IETF 89 I2RS Working group Meeting

When: Wednesday, March 5th, 2014
Location: Blenheim

Chairs:
Alia Atlas (co-chair) (stepping down to become AD)
Edward Crabbe (co-chair)
Jeff Haas (co-chair)

Agenda

Administrivia and Agenda Bashing [(E. Crabbe, 5 minutes)
draft-ietf-i2rs-problem-statement-00 (A. Atlas, 5 minutes
draft-ietf-i2rs-architecture-02 [J. Halpern, 15 minutes)
Yang: Model Gap Analysis (A. Bierman, 15 Minutes)
ForCES: Model Gap Analysis (J. Hadi Salim, 15 minutes)
I2RS Related/Relevant Yang Models Currently in Use (R. Varga, 10 minutes)
Discussion (Group, 10 minutes)
RestConf and Netconf: Protocol Analysis (D. Bogdanovic, 10 minutes)
ForCES: Protocol Analysis (J. Hadi Salim, 10 minutes)
Discussion (Group, 10 minutes)

Information Models: draft-ietf-i2rs-rib-info-model-02 (S. Kini, 5 minutes)
Security: I2RS Security (S. Hares, 10 minutes)

draft-keyupate-i2rs-bgp-usecases-01 (S. Hares, 5 minutes)
draft-bitar-i2rs-service-chaining-01 (N. Bitar, 5 minutes)
draft-white-i2rs-use-case-02 (R. White, 10 minutes)
Various use cases + Juggling Demonstration (S. Hares, 10 minutes)

Notes

1: Administrivia and Agenda Bashing [E. Crabbe, co-chair]

[slides-89-i2rs-13.pdf] [Recording: 4:00 – 6:00]

Summary: Ed Crabbe (co-chair) indicated that people tend to hyper focus on the IETF meetings rather than the daily work on the mail list. The I2RS WG group is late for all deliverables, but there are WG drafts for most items. The I2RS WG will be focusing on completing these deliverables in the next few months.

The agenda is grouped in the following sections:

- existing architecture and problem statement,
- modeling language (yang/ForCES) and protocol (ForCES, NetConf, Yang),
- informational models (RIB), and
- Use cases.

2. Problem statement (draft-ietf-i2rs-problem-statement-00) [Alia Atlas]

[no slides] [6:00 – 7:02, 1 minute]

Alia stated the authors and the chairs believe the I2RS problem statement is complete. Alia requested that anyone that believes the problem statement is missing something should make comments on the list. Ed Crabbe stated that the problem statement is a foundational document which is moving toward last call. Ed also called for all WG members to review the document and post any comments regarding this document to the list.

3. draft-ietf-i2rs-architecture-02 [J. Halpern)

[slides: slides-89-i2rs-7.pdf] [recording: 7:02 – 24:30]

The authors did a paragraph by paragraph review of the documents. The authors found lots of places that were unclear or inconsistent. In the authors edited, they tried to improve the consistency of the architecture document. Ron Bonica participated in this review trying to clarify the operational versus configuration data so that the architecture document is understandable to both the routing folks and the operations people. Between the two groups (operations and routing), there are some discrepancies in the use of terms.

The following issues were addressed in the document: multi-control, security, state storage, and templates. Joel and the co-authors specifically asked for comments on these issues. Multiple control model used in the I2RS architecture is a simple model of collision management where if two things try to write the same object, it is an error. The result of this multi-controller collision may be deterministic (such as the 1st I2RS client always winning), but the multi-controller collision is an error that must be resolved by means outside the I2RS protocol.

Security in the architecture document has been expanded. There is a rule of thumb in designing any routing system ï ñif you do not think about security early, you will not be able to build a secure system.ò There is now a security design team that is trying to review the architecture documentôs security text and create a security document for the people to talk about. The base information is the same, but the assumptions of the environment and external identity servers have been added to the architecture text. The architecture document contains definitions for I2RS client and I2RS identity, roles and scopes (read and write) is contained in the document. Operators have indicated that two things need to be added: 1) mutual authentication of the client and the agent, and 2) confidentiality for the data. The mutual authentication is required because the I2RS client may not be in the same domain as the I2RS agent. When the I2RS client works across the boundary, the I2RS client and I2RS Agent must mutually authenticate each other. In the same manner, if since the I2RS Client and I2RS agent may be in different domains, the I2rs Agent and Client must have Confidentiality. Also if you assume that this I2RS protocol is deployed in the pervasive monitoring world, then you need confidentiality. We are

State Storage in the I2RS architecture document has also been expanded to describe assumptions around storage. The I2RS architecture has a very small requirement for persistent storage. A second state-related change to the architecture is that after an unexpected failure, the agent upon rebooting notifies the client that it has been rebooted. If the I2RS agent loses state after a reboot

and does not know about an I2RS client after a reboot, it cannot inform the I2RS client. Since I2RS WG has stated it does not want to require that I2RS agents and I2RS clients meet after an unexpected failure, this is all the I2RS Agent can be required to do. If the WG via the mail list wishes something different, they should comment on the list. This restriction implies that the start-time/stop-time discussions in earlier document no longer apply to the document. This version of the architecture only allows an I2RS client can only request an I2RS agent to do an operation immediately.

The author team added a section on the basic modeling architecture to the I2RS architecture document. This modeling architecture was set-up to allow a generic modeling description to support the information models proposed. The generic model introduces the concepts of inheritance, or the ability to refer to objects that must be contained in the informational models, modeling language, and the modeling protocol. This generic modeling description does not imply any particular modeling language or a particular information model. This is new text which must be reviewed carefully by the Working Group for substance and wording.

The author team also added a section on templates. Templates are a powerful tool used in networking, but the question for I2RS is whether templates should be used. If the I2RS client uses templates and fills in the values, then the I2RS protocol does not need to know about templates. If the I2RS agent uses templates to reduce the amount of information passed from the I2RS client to the I2RS Agent, then the I2RS protocol does need to know about the templates. The current text describes the second case ï an I2RS Agent with templates that the I2RS client sends information to fill in the template. The templates are simply short-forms of information. The author team wishes to have feedback whether I2RS should have: a) I2RS client-only templates, or b) I2RS Agent templates, or something else. The author team called on the WG to evaluate the text and give feedback. This issue is one of the main issues

Discussion of Architecture document:
Dan Bogdanovic (Juniper): I believe the templates should be strictly agent base because you do not know what dependencies you inside your configuration in the I2RS client. In order to push down a completed template from the client, you must be able to resolve all the dependencies from the client. If you put the templates on the agent side, then you can know that the local dependencies are resolved at the agent, and you know what functions the I2RS agent can do in a single transaction.

Joel Halpern: Iôd like to see this in writing. I suspect I disagree.

Joel Halpern: In summary, the author team hopes to get feedback on four items: multi-control, security, state storage, and templates and put together a final revision. We hope to get WG last call (LC) before the July IETF in Toronto.

