IBNemo BOF
Internet-Draft
Intended status: Standards Track
Expires: January 6, 2016

S. Hares
Huawei

July 5, 2015

Intent-Based Nemo Problem statement
draft-hares-ibnemo-overview-00

Abstract

As IP networks grow more complicated, these networks require a new interaction mechanism between customers and their networks based on intent rather than detailed specifics. An intent-based protocol language is need to enable customers to easily describe their diverse intent for network connectivity to the network management systems. This document describes the problem Intent-Based NEtwork Modeling (IB-Nemo) language is trying to solve, a summary of the use cases that demonstrate this problem, and a proposed scope of work. Part of the scope is the validation of the language as a minimal subset.

The IB-NEMO language is a protocol language for interactions between an application and a network manager/controller. Some would call this boundary between the application and the network management system as northbound interface (NBI), and any protocol language that crosses this as an NBI. IB-Nemo focuses on creating minimal subset of the total possible Intent-Based desired commands. By creating a minimal subset (about 20% of the total possible), the IB-Nemo language can be a simple Intent interface for most applications (hopefully 80%). Part of validation of this language is to to determine what data models should result in the network controller from different use cases. This way as IB-Nemo protocol language is reduced the effort can verify that the critical information is stilled passed.

Table of Contents

1. Introduction

This document describes the problem Intent-Based Network MOdeling (IB-Nemo) language is trying to solve, a summary of the use cases and a proposed scope of work.

IB-Nemo focuses on a minimal size language to express Intent from an application to a network management system (or network controller). Some would describe the interface between the application and a network as the north bound interface (NBI) from the network manager. This paper will utilize that term to indicate the point the IB-Nemo language protocol goes across.

The general idea of "intent-based networking" is that user tells the network what he wants, but not how to do it. Network provisioning can then creatively fulfil the users desire. The key challenge is to provide the user with tools to express what the user wants.

Creation of a minimal size Intent-Based language for Intent requires boiling down the possible alternative to a minimal subset. It is like the creators of SQL, boiling down the potential database commands to a small common subset. The process of creating a minimal size language requires that the language pass some additional information for each applications specific context it operates in. For networking, an example of this additional context may be the name to address mapping for the nodes the applications wants to connect. Some data may be additional attributes bound to the role the application performs. For example, if one of the nodes the application wants to connect to is a mail-spam cleaner, then additional attributes may be listed. Another example may be a "network-nanny" firewall that enforces parental controls. To test SQL language, the language creators run the commands through a set of prototypical database models for a set of use cases.

To test IB-Nemo, the working group must select use cases and develop prototypical data models that should be able to be created in the network management system by use of the IB-Nemo language.

1.1. Where to start

In the spirit of minimalism, this introduction starts with a 5 question FAQ (frequently asked questions) for those who are familiar with the concepts of Intent-Based networking to answer "what is Intent-Based Nemo". If the FAQ answers your questions, jump off to the use cases in this document or the [I-D.xia-sdnrg-nemo-language] along with its management yang modules [I-D.zhou-netmod-intent-nemo].

If you are new to the Intent-Based networking, you'll want to read through the motivation section before looking at the rest of the document.

The purpose of this document is simple: to provide others outside the project with "what, when, where, how, and why" the IB-Nemo network language should be standardized in the IETF as part of the larger Intent-Based network effort.

1.2. FAQ

Q1: There are many industry forums working on an Intent-based policy interface for applications. Why should the IETF form a Working Group to examine an Intent-Based language?

Over the years industry forums have tried to create a mosaic of standards groups where each standards group focuses on it's key role. IETF has focused its efforts on protocols that communicate across the IP network, and management protocols to manage these efforts.

The Intent-Based Network Modeling (IB-Nemo) language is a language which communicates between an application and a network management system that controls traffic through the network. Different forums may call this network management different names (E.g. SDN controller or centralized controller or others).

The IB-Nemo language seeks to provide a minimal set of language statements to pass the intent from an application to the network management system which is controlling the networks.

Q2: Can Intent North Bound Interfaces (NBIs) control more than networks?

A user may use Intent language to control storage or CPU cycles, but an intent-based networking language focuses on networks. Why?

