Internet2 SDN Support
The Programmable Forwarding Environment*

John Hicks
Matt Zekauskas

2017 Internet2 Technology Exchange
Overview

- Motivation
- Where we came from
- Overlay setup
- Onramp procedure
- Use cases
- Future
Motivation

• We need a programmable network where we (and others) can take risks without impacting production
  – Reduce friction from thorough acceptance testing for new projects
  – Use separate hardware to implement an overlay network
  – Support direct remote access for controllers.

• Collocate compute with hardware based switching
  – Support NFV and of SDN exploration
  – Allow for local controllers
  – Support alternative network approaches like ICN
The Programmable Forwarding Environment

• Build a research infrastructure to support SDN as an overlay network
  – Provide a more agile platform for the network and distributed systems research community in an overlay

• Support much of the discipline research needs in the core
  – APIs for dynamic VLAN generation

• Received support from the GENI Project Office to provide OpenFlow 1.3 capable overlay
Overview

• Motivation
• Where we came from
• Overlay setup
• Onramp procedure
• Use cases
• Future
Where we came from

- Operating SDN Networks – base AL2S on OpenFlow
  - Possible to build and operate a reliable Layer 2 and Layer 3 network on top of a SDN substrate
  - Possible to support multiple controllers concurrently on an SDN substrate through software virtualization
  - Vendor implementations of OpenFlow 1.0 were buggy and incomplete
  - Vendor implementations of OpenFlow 1.3 were very slow to appear, as well as buggy and incomplete (especially for hardware supporting many 100GE interfaces)
  - Building a network software stack requires absolutely rigorous testing
  - Supporting multiple controllers concurrently on a production network software stack:
    - Requires significant FTE resources
    - Moves slower than researchers are accustomed
(Former) Controlling a Slice on Internet2

- Request a slice (email: noc@internet2.edu)
- Receive a questionnaire from Internet2 NOC
- Submit questionnaire to Internet2
- Download FSFW; try your controller in that environment
  - [http://globalnoc.iu.edu/sdn/fsfw.html/](http://globalnoc.iu.edu/sdn/fsfw.html/)
- Use mininet simulation of AL2S for correctness testing
- Submit your package
  - Good documentation accelerates process!
  - Good logging accelerates process!
- Internet2 NOC tests your controller on our testbed (AKA iDREAM GENI environment)
  - Problems -> Go back one step
- Internet2 deploys your controller on Internet2 Network
Overview

• Motivation
• Where we came from
• Overlay setup
• Onramp procedure
• Use cases
• Future
Overlay setup

• The Internet2 Programmable Research Environment consists of eight sites on the AL2S backbone:
  – Seattle, Los Angeles, Salt Lake City, Kansas City, Houston, Cleveland, Atlanta and New York
Overlay setup

- Each site contains:
  - Dell Server
  - Corsa Switch
  - Multiple 10GE interconnects provided via AL2S
  - 10GE AL2S port for onramp/offramp
Internal - What it looks like

- AL2S circuits interconnect sites, circuits follow physical infrastructure
  - But this is not required
- Each slice gets its own set of logical circuits
- To the control plane, these look just like direct 10GE adjacencies
- Each slice has its own dedicated management network.
• Each slice gets a dedicated private management network
• A virtual bastion server is provided for access if needed
• Bastion provides limited NAT support to allow OpenFlow connections to remote controllers
• Slice users now have choice to run controller on our network or in their lab.
Overview

• Motivation
• Where we came from
• Overlay setup
• Onramp procedure
• Use cases
• Future
Onramp procedure

A general customer connection scenario
Collect information:

- Determine the topology of the desired slice (which nodes to include)
- Determine where the projects data plane with connect to the overlay (e.g. AL2S circuits between the home institution or project related facilities and the nearest overlay node)
- Determine if the project controller will be internal or external to the overlay
- Determine the Openflow rules needed in the project (used to see if the Corsa implementation of OF 1.3 is compatible with the project’s needs)
Make the request:

