NEMO: Apply Intent for Service Level Network Programming

Helen Chen, Huawei
Pedro A. Aranda Gutiérrez, Telefónica I+D
Tianran Zhou, Huawei
Contents:

• Intent Values
• NEMO Overview
• NEMO Implementation for Intent
• Conclusions
# Application Requirements to Network

## Application

- Discover devices
- Assign interface IDs
- Automatic Configuration
- BYOD
- Centralized TE
- Virtual Network
- Service Chain

## Infrastructure

- Network device
- Network device
- Network device
- Network device

## Automation

- Detect device, link, node failures
- Manage encapsulation / decapsulation
- Optimize path selection
- Maintain network object state variables

## Flexibility

- Easy to Use

## Availability

- Scalability
The Problem

- Complexity
  - Is enemy of scalability
  - Slows down network innovation
  - Is error prone
  - Loses application intent
Requirement for the Declarative NBI

- Declarative network middleware
  - Hides complexity
  - Captures application intent
  - Translates what applications “want to do” to “how to do it”
  - Resolves conflicts
Narrow Waist for Network Service Customization

Transport and Application Protocols

- email
- WWW
- phone
- SMTP
- HTTP
- RTP
- TCP
- UDP
- IP
- ethernet
- PPP
- CSMA
- async
- sonet
- copper
- fiber
- radio

Transmission Media

Non-convergent

Convergent

Non-convergent

Various Network Services

- Multi-site connecting
- Safe Internet (SFC)
- BoD
- QoS for Video Call
- BGP
- OSPF
- RSVP
- MPLS
- vTEP
- ACL
- PBR
- ... Network Primitives

Network resource and function “atomic” models, convergent, platform and technique neutral, upper service and underlying implementation agnostic, easy to use and compose
LEGO Style Network Service Composition Using Atomic Network Primitives

Service Abstraction

Service Modeling

Service Composition

Service Delivery like Building Block

LEGO-Style Service Orchestration

NFV Module  WAN Module  DC Module

Service Composition  Customer Requirement
Contents:

• Intent Values
• NEMO Overview
• NEMO Implementation for Intent
• Conclusions
**Intent Model** is essential. NEMO uses YANG to describe the model and drive the Model Driven Architecture (MDA).

**NEMO Language** is an Intent oriented network DSL (domain specific language), which is a language style network open interface. Operator/End-User or 3rd Party can use it to program network resources and behaviors in their service applications. It is encapsulated in RESTful API to the engine.

**NEMO Engine** is a network middleware, which translates high level service intents to real network instructions base on MDA.

**Supporting tools** such as the NEMO Sandbox facilitates testing, simulation, and deployment.

**NEMO APP**

- **Object Model**
- **Behavior Model**

**NEMO Engine**

- CFN
- Connection
- ServiceFlow
- Operation
- Result

**SDN Controller/Orchestrator**

15 simple statement for hundreds of complex APIs

**NEtwork Modeling (NEMO) for Intent Based Network Programming**

Supporting tools such as the NEMO Sandbox facilitates testing, simulation, and deployment.
NEMO Methodology

Network Abstraction → Network Primitive → Service Level Programming → Compiling and Deployment
Network User Intent Expression

• Two types of network user’s Intent Expressions:
  – Result:
    • Intent expression to describe object’s final result, state, etc.
    • I (do not) want/expect object \{XXX\} would be state \{YYY\}.
  – Operation:
    • Intent expression to describe operations to do.
    • I want to perform operation \{YYY\} on object \{XXX\}.

• Intent examples:
  – “I want to create a Network for HR (Customer Facing Node)”
  – "I want to block the http flow (Flow)"
  – "I want to adjust the bandwidth (Connection) to 10G“
  – “I want to avoid the bandwidth utilization on the connection greater than 80%. (Result)“
NEMO OOR model is highly abstracted from network resources and behaviors for different kinds of network service scenario requirements.
NEMO Models Illustration

CFN

Result

Connection

Service Flow

Operation

Service Chain

vDC

Internet

Network utilization less than 80%

1 G on daytime
10G at night

Deny

Allow

BoD
Contribution to Communities

• Since the beginning, the NEMO project has continued to actively contribute to standard development organizations and open source projects.

