

Demo by ORCA-BEN Project Team on July 7, 2009

Prepared by GPO (Harry Mussman, Nidhi Tare and Henry Yeh) and ORCA-BEN project team on July 8, 2009.

Contents

1. Abstract	1
2. Participants.....	2
3. Demonstration Goals	2
3. BEN Substrate.....	4
4. ORCA Actors, Calls and Containers.....	7
5. ORCA Structure for Demo	9
6. ORCA Calls and BEN Connectivity.....	12
7. Slice Creation.....	15
Options.....	15
Summary	15
Sequence	16
Actor Actions.....	17
8. Resource Descriptions	18
9. GENI Milestones Completed	18
10. Extra Demo: MIN Visualization Engine Demo.....	18
11. Extra Demo: RENCI-P FPGA OpenFlow Project Demo	20

1. Abstract

On July 7 2009, the ORCA team demonstrated to the GENI Project office the ORCA capability to create slices of substrate that include several transit providers, i.e. BEN and NLR. See <https://geni-orca.renci.org/trac/wiki/events/orca-nlr-july-2009>

The demo created several VMs at Duke, RENCI and UNC-CH BEN PoPs and then linked them together with VLANs via NLR and BEN.

NLR VLANs were statically provisioned ahead of time via Sherpa.

BEN VLANs were created dynamically, and included provisioning of Infinera DTN circuits as well as fiber paths using Polatis switches.

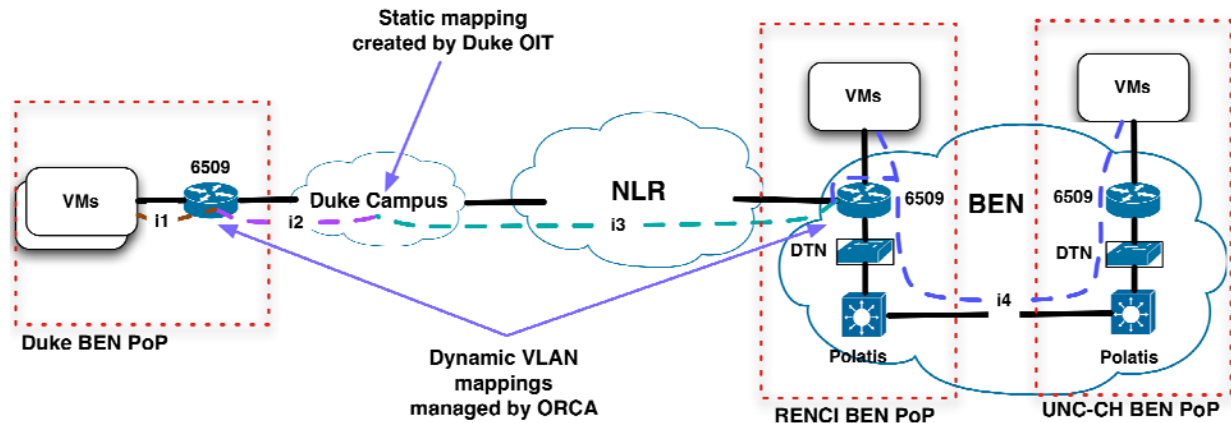


Figure 1-1 Demo Connectivity

2. Participants

Ilia Baldine (RENCI) Co-PI

Jeff Chase (Duke) Co-PI

Aydan Yumerefendi (RENCI Contractor) - policy development, integration, build system and a whole lot more

Yufeng Xin (RENCI) - NDL schema, substrate description, java NDL toolkit, testing

Dan Evans (RENCI) - Network element drivers (Cisco 6509, Infinera DTN, Polatis)

Varun Marupadi (Duke) - SOAP-based distributed container debugging, integration/testing support, stitching (on-going work)

Matt Sayler (Duke) - Eucalyptus integration

Chris Heermann (RENCI) - BEN operations, MIN visualization back-end

Xunlei Wu (RENCI) - MIN visualization front-end

Henry Yeh (GPO) Project Manager

Harry Mussman (GPO) System Engineer

Nidhi Tare (GPO) Intern

3. Demonstration Goals

Demonstrate provisioning of VLANs over a national backbone (NLR) and a regional network (BEN).

Integration of ontology/NDL-OWL as a resource description mechanism.

Demonstration of cross-layer provisioning, with ability to control different types of network elements at BEN PoPs:

- Cisco 6509

- Infinera DTN

- Polatis fiber switch

In more detail, demonstrate the following ORCA capabilities:

- Ability to create slices out of slivers from more than one network substrate provider
 - BEN and NLR
- Ability to create slices using statically provisioned segments
 - NLR tags are statically provisioned
- Cross-layer slice provisioning
 - VLAN tags, OTN circuits and fiber paths
- Multiple network element type integration
 - Cisco 6509 routers, Infinera DTN platforms, Polatis fiber switches.
- NDL integration
 - A slice controller uses an NDL description of the desired slice. BEN site authority uses NDL BEN substrate description.

