MIAMI FL

SEPTEMBER 25–28

A SECURE SDN SCIENCE DMZ

Indiana University

Ben Mack-Crane

Principal Architect, Corsa Technology

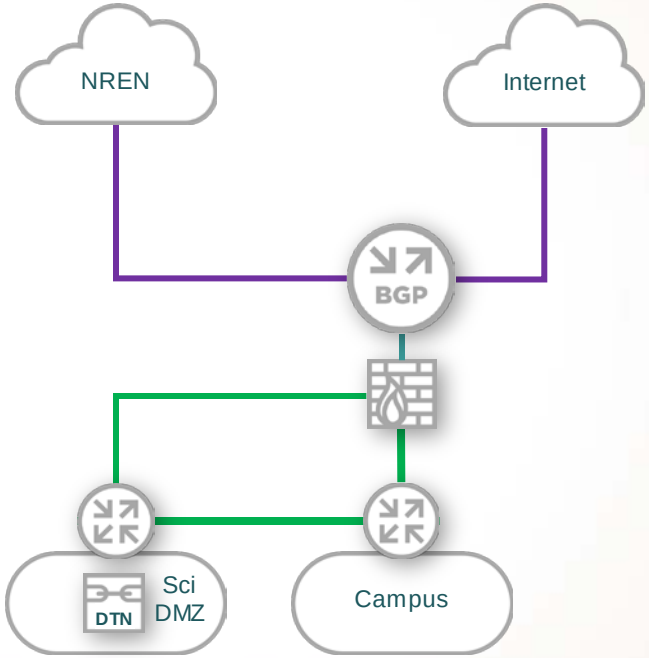
A Secure SDN Science DMZ

CONTENTS

- The Approach
- The Setup
- The Trial

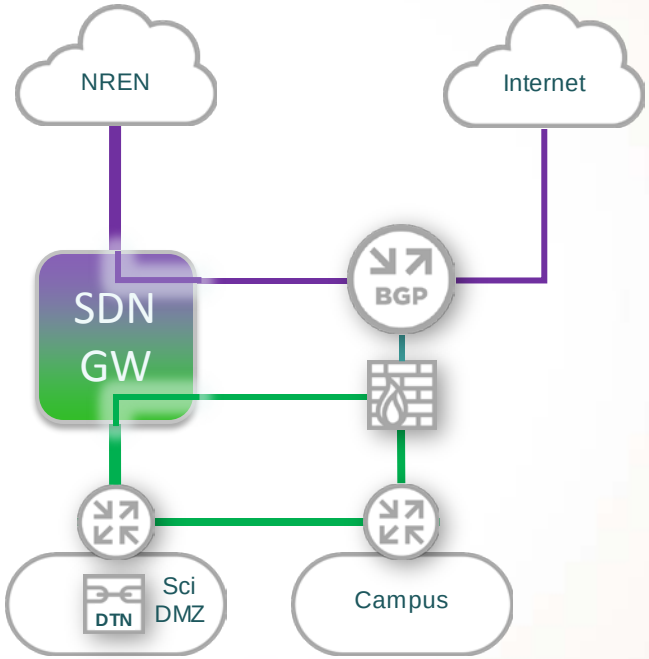
A Secure SDN Science DMZ – The Goal

- Looking for a Science DMZ design that is
  - Easy to understand
  - Relatively easy to deploy
  - Without compromising security (i.e., keeps the CISO happy), and
  - Enhances data transfer performance