• In ONF and IETF, NEMO has contributed to:
  – “Narrow Waist” intent models as common intent NBI
  – Role based intent expressions

• In major open source controller projects, NEMO has contributed to:
  – DSL based intent models.
  – Reference implementation of an intent engine.
Contents:

• Intent Values
• NEMO Overview
• NEMO Implementation for Intent
• Conclusions
NEMO Language

**Resource Access**

<table>
<thead>
<tr>
<th>Entity Model</th>
<th>Node</th>
<th>Node/UnNode</th>
<th>entity_id</th>
<th>Type</th>
<th>Owner: node_id</th>
<th>Properties: key1, value1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>connection/Unconnection</td>
<td>entity_id</td>
<td>Endnodes</td>
<td>(node1_id, node2_id)</td>
<td>SLA</td>
<td>key, value</td>
</tr>
<tr>
<td>Flow</td>
<td>Flow/UnFlow</td>
<td>entity_id</td>
<td>Match/UnMatch</td>
<td>key1, value1</td>
<td>Range(value, value)</td>
<td>Mask(value, value)</td>
</tr>
</tbody>
</table>

**Policy and Event Handling**

<table>
<thead>
<tr>
<th>Capability Model</th>
<th>Query</th>
<th>Query</th>
<th>key</th>
<th>Value</th>
<th>{value}</th>
<th>From</th>
<th>entity_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification</td>
<td>Notification entity_id</td>
<td>Notification</td>
<td>entity_id</td>
<td>On key</td>
<td>Every period</td>
<td>RegisterListener</td>
<td>callbackfunc</td>
</tr>
</tbody>
</table>

**Model Definition**

<table>
<thead>
<tr>
<th>Node definition</th>
<th>NodeModel &lt;node_type&gt;</th>
<th>Property</th>
<th>{&lt;data_type&gt;: &lt;property_name&gt;}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>ConnectionModel</td>
<td>Property</td>
<td>{&lt;data_type&gt;: &lt;property_name&gt;}</td>
</tr>
<tr>
<td>Action</td>
<td>ActionModel</td>
<td>Parameter</td>
<td>{&lt;data_type&gt;: &lt;property_name&gt;}</td>
</tr>
</tbody>
</table>

**Simple syntax**
- 3 primitive groups
- 15 statements
- 36 Keywords

**Programming rather than configuration**
- Function Reusable
- Variable Object operation
- Logic Intent composition

**Infinite programming capability**
- Resource Definition
- Service Programming
- Policy Composition
Using NEMO Language to Create A Hybrid Cloud

IMPORT Node enterprise Type ext-group Property location:"openflow:4:2",ac-info-network:"layer3",ac-info-protocol:"static",ip-prefix:"192.168.13.0/24";
IMPORT Node internet Type ext-group Property location:"openflow:3:4",ac-info-network:"layer3",ac-info-protocol:"static",ip-prefix:"172.168.1.0/24";
CREATE Node dmz Type l2-group Property ip-prefix:"192.168.11.0/24",gateway-ip:"192.168.11.1",location:"openflow:3";
CREATE Node interior Type l2-group Property ip-prefix:"192.168.12.0/24",gateway-ip:"192.168.12.1",location:"openflow:3";

CREATE Connection c1 Type p2p EndNodes enterprise,interior;
CREATE Connection c2 Type p2p EndNodes interior,dmz;
CREATE Connection c3 Type p2p EndNodes dmz,internet;

CREATE Operation o1 Priority 2 Target c1 Condition time>9&&time<18 Action qos-bandwidth 2048 M;
CREATE Operation o2 Priority 1 Target c1 Action qos-bandwidth:1024 M;
From Intent NBI to Service Template

- Education domain user
- Financial domain user
- Web service provider
- Big data system
- Game provider

End user selects

- BOD Template
- HA Template
- Cloud Template
- Big Data Template
- Realtime Template

Face to specific user scenario

Business designer and IT developer

Develop specific service template

use a set of consistent intent primitives to compose service templates

NEMO Language (Network Atom)

NEMO Lib
- VN/VTN
- Service Chain
- BOD
- Customized Forwarding
- Security Group
Accelerate the Template Customization

Watch our Demo at the Huawei Booth

Scenario

Branch A

Branch B

Headquarter

Describe user intent as template.
Apply a template to deploy the service.
NEMO Engine Framework

Operators input

Context information for intent vocabulary mapping

Role-based configuration:
- Business Designer
- OAM Manager
- Infrastructure Designer

Tenants’ Intent

- Each tenant will have a separate virtual network space for self-service resource definition and operation.
- Tenant space isolation;
- Tenant can not see the underlay topology

Level 1 Compiling/Service Building

Language compiling and virtual network generating based on service requirements.

