

Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 15, 2013

A. Atlas
Juniper Networks
S. Hares
Huawei Technologies
J. Halpern
Ericsson
September 11, 2012

A Policy Framework for the Interface to the Routing System
draft-atlas-irs-policy-framework-00

Abstract

A key aspect of the Interface to the Routing System (IRS) is what mechanisms it includes to carry policy information and to enable policy control. This applies both in the protocol itself and in the sub-interfaces associated with the different components of the routing system. Similarly, the data-models associated with the sub-interfaces must be capable of expressing the appropriate granularity for access and authorization-related policy. This document describes the policy framework for IRS.

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 March 15, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. General IRS Policy	4
3.1. Policy between commissioner and agent	6
3.1.1. Identity	6
3.1.2. Security Role	6
3.1.3. Security Model	7
3.1.4. Scope and Influence	7
3.1.5. Resources	8
3.1.6. Connectivity	9
3.1.7. Priority	9
3.1.8. Precedence	10
3.2. Policy between Agent and Local System	12
3.2.1. Local Configuration	13
3.2.2. Removal of IRS-installed State	14
3.2.3. On Reboot	14
4. Policy in a Sub-Interface	15
4.1. Resource Reservation and Three-Phase Commit	15
4.2. Defining IRS Behavior Based on Implicit and Explicit Policy	15
4.2.1. Example of Implicit Policy	16
4.2.2. Passing Explicit Policy	17
4.2.2.1. Explicit policy on Data Forwarding, Resources, and Policy passing	17
4.2.2.2. Example of Explicit Policy	17
5. Acknowledgements	18
6. IANA Considerations	18
7. Security Considerations	18
8. Informative References	18
Authors' Addresses	18

1. Introduction

The Interface to the Routing System (IRS) provides read and write access to the information and state that enable the routing components of routing elements. The IRS is introduced and described in [I-D.atlas-irs-problem-statement] and [I-D.ward-irs-framework].

Policy helps provide filters and control on the level of access to information and state that is enabled by individual protocol interactions. A clear view of the policy features desirable at the IRS is important to shape the architecture and requirements for the protocols and sub-interfaces of the IRS. Policy can be explicitly defined or implicitly assumed in a system, and can be enforced by that system's rules and behavior. Since IRS provides sub-interfaces to routing sub-systems that already have policy defined (implicitly or explicitly), it is important to consider the existing policy mechanisms and how an IRS sub-interface should interact with them.

IRS policy has four different aspects that need to be considered.

1. Policy related to the IRS protocol interactions between different systems.
2. Policy related to the interaction between the IRS Agent and the local system to which the IRS Agent is providing an interface.
3. Sub-interface policy to support scope and influence restrictions and to preserve necessary policy associated with the related routing sub-system.
4. Policy that can be installed or read via a sub-interface's data-model that is associated with the related routing sub-system.

2. Terminology

The following memorable terminology is used in this document.

agent or IRS Agent: An IRS Agent provides the supported IRS sub-interfaces to the local system's routing sub-systems. The IRS Agent understands the IRS protocol and can be contacted by commissioners.

commissioner: A commissioner speaks the IRS protocol to communicate with IRS Agents and uses the IRS sub-interfaces to accomplish a task as instructed by the commissioner's local application. A commissioner can be seen as the part of an application that supports IRS and could be a software library.

scope: The set of information which the particular IRS entity (agent or commissioner) is authorized to read. This access includes the permission to see the existence of data and the ability to retrieve the value of that data. In the context of an interaction between a commissioner and an agent, the effective scope is restricted to the intersection of the scopes of the two entities.

influence: The set of field values which the particular IRS entity (agent or commissioner) is authorized to install. This access can restrict what fields can be modified or created, and what specific value sets and ranges can be installed. In the context of an interaction between a commissioner and an agent, the effective influence is restricted to the intersection of the influences of the two entities.

resources: A resource is an IRS-specific use of memory, storage, or execution that a commissioner may consume due to its IRS operations. The amount of each such resource that a commissioner may consume in the context of a particular agent can be constrained. Examples of such resources could include: the number of installed operations, number of operations that haven't reached their start-time, etc. These are not protocol-specific resources or network-specific resources.

role or security role: A security role specifies the scope, influence, resources, precedence values, etc. that a commissioner or agent has.

identity: A commissioner is associated with exactly one specific identity. State installed by a particular identity is owned by that identity; state ownership can not be transferred. It is possible for multiple communication channels to use the same identity; in that case, the assumption is that the associated commissioner is coordinating such communication. Similarly, an agent is associated with a specific identity.