Show that GENI Milestones have been met:

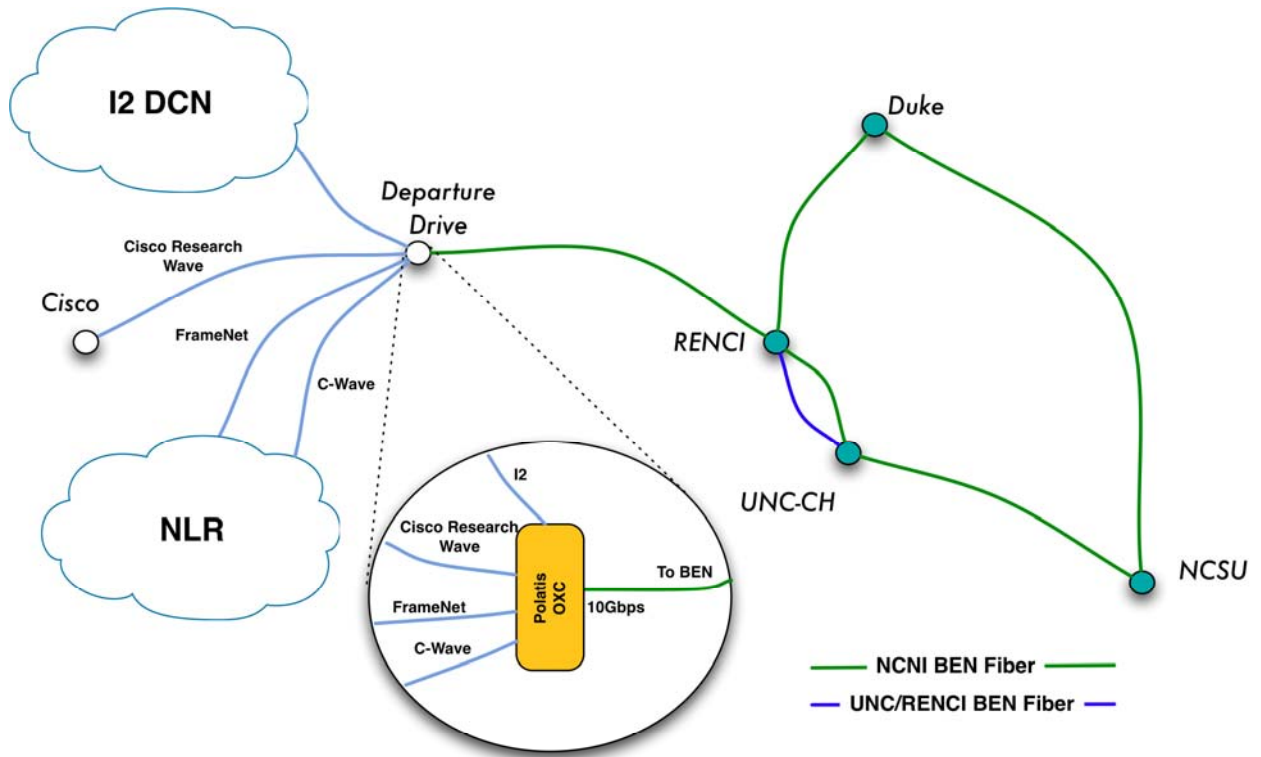
[1c Integration of NDL or other topology description mechanism](#)

[1d Integrate ORCA with BEN including slivering of Infinera DTNs](#)

[1e Complete end-to-end VLAN connections between BEN and NLR](#)

3. BEN Substrate

BEN (Breakable Experimental Network) is an experimental optical network that has been in place for approximately a year. Its purpose is to serve as a testbed for optical networking experiments.



BEN proposed connectivity diagram

Figure 3-1 BEN Topology and Connectivity

BEN Point of Presence (PoP)

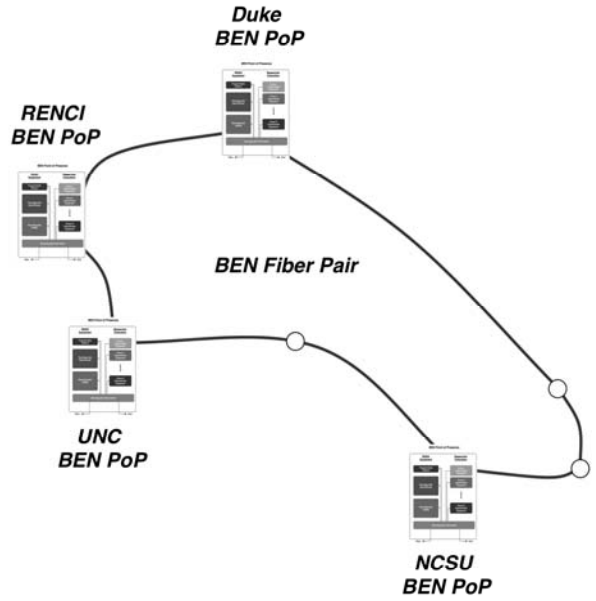
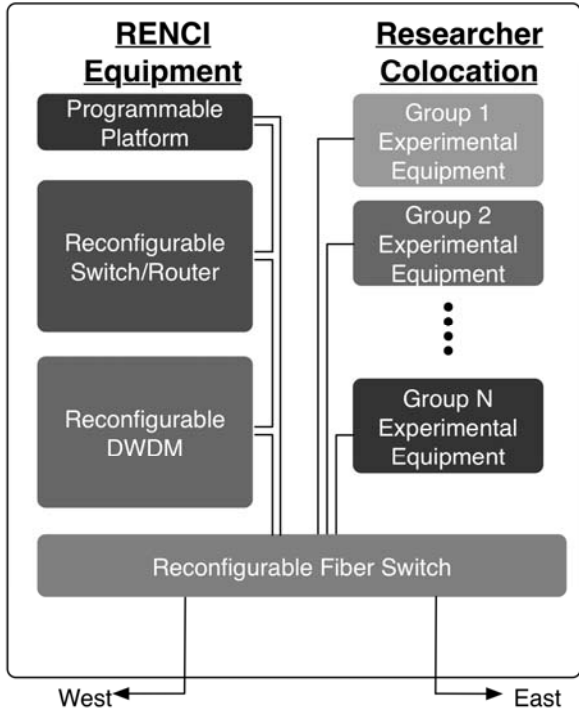