Dan Frost: One quick clarifying question on the multiple control is the requests to the agent be serialized?

Joel Halpern: The thinking is that there are multiple client sending out operations to the agent. What the agent does internally is outside the scope of I2RS? The whole point of the priority

mechanism so that you had a mechanism to order processes so that you did not have depend on any serialization (First-come-First-serve).

Dan Frost: What is the state of the security discussion? Has there any been any discussion around what operator requirements for the security might be? For example, are we considering password bass mechanisms or exchange of keys? Are we also considering a particular solution? One example of a solution could be SRP that has a simple password exchange rather than a complex key exchange. What do operators desire?

Joel Halpern: We have not gotten that far. I can comment on where we are. We have decided to exchange identity information for authentication, and cryptographic techniques to get to confidentiality and authentication of the message that are exchanged. This will have to be cryptographic based, but there is a tendency for me to assume it is mutual TLS certificates. However, once I say this people object that I am assuming things I should not.

Dan Frost: SRP is a very different model than TLS. You have the authentication cryptographically and then you end up with symmetric key exchange with a key that can provide full encryption of the data.

Joel Halpern: The question of which bootstrap protocol has not been decided or discussed in detail.

Alia Atlas: I think with the security consideration in the architecture there is an assumption that there is an authentication and authorization services. One of the hopes (goals) for the security draft is to resolve down to a few options that are acceptable. It is not certain what is mandatory to implement.

Joel Halpern: We will have to have mandatory to implement. We have not gotten far enough in the discussions to have requirements for the communication. Clearly the SRP meets the requirements to have symmetric keys to the communicationôs authentication and encryption. At this point, we need to have security experts.

Ed Crabbe: We will have a more structured security discussion later in the meeting.

3) Yang: Model Gap Analysis (A. Bierman, 15 Minutes)

[slides-89-i2rs-10.pdf] [24:36- 40:00]

Presentation:

Data Models are a contract that binds the information model to the protocol. The Data models allow the agreement between the server and the client on what the server is providing to the client. There is a capability exchange and in some cases capability negotiation that let the client know exactly what the server supports. This is important because requiring the client to guess what the server supports leads to many problems.

Pros for Yang:

Yang is widely implemented for NETCONF (http://trac.tools.ietf.org/wg/netconf/trac/wiki). The goals for Yang to prioritize for readers, writers, and lastly for the people making tool makers. The readers and writers continually use the Yang model so these people must understand it quickly. The Tool makers will determine how to use it once, and build the tools on this understanding. Initially, Yang looked at many existing forms. It was the combination of four different data modeling language to arrive at todayôs yang. The reason yang is popular is its ability to be easily read and understood. The user does not have to put a lot of glue into the model. Unlike SNMP MIBs which require a great deal of glue to be put into the model.

The most important thing was that Yang was able to model the constructs that people put in their routing products. Yang supports object oriented and many different types of modeling configurations. For example, the choice is a new addition from previous data modeling languages. Yang allows for flexible extension by vendors or SDO, and reusable user-definitions. Yangôs extensibility occurs by having ñexternal statementsò which are allowed by all YANG compilers, but not pre-defined. This allows Yang module definitions to be expanded without requiring a new version of the Yang modeling language. The Yang experts believe that all of I2RS model requirements could be done with extensions to Yang without requiring a new version of the yang language.

Yang reusability is important since Yang is used for the local configuration. I2RS supports both configuration and operational state. I2RS he operational state overrides local configuration for a period of time. After I2RS completes, the operational state is restored. Andy Bierman suggested if you do not have a good correlation between the I2RS state and the configuration, it may be difficult to manage in his experience. One of the biggest problems for operations staff members have with tools that use multiple modeling languages is the use of multiple names for the same functions. Operators get confused between different variants of names. Since Yang is used for the local configuration, it provides a single modeling definition for both local configuration and I2rs. Yang also has a concept of group that associates a set of definitions (typedefs). Yang group does not provide inheritance, but does provide functions across all data modeling. Sample data models are:

- http://tools.ietf.org/id/draft-clemm-netmod-yang-network-topo-00.txt
- http://www.ietf.org/id/draft-huang-netmod-acl-03.txt

See the models in the netmod page. The OpenDay Light Yang models are also relevant.

Cons for Yang:

Yang (RFC6020) has some NETCONF specific features. A revision to the Yang data modeling is in progress so it is a good time to make changes. These adaptations can make it useful for I2RS and RESTCONF. Examples of these netconf specific features are: XML attributes for insertion operations, and XPath mapping definitions.

YANG is not object oriented, and it does not support derived complex data types. Andy wishes YANG had derived complex data types to support a type of template.

Gaps for Yang:

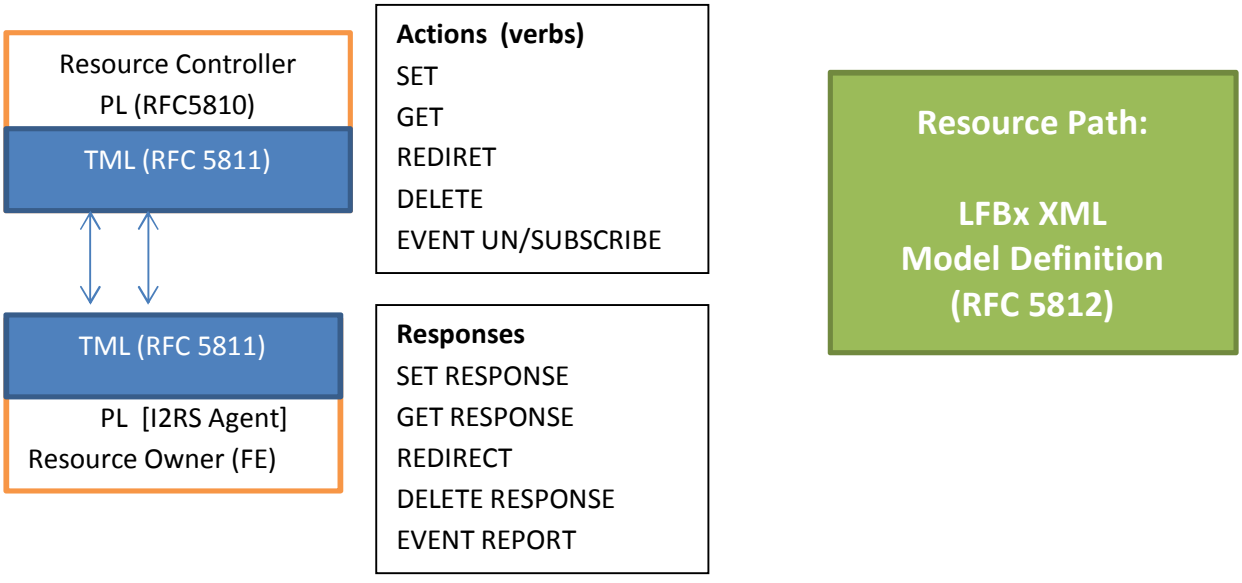
Yang does not support editable operational state. Yang document read by a human reader outside the context of a protocol can be easily understood. Yang can provide this editable extension easily with the code:

```
list some-state {
        config false;
        I2rs: editable-state;
        é
}
```

The Yang tag would signal between the server and the client an editable function. It is important to differentiate between operational state that cannot be edited (For example, temperature gauges) from state that can be edited.