3. General IRS Policy

IRS needs its own implicit and explicit policy. This section articulates some of the those key concepts and policy decisions. The IRS policy applies to interactions between the agent and commissioners and between the agent and the local system.

The agent's externally perceivable behavior and associated policy is a key aspect of IRS that must be described. The commissioner's behavior and functionality is specifically out-of-scope except where

it needs to be described with respect to the agent's behavior and the IRS protocol.

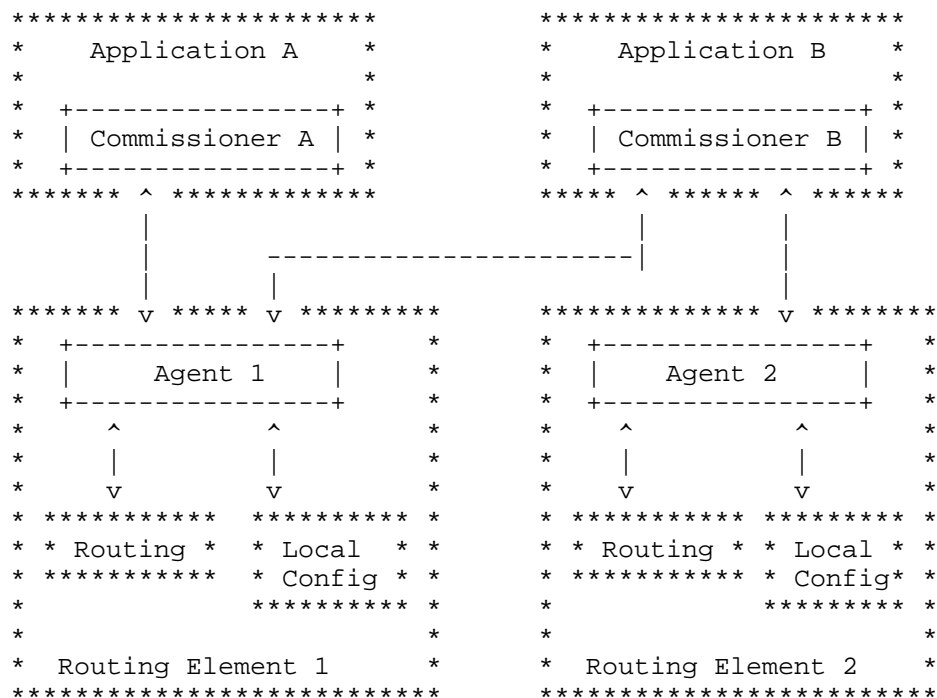


Figure 1: Architecture of commissioners and agents

As can be seen in Figure 1, a commissioner can communicate with multiple agents. The application associated with a commissioner may have multiple tasks it is accomplishing (separate functions, short-term versus longer-term, etc) and each such task may involve a set of agents which may or may not differ.

As can also be seen in Figure 1, an IRS Agent may communicate with multiple commissioners. Each commissioner may send the agent a variety of install operations. The set of install operations received by an agent may overlap and conflict. No simple protocol or policy mechanisms by an agent can completely avoid indirect interactions between different install operations. The functional partitioning between the different commissioners must be done to avoid undesirable indirect interactions.

3.1. Policy between commissioner and agent

Multiple commissioners can communicate with the same agent. The agent must have policies to manage the resulting complexity. Implicit policy includes the assumptions about communication between the commissioner and agent. Explicit policy includes mechanisms to arbitrate between different commissioners, between operations of the same commissioner, and to manage state owned by an commissioner inside the agent.

3.1.1. Identity

By definition, a commissioner is associated with exactly one identity. An agent will store data that is owned by a particular commissioner, based upon that commissioner's identity. Since a commissioner can communicate via multiple transport channels and no channel needs to be active for the agent to have associated state, the commissioner's identity is used to identify the ownership of the data stored by the agent.

Similarly, by definition, an agent is associated with exactly one identity. An commissioner may also store local state associated with a particular agent. The agent's identity can be used to identify ownership of the data stored by the commissioner.

The details of what constitutes an identity can be dependent upon the specifics of the IRS protocol and selected security mechanisms. However, there are some critical considerations for identity that do impose constraints.