Many operators supporting this work want to control virtual networks, service-based forwarding in networks or data center networks, home-networks, and mobile networks. If Intent based networking is successful, then the community may turn to controlling networks plus storage plus CPU. The group is starting with what they know.

The [I-D.xia-sdnrg-nemo-language] focuses on three basic components: logical node, logical link, and a logical data flow.

Q3: Why a minimal size language? How will you control all of the network management devices that control the network?

The purpose behind the minimal language set is provide a very simple language that most applications can use for simple operations. Often in languages, most users (say 80%)of the language utilize only 20% of the commands. We’ll call this within this paper as the 80/20 rule of languages. To be available for most applications, the language must be standardized, interoperate between different implementations, and have management interfaces.

The IB-Nemo language [I-D.xia-sdnrg-nemo-language] allows groups of applications to simplify the interface by providing the capability to transfer a data model that can store common information (e.g. names or addresses) for nodes and links plus rate of data flow (e.g. 10Gigbit). As an example, an application for a home-network on a cable network can simply load one set of data from a library and pass them to the network management system. Applications for virtual networks for a company could load a different set of data from a library and send it to the network management system.

The goal of this language is not to support all possible Intent language commands nor all network management systems. The intent is to work within the 80/20 rule.

Open-Daylight (ODL) has three Intent-Based Code projects:

o Network Intent Composition (NIC) (https://wiki.opendaylight.org/view/ Project_Proposals:Network_Intent_Composition) (ODL:NIC),

o Open Daylight Nemo (ODL Nemo) https://wiki.opendaylight.org/view/ NEMO:Main, and

o Group Based Policy (ODL-GBP) (https://wiki.opendaylight.org/view/ Group_Based_Policy_(GBP)).

The ODL-NIC project is creating a Intent based interface that provides all necessary Intent. The ODL Nemo project is focusing on creating a minimal size language interface using the 80/20 rule of languages. The ODL-GBP sees Group-based policy as the automation of Intent by creating contracts between groups of endpoints.

Q4: Is it time for IETF standardization?

An Open Source release of the Open Daylight code for IB-Nemo (ODL Nemo)under the Open Daylight Nemo will occur in July of 2015. A demonstration of this was shown at ONS 2015. The Open Daylight code for the Network Intent Composition (ODL-NIC) and the Group Based Policy (ODL-GBP) was released with the Lithium release in June 2015. Demonstrations were shown at ONS 2015 of these three source projects. Both ODL-NIC and ODL-GBP are full-features (aka non-minimal size) north-bound interface.

The Open Daylight code base has been transitioned to products with a number of vendors. Some of the ODL source code is headed for the Open Platform for NFV (OPNFV) project (https://www.opnfv.org). The IB-Nemo project team is working on the OPNFV Movie project (https://wiki.opnfv.org/movie) to provide use cases that will allow matching the ODL code bases with the OPNFV deployments. Much of the open source code from ODL and OPNFV open source projects has moved into the product code bases of vendors.

Now is the time for the IETF to begin to standardize the interoperability of the IB-Nemo interface as the code enters these open source bases.

Operators in carrier and cable (MSO) see this as a key way to speed up provisioning by obtaining their users desires via the Intent Interface. Operators like Telefonica wish to plug IB-Nemo into their Net-IDE interface.

Q5: What data models will IB-Nemo focus on?

IB-Nemo is focusing on the data modeling that will allow development of a minimal size language. The process of developing a reduced set of language commands involves choosing the use cases that must be solved, and then attempting to design the language to pass the right information from the application to the network management system.

The best way to validate the language is to have prototypical application use cases and then use the language to pass the intent plus the additional contextual information needed in order for the network management system to create the virtual network needed. A good way to summarize the information the network management system stores is in a yang data model. Therefore, the working group scope includes the creation of these data models to test the language. Long-term these test cases can be used to test language implementations.

Like All protocols, IB-Nemo will be created with yang data modules to configure and manage the protocol. However, these are different than the modules used to validate the subset of interoperable commands.

These data models are not information models for generic Intent-based or declarative policy. SUPA is working on generic information models for Event-Condition-Action (ECA) and declarative policy. As these models develop, it is hoped their insights on policy may help those working in the Intent-based policy.