Figure 3-2 BEN PoPs

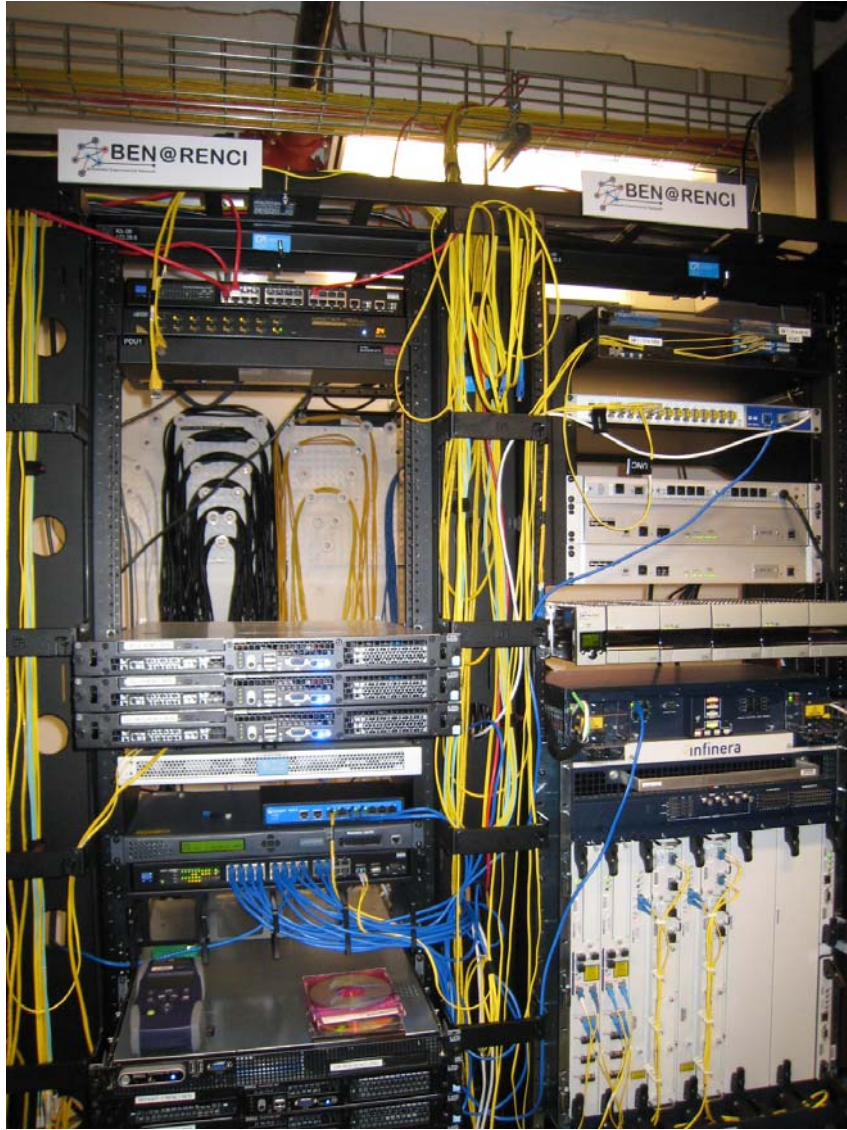


Figure 3-3 BEN PoP at RENCI

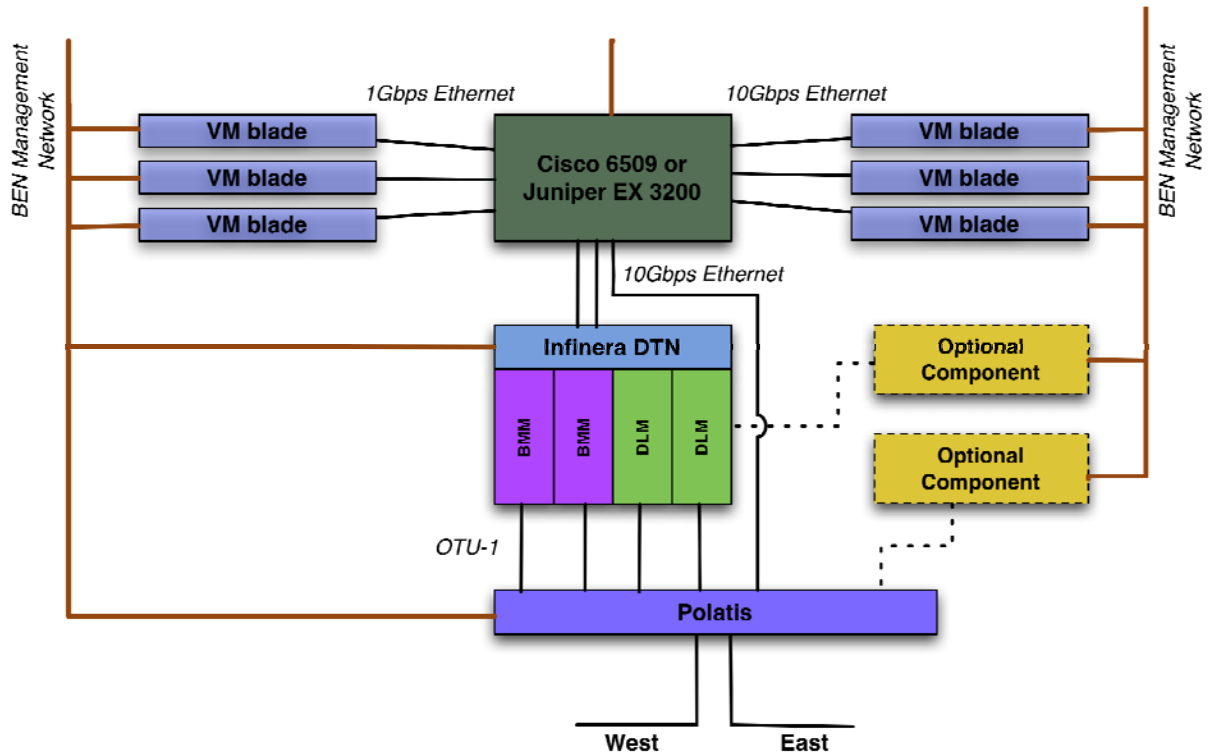


Figure 3-4 BEN PoP Structure

4. ORCA Actors, Calls and Containers

ORCA is a control structure based on the use of three types of actors, that are represented by JAVA objects that run in containers:

- Authorities
- Slice Controllers
- Brokers

Authority actor:

- Controls some portion of substrate for an edge site OR network domain
- Under a "Management Authority" (MA)
- Partitioning of MA-owned substrate across its aggregates/authorities is fluid.
- Authority has power to manipulate substrate components under its control.

Slice controller actor:

Properly speaking, the actor is a “service manager” and the controller is a plugin. We sometimes blur the distinction...

Runs on behalf of experimenter.

Unprivileged, modifiable by user (in principle)