An identity is not tied to a single communication channel. A commissioner may use multiple IP addresses; an identity should not be tied to a specific IP address. If the commissioner or agent is associated with a system that may be mobile, that should be considered in its identification. Finally, the syntax and semantics for identifiers used for a commissioner and for an agent may be different.

3.1.2. Security Role

In the context of an agent, each commissioner will have a security role. The commissioner's identity and associated security role will have to be verified via an acceptable security mechanism. A variety of such mechanisms are anticipated to meet different security and operational objectives. Example mechanisms might include a role assertion from the commissioner to the agent that the agent can cryptographically verify or having the agent to use an already trusted protocol to verify the commissioner's security role and

identity.

An agent must know the scope, influence, and resources associated with each particular security role. This information may vary across different agents even in the same network or it may be consistent across different agents in the same network. The latter can be enforced by having a commissioner that is authorized to influence the meta-data model of security roles on the relevant set of agents.

A security role also defines what precedence values (See Section 3.1.8) a commissioner can use.

3.1.3. Security Model

As described above, roles identify the scope, influence, and resources allowed to an IRS Commissioner. The policy model therefore needs to include these roles. The question of the bindings of identities to roles, and the selection of identities are protocol specific matters outside the scope of this document.

The policy model for roles needs to address two dimensions. It needs to create the roles themselves. This should allow for use of techniques like inheritance, presumably with some rules on how role definitions can augment or restrict the inherited definitions.

The security model also needs to define, by reference to the policy model itself, the scope and influence of the role. The question of defining the resources of a role is for further study. The role definition needs to indicate what types and instances of data can be observed and what information about those instances entities with that role can observe. The security model also needs to define which data items can be modified, and what modifications (ranges, specified values, or other assertions that must be met) are permitted.

3.1.4. Scope and Influence

Scope and influence are specified as part of a security role. A security role may be defined and managed in an external repository, centralized within an administration. The security role definitions must be accessible to an agent.

In the context of an interaction between a commissioner and an agent, the effective scope or influence is restricted to the intersection of the scopes or influence of the two entities.

What information a particular commissioner is authorized to read is known as the commissioner's scope. A scope includes the ability to see that particular data exists and to read the same data. The scope

can have its constraints specified in terms of specific portions of data models.

Similarly, what information a commissioner can install may be constrained. This is known as its influence. The influence is specific in both the parts of the data models and in the set and range of data that can be installed. For example, a commissioner might be able to write static routes in the RIB data-model for prefixes in 10.0/16.

While the commissioner's behavior and functionality is specifically out-of-scope, it is useful to describe the same scope and influence concepts for an agent operating in the context of a commissioner.

An agent's scope is the set of data that the agent can read or have access to. An agent would generally learn such data because the commissioner has sent that data to the agent in an operation.

An agent's influence is the set and range of data that the agent is allowed to provide to the commissioner and that will be accepted by the commissioner. For instance, commissioner B may accept next-hop change notifications for prefix 10.0/16 from agent 1 but not from agent 2.

3.1.5. Resources

When a commissioner sends operations to an agent, those operations can consume resources. Therefore, it is important that the agent have policy to limit the resources available to a particular commissioner. This is based on the commissioner's identity and security role. Such resource policy specifications need to be provided in a data-model that can be modified by appropriately authorized commissioners or local configuration.

Examples of such resource constraints include:

- Number of installed operations owned,

- Number of operations that haven't reached their start-time, and

- Number of event notifications registered for.

As discussed in Section 3.1.7, a commissioner can specify priorities for the operations it sends.

If compute resources are considered, it is not the intent to try and determine the computation associated with particular operations. Instead, the constraint could be on amount of compute-time given to a

commissioner every pre-defined period. This can provide a mechanism for fair sharing of compute resources between commissioners.

3.1.6. Connectivity

A commissioner does not need to maintain an active communication channel with an agent. Therefore, an agent may need to open a communication channel to the commissioner to communicate previously requested information. The lack of an active communication channel does not imply that the associated commissioner is non-functional. When communication is required, the agent or commissioner can open a new communication channel.

State held by an agent that is owned by a commissioner should not be removed or cleaned up when a commissioner is no longer communicating - even if the agent cannot successfully open a new communication channel to the commissioner.

3.1.7. Priority

The motivating example for priority is when a single commissioner is sending operations to accomplish multiple tasks. For example, one task might be long-term and another task might deal with unexpected requests that are more important. In this case, the commissioner may wish to provide a hint to the relevant agents as to which operations should be done first.

