Evolving towards Software Defined

Eliminating barriers to adoption of IT Innovation
Agenda

• 3 Keys to enabling innovation and transformation
  › Programmable data plane
  › Vertical disaggregation of network solutions
  › Implementation independent integration language

• Intent based networking

• Potential applicability of IBN to Platform Lab’s BCP project

• Unsolved Problems we need to solve: System architecture, scaling,
Something new? Too Scary!

- Millions of dollars of integration and capital equipment cannot be undone if it’s a disaster.
- Must change both hardware platform and operating software simultaneously.
- Changes to operations, training, processes too disruptive
- Can't succeed until risk is reduced and clear benefit is identified

Dangerous Road. Can't Backup!
The Big Three Barriers to network innovation

1. Data Plane is Not Programmable
2. Integration work is Entirely Implementation Specific
3. Solutions are vertically integrated
Data Plane Programmability is coming

- Many startups and established vendors working on this
- New instruction sets and languages (e.g. P4) allow downloading of new “wire-speed” features to deployed device.
- Designers can choose to deploy new logic in device or controller
Disaggregation can enable risk-free changes

Vendor x User Interface
Vendor x Management
Vendor x Operating Software
Vendor x Devices
Vendor x Silicon

Vendor a User Interface
Vendor b Management
Vendor c Operating Software
Vendor x Devices
Vendor y Silicon
Intent: Model the application, not the network

• User creates implementation independent description of what applications need from the network.
• Users describe what they need in terms they understand
• Automation and Experts help guide the translation to terms the provider can fulfill
• Cost and risk of trying or changing solution components becomes minimal.

Intent: Take me from A to B
Review of IBN concepts and development status

INTENT BASED NETWORKING
Why Intent?

1. Eliminate “Test Drive” cost and risk
2. Eliminate Vendor Lock-In
3. Make Solution Components Fungible
4. Enable “programming the network” for Non-Experts
5. Allow Write-once, Run-anywhere Infrastructure Integration
6. Support Dynamic Behaviors of Network Applications and Resources
### Intent-based Operating Model

- Describe the problem
- Model the Workload requirements
- Tell me what you need
- Make my headache stop
- I need a virtual network (logical isolation) for VMs 1, 2 & 3
- 99% of network “users” only have to understand their business and workloads

### Traditional Network Operating Model

- Describe the solution
- Model the Network
- Tell me what to do
- Give me an aspirin
- I need, e.g., VXLAN tunnels, full L2 mesh between VMs 1, 2 & 3,
- 100% of network “users” need to be experts in networking as well as their business and workload verticals.
Intent is a Virtual World
Label-mapping Makes It Real

• Elastic, infinite, extensible, reliable, available, simple. No corner cases 😊

• Intent relationships can be described between virtual Objects and Object groups

• Intent statements apply run-time extensible set of modifiers and predicates to relationships between objects/groups

• You don’t get to specify or touch underlying resource pool

• Extensible Framework: Add one use case at a time
Goal: Unifying Common NBI Shim

- OpenStack
- OPEN-O
- OPNFV
- OPEN MANO
- OSM
- MEF LSO
- IMTC UCC
- Congress
- OSSDN Atrium
- CORD
- ODL NEMO
- ONOS & ODL NIC
- ODL GBP
- ONOS Intents
- VTN
- Transport SDN
- Other…

COMMON INTENT NBI
Narrow Waist Interoperability Demarcation

- VAS controllers
- Compute controllers
- Orchestrators
- VFM
- Analytics
- HPC clusters
- OSS/BSS
- LSO

SaaS/PaaS/IaaS

- Vendor SDN Controllers
- Legacy Equipment Adaptation
- Open Source SDN Controllers

Other

COMMON INTENT NBI
## Any Use Case That Can Be Described Can Be Split Into Intent+ Mappings

<table>
<thead>
<tr>
<th>Intent</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Changes in management-plane time, human time, and minutes/hours</td>
</tr>
<tr>
<td>• Does not change based on state of network, endpoints, users.</td>
</tr>
<tr>
<td>• Independent of protocol, media, vendor, etc.</td>
</tr>
<tr>
<td>• Easily understood and authored by non-experts</td>
</tr>
<tr>
<td>• Simple test to determine whether desired state is portable enough to be intent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Changes in control-plane time, real-time, and sub-second</td>
</tr>
<tr>
<td>• Changes whenever the state of the network or resources changes.</td>
</tr>
<tr>
<td>• Specific to resolving abstract intent to protocol, media, vendor, etc.</td>
</tr>
<tr>
<td>• Requires deep understanding of technology, networks, etc.</td>
</tr>
</tbody>
</table>
The Intent Killer App – Solving the Multi-Writer Problem