May be provided as an extensible service, or launched by experimenter.

Broker actor:

Properly speaking, a broker is a service within a clearinghouse, e.g., a “ticket granting service” or engine

Accountable delegation is a core element of security architecture.

Broker has only the powers delegated to it from substrate providers.

Brokers assist with resource discovery and coordinated arbitration across providers.

RPC calls between actors allow the authorization and assignment of resources from Substrates to Researchers:

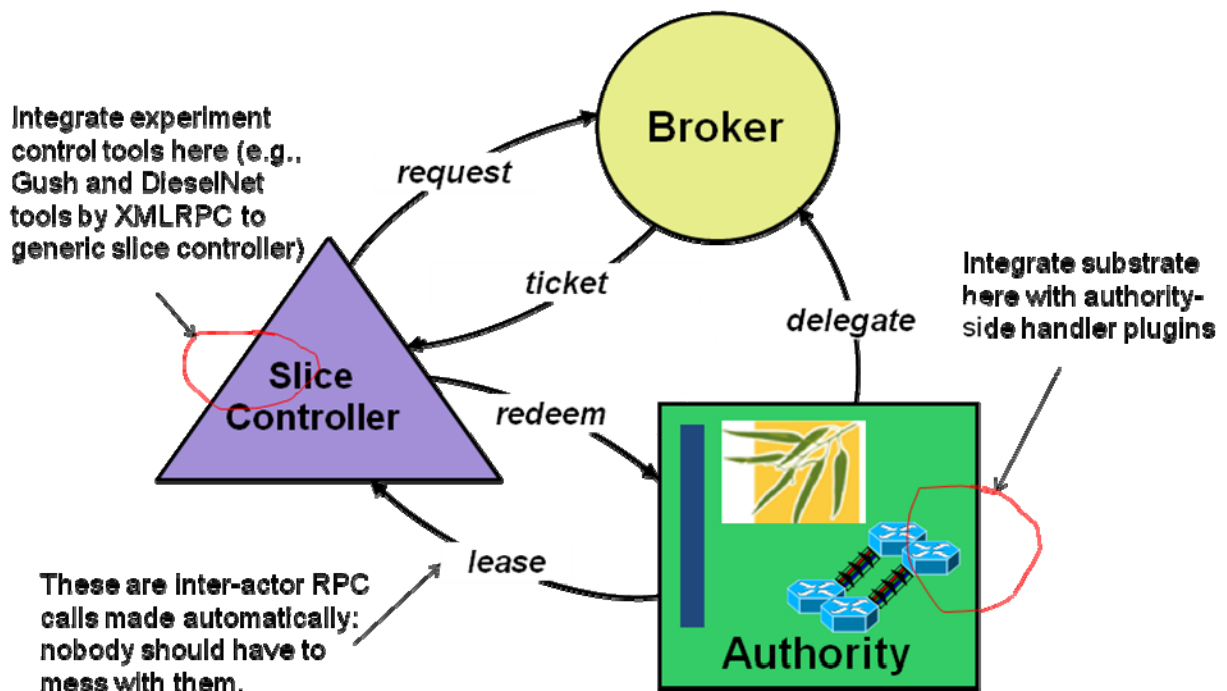


Figure 4-1 ORCA Basics

Actors run within containers

JVM, e.g., Tomcat

Per-container actor registry and keystore

Per-container mysql binding

Actor management interface
Useful for GMOC and portals
Not yet remoteable

Portal attaches to container
tabs for different actor “roles”
Dynamic loadable controller plugins and views (“Automat”)