IB-Nemo work plan does not focus on being an automation architecture or protocol. ANIMA is working on this in the IETF.

```
               context library
                        :
        +----------:-----+
        |  application    |
        +-------||--------+
                || http with IB-Nemo
                || language
   +-----------------------------------------+
   | network    ............   +=======+    |
   |management  : Nemo      :   | Nemo  |    |
   | system     : Intent    ===== Models|    |
   |            : Engine    :   | for   |    |
   |            ..............  |content|    |
   |            :yang models :  +=======+    |
   |            :to configure:               |
   |            :Nemo Intent :               |
   |            :Engine      :               |
   +-----------------------------------------+
```

1.3. Definitions and Acronyms

ETSI: European Telecommunications Standards Institute

Intent-Based Interface: An interface which tells what what to do (go to store) rather than how to do it. (Travel 5 miles down this road to SAMS Club store)

Intent-Based Language: A intent-based interface consisting of a protocol that carries a set of Intent commands from the application to the network management system.

NETCONF: The Network Configuration Protocol

NFV: Network Function Virtualization

ODL: Open Daylight project

ODL NIC: ODL Network Intent Composition

ODL Nemo - ODL Network Intent Nemo

ODL GBP: Open Daylight Group Based Policy project

ONF: Open Network Forum

RESTCONF:REST-like protocol that provides a programmatic interface over HTTP for accessing data defined in YANG, using the datastores defined in NETCONF.

2. Motivation for Intent Interfaces

The IP networks within Carriers, Data Centers, Cloud provider, and Enterprises continue to grow in size and complexity. Simultaneously, the services that are demanded by customers, particularly the upper layer applications, are becoming more and more complicated. The users of these service demand that the services be available to mobile devices (E.g. iPADs, smart phones) as well as their desktops. New applications that demand these services have a short life span (months rather than years). The current rigid service models are lacking the flexibility to meet this combination of requirements and scenarios.

Recent efforts have looked to open source and open APIs for the IP devices and networks as a way to replace the rigid service models with fast-paced development. ETSI’s NFV group, CableLabs DOCSIS (docsis.org), and ONF Intent-Based NBI (North-Bound interface) are industry forums looking at Intent based open APIs. OPNFV Movie project (https://wiki.opnfv.org/movie) is examining the intent-based use cases for OPNFV (https://www.opnfv.org/). The use cases in this document encapsulate many of the use cases discussed with operators and vendors individually or within these forums.

The idea of Intent-based networking can be summed up in a simple phrase: "Do not tell me what to do, tell me what you want". Traditional networking configures devices, network protocols, and topologies within a network. It is network-device centric. Intent-based network focuses on the applications (or application workload) and their interactions. It is application-centric. In Intent-based networking, the network provisioning or network automation can work many ways as long as it provides the application the requested service.

Intent-based network models present the network as the application would see it. Intent-Based Nemo utilizes the application-centric view in its modeling of a network. These models may hide details the application does not need to know.

2.1. Challenges in Intent-Based networking

The challenges in Intent based networking are to:

1. create a common definition of intent,

2. create a common architecture of a Interoperable Intent-Based Northbound API,

3. create a standard and interoperable way for the applications to communicate with the network, and

4. create a way to reduce the complexity that the context places on the intent engine.

The ODL projects, the Distributed Management Task Force (DMTF - www.dmtf.org), Open Networking Foundation (ONF) Intent-Based Northbound interface(NBI) working group (ONF Intent NBI WG) (https://www.opennetworking.org/technical-communities/areas/services/1916-northbound-interfaces), and OpenStack Congress (https://wiki.openstack.org/wiki/Congress) are working on definitions of Intent.

The IETF SUPA BOF (http://tools.ietf.org/bof/trac/) proposes to create IETF Working group which will create a generic declarative information policy model as well as a generic Event-Condition-Action (ECA) policy model. The authors of the SUPA BOF policy drafts are familiar with the DMTF work, the ONF NBI WG effort, and the OpenStack Congress model.

ONF Intent NBI WG (http://www.onfsdninterfaces.org/) and ODL-NIC project are working on common architecture principles for the Intent-Based Northbound API (https://wiki.opendaylight.org/view/Network_Intent_Composition:Main) with work to define application end points (https://wiki.opendaylight.org/view/Network_Intent_Composition:Dynamic_Attributes).