Communication from a commissioner can come across multiple channels, so simply specifying that operations be done in order is not sufficient. Additionally, all operations may not be immediately carried out, due to varying start-times or other constraints. With these factors and this motivating example, it is useful to introduce the concept of prioritization for operations sent from the same commissioner.

By introducing the concept of priority for operations, a commissioner can accomplish multiple uncorrelated tasks that affect the same agent with the specified prioritization.

A default priority can be specified for each particular communication channel. In addition, an IRS operation can specify a priority to use instead. Priorities between operations from different commissioners need not be compared.

The priority can be used by an agent to determine which operation from a commissioner to execute next.

3.1.8. Precedence

A mechanism is needed for the agent to determine what state to install when there are overlapping install operations. An install operation may overlap with locally-installed configuration state or with a previous install operation that was requested by a commissioner. The mechanism to resolve this is termed "precedence". No simple mechanism can fully handle indirect interactions; considering such interactions is out-of-scope. Indirect interactions must be considered when different commissioners are given their tasks.

A critical aspect of precedence-based decisions is that preference is only given based on arrival time of the install operation when multiple commissioners use the same precedence value.

Each install operation has a precedence associated with it. This precedence may come from the default associated with the commissioner, with the specific communication channel, or with the specific operation. The range of possible precedence values that can be used is known based on the commissioner's security role. The determination of the precedence associated with any operation is a policy decision at the agent, but may utilize any or all of the information described above.

When an install operation is executed, the agent first determines if there is overlapping existing IRS-installed state. If not, the agent must determine if it overlaps existing local-configuration state. Local-configuration state will also have a precedence associated with it so that the agent can make an appropriate decision.

A commissioner can specify whether an install operation should be store-if-not-best. This allows a commissioner to determine what happens when an install operation doesn't win the precedence comparison. If store-if-not-best is specified, then the install operation succeeds and the associated installed state is stored but not actively installed by the agent. If store-if-not-best is not specified, then the install operation will fail.

The store-if-not-best flag is stored with the installed operation's precedence. If the agent determines that an installed operation must be preempted, then the agent consults the store-if-not-best flag. If store-if-not-best is specified, then the agent stores the preempted operation and does not notify the associated commissioner. If store-if-not-best is not specified, then the agent notifies the associated commissioner of the preemption and removes the previously installed state.