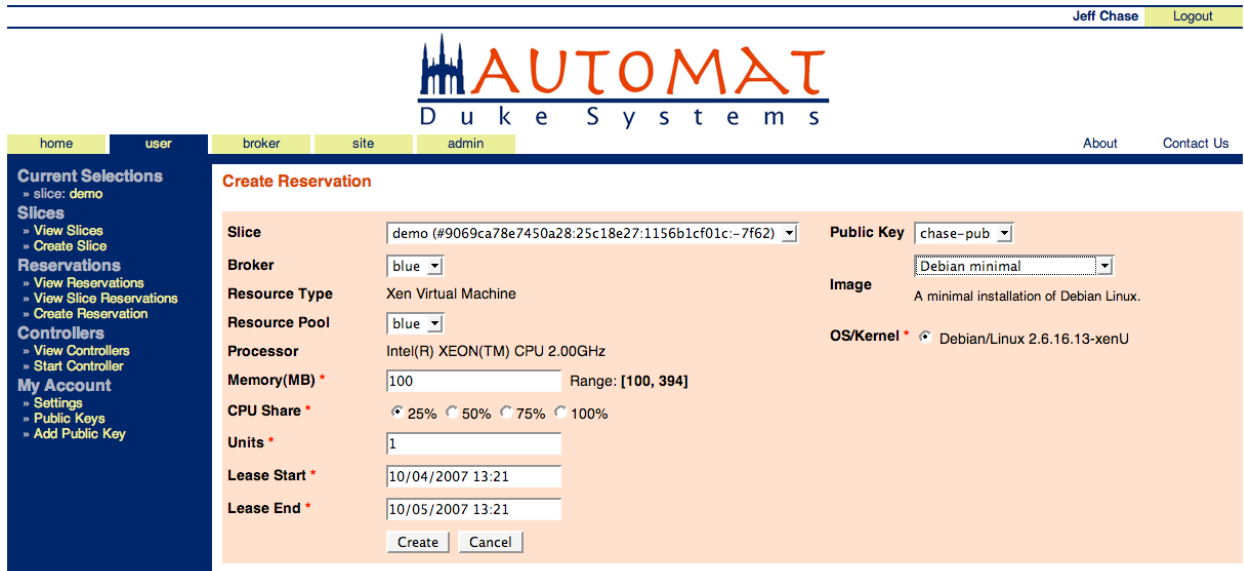


Figure 4-2 Automat Portal into Container of ORCA Actors

5. ORCA Structure for Demo

For the demo, three ORCA containers were setup at RENCI, Duke and UNC.

These containers were populated with:

- Six ORCA authority actors, in containers at RENCI, Duke and UNC
- Two ORCA broker actors, in a container at RENCI
- One ORCA slice controller (service manager) actor, in a container at RENCI.

Various drivers were built to allow the authority actors to control the substrate.

An “administrator” with all privileges can log into a container using the portal and configure the container, and all actors in containers, and establish relationships between actors.

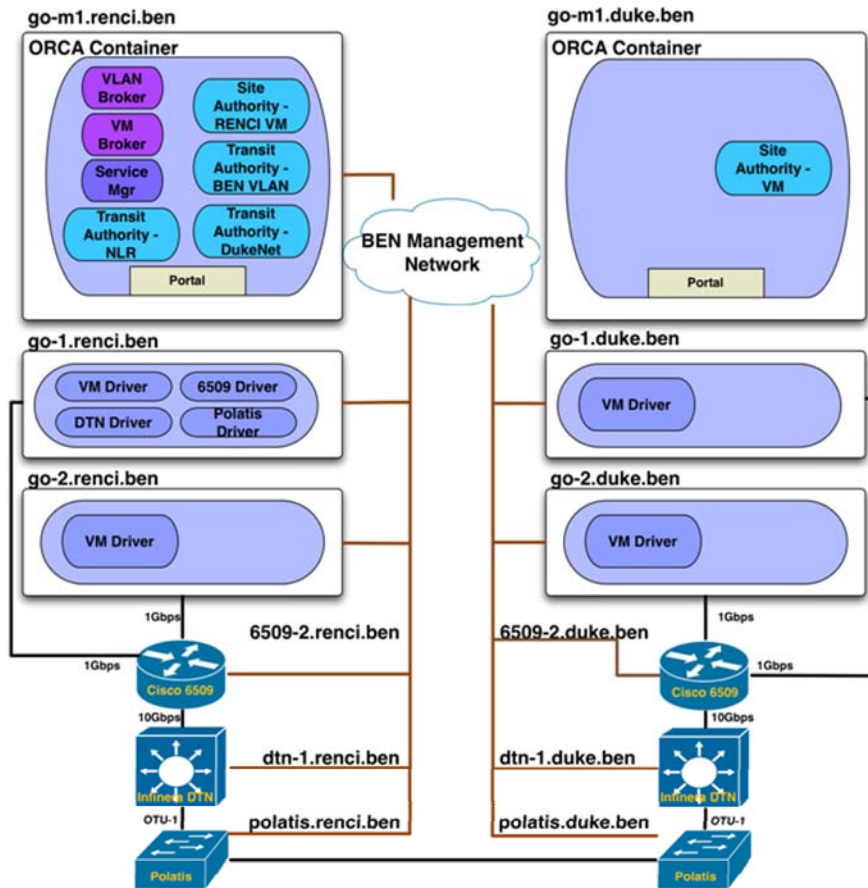
A “researcher” can log into the container with the slice controller (service) manager and use it to obtain the desired paths and resources.

Here are details on the several site/transit authorities:

Name	Substrate	Resources delegated to clearinghouse	Container	Description
UNC-CH VM	3 blades (go-[12].unc.ben)	VMs	UNC-CH	Allocate and configure VMs on the substrate using XEN
Duke VM	2 blades (go-[12].duke.ben)	VMs	Duke	Allocate and configure VMs on the substrate using XEN
RENCI VM	3 blades (go-[123].renci.ben)	VMs	RENCI	Allocate and configure VMs on the substrate using XEN
DukeNet	Cisco 6509 at Duke BEN PoP	VLAN tags	Duke	Allocate VLAN tags, map between NLR tag and BEN tag
BEN	Cisco 6509's, Infinera DTNs and Polatis switches at UNC and RENCi	VLAN tags	RENCI	Configure cross-layer paths between sites, map NLR VLAN tag to BEN VLAN tag
NLR	Null	VLAN tags	RENCI	Holds a number of preconfigured VLAN tags across NLR.