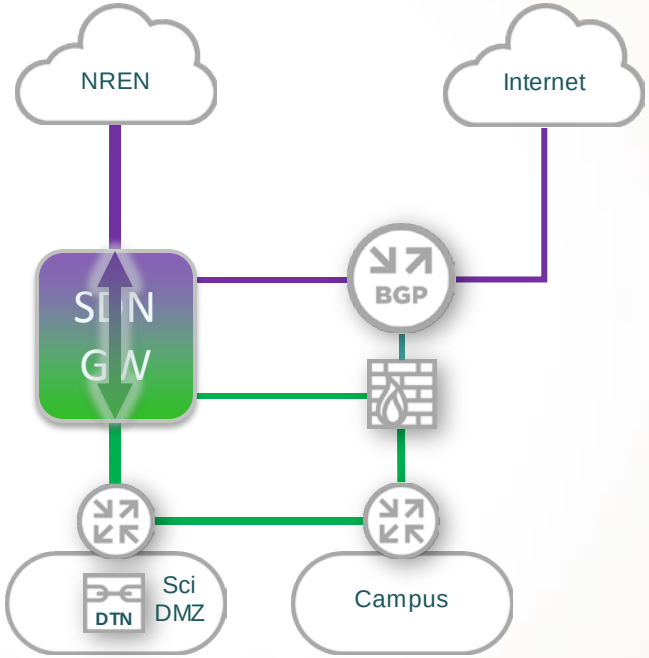
A Secure SDN Science DMZ – The Approach

- Add an SDN Science DMZ Gateway
- Don't change the (logical) topology
  - Maintain existing IP peering relationships
  - Maintain the traffic pattern that secures the campus network
- Configure static L2 connections
  - SDN gateway acts as "bump in the wire"
  - Or two bumps
  - No change to routing topology



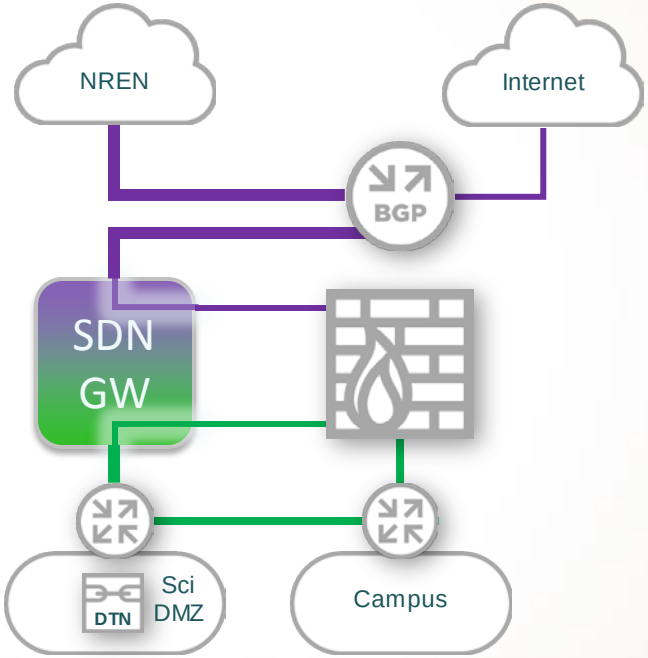
A Secure SDN Science DMZ – Selective Routing

- Improve throughput for high volume data transfers
  - Selectively route authorized data transfer flows directly to the DMZ
  - Bypass border router and firewall
- SDN datapath controls
  - Recognize authorized data transfer flows and install per-flow routing
  - Packet headers rewritten as though they passed through router/firewall



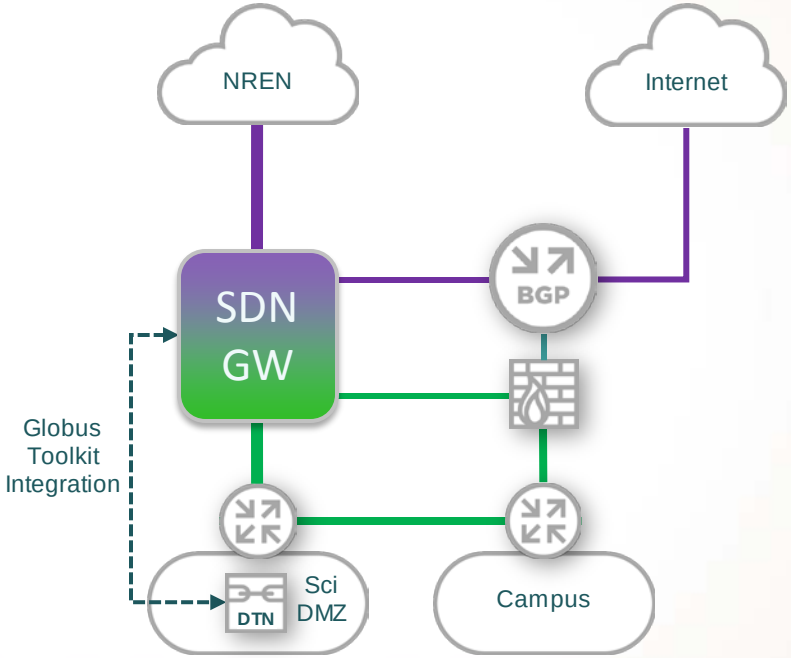
A Secure SDN Science DMZ – The Approach (alternate)