IB-Nemo seeks to simply apply this evolving work by creating an interoperable minimal size language operating as a protocol between the application and the network management system (or network controller). The IB-Nemo language interface reduces the complexity of the full intent-based NBI (northbound interface) by supporting a portion of the commands most often used. The people on the ODL Nemo project https://wiki.opendaylight.org/view/NEMO:Main. have selected a small set of commands and created an open-source prototype. The IETF work is to review and standardize the set of commands to make sure it provides an interoperable set for all applications.

2.2. Roles and User specific network information

Authentication, Authorization and Accounting (AAA) protocols such as Diameter and Radius pass information on the access permissions that certain users or user programs have to a network or virtual network. Group based policy suggests that a group of users may share a set of policy that determines the access to the network or a virtual network. Role-based network access suggests that roles can better summarize what access the user or user programs have to the network. Since IB-Nemo is trying to use prototypical use cases to test the ability of the IB-Nemo language to pass enough information to create the appropriate data models in the network management system, it is natural to use the role-based concepts of summarization to describe these data models.

The contextual information is the characteristics which make groups of applications unique when operating over the network. Logically most of this information may be associated with roles. For example, if you have a set of users in a home communicating over a home network the characteristics which are unique is a set names and address for devices, links, and policy within the home. If it is a virtual network for a company, the unique information the names, addresses, links, and bandwidth expected on the links along with security issues. As these examples show, Intent networking can be seen to be a few prototypical application-centric network topologies plus a set of unique information (which could be called context). Both the home network and the virtual network are creating a virtual network for the applications running over the network.

2.3. What is a simple Intent-Based Interface?

What is a simple interface? It is said that 80% of the applications only use 20% of the commands in any open API. This paper calls this the 80/20 rule of networking. A simple Intent-based interface only supports these 20% of the total Intent-Based commands in a north bound interface (NBI). The challenge in any Intent-based interface is to create a simple interface that serves 80% of the applications that is easy to use and similar to a human being’s natural language.

The challenge is that different industries may have a different 20% of the commands that are commonly used. The Nemo Project teams in the ODL Nemo project and OPNFV Movie project are seeking uses cases to determine if there is common set of use cases that vary just by context. For example, a global L3VPN for a company with three sites may be similar to a three site L3VPN across a cable network.

After getting a set of uses cases, creating a simple interface is the four step repetitive process:

1. find use cases,

2. develop prototype code,

3. do early testing at proof of concept demonstrations and hack-a-thons

4. work with many vendors to clarify language to make the language small and interoperable, and

5. go back to step 1

Where is Nemo is this process?

IB-Nemo has gone through steps 1-3. Use cases are listed below, and the OPNFV project is working on use cases. IB-Nemo's ODL Nemo project is developing the code for the open source (ETA July release). IB-Nemo is at a stage where it needs to work in a standards body to create a small, efficient, interoperable protocol language.

The standardization through an IETF WG will help IB-Nemo to work on step 4.

2.4. Intent-Based NBI Open Source is heading toward Products

The following are Open Daylight Projects:

Open Daylight Group Based Policy (GBP)
https://wiki.opendaylight.org/view/Group_Based_Policy_(GBP)

OpenDaylight Network Intent Composition (ODL-NIC)
(https://wiki.opendaylight.org/view/
Project_Proposals:Network_Intent_Composition), and

Open Daylight Network Intent Composition: Nemo
https://wiki.opendaylight.org/view/NEMO:Main.

These are open-source coding efforts creating an intent-based northbound interface for intent-based networking.

The ODL Group Based Policy (GBP) views policy as a contract between two endpoints, and sees its work as the automation of Intent.

ODL-GBP was released in the ODL Lithium release in June of 2015.

The ODL-NIC project is creating a Northbound interface (NBI) for network orchestration systems, SDN applications, and Network operators. It may be defined as RESTCONF [I-D.ietf-netconf-restconf] protocol and/or Java APIs. This extensible interface will be designed to allow any and all new intent expressions to be exposed as part of a consistent and integrated single NBI to SDN applications. The singularity is necessary for the Composition Function to provide a comprehensive capability to manage network resources and resolve conflicts across application's intents. In a sense, the ODL-NIC project is suggesting a thin waist of a single API at the entrance to the networking layer, just as the IP protocol presents a thin waist of a single API at network layer.