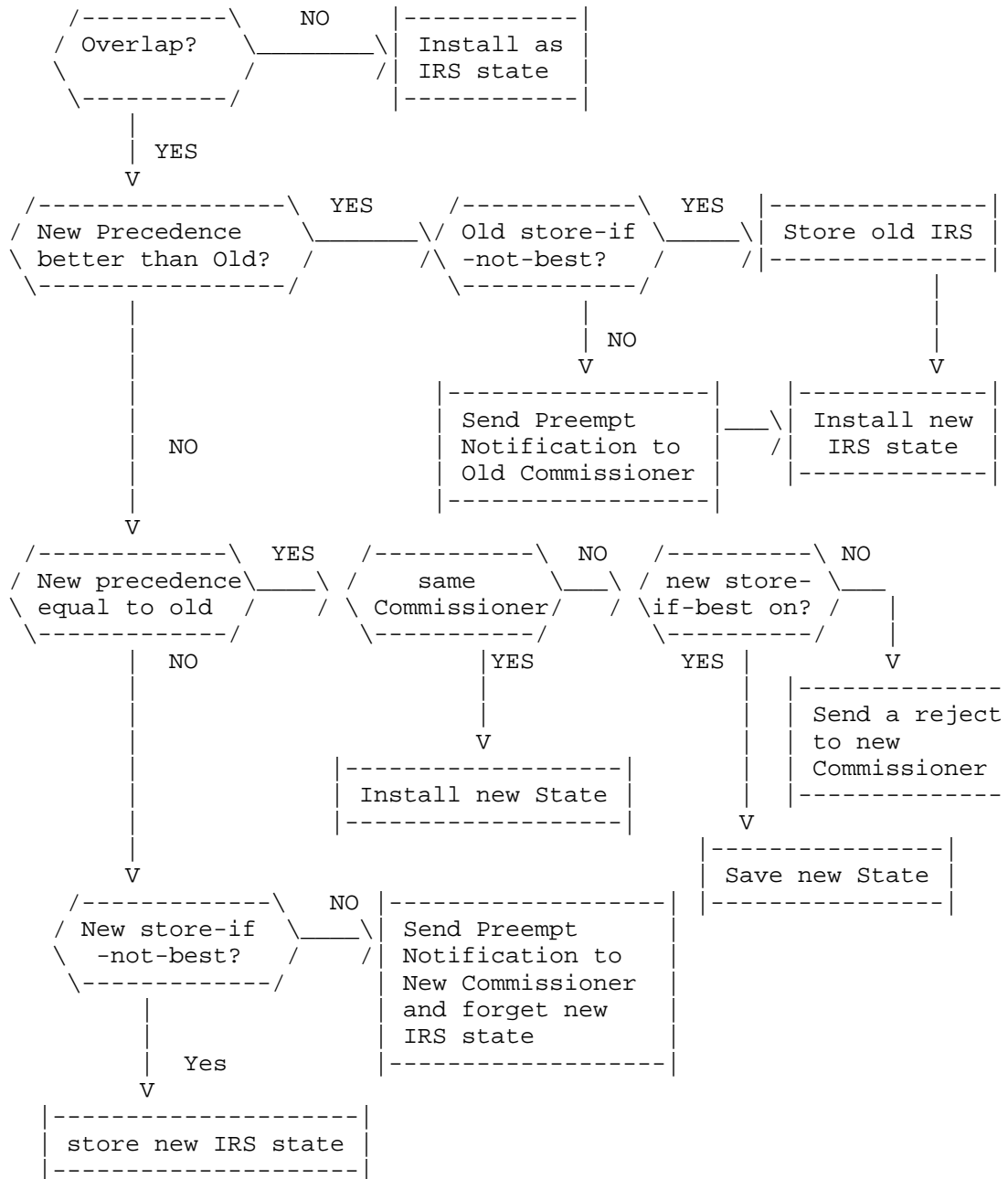


Figure 2: Precedence Decision-Making

If the overlapping new operation has a precedence that is better than

the existing state, then the agent should preempt the existing state and act according to the existing state's store-if-not-best flag. If that store-if-not-best flag is set, the agent will store the old state and install the new state. If the store-if-not-best flag is clear, the agent will send a preemption notification to the Old IRS Commissioner, install the new IRS state, and forget the old.

If the overlapping existing state has the same precedence and the same commissioner associated, then the agent completes the install operation; otherwise, the agent must reject or store the install operation, based on the store-if-not-best flag.

If the new overlapping operation has a precedence that is worse than the existing state, then the agent must reject or store the install operation, based on the state of the new store-if-not-best flag. If the store-if-not-best flag is set, then then the agent will store the new IRS state. If the store-if-not-best flag is clear, then the the IRS agent will send a preempt notification to the new commissioner and forget the new IRS state.

This decision process is illustrated in Figure 2.

When an uninstall operation (e.g. remove) is done, the stored state with the next best precedence should be selected and installed.

A consequence of the precedence policy mechanism is that a commissioner must be able to handle its installed operations being preempted at any time, either explicitly or simply by having the active state changed. Such preemption can be minimized by appropriate separation of tasks, with their associated install operations, between the local systems of the commissioners and by knowledgeable local system configuration.

3.2. Policy between Agent and Local System

It is critical to understand and clearly specify how IRS interacts with local configuration. The key questions are:

1. What happens when Local Configuration overlaps with IRS installed state?
2. What happens when IRS installed state is removed?
3. How is state recreated when a local system reboots?

A consequence of using IRS is that the local system's state may not be synchronized with the local configuration. Since this is a change in understood behavior, any discrepancies should be clearly visible

to the operator with an associated explanation.

Logically, the local configuration is essentially modeled as a local commissioner, with its own precedence, identity, and security role and immediate permanent install operations. The key differences are both that all relevant local configuration state need not be cached by the agent and that reboot imposes the need to process local configuration state before any other IRS-installed state.

3.2.1. Local Configuration

The local system's local configuration may have overlapping influence with that of one or more commissioners using an agent. Therefore, explicit and implicit policy interactions must be specified. The mechanism that IRS provides for deciding between overlapping install operations is "precedence". This same mechanism can be used to decide between local configuration and an IRS operation. Local configuration can specify the precedence value to be used for the local system.

