Network Intent Composition Proposal

Name

Network Intent Composition (NIC)

Repo Name

nic

Description

This Network Intent Composition project will enable the controller to manage and direct network services and network resources based on describing the "Intent" for network behaviors and network policies. Intents are described to the controller through a new NorthBound Interface, which provides generalized and abstracted policy semantics instead of Openflow-like flow rules. The Intent based NBI allows for a descriptive way to get what is desired from the infrastructure, unlike the current SDN interfaces which are based on describing how to provide different services. This NBI will accommodate orchestration services and network and business oriented SDN applications, including OpenStack Neutron, Service Function Chaining, and Group Based Policy. The Network Intent Composition function will use existing OpenDaylight Network Service Functions and Southbound Plugins to control both virtual and physical network devices. The Network Intent Composer will be designed to be protocol agnostic such that any control protocol can be used such as Openflow, OVSDB, I2RS, Netconf, SNMP etc.

One outcome of this project will be to satisfy the reference implementation of the ONF NBI Virtualizer Function being defined within the Open Networking Forum's NBI Working Group and to build a reference design for the ETSI NFV ISG, SDN Work Item.

Another key outcome of this project will be a solution for composing multiple Intent-Driven network policy requests from various SDN applications into a consistent set of programmed network actions, monitors, and responses. The goal is to solve the multiple-writers/multiservice-SDN problem with an intent-only distribution of the controller exposing intent as intermediate language.

This project is intended to leverage the expertise of the OpenDaylight community in order to rapidly expose an evolving set of services to the applications, including L4-L7 service functions, network end points, multi-tenant domains, security services, QoS and traffic engineering capabilities, and network infrastructure elements.

Goals

The proposal is to incrementally build from a small number of simple use cases to supporting virtually any imaginable use case. The project goals include:

- Creating an extensible NBI that can be used to provide simple, business-centric descriptions of network intent and automate the process of configuring the available devices and systems that implement the intent.
- Building advanced composition logic for identifying and resolving intent conflicts as well as optimizing the resulting rules for better resource utilization, network performance and minimizing state change (solving the multiple writer/multi-service SDN problem.
- Developing a modular, pluggable software architecture that makes it easy to extend the kinds of intent supported and the types of southbound protocols supported without having to comprehend all of the logic supporting different models.
- Embrace diverse community needs and seek help in inventing, designing and implementing an intent based system
- Develop and enforce strong use-ability requirements to ensure a solution that appeals to a broad set of use cases.
- Design the solution based on use cases prioritized within the diverse community of contributors.
- Work closely with other organizations including ETSI, ONF, and OpenStack to create an implementation consistent with the work they are doing on architecture and interface definition.

Scope

The proposed project will refine the following functional framework into a design and implementation within the OpenDaylight Controller architecture. Deliverables will include enumeration of the intended use cases and phasing of the implementation to address those use cases.

The Intent NB Interface The Intent NBI is exposed for Network orchestration systems, SDN Applications and Network Operators. It may be defined as RESTCONF, and/or Java API's. This extensible interface shall 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.

- The Intent Interface would provide abstractions supporting "programming" of intended service policies including:

- Multi-tenancy and a hierarchy of virtual networks independent of infrastructure
- Intents which can utilize attributes such as User, Device Type, Location, Application, Network Service, Network State, and Time.
- Intents which can be applied to addressable end-points to manage:
- Connectivity, Network access, Resource access, Service access
- QoS, CoS, Bandwidth, Latency, Jitter
- Diverse micro-flow relationships for services such as SIP, UCC, streaming video, and content optimization.
- Security and privacy
- Intents applied to Network Infrastructure and Network Service Functions to manage:
- Resiliency, scale-ability, function types, function locations
- Broadcast domains, addressing domains, controlled domains
- Diverse multicast and group based transport services
- Support for NFV SFC and advanced traffic steering policies
- Border functionality for attaching legacy network domains to SDN controlled fabric.

Intent Information Elements The Intent Composer uses various logical databases of information needed to affect the Intents. The following are examples of Intent Information Base and not a recommendation for the exhaustive list. Development of such detailed design is part of the work product that the project will create:

- The Network Service Intent DB maintains policies and information affecting Network Service Functions (virtual or physical). Network Service Functions include special function routers, firewalls, load balancers, WAN accelerators, NAT's, IPS/IDS/DPI's, DHCP servers, Domain Name Servers, Directory Servers, etc. The Virtualizer Function uses or exposes the APIs for CRUD operations on Network Service Functions and their attributes: locations, capabilities, groupings, and state.
- The End Point Intent DB maintains policies and information affecting end points. An End Point is an addressable network element, which can be a provider or consumer of data traffic within a network connection or communication session. Thus an End Point can be a business application running on a server and/or a personal device with its associated user. The Virtualizer Function uses or exposes the APIs for CRUD operations on End Points, groups of end points, and their attributes: user, business app, location, device, permissions.
- The Tenant Network Intent DB maintains policies and information for tenant based Network as a Service use cases such as the Openstack Cloud Operating System. A Tenant Network Agent uses or exposes the APIs for CRUD operations on tenant virtual network components: networks, subnets, ports, VTEPs, vSwitches. The ODL OVSDB and VTN components could provide the Tenant Network DB and agent for issuing Openstack Neutron Intents into the Intent Compilation engine.
- The Network Security Intent DB manages policies and information to affect network security intents. Network Security Intents affect End-Points, SDN applications, business applications, Network Service Functions, Controller components, and network operators. Network security intents for traffic include: inspection, remediation, encryption, etc. The network security intents control authentication, access, privacy of communications, audits, roles and privileges. The Virtualizer Function exposes the APIs for CRUD operations on these types security services.
- The QoS & Traffic Engineering Intent DB for managing policies and information regarding QoS automation for applications, End-Point sessions and other specific traffic classes (Elephant flows etc.) identified by the Network Operator. The Virtualizer Function exposes the APIs for session and flow QoS: CRUD, classification, marking and other operations on sessions or flows and their attributes: media element, user, CoS/QoS, bandwidth.
- The Network Infrastructure Intent DB maintains policies and information to affect the operating behavior of the physical infrastructure. These are often referred to as Network Operator Intents. These Intents include the definition of Policy Domains, L2 and L3 traffic domains, addressing domains, network resiliency, scalability, fault tolerant traffic engineering, firmware management, etc.

Intent Compilation The Intent Compilation Engine compiles Intent Language Policy Expressions into protocol specific rules. The initial phase of this project will create an Openflow Rules Compilation Engine and some utilization of SB protocols such as NetConf and SNMP for device specific configuration access. Future phases could encompass other SB protocol plugins such as I2RS, PIF, OVSDB, LISP, BGP, TL1, etc. to enable broader application of data plane controls.

The Intent Compilation Engine includes the following functions to ensure complete consistency of the data plane.

- Creation of specific Openflow rules for all affected devices
- Installation and maintenance of those rules across affected devices
- Policy and Rule conflict detection
- Policy and Rule conflict resolution and notification of irreconcilable conflicts
- Enforcement of policy prioritization
- Rule optimization:
- Combining multiple rules to optimize table space optimization
- Splitting rules where necessary to take advantage of hardware table pipelines

It is envisioned that the Intent Language defined by this project could utilize other SDN programming languages and policy models such as Maple, Frenetic, Pyretic, Flowlog, GBP etc. Thus this Intent Compilation Engine could take advantage of other policy compilers to extend its capabilities.

Resources Committed (developers committed to working)

- Duane Mentze – duane.mentze@hp.com ODL username: dmentze
- Devon Dawson - devon.dawson@hp.com
- Shaun Wackerly – wackerly@hp.com ODL username: wackerly
- Uyen Chau – uyen.chau@hp.com ODL username: uchau
- Rex Pugh – rex.pugh@hp.com ODL username: Rex.pugh
- Dave Lenrow – david.lenrow@hp.com
- Masashi Kudo – m-kudo@cw.jp.nec.com
- Hideyuki Tai - hideyuki.tai@necam.com ODL username: hideyuki
- Shigeru Yasuda - yasuda@uxd.fc.nec.co.jp
- Cathy Zhang –Cathy.H.Zhang@huawei.com
- Louis Fourie - louis.fourie@huawei.com
- Kanika Gupta - kagupta@ciena.com
- Mallik Kore - mkore@ciena.com
- David Bainbridge - dbainbri@ciena.com
- Mathieu Lemay - mlemay@inocybe.com ODL username: mlemay

- Colin Dixon - colin@colindixon.com ODL username: ckd
- Helen Chen - Helen.Chen@huawei.com
- George Zhao - george.y.zhao@huawei.com

Initial Committers

- Duane Mentze – duane.mentze@hp.com ODL username: dmentze
- Devon Dawson - devon.dawson@hp.com
- Shaun Wackerly – wackerly@hp.com ODL username: wackerly
- Uyen Chau – uyen.chau@hp.com ODL username: uchau
- Rex Pugh – rex.pugh@hp.com ODL username Rex.pugh
- Dave Lenrow – daveidlenrow@hp.com
- Masashi Kudo – m-kudo@cw.jp.nec.com
- Hideyuki Tai - hideyuki.tai@necam.com ODL username: hideyuki
- Shigeru Yasuda - yasuda@uxd.fc.nec.co.jp
- Cathy Zhang –Cathy.H.Zhang@huawei.com
- Louis Fourie - louis.fourie@huawei.com
- Kanika Gupta - kagupta@ciena.com
- Mallik Kore - mkore@ciena.com
- David Bainbridge - dbainbri@ciena.com
- Mathieu Lemay - mlemay@inocybe.com ODL username: mlemay
- Colin Dixon - colin@colindixon.com ODL username: ckd
- Helen Chen - Helen.Chen@huawei.com username: helenc878
- George Zhao - george.y.zhao@huawei.com ODL username: gzhao

Vendor Neutral

This project will consist of contributing some existing code as well as developing new code. Code will be made available for review by ODP and Linux Foundation after it has been approved by contributing organizations.

Meets Board Policy (including IPR)