- Add an SDN Science DMZ Gateway
- Don't change the (logical) topology
  - Maintain existing IP peering relationships
  - Maintain the traffic pattern that secures the campus network
- Configure static L2 connections
  - SDN gateway acts as "bump in the wire"
  - Similar to SciPass developed at IU, but without the IDS load balancer



A Secure SDN Science DMZ – Authorizing Flows

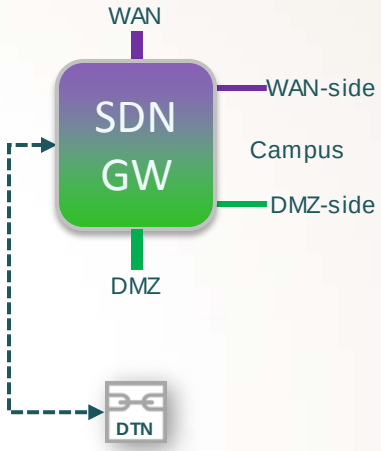
- Recognize authorizied data transfers
  - Method depends on policy
    - Learn flows using DTN source address
    - Recognize transfer control protocol
    - Integrate with transfer control system
- Globus/gridftp integration
  - Callout functions at key points in data transfer process, for example
    - post_connect() or post_accept() – install bypass flows
    - post_close() - remove bypass flows



Setting up an SDN Science DMZ

Goal: Provide configuration guidance and automate as much as possible

- Switch installation, cabling and commissioning (as usual)
- Configuration of DTNs
  - As usual, plus (optional) configuration of VLAN interfaces
- Install and start SDN Controller
  - Configure interface/VID roles (WAN, Campus WAN-side, Campus DMZ-side, DMZ)
  - Controller provisions baseline connectivity (e.g., VLAN connections)
  - Configure secure connectivity to DTNs (e.g., for Globus integration)
- SDN Operation
  - Automated selective routing for authorized data transfers
  - Potential for coordination with other SDN Controllers (e.g., OESS)



Setting up an SDN Science DMZ – Extras

As with many networking solutions, there are opportunities for enhancement

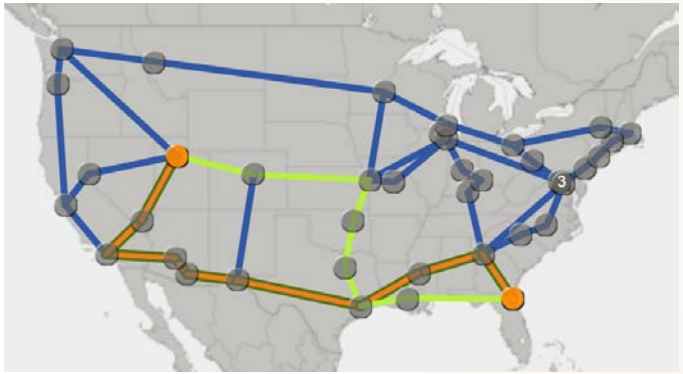
- Potential for BGP integration if there are multiple routes to WAN
- Integration with fault recovery strategies
  - "Bump in the wire" approach intended to simplify this, but…
  - There may be benefit in other approaches
- Integration with OESS to use dynamic AL2S circuits

The Trial

Goal: Investigate SDN Science DMZ configuration and performance

- We are setting up SDN switches, with attached DTNs, on three campuses
  - University of Utah (Joe Breen)
  - Florida International University (Jeronimo Bezera)
  - Indiana University (Uwe Dahlmann)

- Develop and test configuration automation
- Evaluate effectiveness of traffic management tools (e.g., flow shaping)
- Testbed for software integration



