption of NetConf; a significant amount of per-vendor customization is still needed. The transport mechanisms that are currently defined (e.g.,

SOAP/BEEP) for NetConf are not those commonly used by modern applications (e.g., ReST or JSON).

NetConf primarily facilitates configuration rather than reading of state or handling asynchronous events.

NetConf matches up to the key needed aspects as follows:

Multiple Simultaneous Asynchronous Operations: Not Possible

Configuration Not Re-Processed: Not Possible

Duplex: Not Possible - strict pull model.

High-Throughput: Unlikely - Can depend on configuration size

Responsive: Unlikely - Can depend on configuration size

Multi-Channel: Not Possible

Authors' Addresses

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

Email: akatlas@juniper.net

Thomas Nadeau
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: tnadeau@juniper.net

Dave Ward
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: wardd@cisco.com