Table 1: Description of site authorities

Table 5-1 Site Authorities



Container 3
UNC just
imitates Duke...

Figure 5-1 ORCA Structure

In addition to the ORCA containers, there were VM substrates at RENCi (3 blades), Duke (2 blades) and UNC-CH (3 blades).

Each VM was stood up with 2 interfaces. One was bridged into the management network and the IP address assignment on it was done by ORCA out of a pool of addresses we gave it. The second interface was bridged into the BEN dataplane and left unconfigured.

6. ORCA Calls and BEN Connectivity

To create several VMs at Duke, RENCI and UNC-CH BEN PoPs and then link them together with VLANs via NLR and BEN, the following ORCA calls were used:

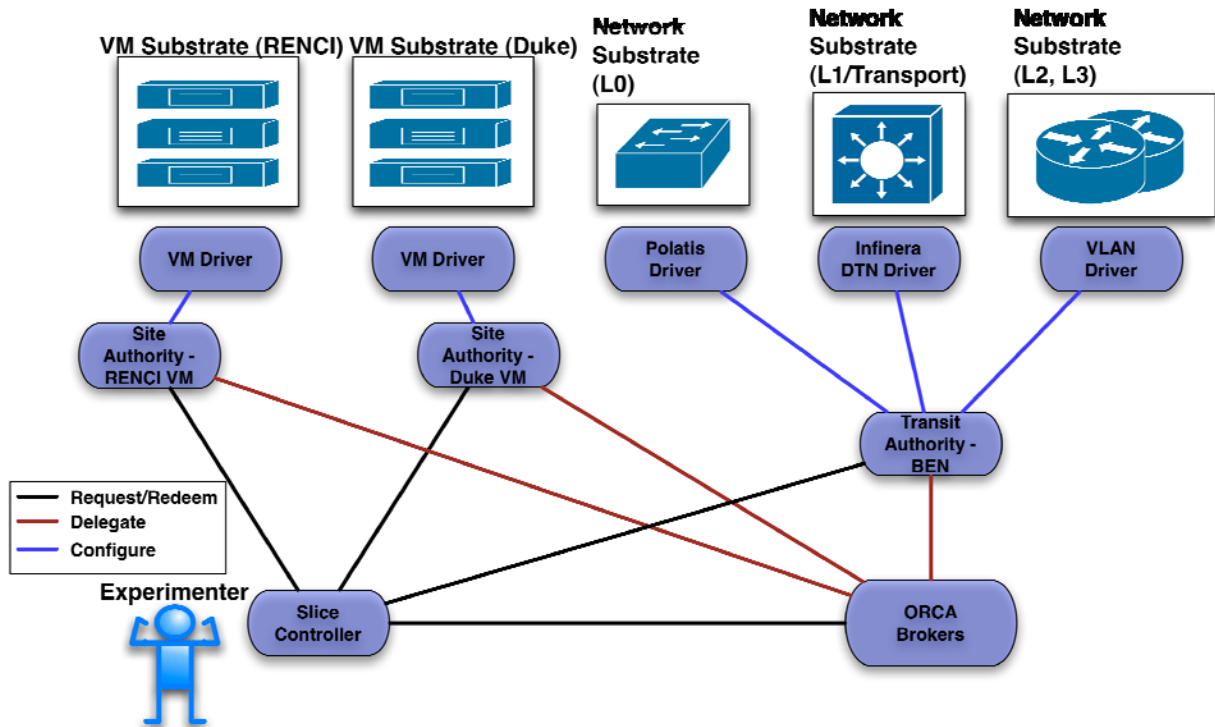


Figure 6-1 ORCA Calls

They were done in the following sequence, which was determined by the Slice Controller:

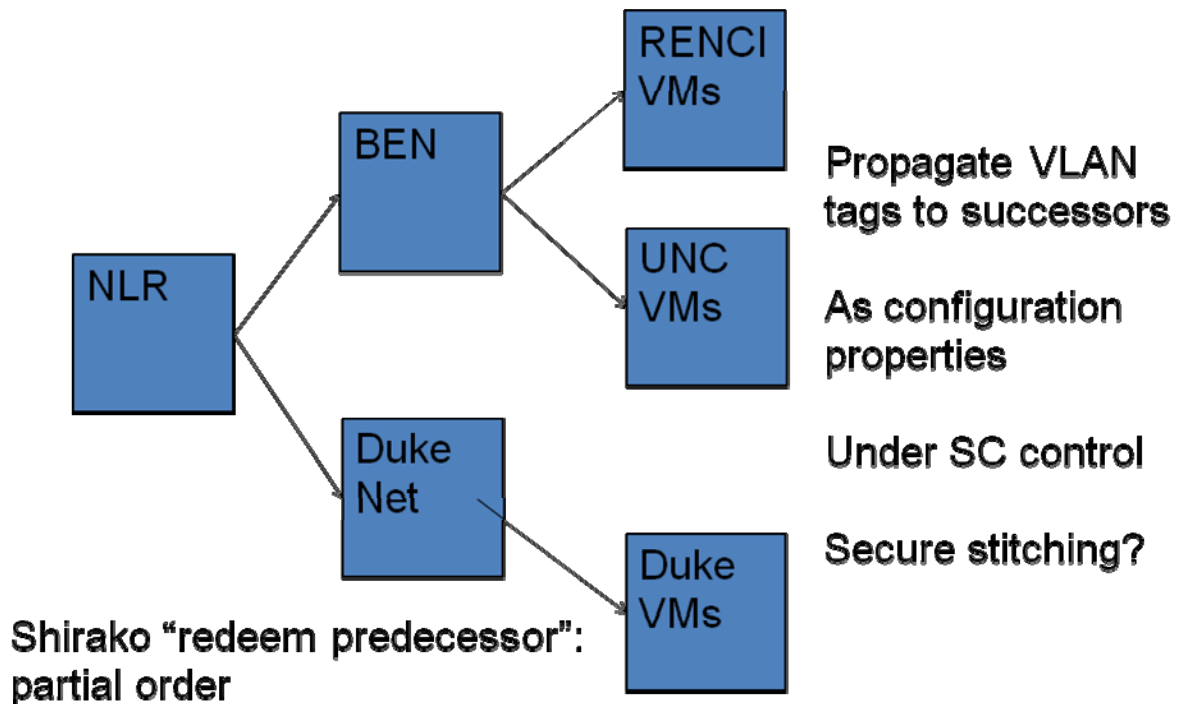


Figure 6-2 ORCA Call Sequence

Since the NLR connection is currently pre-configured using Sherpa, the NLR transit authority returns that fixed configuration.

Since the DukeNet connection is currently pre-configured using Sherpa, the DukeNet transit authority returns that fixed configuration.

The BEN Transit Authority dealt with equipment (Polatis, DTN and 6509) in multiple PoPs, handling three network layers, to setup the desired VLANs.

The VLAN were connected with translations provided by the 6509s.

VLAN Mapping Details

Two types of VLAN tag mappings are assumed: *static* and *dynamic*. Static mapping is considered pre-existing to the demo and known to the CF. The slice manager performs dynamic mappings between VLANs to stitch together slivers of different networks (e.g. BEN and NLR).

NLR VLAN tags are statically provisioned ahead of time using NLR's Sherpa tool to connect RENCi 6509 BEN router to the edge of the Duke campus cloud. A static

mapping is also created for the NLR tag to the Duke cloud tag, which is visible from the Duke BEN 6509.

NLR tags are issued in tuples < tag1, tag2 >, where tag1 is the tag visible on the BEN 6509 at Duke and tag2 is the tag visible on the BEN 6509 at RENCI. The same NLR VLAN tag can be seen as two different tag numbers¹ due to static remapping across the Duke campus cloud.

If VMs are provisioned at a site, *all* client-side ports on site's 6509 are provisioned with the same given BEN VLAN tag (the ids of these ports are known to the site authority). Currently we do not limit provisioning VLAN tags to ports to which physical hosts with VMs for this slice are attached to.

Figure 1: VLAN Mapping example

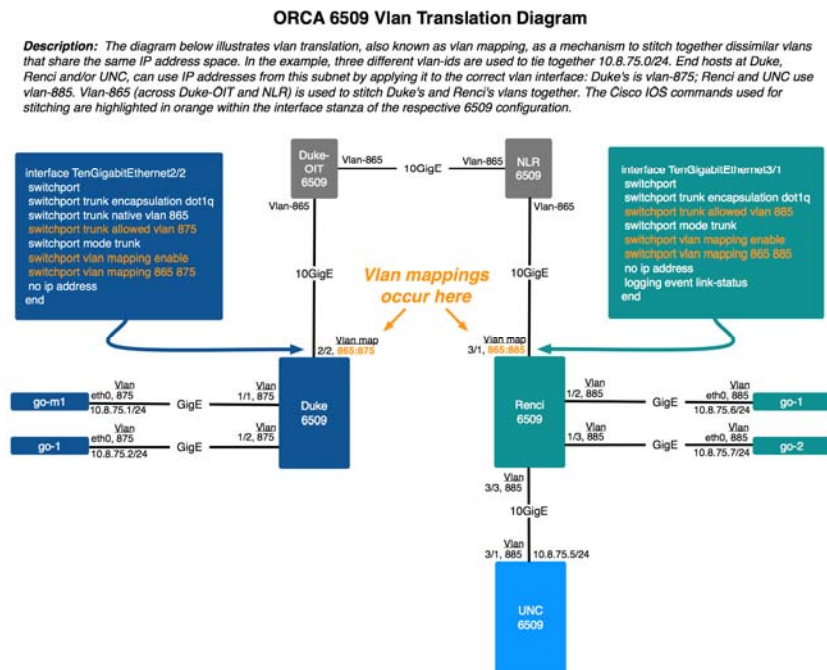


Figure 6-3 VLAN Mapping Example

Figure 6-3 shows a working example of VLAN map operation using NLR and BEN. The port identifiers are the ones that will be used in the demo. Note that VLAN remapping takes place at the 10G interfaces.

¹ We will have two policies for NLR tags – one that assumes that tags on both sides are the same and another that allows for tuples. We will use whichever one is ready by demo time.

During the demo, there was a login to each VM via the management interface, to configure the data plane interface with an IP address from the pool of private addresses, and then to demonstrate a ping across the slice, plus slice isolation.

The appropriate reservation of resources for both slices could be seen in each piece of BEN equipment (Polatis, DTN and 6509).

After the slices were set up with the help of the ORCA web portal and the Slice Controller, they were subsequently torn down. It could be seen that all resources were properly released after both slices were torn down.