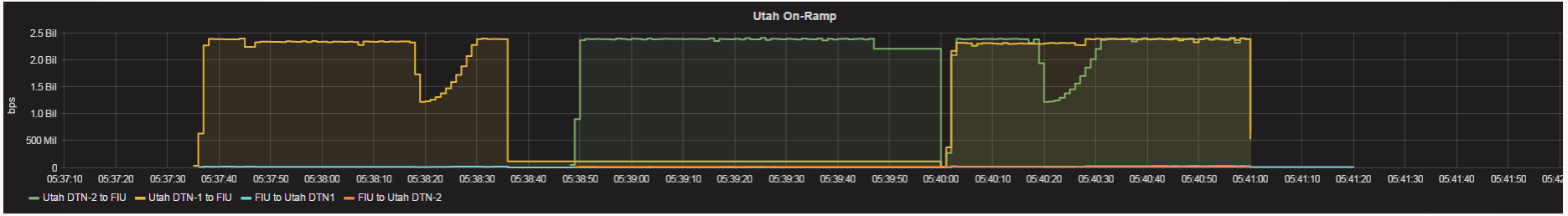
Trial Configuration

- DTN configuration
  - Internet2 CentOS image
  - Ansible playbook for additional configuration (Globus, VLAN subinterfaces)
- Switch configuration
  - Corsa SDX pipeline with flow entries implementing default L2 connectivity
  - Simple Ryu controller to capture performance data (on-switch VM)
- Internet2 configuration
  - Utah-to-FIU AL2S connections to test high volume transfer performance and traffic management
  - Loopbacks for testing campus connectivity to Internet2 (regional network)

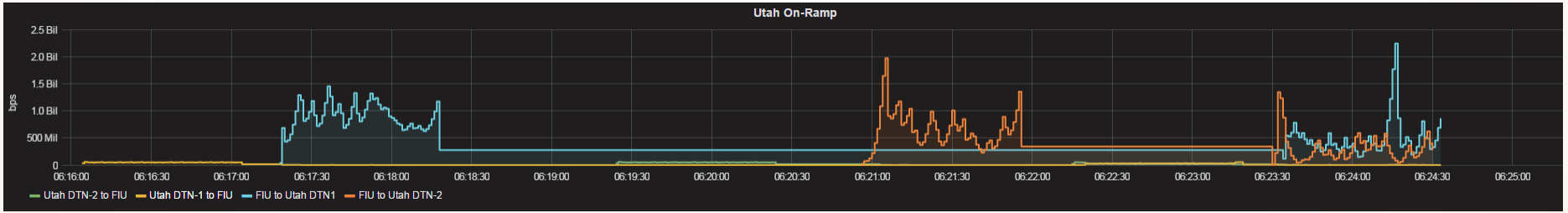
Trials (and tribulations)

- Baseline testing
  - Ansible playbooks to do local campus testing and end-to-end testing
  - Initial tests being done with bwctl/iperf3
  - Loading measurement data into InfluxDB, viewing with Grafana
- Working toward a stable baseline
  - Default DTN configuration achieves roughly 2.5Gbps transfer rate with 60ms RTT
  - Challenge to achieve this performance in both directions between Utah and FIU
  - Challenge to test campus to Internet2 (AL2S loopback problems)

Trials (and tribulations)



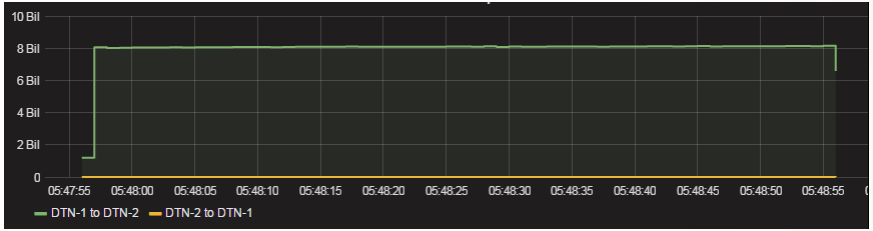
Fairly clean transfer Utah-to-FIU, but negligible rate FIU-to-Utah



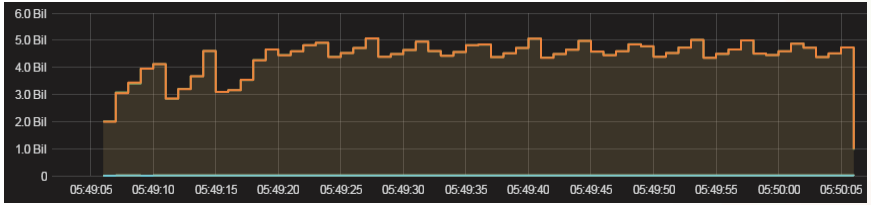
Negligible rate Utah-to-FIU, choppy transfer rate FIU-to-Utah

Trials (and tribulations)

- Local test between DTNs at one campus
  - Stable transfer rate



- Same campus test looped at Internet2
  - Packet retransmission every few sec
  - What is causing this?



The Ongoing Trial

Goal: Investigate SDN Science DMZ configuration and performance

- Upgrade campus switches
  - More comprehensive measurement capabilities
  - More advanced traffic management functions
- Develop and test configuration automation
- Evaluate effectiveness of traffic management tools (e.g., flow shaping)
- Testbed for software integration