ODL NIC project was released in June of 2015.

The ODL Nemo project has created a minimal language interface as part of this effort that funnels all new intent expressions into a consistent and integrated single NBI for SDN applications. The original language has 15 language statements in three groups. Group 1 describes nodes, links, and flows between nodes. Group 2 deals with operational checks (query, notification, policy, connect, disconnect, session (start), and commit (end of commands). Group 3 defines the model that provides the context for nodes, links, flows and policy.

ODL Nemo project is due to be released in July of 2015

ODL open source code is currently finding its way rapidly into other sources (E.g. OPNFV code base) and into products that are within 6 months to a year of release.

2.5. IB-Nemo Intent NBI is Synergistic to NETCONF and I2RS

The IETF netconf [RFC6541] and restconf [I-D.ietf-netconf-restconf] protocols provide a network interface to the configuration and status information within IP network devices. The IETF I2RS (Interface to Routing System) WG is creating a highly dynamic network interface to the routing system which can inject or retrieve state regarding routing state, topologies, filters, and operational state. The PCE Working Group has protocols and methods to pass routing for calculation. Each of these interfaces and protocols have a purpose in managing and enhancing IP network infrastructures.

Intent Based NBI is synergistic to these IETF interfaces to the devices. Synergistic means that sum of Intent Based Nemo language + NETCONF + RESTCONF + I2RS + PCE is more than any of the parts alone. Intent Based Nemo language can signal from the application/user to a central client which configures, manages, and monitors network devices through these protocols.

2.6. Rest of Document

Based on this motivation, the next sections discuss:

- o What is Intent-Based NEMO Language Interface?
- o The Scope should the Intent-Based NBI work
- o Summary of Use cases for this scope
- o Gap Analysis and where IB-Nemo fits
- o Transition from IRTF to IETF

3. Intent-Based NEtwork MOdel (IB-Nemo) Language interface

A protocol based on language best resembles a natural language. To determine what form the language should take, the authors of [I-D.xia-sdnrg-service-description-language] analyzed customer technical requirements to determine the design considerations for such a language. They conclude that an intent-based language should have the following abilities for virtual network devices:

- o Be able to describe customer traffic which can be identified as flows,
- o Be able to describe access nodes, virtual networks, servers, and other network entities as the end-customer perceives these devices;
- o Be able to describe QoS, SLAs, and other relevant properties;
- o Be able to describe logic that combines a few demands together with certain choices for specific circumstances;
- o Be able to describe the network so the network customers can describe their demands; and
- o Be able to be extended.

The Open-Daylight Network Intent Composition project (https://wiki.opendaylight.org/view/Network_Intent_Composition:Main) has begun an open-source project for a North-Bound Interface (API) from orchestrator to controllers that provides abstracted policy syntax rather than open-flow rules.

The Affinity chaining proposal (https://wiki.opendaylight.org/images/3/30/

Affinity_Service_Chaining_Proposal_ODP_7-23-2013.pdf) suggests structure similar to IB-Nemo’s structure (node, links with end-points, and flows). This open-source project has suggested requirements similar to requirements noted by [I-D.xia-sdnrg-service-description-language].

4. Scope

4.1. Inside Scope

The initial scope of this IB-Nemo work has focused on:

1. creating minimal size language for north bound interface for Intent-Based Networking with a modelling mechanism that handles user context,

2. selecting use cases and associating them with prototype applications in order to determine the subset of commands that needs to be included in IB-Nemo language,

3. validating the IB-Nemo language by creating data models (which should exist in the network management system) for each application use case to determine if the language can help a network management system create the right data model

4. creating a management data model to manage this Intent-Based Networking language, and

5. working with other forums to refine a definition of intent so that the minimal size language serves a wide range of use cases (target of 80% of known use cases) with an interoperable interface.

4.2. Outside of Scope

The following things are outside the IB-Nemo scope:

- The creation virtual networks using I2RS or netconf/restconf to directly connect to a yang model is outside the the scope of the proposed IB-Nemo work.

- The creation of a service-layer interface using I2RS and yang data models is outside the scope of the proposed IB-Nemo work.

- The creation of a language to communicate from a security network management system to the network security devices is outside this scope.