Sequence

1. Site authorities delegate resources to the ORCA-BEN clearinghouse:
 - a. RENC1, Duke and UNC-CH VM site authorities delegate a fixed number of VM instances they are able to create.
 - b. BEN transit authority delegates a fixed range of VLAN tags available for connecting sites on BEN.
 - c. NLR transit authority delegates a fixed range of VLAN tags available for traversing NLR. These tags have been pre-created using Sherpa and statically remapped through the Duke campus network to reach BEN 6509 at Duke. Depending on the policy, each NLR tag can be seen as two distinct tag numbers – a Sherpa-issued number is visible at RENC1 6509, a remapped number is seen at Duke 6509.
 - d. DukeNet transit authority delegates a range of VLAN tags.
2. Demo-specific slice manager passes the following to the ORCA-BEN clearinghouse as part of a request for a VLAN-based slice between RENC1 and Duke using BEN and NLR substrates
 - a. Number of VMs to be created at UNC, Duke and RENC1
 - b. NDL of the BEN VM-to-VM sliver
3. Brokers inside ORCA-BEN clearinghouse grant tickets for
 - a. A single NLR VLAN tag
 - b. A single BEN VLAN tag
 - c. A DukeNet VLAN tag
 - d. A number of VMs to be created at Duke
 - e. A number of VMs to be created at RENC1
 - f. A number of VMs to be created at UNC-CH
4. The redeem order is important as it allows us to not deal with the ‘stitching’ problem:
 - a. NLR transit authority (to get a single tag or a tuple, depending on policy)
 - b. BEN transit authority (to get a BEN tag and create a mapping in RENC1’s [and, optionally, Duke’s] 6509 between NLR and BEN tag). It uses the passed NDL to create a multi-layer sliver across BEN.

- Prerequisite: step (a)
- c. DukeNet transit authority (to get a DukeNet tag and create a mapping between NLR tag [tag visible from Duke] and DukeNet tag)
 - Prerequisite: step (a)
- d. RENCi VM site authority (to create VMs and enable BEN tag on all client side ports of RENCi's 6509)
 - Prerequisite: step (b)
- e. Duke VM site authority (to create VMs and enable BEN or DukeNet tag on all client side ports of RENCi's 6509)
 - Prerequisite: step (b) or (c)
- f. UNC-CH VM site authority (to create VMs and enable BEN tag on all client side ports of UNC-CH's 6509)
 - Prerequisite: step (b)

Actor Actions

NLR TA

1. Allocate a VLAN tag

BEN TA

The configuration actions of this site authority are based on the NDL slice request:

1. Allocate a VLAN tag
2. Unless exists already: provision fiber path between two sites (unless already provisioned by another slice) using Polatis switches
3. Unless exists already: provision 10G circuit between two sites (unless already provisioned by another slice) using Infinera DTN
 - a. We will use explicit mode of setup (no GMPLS)
4. Provision VLAN tag on 10G ports of 6509's facing each other as well as all client ports
5. Map the NLR vlan tag to the allocated tag on the NLR-facing 10G port.

This TA also performs the mapping between NLR tag and BEN tag.

DukeNet TA

1. Allocate a VLAN tag
2. Enable the VLAN tag on all client ports of 6509
3. Map the NLR VLAN tag onto the allocated tag on the NLR-facing 10G port.

VM SA (RENCi, UNC-CH and Duke)

1. Add vlan to the trunked interface connected to the 6509. Create vlan interface on the host
2. Create a slice bridge for the VLAN on the physical host connected to the vlan interface
3. Instantiate VMs with two interfaces (management and experimental) and connect to the bridge

4. Create VMs using XEN with standard images.

Portal requirements

- The portal allows requesting a number of VMs at each site.
- The portal displays NDL of BEN sliver request (editable text box)
- The portal displays IP addresses of allocated VMs at each site.
- The portal provides an XML-formatted page (format TBD) where the slice configuration is reflected (information about VMs, as well as NDL of the BEN sliver).

8. Resource Descriptions

An NDL-OWL schema was used for substrate descriptions and resource requests.

In particular, the BEN substrate was specified in an NDL-OWL file, that was loaded into the BEN Transit Authority. See <https://geni-orca.renci.org/trac/browser/orca/network/trunk/src/main/resources/orca/network>

After reserving a ticket specifying a “VLAN through BEN”, the Slice Controller also used an NDL-OWL file to completely specify the VLAN it wanted. See <https://geni-orca.renci.org/trac/browser/orca/network/trunk/src/main/resources/orca/network>

9. GENI Milestones Completed

The demo showed that the following GENI Milestones were completed:

- [1c Integration of NDL or other topology description mechanism](#)
- [1d Integrate ORCA with BEN including slivering of Infinera DTNs](#)
- [1e Complete end-to-end VLAN connections between BEN and NLR](#)

10. Extra Demo: MIN Visualization Engine Demo

The Multi-touch Collaborative Network Management (MIN) system was demonstrated as part of the ORCA-BEN demo.

It permits an operator to interact with a large touch screen, and get real-time data on network configuration, assignments, paths and flows.

This is a project internally funded by RENCi (a seedling prototype).

See <http://vis.renci.org/multitouch/>

Participants:

Chris Heermann (RENCi) - BEN operations, MIN visualization back-end

Xunlei Wu (RENCi) - MIN visualization front-end

11. Extra Demo: RENCI-P FPGA OpenFlow Project Demo

The RENCI-P (RE-configurable Networking and Computing Platform) project is based on the Sixis FPGA platform that provides 10x 10Gbps ports and 7 FPGAs (4 compute, 2 I/O and 1 supervisory).

Sixis is a local RTP startup. Their newest platform is even more impressive: more FPGA real estate, 12 ports with QSFPs that allow the use of optics.

The project goal is to complete the port of OpenFlow to the current Sixis platform to investigate any possible throughput issues. Then (if we can find the funding), to extend the capabilities by adding precise measurements (jitter and delay to enable protocol testing) as well as virtualization support.

Participant:

Dan Bedard (RENCI) RENCI-P hardware engineer