Yang does not support a tag-based notification for variables. This feature can be added as an extension in the same manner as extensible state.

ForCES: Model Gap Analysis (J. Hadi Salim, 15 minutes)
[slides-89-i2rs-8.pdf] [40:26 ï 55:13]



The PATH within ForCES is a Data model. ForCES is not an RPC model, and has the syntax

[<Verb> <noun> [args]]

ForCES has a protocol (verb) and a data model (nouns describing resources). The data model is comprised of logical functional blocks. Forces is closer to REST than RPC.

LFB Classes:
Forces modeling is object oriented. The LFB class definitions have:

- Datatype definition ï to describe the resources (like header types)
- Components ï describing resources (using Datatype definitions)
- Resource capabilities ï (use Datatype definitions)
- Events monitoring and reporting on components or resources

Multiple I2rs clients can be modeled as listening to events. Multiple instances of a class can be instantiated (RIB instance 2) ï with own state/configuration, capabilities and events. ForCES does distinguish between state and config.

Datatype definitions are formal constraints for validation of defined attributes. This includes: atomic types of variables, complex/compound types, grouping of types into structures or tables, hierarchical tree semantics, and aliasing (via symlink shared infrastructure). ForCES Data Model can express whether data values are optional or required, and whether access is read or write. Each component data values can have default values assigned.

LFB Class definitions include components, capability (advertised by resource owner), events (triggers or reports). **LFB Class can be extended** by: a) inheritance and extension of parent class, b) inheritance and extensions of data, c) backward and forward compatibility of LFB classes and defined data structures. FoRCES Model supports the versioning of LFBs. Ex

Sample component definitions are:

component _id 1:
INSTANCE_NAME type string[N}, read-write
component_id 2:
ROUTER_ID type unit32, *read-write*
component_id_3:
optional interface-list array of type ifindex, read-write
component_id_4:
rib-list array of type rib*, read-write*

Example Capability

capability_id_27
NH_CHAIN_DEPTH type unit16

Example Capability

Event_id_1: monitor Routes Table,

- advertise route that changed

Event_id_2

- advertise route added

Event_id_3:

- advertise route deleted

Event_id_4: monitor NextHop Resolution

-advertise next-hop + changed status

GAPS with Forces:

- overhead in table dumps or bulk sets when tables have holes
  - requires use of ILV per table row (64 bit overhead)
  - RIB model is large model
- New Data Models may need to be crated for RIB Info
- Union types may require some rethinking (see discussion at:
  - http://www.ietf.org/mail-archive/web/forces/current/msg04668.html

Summary:

- Pro: Object-oriented, extensible, simple interfaces, and
- Con: Small changes need to meet I2RS specification

Questions: [55:13 -1:16]

1) **Dan Frost:** Can you comment on the security on ForCES and the data model granularity?
2) **Jamal:** The security issues are related to the protocol, and not the data model. I will describe later how some pieces can be added. ForCES mandated IPSEC between the controller and the forwarding engine. IPSEC allows authentication, confidentiality and privacy.
3) **Linda Dunbar:** ForCES seems to be different than I2RS. I2RS manages routers which already have downloads that contain all the knowledge. In a sense, these routers have working brain. ForCES is loading raw images into hardware boxes. To continue the analogy, ForCES works with devices that do not have a brain. Can you carve out a small piece to manage I2RS?
4) **Ed:** The short answer is ñyesò. You can carve out a small piece to manage.
5) **Jamal:** ForCES has two parts: the Data Model and the Protocol. Perhaps we have some gaps in the protocol, but the data model is sound. We started with the assumption that we needed to control. We have a resource owner (like an I2RS agent) and someone who is managing the resource (like an I2rs client). If you look at it this way, you can map it to I2RS.
6) **Benoit Claise:** Andyôs report has many implementations. Where is ForCES implemented?

Jamal: ForCES has been around for a few years, and we have several deployments. Not by some big companies, but we have deployments in other companies. The initial intension of ForCES was counter to big companies.
Ed Crabbe (co-chair) Are there reasonable open source implementations? It is an important point.
Jamal: There are implementations at large deployments. Is it a requirement to have open source because GMPLS does not have open source implementations?
Ed Crabbe: I will take that back. Is it mandatory? No. Is it important? Yes.
Benoit Claise (Cisco) In the URL from Andy, there are many open source implementations?
Jamal Salim: RFC 6358 has open source implementations from many people.
Ed Crabbe: (co-chair) This is not a productive conversation. Letôs go to the next question. Please take this conversation to the list.

7) **Ron Bonica (Juniper):** Iôd like each of you to comment on a trade-off we have before us. One day theyôll be systems with netconf and I2rs. Local configuration will be modelled in netconf as it is today. There will be lots and lots of people who will need to understand that modeling language in operations group in the future. We have an economic trade-off between having those people understand two modeling languages (netconf and ForCES) or modifying the local modeling language. Could each of you comment on the economics of the trade-off?
Andy Bierman: It is a very important question. I am concern how the operational staff figure how what happens when the I2RS client does the wrong thing. The operational state will need to be compared to the configured state. (This contention in operational state reminds me of the /etc/conf* files that do not scale very well as a model.) Contention between I2RS and the local configuration will need to be debugged by operators. Similar models will be helpful to operators who are being debugged.
Jamal Salim: What are the real requirements for the protocol? Is it control or operational state? f the requirements for I2RS are config, this is once per day. If it state based at once-per second.
Ed Crabbe: I disagree with that definition of empheral state.
Jamal: The configuration differences is between latency and throughput.
Ed Crabbe: I believe the difference between config and I2RS is persistence versus the frequency of update. Ronôs point is well taken. At some point, we will have to interact with the configuration Database. If someone puts a static route in, you will have to interact with configuration.
Ron Bonica: It seems cost effective for a few dozen of us to make the changes to netconf to fit the bill, than to have a few thousand operators learn 2 modeling languages.
Jamal: I think the solution is to be based first on the requirements for I2RS protocol.
Andy Bierman: I think that the data types will be shared by NETCONF. In fact in some cases, the leaf will be copied into operational state. This shows up when you have static ARP entries. The configuration representation tends to be close if not exact to the operational state.