5. Use cases for Intent-Based IB-Nemo

The following use cases are described in this section:

1. Virtual WAN

2. Virtual Data Center

3. Bandwidth on Demand

4. Service Chaining

5.1. Virtual Wide-Area Network (WAN)

Description: Enterprises want to set up their own virtual WAN for more control and optimization.

User Intent: Create virtual Wide-Area network between offices.

Network management systems do the following:

1. Deploy virtual routers and links for a customized topology.

2. Identify flows.

3. Steer flows though different path. (E.g. real-time flow to go through a shortest path, and backup flow to go though a broadband path but may have more hops.)

The network management data system should have a data model that captures this information. IB-Nemo needs to pass this information in the IB-Nemo language.

Network operator: Creates web portal for business customers to request a WAN connecting offices. Interface request corporate ID, security ID, and a link to the payment system.

The sub-cases of this general use are the following.

Home LAN attached to Corporate Network

parental controls for child travelling outside the home

Details can be found in (draft-hares-nemo-usecases-00.txt)

```
          ==== real time (R1-)
          **** broadband
                  ....................
                  :   Virtual LAN    :
                  : (real time path) :
                  :                  :
        +-------+ : (real time path) : +---------+
        |      ===========================       |
        |       | :     e      f     : |         |
        |Beijing|-----R1- - - - - R2---| London  |
        |office | ***a|* \b  c / | d : | office  |
        +-------+ :   | * \   /  |****>+---------+
                  :   |  * \ /   |*  :
                  :   |    / \*  |*  :
                  :   |   /   \* |*  :
                  :   |  /     \*|*  :
                  :   R4- - - - R3   :
                  ....................
```

Figure 5-1:

5.2. Virtual Data Center

Description: User (corporate or home) creates a virtual data center with network. The virtual data center has a front-end network of router to exterior firewall to DMZ LAN to interior firewall to computing user.

User Intent: Corporation wants to buy want to buy Cloud computing inside a virtual data center with secure computer cluster.

Network Operator service provider: Defines secure cluster network as the following network topology:

o router connected to network,

o exterior firewall,

o DMZ LAN,

o interior firewall,

o interior secure LAN with compute clusters.

The network management system must have a data model with this information. IB-Nemo must pass this information to the network management system.

Network Operator:Creates Web portal for business customers to put in request with corporate ID and level of security for cloud cluster. User will have to provide corporate accounting and security IDs.

Context: Corporate context puts in the amount of computing power and the virtual topology for security. The template of Secure vDC will be set-up with router, exterior firewall, DMZ, interior firewall, LAN. Both the Corporate context and the secure vDC context will be loaded into the customer’s context for processing.

Operator automation: Based on the context with Intent, corporate context, secure vDC context, the operator automation series will place the virtual cluster in a data center, and set-up the vDC and the Cloud computer clusters. The Corporate customer IDs that are pushing data to this vDC will have the vDC defined in the Corporate culture.

Specific use cases from this prototypical use case are:

o User gets clean mail services with firewall and spam mail cleaner

o SMB Manufacturing network with Virtual DataCenter

o SMB with Sales-Marketing accounting on Virtual Data Center

These are described in (draft-hares-nemo-usecases-00.txt)

```
              (internet )
             |
   ..........|...................
       +-----|------+
       |  router    |
       +-----|------+
             |
       +-----|------+
       |  firewall  |
       | exterior   |
       +-----|------+
             |
      ===============
       DMZ   |
       +-----|-------+
       |  firewall    |
       | interior    |
       +-----|-------+
             |
       ( protected  )
       (  cloud     )
```

Figure 5-2

5.3. Bandwidth on Demand

Description: The corporate user wants to create a virtual link between remote offices and headquarters that has bandwidth that can be adjusted based on time of day.

User Intent:Corporation wants to connect branch office with corporate office with 10G of bandwidth for data flow 8am to 6pm, and 1G of bandwidth from 6pm to 8am.

User interface: A web portal allows him to login (corporate ID and security IDs) and indicate this intent via a graphic picture of his network that allows him to indicate on-demand bandwidth size and time of day.

Network Operator:Creates Web portal for business customers to put in request with corporate ID and level of security for entrance into the corporate intent site. The Web portal allows for prototypical use case (virtual WAN, Virtual DC, Bandwidth-on-demand Virtual Private Network (VPN), Service Function Chaining (SFC). The network operators store enough application-level topology that the the users intent is defined.