A precedence value that causes the desired behavior can be specified as follows. (MAX is the highest precedence given to a commissioner. MIN is the lowest precedence given to a commissioner.)

MAX+1 Precedence: If the local configuration has a precedence higher than that given to any commissioner, then state from the local configuration will always be installed. If any IRS-installed state is therefore preempted, the agent will notify the associated commissioner.

MIN-1 Precedence: If the local configuration has a precedence lower than that given to any commissioner, then IRS-installed state will always override local configuration. That this preemption has occurred should be reflected in how the local system displays its state.

Other Precedence: The local configuration can have higher precedence than that given to some commissioners, lower precedence than that given to other commissioners, and equal precedence to that given to other commissioners. Then some local configuration state may be preempted by IRS-installed state while some IRS-installed state can be preempted by local configuration.

Local-configuration wins all precedence ties.

Just as an agent must check to determine if an install operation overlaps with existing installed state, the process of committing local configuration must check to see if there is overlapping IRS-

installed state.

What the process of committing local configuration is can vary by local system. Well known examples are when a return is sent to the CLI and when an explicit commit command is specified. How the proper checks for interaction between the agent and local configuration are done is a local system matter.

Similarly, when an agent checks to see if an install operation overlaps with existing installed state, the agent must determine if it overlaps with existing local configuration.

If the precedence associated with local configuration is changed, then it is retroactive. All local configuration state stored by the agent must be updated with the new precedence value and installation decisions made for overlapping data. This change could be very disruptive.

3.2.2. Removal of IRS-installed State

When a piece of local configuration is removed, the local system goes back to the appropriate system default. However, when an operation removes some IRS-installed state is removed, the correct behavior is not to just go back to the system default. Instead, any stored state must be considered - whether that comes from local configuration or stored IRS install operations that didn't have the highest precedence. If there is any stored state, then the highest precedence of the options is selected and installed. That existing overlapping state might come from the local -configuration.

If IRS's implicit policy were to just go to the system default, then the local configuration and the local system state would not be synchronized and there would be no remaining IRS-state to explain the discrepancy. Since IRS state can also be stored and not installed, the same mechanism can be used for stored IRS install operations and for local configuration.

3.2.3. On Reboot

When the local system reboots, only persistent IRS-installed state is preserved by the agent. The implicit policy for IRS is that the local configuration is read and installed first. After the local system has its local configuration installed, the persistent IRS install operations are executed to bring the system to the persistent state.

4. Policy in a Sub-Interface

It is critical to consider how policy influences a sub-interface when defining the sub-interface and its associated data-model(s). There are several different aspects to consider.

How are scope and influence policy specified in the data model? What granularity levels are necessary for the particular sub-interface?

How does the implicit policy in the associated routing sub-system effect what IRS can be allowed to influence?

Are the implicit policies of the associated routing sub-system captured in the semantic content of the information model, data model, and description?

What explicit policy communicated in the associated routing sub-system needs to be included in the data-model? What indirection and abstractions are needed?

4.1. Resource Reservation and Three-Phase Commit

Some agents and sub-interfaces may offer the ability to reserve resources required by operations before the operation start time. There are two aspects to how to support this.

First, if the agent can do time-aware resource reservation, then an install operation can specify "reserve-only" to prompt an acknowledgement or failure as to the ability of the agent to confirm the reservation. Then the commissioner can either send an operation to commit the reservation, which causes the associated install operation, or to remove the reservation. A "reserve-only" operation will have its reservation expire at the end of its associated life-time.

Second, part of a sub-interface's data-model may be to request a reservation with a known start-time and duration. An example might be reserving a specific bandwidth on a path for an LSP between two devices. It is important to consider whether a particular sub-interface should offer a time-based reservation service as part of its data-model.

4.2. Defining IRS Behavior Based on Implicit and Explicit Policy

The semantics in a data-model must respect and describe the implicit policy of the associated routing sub-system. This doesn't imply that the data-model components should instantiate it or allow reading or

writing.

Policy Routing systems must deal with the verification, reading and installing of routes from sources such as EGP, IGP, and static routes. Policy routing may also control forwarding and the monitoring of data forwarding; and data resources. The explicit policy examples are given for the routing framework. It is assumed the reader can extend this framework to the data forwarding and data resource arena.

4.2.1. Example of Implicit Policy

The ISIS protocol specification uses implicit policy to set constraints on level 1 peers. Due to this fact, many ISIS implementations only let one level 1 ISIS peer associate with one Level 2 peer domain.