Tom Petch: If I was writing an informational model, I would use **Yang** every time. If I was writing a data model, I would go for FORCES every time. The issue that has just risen regarding configuration versus operational state is founded on configuration. The configuration is put in the router to get it running, and the operational state occurs afterward. Yang has done a lot of work in the last year of coming up with Data Models for existing models within the routing technology (such as interfaces table within the RIB). Yang has come with lots of odd results. In a sense you come with 2 RIBS ï one static routes and one dynamics routes goes in a separate. This comes from the basic assumption that Yang and NETCONF is about modeling configuration and not running state. This means that all the other information (from dynamic protocols (BGP, ISIS, OSPF)) needs to go someplace else. For me, this is a problem when you run into data models. If I2RS goes into a Data Model, Yang is a problem.
Jamal: I need clarity here. I have a data model and whether it is configuration or dynamic state ï it does not matter to me.
Tom Petch: This is true for Forces, but in Yang it is fundamental to know which type of state it is. The premise of NETCONF is that it writing configuration to get the box working in the first place. Everything that is not configuration is outside the range of NETCONF, and has it impact as Andy knows full well.

Andy Bierman: NETCONF is configuration is entirely at the layer 9 (political layer) so that the charter would go through. NETCONF fully supports monitoring, notification, and actions. Yang has the ability to model all of NETCONFôs protocol operations, and NETCONF is used quite extensively for monitoring.
Ed Crabbe: To the discussion of configuration versus empheral data, there is a discussion in RESCONF (?) about whether you have multiple named data stores with some identifier. Ulitmately it is not just configuration. Both protocols (ForCES and Yang) are capable of modeling configuration plus empheral datastore.

Jamal: I do not see the split between configuration and empheral data.
Tom Petch: With ForCES there is no split, but with YANG there is a fundamental split that causes problem.
Ed Crabbe: This issues is not with the data model, but the protocol.
Tom Petch: If you mean Yang for informational model, I agree. If you imply Yang for Data model, I disagree.

8) **Dan Bodonavich (**Juniper) : How much is your data modeling language without the ForCES protocol?
Jamal: The ForCES protocol is tied to the model. ForCES has a concept of a PATH. A PATH points to something that is defined in the model. The basic verbs are the same in HTTP: SET, GET, Event, etc.

Dan Bodonavich (Juniper): I want to do a post/patch with ForCES. How do I do it?
Jamal: The patch is like a replace? This is built into FoRCES.

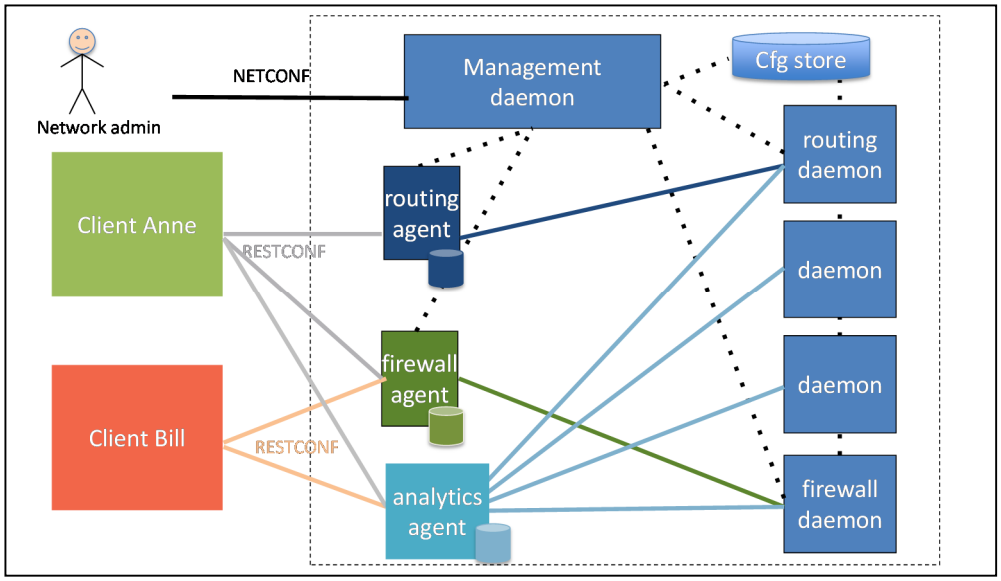
9) Nagica (?) (Cisco): I have comments that I would like to add presentation on ForCES. I may ask stupid questions. How is ForCES different than OPenFlow? I2RS is about controlling the routing plane.
Jamal: You asked the same question as Linda Dunbar earlier. We started controlling forwarding engines. You consider ForCES controlling something that process packets. Instead consider it as a protocol that handles a resource which could be anything. It could be forwarding planes, or it could VMs. It is simply about controlling a resource.
Nagica: If I look at open-flow, ForCES, and I2RS. Each of these technologies are imply interfaces to communication process that speak on the other end to a routing system. The routing system is about forwarding data or other structuresé
Jamal: I do not care if it is processing packets, or intending to process things like packets.
Ed Crabbe (co-chair): What was the question?
Alia Atlas: We are talking about whether the syntax and modeling language of ForCES is useful. It is not the content. Forces and Yang were developed for a particular application, but the base of each piece of the technology that is being examined now is how the modeling language works for a data model. We are looking
Nagia: It is discussion is about the formal modeling language for data models. These models must be read by humans and machine-based tools. Is this correct?
Jamal: I care about human who are programmers, and machines.
Alia: We do not have specific requirements on the human entities, but we do care that the uses of h the I2RS protocols are supported.
Nagia: Imagine that the Network Admin is not a person, but automatic. It is important that the machines and humans can equally use the data-modeling language.
Jamal: Are we talking about an CLI or what?
Alia: I2RS is talking about a network application that can communicate and automate the process. The network application is not a proxy for the human typing.
Nagia: I think there should be independence between the data model and the protocol.
Ed: This what we will get to in -2 minutes.
[N…sin}: It has been stated today that Yang is not object oriented. I would like to RFC6095 which implements XSD information types, and introduces inheritance and recursive-ness.
Ed: This is one of my questions that we did not get time for.
[N..x]: Yang is not object oriented, but there are extensions which allow object-oriented behavior modeling.

RestConf and Netconf: Protocol Analysis (D. Bogdanovic, 10 minutes)
[slides-89-i2rs-0.pdf] [1:16:00 -]

Definitions in Protocol
The following definitions are important when discussing the protocol:

- persistent data store ï data store on network element where management system writes configuration data. Persistent data store must survive the reboot process and provide access to historical configuration changes.
- ephemeral data store: Data stores on network element where management system writes temporary configuration data. The ephemeral data store does not survive the reboot process or provide access to historical configuration. The ephemeral configuration data is not verified.
- Operational state ï the state of the control daemon. The operational state can be changed by reading persistent or ephemeral configuration store, control protocols or via APIs.

I2RS must change the operational store, the ephemeral data store, and interact with the persistent store.