SDN Apps That Render Openflow

- UCC Domain Logic
- Streaming Media Domain Logic
- Flow Rule Logic
- Network State
- Topology
- Inventory

- OpenFlow Multiplexor
- Multi-Writer Conflict
- Forwarding Table

SDN Apps That Push Intent

- UCC Domain Logic
- Streaming Media Domain Logic
- Intent Media Logic
- Conflict Resolution
- Flow Rule Logic
- Network State
- Topology
- Inventory

- OpenFlow Multiplexor
- Forwarding Table
Intent Based SFC/NFV

Add 10.1.1.23, 10.1.1.24, 10.1.1.25 to Firewall-xyz object group (pool).
Intent Based System Architecture

Intent Plane

Intent NBI

Mapping NBI

Intent Engine

Platform independent

Platform dependent

Sync

Intent NBI Mapping NBI

Sync

Intent Repository (IR)

Mapping Repository (MR)

SDN Controller Infrastructure (ODL/ONOS)
External sources of truth feed real-time mapping
NBI Specifics - Intent NBI Atoms

- Object
- Object Group
- Modifier

RESTCONF CRUD operations on above items
YANG model based

- Intent objects and their relationships form graph
- Graph theory can be applied for resolving aggregate requirements, config, minimal update, multi-path routing, etc.
Intent and BCP

• BCP will control a superset of systems that includes network and cloud computing infrastructure (in order to further support MEC applications for drones, robots, autonomous vehicles, etc.)

• It makes sense to build this next generation automation/control system using an intent based interface
  › ONF intent NBI work is becoming de facto standard way to interface with network controllers supported by work in ONF, ODL, ONOS, etc.
  › Architectural benefits including modularity, composability, portability, future-proofing, migration enabling, multi-vendor, controller agnostic, protocol agnostic, etc., etc.

• CRI would like to explore working with platform lab to solve some of the problems that we know stand between our current prototyping, and a deployable hyper-scale control system.
Sample Design and Research Problems

• How do we balance the centralized/global state sharing versus the distributed/local state.
  › Fully autonomous won’t work. Fully Centralized won’t work.

• We have stated that Intent is global in nature, and changes relatively slowly (e.g. human/policy timescale)
  › We can replicate this slow changing, low volume data at massive scale

• Much of the rendering logic will be pushed out to small, shared-nothing intent domains each with a smaller number of objects and devices to control. The system scales-out linearly to the extent we are able to live with shared-nothing

• We need a way to efficiently, coherently distribute the bare minimum of shared state information.
Transit Path Advertisement and Scheduling

- End-to-end deployment of intent can and will cross multiple disjoint intent domains.
- Some higher level (logically centralized cooperative intelligence) logic within the intent stack must understand the available ingress/egress paths available for stitching together end-to-end service behaviors across multiple otherwise autonomous, shared-nothing domains.
- Exactly how does an intent domain advertise any/all interconnecting network interfaces with adjacent intent domains. What resources capabilities need advertising and how are they interpreted by central logic.
- Do we need one or more additional controller-of-controllers layer to stitch end-to-end across these meta-domains for max scale?
- Looks like BGP with constraint routing problem space, but needs new solution? Fast-reconvergence based on global view “memory”?
E2E Path Computation Across Intent Domains
Mobile Edge Service scheduling and State Handoff

- Lower Compute Cost, Higher Latency
- Pre-provision service image
- Skip Edge Nodes if Speed > X?
- Transfer State on handoff

State Service O.S. Container Virtual Machine
Mapping Service Replication, Compression, Synchronization

- First cut can make great progress with OTS distributed key-value stores and dense state exchange.
- Ultimate scale will require optimized, multi-path aware transactional systems and sparse/summarized state exchange.
- Need to invent, model, simulate, prove techniques to achieve global telco and web scale.
Minimal update to global rendering

- When a change occurs to the state of infrastructure, intent or mappings, the intent engine has to compute and push new rules to adapt the network to the new combination of inputs.
- The naïve implementation recomputes everything from scratch, possibly resulting in massive thrashing of traffic in-flight with resulting dropped sessions, etc.
- The problem that needs to be solved is to build a rendering engine that can generate assembly-language (e.g. openflow rules) for many network devices at scale in response to state changes that minimize the disruption to the existing state of rules satisfying the aggregate end-to-end requirements.
Power of implementation-agnostic Model

**Intent Data**
- Portable. Implementation/state independent
- Scale-able. Compact, global metadata
- Compose-able. Common, general model
- Understandable. No army of experts
- Secure-able. No flow-tables, topo, inventory
- Write-once, run anywhere
- Future-proof. No more integration expenses here