This policy is not encoded in any local configuration directly, but is rather included as an implicit policy. When local configuration policy is checked (prior to a configuration commit), this local policy is checked. If the configuration input from a CLI is in error, the input will be rejected, and the CLI will warn the user. Similarly programmatic interfaces for the local configuration cause the implicit policy to be checked.

IRS data models guide the commissioner in an interoperable interaction with the reading and installation of data at a particular agent. The IRS data models must contain both the implicit policy and the explicit policy. Although an agent may not report the IRS implicit policy in the protocol, the commissioner must know of the existence of the implicit policy.

This knowledge allows the commissioner to know the implicit policy interactions on different systems in a heterogeneous network. For example, assume a situation where a commissioner is talking to two agents - one on system A and one on system B. The routing process on system A has has different implicit rules for the ISIS Level 1 peer to Level 2 peer connection than the routing process on system B. Routing process A is built to allow one level 1 ISIS peer associated with 2 ISIS Level 2 peers. Routing process B upholds the standard implicit policy that 1 level ISIS peer can only be associated with 1 ISIS Level 2 peer. The commissioner setting up the ISIS peering in a network containing system A and system B must know that System A will allow a level 1 peer to connect to 2 ISIS Level 2 peers. When the commissioner's scope allows it to read data from system A and system B, it should not flag the difference in ISIS level 1 peer connections as a problem. Instead the commissioner will need to determine if the use of the different configurations can cause a network problem.

4.2.2. Passing Explicit Policy

Routing systems' explicit policy controls protocols, associates/deassociates interfaces, route verification policy, route forwarding policy, route aggregation policy, and route deaggregation policy. All of this policy can be found in the detailed configuration specification of a routing process. However, even via CLI, it is rarely possible to configure all the possible options. Other configuration mechanisms do not have public models for all the private router configuration. The developers of a routing system often have a complete policy model either in formal modeling languages or informal language.

Explicit policy contained in an IRS data model is the detailed configuration model at the deepest level that an agent can access. This detailed configuration model may come from IETF Standards and/or the vendor specific configurations. The public data models must specify a vendor specific tree where the individual configuration is plugged into.

4.2.2.1. Explicit policy on Data Forwarding, Resources, and Policy passing

Forwarding policy has to do with the data flow may also be controlled by an agent. If so, the explicit policy must be placed in a data model along with the implicit policy.

Lastly, protocols have begun to pass explicit policy about passing policy. Examples of this type of policy are BGP ORFs, BGP Flowspecs, and ISIS policy passing. Commissioners must know the implicit policy and explicit policy this policy impacts, and the precedence between these policy. Due to the extensive use of BGP ORFs and the growing use in BGP Flowspecs policy, early data models for BGP should describe the implicit policy, explicit policy, policy precedence for the BGP ORFs and BGP FlowSpecs, and how they interacts with other BGP, route policy and preferences. This information should be placed inside an IRS Data Model for an Agent supporting these features.

These explicit models for BGP policy are not trivial, but these models exist today. Frequently, IRS data models may be simply a casting of existing implicit policy and explicit policy into a common standard form so that programmatic interfaces may interact with a routing element.

4.2.2.2. Example of Explicit Policy

There are two clear explicit policy pieces for ISIS. First is the peer level. Second is the policy of the external routes to be

redistributed into and out of ISIS.

5. Acknowledgements

The authors would like to thank Ross Callon, Adrian Farrel, David Meyer, David Ward, Rex Fernando, Russ White, Bruno Risjman, and Thomas Nadeau for their suggestions and review.

6. IANA Considerations

This document includes no request to IANA.

7. Security Considerations

This is empty boilerplate for now.

8. Informative References

[I-D.atlas-irs-problem-statement]
Atlas, A., Nadeau, T., and D. Ward, "Interface to the
Routing System Problem Statement",
draft-atlas-irs-problem-statement-00 (work in progress),
July 2012.

[I-D.ward-irs-framework]
Atlas, A., Nadeau, T., and D. Ward, "Interface to the
Routing System Framework", draft-ward-irs-framework-00
(work in progress), July 2012.

Authors' Addresses

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

Email: akatlas@juniper.net

Susan Hares
Huawei Technologies

Email: shares@ndzh.com

Joel Halpern
Ericsson

Email: Joel.Halpern@ericsson.com