Reviewing RESTCONF and NETCONF capabilities for I2RS use

- Yang is assumed for this presentation to be the Data (Modeling) Language for I2RS
- Assumptions: I2RS Agents and Clients have access to Yang Data Models
- The administrator will configure the device through NETCONF and RESTCONF provide mechanisms to configure data that is stored in the ephemeral data store.
- Configuration model has service templates exposed to clients via agents for routing or filtering of data. Since these configurations templates can access all variables (? in

persistent, ephemeral, and operational data stores), it can verify all variables. If all variables provided, it is valid (yes) if not the response to the request is rejected (no).

- The management demon provides authentication for the client. Via Client policy the I2RS agent must define policy for which clients can access which data.
- Filter model is contains rules expressed as:
  - [filter-id][match-func][action-func][address-family]
- netconf and restconf do not modify operational state

NETCONF (RFC6241)

Operates on: RPCs
XML use: configuration data and protocol message

NETCONF has the following 4 layers:

1) content: contains configuration and notification data
2) operations: with base protocol actions to retrieve and edit data (<get-config>), and then commit the data (<commit-config>) to actually configure data into persistent store.
3) RPC message layer ï (<rpc> <rpc reply>)
4) Secure Transport (SSH, SSL, Beep)

If you want to have multiple configurations, the two-stage configuration of data from ephermeral store persistent store makes sense. You can distribute the configuration to all nodes, and then enact all the configurations at once on all devices.

pros/cons:

Pro: IETF configuration protocol, select data retrieval with filter, data validation & verification

Con:

- Canôt modify operational state directly,
- multiple configuration data stores,
- commit model (does not work well in high frequency system), and
- RPC based (hard to debug).

RESTCONF (draft-bierman-netconf-restconf-04)

What: Simplified interface operation on (NETCONF) resource-oriented device abstractions.
Operates on: RPCs
XML use: configuration data and protocol message

Pro:

- Unified data store: contains both configuration and operational data;
- atomicity of transactions: each REST call is one transaction;
- simplified defaults handling: due to atomicity of transactions;
- allows multiple edits (with PATCH) within a single message: with ack/fail on results;
- provides abstracted simplified config model;
- supports XML and JSON;
- streaming via server-sent-events ï which user can decide which stream to listen to;
- edit collision detection.

Con:

- No network locking model (so you do not want to use across multiple devices),
- canô modify operational state of network device,
- using Jason only simple meta-data is supported.

For example: if agent providing statistics reboots, the status can be sent via another channel. This feature will enable a second channel to notify information on server-side events.

Gap of REST CONF AND NETCONF:

- Both protocols lack a mechanism to install operational state on the devices.
- Suggest operations
  - <edit-operational> in NETCONF,
  - <put> or <post> or <patch> to the operational resource in RESTCONF.

Discussion:

[1:26:00 – 1:27:00]

1) Ed: Why do you think RESTCONF cannot support network configuration of multiple devices?

Dan: The problem is with rollbacks on a device. RESTCONF cannot support roll-back, and I2RS supports changing multiple devices in the system. Let me give you an example of setting up a VPN on two devices. On the first device, the VPN set-up works. On the second device, the VPN does not succeed.

Ed: This just means the protocol itself does not support some type of asset semantics. As long as you are willing to track a dependency at the upper level, RESTCONF can perform this task.

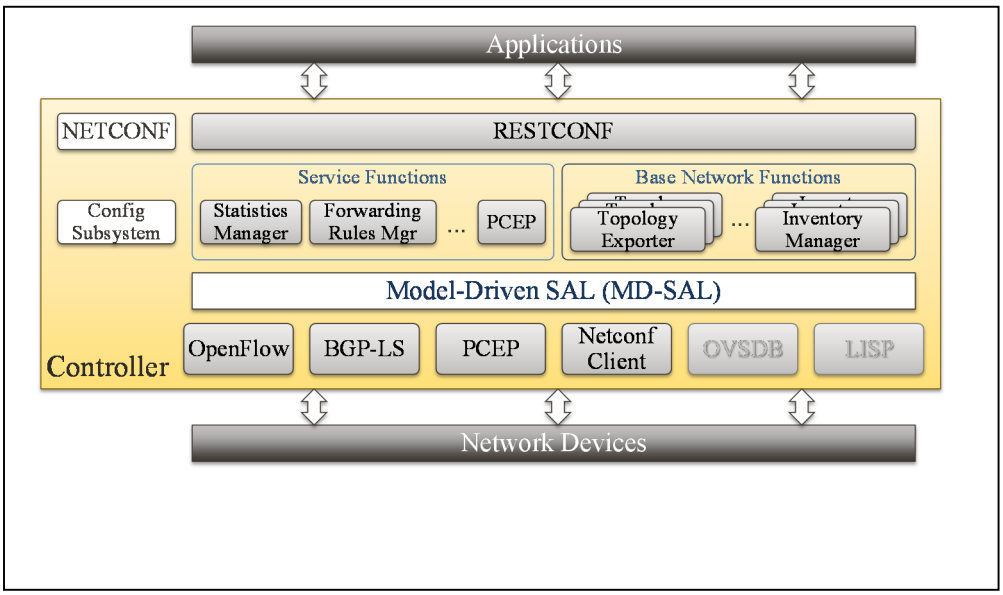
Dan: This is correct. In NETCONF, it is built in. In RESTCONF, you have to place it at a higher level.

2) Dan: It is only using an adjacent only in the META Data. If you modify metadata, you need additional changes (which the netmod WG is working on).
3) Ed: The single data store for RESTCONF may be changing.
4) Dan: This is true, but this presentation gives the current state of information. [time: 1:28-1:31]
5) Andy Bierman: I do want to point out that the data stores you are indicating are configuration.
    a. The NETCONF flag ñConfig = trueò flag is very special to NETCONF. The NETCONF protocol allows data store validation and integrity checks across the entire data store. This features allows NETCONF to have a concept of configuration being valid. So, NETCONF has carved out a space called the running configuration.
    b. The NETCONF Flag ñConfig = Falseò is everything else.
    c. I2RS will create a different data store whose flag is ñOperational = TRUEò, and this data store will have completely different rules on validation checks. Al the validation checks only apply to ñConfig=Trueò

Ed: Each of the data stores may have a different semantic.