Operator automation: Based on the data passed by Nemo providing the Intent and the data regarding the corporate virtual data center, the provisioning software will automatically will allocate bandwidth between these two sites at the rate indicated. The access router/ switch can optionally limit at a rate over this value.

Corporate Virtual Data Center information: includes the IP address, DNS names, and application addresses (Transport Ports, application identifiers) of subnet with application works on, and the applications transferring data. The corporate data also includes information on whether L2VPN or L3VPN is used by the customer.

```
          ==== 8am to 6pm 10GB
          **** 6pm to 8am 1BG
                  ....................
                  :   VPN            :
                  :                  :
            +-------+ :     daytime      : +---------+
            |      ===========================      |
            |       | :     e      f     : |         |
            |Branch |-----R1- - - - - R2---| HQ      |
            |office | ***a|* \b  c / | d : | site    |
            +-------+ :   | * \   /  |****>+---------+
                  :   |  * \ /   |*  :
                  :   |    / \*  |*  :
                  :   |   /   \* |* night time
                  :   |  /     \*|*  :
                  :   R4- - - - R3   :
                  ....................
              Figure 5-3:
```

The following use cases are specific examples of this prototype use case:

Home Network gaming system

Home Security system zoom-in

Application Big Data or SAP Transfers at night

Database applications contact other database applications

5.4. Service Chaining

Description: Apply several virtual network functions, such as firewall, load balancer, WAN optimization between virtual private cloud and the internet.

User Intent: User has a private cloud and wants to get a secure interface to the Internet.

Network Operator network management system defines the secure access ring of protection around the private cloud to be the following virtual network topology:

o firewall

o load balance

o DPI inspection

Network Operator:Creates Web portal for business customers to put in request with corporate ID and level of by clicking a request.

Corporate Information: Corporate context has the topology of private cloud, and the access points. The network operator will access service chaining to through a virtual access ring.

Operator automation: Based on the context of the network topology of the private cloud’s link to the carrier network and the access points to service chains, the network automation sets up the traffic flow so that the traffic to and from the private cloud flows through a firewall, load balancer, and DPI inspection.

```
           (internet )
           |
 ..........|...................
     +-----|------+
     |  firewall  |
  |--|  function  |--------|
  |  +-----|------+        |
  |        |               |
  |  +-----|---------+     |
  |  | load balancer |     |
  |  | function      |     |
  |  +-----|-------|-+     |
  |        |       |       |
  |    +---|-+ +---|--+    |
  +----|DPI 1| |DPI 2 |----+
       +---|-+ +---|--+
           |       |
         ( private Cloud   )
         ( for corporation )
```

Figure 5-4

The specific use cases for this prototype are:

Providers access edge box replaced by service chaining for wired and wireless (LTE and Wifi)

Corporate access edge box replaced by service chaining for wired and wireless

Wifi offload of LTE does service chaining to replace mobile services

6. Gap Analysis and where IB-Nemo Fits

6.1. IETF Working groups Gap Analysis

No working group is working on an Intent-Based NBI.

SUPA proposes to create an information model for generic and intent-based policy. IB-Nemo will use this generic intent-based policy to help guide their creation of the minimal size intent based NBI.

NETCONF and NETMOD are not creating an intent-based interface.

6.2. ODL Open-Source

ODL network intent composition (ODL-NIC) is creating a full intent-based North Bound Interface. ODL Nemo is creating a minimal size NBI (20% of command that serve 80% of applications) in open source. The IETF IB-Nemo work will create an interoperable protocol based on the IB-Nemo language with its context models.

OPNFV Movie project (https://wiki.opnfv.org/movie) is defining the use cases for Intent-Based networking. IETF IB-Nemo will expand on these use cases, and exchange information beyond just the Network Function Virtualization into Cable networks (MSO) or carrier networks.

6.3. Open Stack Policy initiatives

None of the Open Stack Congress work focuses on Intent networking or intent-based policy.

Open Stacks policy includes network, compute, and storage. Its work combines automation (scheduling of resources, monitoring cloud services, Event-Condition-Action (ECA) policy, ECA based management), store-related policy, and meta-data policy storage. The projects are:

OpenStack has Congress (https://wiki.openstack.org/wiki/Congress) with its Congress initiative aims to provide an extensible open-source framework for governance and regulatory compliance across any cloud services (e.g. application, network, compute and storage) within a dynamic infrastructure.