Level 2 Mapping/Resource Placement

Virtual Network to Physical Network mapping.

Level 3 Translation/Configuration Render

Using renderer to generate real device configurations

Sandbox Debugging

Real Network Provisioning
NEMO Engine Implementation

Intent Engine
- Object Mapping
  - CFN Mapper
  - Connection Mapper
  - Flow Manager

Operation Resolver
- Condition Monitor
- Constraint Resolver
- Action Resolver

Result Monitor

Query & Notification Handler

Generic Data Store and Event Bus

RESTful API

GUI/External Applications

Language Parser
- Tenant Manager
  - Tenant Auth.
  - Transaction
- VNSpace Mgt

Physical Network Manager
- Device Manager
- Link Manager
- SP Compliance

Tenant Manager

Result Monitor

Renderers
- OpenFlow Renderer
- CLI Renderer
- 3rd-Part Renderer

Implemented

Next Step
Basic Use Case 1: Data Center Interconnection

- The enterprise wants to inter-connect their multiple sites with layer 2/3 connections.
- The basic connection types include
  - point to point, point to multi-point, and the full mesh.
- It is also desired that the user can construct the topology freely by composing the basic connection types.

Diagram:
- a) Point to Point
- b) Point to Multi-Point
- c) Full mesh
Basic Use Case 2: Bandwidth on Demand

- The enterprise needs a connection between the branch and the headquarter offices. There is requirement to adjust the bandwidth of the connection on demand.
  - The adjustment can be triggered when "conditions" are met. For example, the bandwidth will be adjusted when the timing condition meets. At night, more bandwidth is required for backup traffic. The bandwidth adjustments can also be triggered by traffic bursting.
- This use case can also be generalized to more resource (node, connection) reservation policies.

Condition: From 8am To 6pm
Bandwidth: 10G
Condition: From 6pm To 8am
Bandwidth: 1G
Basic Use Case 3 : Service Chaining

- A virtual private cloud or an enterprise site wants to apply several virtual network functions, such as firewall, load balancer, WAN optimization before accessing the internet.
- The operations include:
  - Identifying flows to be applied for VNFs. e.g. outgoing flows, incoming flows
  - Describing connections with chained service functions.
  - Applying policies to steer flows to go through different service paths
- The intent based service chaining should be topology agnostic, and only show information that users need.
Basic Use Case 4: Virtual Private Cloud

- Operator provides virtual private cloud services for enterprises. A typical architecture is to allocate two zones in the vDC for security issues.
  - DMZ provides http/email/video services to be accessed from internet.
  - Interior provides compute and storage resources for internal use.
  - Internet access cannot reach the interior.
  - User can also describe connectivity to exterior, the internet and the existing site.
Compound Use Case: Hybrid Cloud

- A hybrid cloud is a typical cloud configuration for enterprises to combine the existing site and new virtual private cloud in a public cloud center. This use case is usually a combination of virtual private cloud, data center inter-connection, service chaining and bandwidth on demand.
  - For secularity purposes, the vDC is usually separated into two zones.
  - There will be a service chaining service between the internet and DMZ.
  - The interior zone provides compute and storage resources for existing enterprise site, which might be a private cloud.
  - There will be requirement to dynamically adjust the connection between the existing site and the vDC, so that bandwidth could satisfy at cloud bursting or backup scenarios.
Conclusions

- Intent is important to network service level programming
- NEMO provides a set of intent implementation, including language style interface, intent data model, intent engine and supporting systems
- NEMO suits well with the NetIDE use cases
  - Simple, human-readable way to express the relationships between different SDN applications
Thank you

Copyright©2015 Huawei Technologies Co., Ltd. All Rights Reserved.
The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

The NetIDE project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619543.