I2RS Related/Relevant Yang Models Currently in Use (R. Varga, 10 minutes) [slides-89-i2rs-11.pdf] [1:32-

Co-authors: Robert Varga, Anton Tkacik, Jan MedVed



- Why Yang: :
  - Network device management protocols are: SNMP, TL1, CLI, and NETCONG
    - NETCONF was the best candidate
  - NETCONF had an extensible DDL (or as Andy pointed out a IDL)
    - XML Information Set
    - Augments, Extensions
    - Data-dependent structure ï with whenò statement
    - Early validation ï had an Ranges, ñmustò statement
    - Backwards compatible (with SMIv2 MIB)
  - Netmod is working on Standardized models being worked on
- What models
  - ~110 models defined in OpenDaylight
    - https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:Model_Reference
  - 3 models from RFCs
    - RFC6021
    - RFC6022
  - 8 models from current drafts
    - draft-clemm-i2rs-yang-network-topo
    - draft-ietf-netmod-iana-afn-safi
    - draft-ietf-netmod-iana-if-type

- 8 models ï not in drafts
  - ~10 models dealing with IETF protocols (BGP, PCEP)
  - ~27 models dealing with OpenFlow (1.0, 1.3)
  - ~35 internal wiring models
  - 15

[Discussion] [1:36:07]

1) Jamal: Why do you need 27 models for OpenFlow (1.0, 1.3)
   Robert: I do not know. Tony may be able to help
   Ed: I want to take this question offline. It is totally irrelevant for this conversation.

- **Inventory Model**
  - Inventory model is similar to topology model
  - Represents manageable endpoints connected to controller
    - Simple base model, only skeleton concepts
    - Node ï manageable endpoint, logical node
    - Node Connector ï connection point present on logical node (interface, port, etc.)
  - Extended by technology-specific models
    - Flow-capable Node for OpenFlow 1.3 to include capabilities and Flow Tables,
    - NETCONF Node ï extended to include capabilities, and pass-through access to the underlying device
- PCEP protocol model
  - Message-level model of PCEP PDU as Yang models
    - Based in part on draft-cmfg-pce-pcep-grammar
    - RFC5440, RFC5441, RFC5455, RFC5521, RFC5557
    - draft-ietf-pce-stateful-pce augmentation-{02,07}
    - draft-{crabbe,ietf}-pce-pce-initiated-lsp-00 augmentation
  - Message is a Notification
  - Augmentation of topology model with list of reported LSPs
  - RPC model for invoking PCInitiate/PCUpd requests
  - Tool-generated DTOs, extensible parser
    - https://wiki.opendaylight.org/view/BGP_LS_PCEP:Models
    -

Discussion: [1:37:30 – 1:40:00]

1) Ed Crabbe: One of the reasons I asked Robert to present, I think this is an interesting state studies. It is an example of how to work in yang. Please look at this information.
   [N---(?)] (Cisco): You explained *why* Yang nd *what* youôve done with Yang, but you have not explained *how* you did.

   Ed Crabbe: I think this is expressed in the models themselves (Daylight created models).
   [N---(?)] (Cisco): Usually ñwhy, how, and whatò is how we design and architect systems. I want to understand the usability and how ñfastò you developed these interfaces.

[Robert]: Yang defines your Yang Contract. We through this into a Yang tool which spits out the Java interfaces. At this point, the Java producer and consumers can be developed in parallel. Humans can easily quickly debug Yang files. You can integrate easily because you an enforcer with netconf in the middle.

Jamal: ForCES is ñsimplierò than Yang. The contract is set/get, and you can the information fixed. The Model defines what you are consuming and producing. The validation is built in. We can validate as a middle box (blind to specifics) because we are using a model. If you have a model, you can validate. My apologies to you if earlier I was a bit hostile on OPEN Flow question. To be specific on Open Flow, Evangelos defined a draft which provided the specifics of how to do an OPEN Flow data path. This OpenFlow data path becomes a specific LFB class.

Straw Polls: [Ed Crabbe] [1:40:00-1:46:00]

1) From a modeling perspective:
    a. Who thinks they have an adequate understanding of FORCES to make an informed decision right now?
    b. Who thinks they have an adequate understanding of Yang to make an informed decision right now?
2) From a Protocol perspective,
    a. Who thinks they have an adequate understanding of FORCES to make an informed decision now?
    b. Who thinks they have an adequate understanding of NETCONF to make an informed decision right now?
    c. RESTCONF?

Ed: I asked people to review FORCES careful so we can make an informed decision. I understand that YANG/NETCONF/RESTCONF is more popular, but this is not a good reason to choose this protocol. We need to have an informed discussion.

Thomas Narten: This is useful. Are there people who understand FORCES and NETCONF, and no overlap between the two groups? Iôd like to understand if there are people who understand all three, and we can make an informed decision between the three protocols.

Ed: This is what Iôm trying to determine here. Who understands both? All 10 of you deserve congratulations. Just as an exercise, and as a data point for meé

Who favors: YANG plus NETCONF/RESTCONF as the protocol for I2RS? [chair counts]

Who favors: FORCES? [Chair counts]

ForCES: Protocol Analysis (J. Hadi Salim, 10 minutes)
Discussion (Group, 10 minutes) [1:46:00 ï 1:58:00]

1) Forces <verb> <noun> [args]
    a. protocol (verbs) ï tied to the model
    b. Data model (nouns) ï path

2) Force protocol has the following characteristics:
    a. transport independent (transport middle layer) ï can run over any transport
    b. simple verbs: set, get, redirect, delete, event subscribe/unsubscribe;
    c. optional transactional: roll-forward, roll-back;
    d. Execution model
        i. Atomic - all at once;
        ii. batch ï of things, then if fail it is success;
        iii. high throughput ï 100K table rolls to FIB (important)
            1. Does I2RS require an extremely fast update of the RIB
            2. Binary encoding;
    e. Security ï based on the transport
    f. Traffic sensitive heart beat ï occurs in bi-direction state;
    g. optional high availability ï with hot standby as well as cold standby;
3) Examples:
    Get /RIB/2/Interfaces/1 ï client request to get entry with ifindex 1 from RIB resource on controller;
    GET /RIB/1/Interfaces ï get the who interface table
    Del /RIB/1/Rib/Routes ï flush every route in the RIB table
    Set /RIB/1/Rib/routes (route entry contents) ï create or update RIB on instance 1 with a new route;
    REPORT /RIB/10/Rib/RouteAdded {route entry contents}

4) Gaps
    a. Directionality: Forces assumes the Resource owner (RIB Manger/agent) will associate with the Resource controller.
    Fix: It will require a change to allow client to agent connection.
    b. Client very smart and the agent dumb: FORCES state is stored in the client.
    Fix: Requires: small addition to create partial table: Table create, exclusive create and Append
    c. authorization and authentication: Assumes of a single controller.
    Fix: An identity server (such as OpenStack uses) will allow the agent and client to each get an identity and key. If we extended SCTP, we could use certificates. The client needs to have authorization to be able to change certain pieces of the agent.
    d. multi-headed control is missing:

Fix: LDB requires protocol ID

e. RFC 5811 TML ï may need to be a goo

Forces Summary:

Cons: Not RPC and some protocol changes

Pros: Not RPC, simple and extensible protocol;
Designed for High throughput and low latency;
Capability discovery/negotiation;
Allows for publication and subscription to events;
Rich Transactional events;

Discussion:

D. Bogdanovic: The other piece that is missing is the authorization. You might have a single agent and multiple clients. The group can be authenticated, but they are not authorized to use resources in the agent.