SolverScheduler (Nova blueprint): The SolverScheduler provides an interface for using different constraint solvers to solve placement problems for Nova. Depending on how it is applied, it could be used for initial provisioning, re-balancing loads, or both.

Gantt: A scheduler framework for use by different OpenStack components. Used to be a subgroup of Nova and focused on scheduling VMs based on resource utilization. Includes plugin framework for making arbitrary metrics available to the scheduler.

Neutron policy group: This group aims to add a policy API to Neutron, where tenants express policy between groups of networks and ports. Policy statements are of the form "for every network flow between groups A and B that satisfies these conditions, apply a constraint on that flow". Constraints are currently are allow or deny, but this may expand.

Open Attestation: This project provides an SDK for verifying host integrity. It provides some policy-based management capabilities, though documentation is limited.

Policy-based Scheduling Module (Nova blueprint): This effort aims to schedule Nova resources per client, per cluster of resources, and per context (e.g. overload, time, etc.).

Tetris: This effort provides condition-action policies (Event-Condition-Action policy). It is intended to be a generic condition-action engine handling complex actions and optimization. This effort subsumes the Runtime Policies blueprint within Nova. It also appears to subsume the Neat effort. Tetris and Congress have recently decided merge because of their highly aligned goals and approaches.

Convergence Engine (Heat): This effort separates the ideas of desired state and observed state for the objects Heat manages. The Convergence Engine will detect when the desired state and observed state differ and take action to eliminate those differences.

Swift Storage Policies: A Swift storage policy describes a virtual storage system that Swift implements with physical devices. Today each policy dictates how many partitions the storage system has, how many replicas of each object it should maintain, and the minimum amount of time before a partition can be moved to a different physical location since the last time it was moved.

Graffiti: Graffiti aims to store and query (hierarchical) metadata about OpenStack objects, e.g. tagging a Glance image with the software installed on that image. Currently, the team is working within other OpenStack projects to add user interfaces for people to create and query metadata and to store that metadata within the project's database. This project is about creating metadata, which could be useful for writing business policies, not about policies over that metadata.

7. From Open Source and IRTF to IETF

As discussed above, the open-source work for ODL-NIC had its first release in June of 2015 and ODL Nemo plans its first release in July of 2015. The movement of these code sources to OPNFV (https://www.opnfv.org/) will happen rapidly, aided by the OPNFV Movie project (https://wiki.opnfv.org/movie) use case work. In order to get an interoperable minimal size (80/20 rule) IB-Nemo language, it is important to standardize the language in IETF. As part of the standardizing of the language, the work also needs to standardize Yang modules to configure the Intent-Based Engine plus Yang modules for the storage of Context specific data (see figure x-x).

Initial concepts for IB-Nemo have been presented in IRTF's NFVrg and SDNrg to obtain initial review.

8. IANA Considerations

This draft includes no request to IANA.

9. Security Considerations

The security in a Intent-Based interface may require that most Intent-Based Networking operate across a secure transport security with encryption. However, some use cases (in-home only) or some limited data may allow an unsecured transport.

10. Informative References

[I-D.ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-06 (work in progress), June 2015.

[I-D.xia-sdnrg-nemo-language]
Xia, Y., Jiang, S., Zhou, T., and S. Hares, "NEMO (NEtwork MOdeling) Language", draft-xia-sdnrg-nemo-language-02 (work in progress), May 2015.

[I-D.xia-sdnrg-service-description-language]
Xia, Y., Jiang, S., and S. Hares, "Requirements for a Service Description Language and Design Considerations", draft-xia-sdnrg-service-description-language-02 (work in progress), May 2015.

[I-D.zhou-netmod-intent-nemo]
Zhou, T., Liu, S., Xia, Y., and S. Jiang, "YANG Data Models for Intent-based NEtwork MOdel", draft-zhou-netmod-intent-nemo-00 (work in progress), February 2015.

[RFC6541] Kucherawy, M., "DomainKeys Identified Mail (DKIM) Authorized Third-Party Signatures", RFC 6541, February 2012.

Author’s Address

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com