• Contact the GRNOC to request a ‘slice’ on the overlay network with the following information:
  – Provide the topology (nodes & links)
  – Include the locations of your external data plane connections
  – Provide OF rules needed for the project
  – Controller information:
    • Internal – We will provide the address and port to connect to your VM
    • External – IP address of the controller

[16]
Make the request (cont.):

• PI and/or Co-PI
• Short abstract of the project including desired outcomes
• Indicate if the project is funded
  • If so, by whom and provide grant # if applicable
• Duration of the project
Ongoing support:

- The Internet2 NOC will provide support for the initial setup and to ensure that the components are configure properly. However, once initial connectivity is established the Internet2 NOC will only intervene on infrastructure and/or security related issues. The Internet2 NOC will not, for the most part, help with debugging controller applications.

- If customers feel that there is an infrastructure related issue, they are encouraged to open a ticket with the Internet2 NOC. The ticket should reference the overlay project and have a description of the issue.
Overview

• Motivation
• Where we came from
• Overlay setup
• Onramp procedure
• Use cases
• Future
Steroid OpenFlow Services (SOS)

• SOS is a paradigm for network services delivery that enables operators to deliver network services without any setup requirements on user machines.
• SOS utilizes OpenFlow to redirect application specific traffic to application specific service agents.
• SOS also rewrites packet headers for a service to remain seamless to users.
• The SOS service optimizes large volume TCP downloads across a large delay-bandwidth-product wide area network.
• SOS service agents on both ends of the connection
  – seamlessly terminate a user TCP connection
  – launches a set of parallel TCP connections
  – leverages multiple paths when available to maximize throughput.
Clemson and Utah Genomic Transfer Test Environments

Environment 1: Dedicated circuit between production facilities
Environment 2: OpenFlow SOS Testbed
Apr 2015
Distributed SDX Interconnection
Development and Experimentation

SDX Overlay AM
- GRAM
- MAX Aggregate Manager
- StackV
- OpenFlow Driver

SDX AM
- StarLight SDX
- Switch

Starlight ExoGENI Rack

SDX AM
- GRAM
- MAX Aggregate Manager
- StackV

SDX AM
- NSE API
- AWS API
- NSI API
- OSCARS (NSI)
- Amazon Web Services

Washington International Exchange (WIX)
- SDX AM
- NSE Browser MLXe

MAX AWS (US East Region) Direct Connect Service
- AWS API
- NSI API
- OSCARS (NSI)
- Amazon Web Services

MAX InstaGENI Rack

Internet2 SDN Overlay

- SEAT
- SDN-Overlay AM
- SALT
- CLEV
- NEWY
- KANS
- LOSA
- ATLA
- HOUS

Southern Crossroads (SOX)
- SDX AM
- Corsa

SOX InstaGENI Rack

SDX InstaGENI Rack

- AL2S Circuits Connecting Exchange Points to SDN Overlay
- Range of 20 VLANs (outside of AL2S GENI AM VLANs)

- VLANs connecting GENI Racks to SDX
- Range of 20 VLANs, need to sync up with above VLANs
- Need to be configured as a new link in the GENI Rack AM (without peering interface for now)
SDNTrace

• End-to-End Network troubleshooting requires the visibility on a hop-by-hop basis
• End-to-End troubleshooting should also be able to look at “flows of interest” in “virtual paths”.
• This demo is the an exploration into using SDNTrace, perfSONAR, and other tools to look at these “virtual paths” on a network hop-by-hop.
• The exploration will validate the “virtual path” by starting a client and dynamically placing the tools in the “virtual path”.
SDNTrace
Overview

- Motivation
- Where we came from
- Overlay setup
- Onramp procedure
- Use cases
- Future
Future - next steps

- Continue to support researchers
- If you have a project that could use this programmable research environment – let us know!
- Looking for SDN projects using technology other than OpenFlow
- Look toward more automatic slice creating (based on experience)
- Possibly deploy 100G path on designated nodes for high bandwidth applications