Jamal: Good point.

Information Models: draft-ietf-i2rs-rib-info-model-02 (S. Kini, 5 minutes)

[draft-ietf-i2rs-rib-info-model-02] [1:58:00- 2:01

Changes from version 01 to version -02 include:

- Removed the use of MULTI_TOPOLOGY_ID from being a key to the RIB
  - The routes will be calculated within the client and downloaded to the RIB as any other route.
- Removed the rpf-check-interface variable
  - rpf check is supported, but this variable is not supported due to lack of use case for this rpf-check.
  - If a strong use case exists for the RIB, we will add the use case later.
- Removed inter-domain extensions to RIB
  - The discussion on the list is that this information belongs either in the client or in the inter-domain protocol section
- Removed optimized exit control section
  - This section was a combination of a use case and applicability statement. This has been removed because this will go into a separate draft.
- Removed Next Hop content ambiguous example associated with ñaddressò option
  - Cleaned up the section by adding the following separate options:
    - interface + IP address
    - Interface + MAC address
- Updates to load-balancing section:
  - Version 1 had a percentage value to split between multiple next-hops;

  - Version 2 gives the load balancing as a proportion of all the resolved nexthops total weights

- Future work plan includes
  - Capability modeling
  - Model language ï will be changed to UML
  - Writing a separate draft on routing policies,
  - Split use-cases to separate drafts.
  - We will be writing applicability draft.

- **Discussion:** Ed: Please take all discussion to the list.

Security: I2RS Security (S. Hares, 10 minutes)

[draft-hares-i2rs-security-arch-00.txt] [2:00 ï 2:10]

- Security work is progressing in an active small discussion on Fridayôs at 5pm ET/2PM PT/ 6a Saturday.
- This discussion group was started a little slower because we wanted to finish the security comments in the architecture draft. The architecture draft now contains the initial security discussion. Joel indicated two of the discussion points: mutual authentication and confidentiality. This small group is trying to find additional issues. \
- This small group is looking for security people to join us.

- Document [draft-hares-i2rs-security-arch-00.txt] is an initial start for the WG
  - The current list of issues are the following:
    - Mutual authentication
    - Confidentiality
    - Role Based Security
  - The small group is not looking the following relationship:
    - Application-I2RS Client
    - I2RSAgent-Routing System
  - The small group is only considering the security between the I2RS client and I2RS agent in the protocol.
  - We are looking the impact of security (when/where/how/why) for the full impact of security for I2RS environment
  - The document contains questions for the WG.

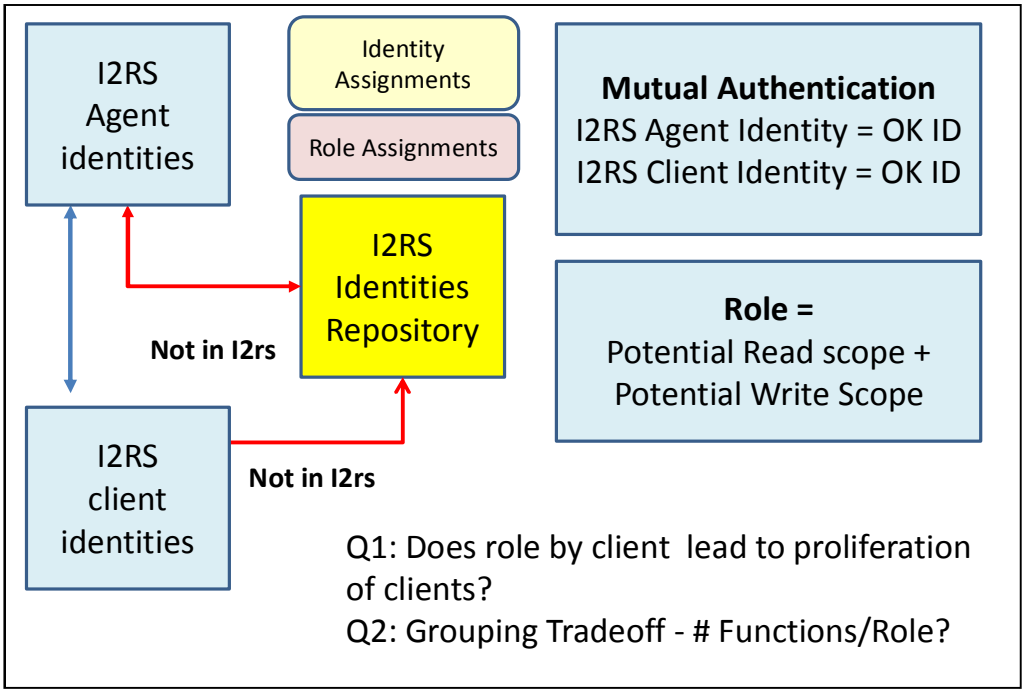
  - Definitional References to look at From RFC4949 are (see figure below)
    - Access control is: Authorized use of resources by authorized entities (users, programs, processes). This includes security which is
      - Role based AC: (RBAC) limits by access by a combination of role + identity.
      - I2RS has a client and agent identities which are exchanged and authorized. Once both are authorized, you can consider whether roles are a match .
      - Roles are simply potential read scope + potential write scope

- Scopes are a set of variables. E.g., "I want to read the BGP notification state".
- Once you have the same read scope and the same write scope, you are doing the exact same things. Whether you have stream 1 across a TCP session or stream 2 across an STCP session, you are attached to the same client doing the same change.
- How does this impact changing data? The small security group would like input from people whose first job is security, and the second job is networking.

  - Confidentiality: data is not disclosed to system entities unless they have been authorized to know, and data is not disclosed to unauthorized individuals, entities or processes.
  - Mutual Authentication : Moving from mutually suspicious state to mutually authenticated by identify and verifying identity and role

- Environmental issues that impact
  - Transport requirements are looking at the:
    - Security Implication on multi-stream model of I2RS,
    - Does the role have a security impact on publishing broker or subscription to events
  - Auditable Data Streams
    - Can the auditable data streams be option?
    - Can we audit either a full stream or a partial stream (filtered audit on filtered events)
    - There is a great deal of concern for auditability, so you have a log of what was changed.
    - From one operator I had an example: If I hand over changes to an automated process. How do know if it works? And how do I handle failures?
    - Please look at: drafts: draft-clarke-i2rs-traceability-01
  - Privacy
    - Is encryption oiptional or mandatory
  - What about stacked I2RS agents
    - i2rs clientð i2rsagent/i2rsclient --- I2rs Agent

- Next steps are:
  - Goal: Quick feedback to Architecture document
  - Goal: 1st Draft completed by 05/15/2015
  - Discussion and adoption afterward
  - Send mail to shares@ndzh.com if you want to join security discussions



Discussion**:** Jeff/Ed - we need discussion for this presentation on the list.

draft-keyupate-i2rs-bgp-usecases-01 (Keyur Patel , 5 minutes)

[2:12 ï 2:14]

I2RS needs to know about BGP. John and I chair IDR. And we get requests to put more routing info in BGP. This use-case is about protocol, route-manipulation, diagnostics, events, and filtering of overlapping BGP Traffic Engineering Routes. For details come to IDR on Thursday, have some interesting drafts there, and some of the drafts

The scope of the I2rs is:

- not to replace any existing configurations mechanism,
- not to replace an existing protocol mechanisms.