**Mapping Data**
- Changes with platform, infra, state
- Fast changing, locally meaningful
- Segmented, per domain
- Requires implementation expertise
- Exposes more powerful abstractions
- Maintenance per-implementation
- Remaining subset that changes as you operate or move platforms.
Intent Levels The Playing Field

Vendors Who Can Compete on Price/Performance/Innovation Win

Operators are primary beneficiaries

Network Effect drives ecosystem

Virtuous cycle of vendors supporting IBN and operators asking for IBN
Thank You

• Next Question:
  How can we collaborate and contribute to solving these problems in BCP context?
Simple Connectivity Use Case: Bob’s Internet

Labels not understood by intent syntax resolved by mapping service

Bob is ww.xx.yy.zz, Internet is <complicated Wildcard expression>

Application

Facebook Information Service (Label Manager)

Mapping NBI

Intent Engine

Lookup bob?
Bob is ww.xx.yy.zz
Lookup Internet?
Internet is <complicated Wildcard expression>

Bob is allowed to access the internet

Labels not understood by intent syntax resolved by mapping service
IBN reduces SDN Attack Surfaces

“Fine Grained” NBIs Exposed

Common Intent NBI Exposed

What I need
Intent Levels The Playing Field

Vendors Who Can Compete on Price/Performance/Innovation Win

Operators are primary beneficiaries

Network Effect drives ecosystem

Virtuous cycle of vendors supporting IBN and operators asking for IBN
OSSDN Boulder – Intent Demarc

OSSDN Boulder

Intent Engine

Label Manager (Mapping)

NFV VFMs

UCC Call Controllers

DSL Compilers/Interpreters

Vcenter Connector (MQ bus)

IPAM Connectors

OpenStackCongress Connector

Inventory

Flow rules

Routes

Topology

Link State

Intent Repository
Over-Prescription Yields Fewer Solution Choices

Hey Bro. I have a terrible headache. Do you have any aspirin?

Sorry Pal. No Aspirin.
Any Use Case That Can Be Described Can Be Split Into Intent+ Mappings

<table>
<thead>
<tr>
<th>Intent</th>
<th>Mapping</th>
</tr>
</thead>
</table>
| • Changes in management-plane time, human time, and minutes/hours  
• Does not change based on state of network, endpoints, users.  
• Independent of protocol, media, vendor, etc.  
• Easily understood and authored by non-experts  
• Simple test to determine whether desired state is portable enough to be intent | • Changes in control-plane time, real-time, and sub-second  
• Changes whenever the state of the network or resources changes.  
• Specific to resolving abstract intent to protocol, media, vendor, etc.  
• Requires deep understanding of technology, networks, etc. |
**Intent Based Service Function Chaining**

- **Network Manager**
  - Create(Internet)
  - Create(Sales)
  - Add(Internet, 10.1.1.25, 10.1.1.26)
  - Add(Sales, 10.1.1.27, 10.1.1.28)

- **Service Policy Manager**
  - Service_A=Path(Sales,Firewall,Internet)
  - *Service_A=Path(Sales,NULL,Internet)*

- **VNF Manager**
  - Create(Firewall)
  - Add(Firewall, 10.1.1.23, 10.1.1.24)

- **Mapping**
  - Read(Firewall)
  - (10.1.1.23, 10.1.1.24)
  - Read(Sales)
  - (10.1.1.27, 10.1.1.28)
  - Read(Internet)
  - (10.1.1.26, 10.1.1.27)
SDN Controller “Intent Engine”

Mapping Info Source System

Mapping

Mapping Lookup Index (Write)

Mapping Lookup Result (Write)

Map_Gen API

Mapping

Intent Engine

Consumer Service Manager

Intent NBI

Mapping Lookup Index

Mapping Lookup Result

Map_Read API

SDN Controller

Instruct Network

Intent NBI Handler

Controller-Specific SBI Handler

Map_Read API Handler

Intent Active Loop

Intent & Mapping Repo
Getting From Consumer model to producer model

External Systems

Consumer-Provider Agreements

Dynamic Updates

Mapping Sources

Mapping

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
</table>

Consumer Terms

Provider Terms

Standards

Configure Resources

Consumer

Provider (e.g. SDN controller)

External Systems

Lookups

Standards

Mapping Sources

Mapping (e.g. SDN controller)
Consumer-provider interactions using Intent NBI

Consumer (System)

Intent NBI: Requests for Network Service (Restricted to “What”), Notifications

Knows Service Needs (“What” and “Why”)

Provider (System)

Determines Service Delivery (“How”)
Architectural representation of Intent NBI and mapping

- **Consumer**
- **Provider** (e.g. SDN controller)
  - **Mapping**
    - Translates consumer terms to provider terms
  - **Configure Resources**
  - **Mapping Lookups**
  - **Intent NBI**