The document is the communities understanding of how i2rs can be used in the context of BGP.

The current draft: merged BGP use cases from: keyupate-i2rs-bgp-usecasees-00 and draft-white-i2rs-use-cases-00 as per WGôs feedback. Version 1 has removed BGP Protocol configuration and Policy Configuration per WGôs feedback.

Ed: Working Group adoption stalled. I think this is mature. We're going to do an on-list call.

Keyur (Cisco): Feedback from Working Group was 2 main points:

1) Merge with Russ's draft. Done that.

2) Take config out. Done that.

So we want to re-issue the adoption call.

Ed: Will do that after meeting.

draft-bitar-i2rs-service-chaining-01 (N. Bitar, 5 minutes)

[slides-89-i2rs-4.pdf] [2:14 ï 12:18]

- Objective was to define use-cases for service chaining and required information that could be controlled.

- Scope defined as setting
  - Multi-tenancy service chaining
  - Service topology discovery and maintenance
  - Service node monitoring
  - Controlling the routing on a service chain path,
  - Opaqueness to the actual service provided

- Updates:
  - Addressed comments from Alia on
    - Wording on service node model as part of service topology description.
    - Changed ñforwardò to ñrouteò where appropriate (FIB-to-RIB)

  - Addressed comments from Dave McDysan.
    - One comment was related to OpenFlow, but not within this draft's scope.
    - We did make wording clearer on encapsulation regarding the encapsulation being: IP plus shim header
    - Added information on packet mirror to indicate how packet mirroring is done and over what types of interfaces.
- Next Steps
  - We had new comments [xx] just the week from IETF from Jeff Haas and others.
  - We will address the comments by next week.
  - Add Dave Allan back into the Acknowledgements.

- Key Next Step:
  - We need to figure out how this fits between I2RS and SFC. This draft was developed during the I2RS interim in Sunnyvale. At that time, NFC was the suggestion.
  - Now, we have a SFC WG with a substantial portion of its charter on management. I2RS is about management.
  - Should we leave this document in I2RS or put it in SFC.

Discussion:

- Jeff: Quick poll of room as to how closely following SFC? [lots raised hand]
  - How many thinking this is the right place to SFC OAM? [Lots fewer]

- Dean: Both SFC and I2RS are network management, but from different levels.
  - SFC is for multiple Boxes.
  - Is I2RS for controlling one box whereas SFC is the whole chain?

- Ed: I don't necessarily agree that summation of I2RS.
- Ed: Who wants this in SFC? Nobody.
- Nabil: I'm worried that both groups (SFC, I2RS) will work on this problem. In the end, we need to control the service chain (virtual and physical) and we'll end up on different paths.
- Jeff: We'll have to take this to the list.
- Dan Romascau (Avaya): The two WG are dealing with different management layers. OAM layer is different from the Provisioning layer.
- Please do this on the list.
- Tom Narten (co-chair): This is relevant to SFC. Chairs are aware. But issue is where we put this. I think parts of this are relevant to SFC.

draft-white-i2rs-use-case-02 (R. White, 10 minutes)

[draft-white-i2rs-use-case-02] [2:19-2:21] Russ White (Ericsson)
[slides-89-i2rs-5.pdf]

- We want to move this (very basic) use-cases draft into the WG adoption. Time for it to move forward.
-

Various use cases + Juggling Demonstration (S. Hares, 10 minutes)

[2:22-2:26]
[draft-ietf-hares-i2rs-i2rs-use-case-vn-vc]
[draft[ietf-ji-i2rs-ccne-services-01]
[draft-chen-i2rs-ldp-mpls-use-cases]
[draft-huang-mpls-te-link-usecases-01]

- Introductions:
  - Goal here is 5s use-cases in 5 minutes.
  - Use model started simple with the RIB use cases, and then we did BGP. Then Nabil gave us service chaining.
  - Now, I'm going to give you Virtual Circuits.
- draft-hares-i2rs-use-case-vn-vc
  - **What is it:** This is a use case where a client can set-up both virtual circuits and virtual networks.
  - **What do we need to add to I2RS:** We need a collector, traffic matrix, and PBR.
  - **Other Drafts:** There are a couple of PBR drafts out, and I have a PBR related draft. We hope to combine this work.

-
- draft-ietf-ji-i2rs-ccne-services-01
  - **What is it:** This is the Centralized Computation controller.
  - **What do we need i2rs to do:** Collect RIB, get info via RR, PBR. sounds familiar?
- draft-chen-i2rs-ldp-mpls-usecases
  draft-huang-mpls-te-link-usecases-01
  - **What is it:** LDP and MPLS-TE is used to create a set of network topologies using LDP and MPLS-TE for a LSP path.
  - **What I2rS needs to do:** This can collect LDP and TE info and traffic stats. You can push proactive changes. Again looks about the same. Now have basics want feedback on these drafts.
- DC Traffic Steering.
  - **What is it:** Original models had traffic steering between data-centers, but didn't necessarily have traffic steering from one data-center to different core networks. This I2RS requirement which comes from the NVO3 space.
  - **What I2RS needs to do:** collect routing info, traffic matrix, RIB info.
- **Mobile Backhual:**
  - What is it: Creation of topologies for Mobile backhaul use.

Discussion:

Jeff: What we want to do with the use-case docs is clean up the in-charter ones up and progress. Also putting together a requirements doc that summarizes the requirements we're seeing across use-cases whether in-scope or not. We are not sure yet what fate of out-of-charter use-case draft is. We may adopt them, or may leave as individual submissions.

Ed: nice thing with model-based networking is that use-cases lead directly to a model. Once we select the underlying transport system, it is relatively a simple progression, but trying to stay in charter as we're behind.

David Wood (Juniper): How do you collect the traffic matrix?

Ed Crabbe: I guess it's a Question for Sue. But presumably you can use IPFIX Netflow, interface/LSP utilizations, etc. Sue has a look of consternation on her face.

Sue: Yes, there are more places that you can collect stats. The info models are the place to start to define it. That's why we started with RIB info model.

Ed: We're not going to mandate how you collect it.

David Wood: This was more of a comment that collecting traffic matrix is quite hard. Slide said "collect traffic matrix". My comment is that that's a hard thing to do.

Adjourn: 2:30