geni
Exploring Networks of the Future

GENI Monitoring

Topology in the Mesoscale

Chaos Golubitsky, GPO

GEC12 Nov 3, 2011

www.geni.net

Sponsored by the National Science Foundation

Topology and the Mesoscale

- Layer 2 mesoscale network is different from commercial networks:
  - Topology contains intentional loops, for use by experimenters in path selection experiments
  - Experimenters know how packets are switched and routed through the multisite network
  - Experimenters modify packet switching and routing, independently from other experiments
- Mesoscale has been operational for a year now
  - Examples of network problems seen so far in the wild
  - Implications for future monitoring and debugging

Uncontrolled VLAN Bridging

- What can happen?
  - Mesoscale core consists of two VLANs
  - VLANs are bridged at each campus by an OpenFlow-controlled VLAN
  - If bridge fails open, traffic leaks between core VLANs
- Why does it happen?
  - Interface misconfiguration
  - OpenFlow VLAN fails open (default config on HP)
  - VLAN loses its OF configuration entirely and reverts to learning switch
  - Experiment misconfiguration

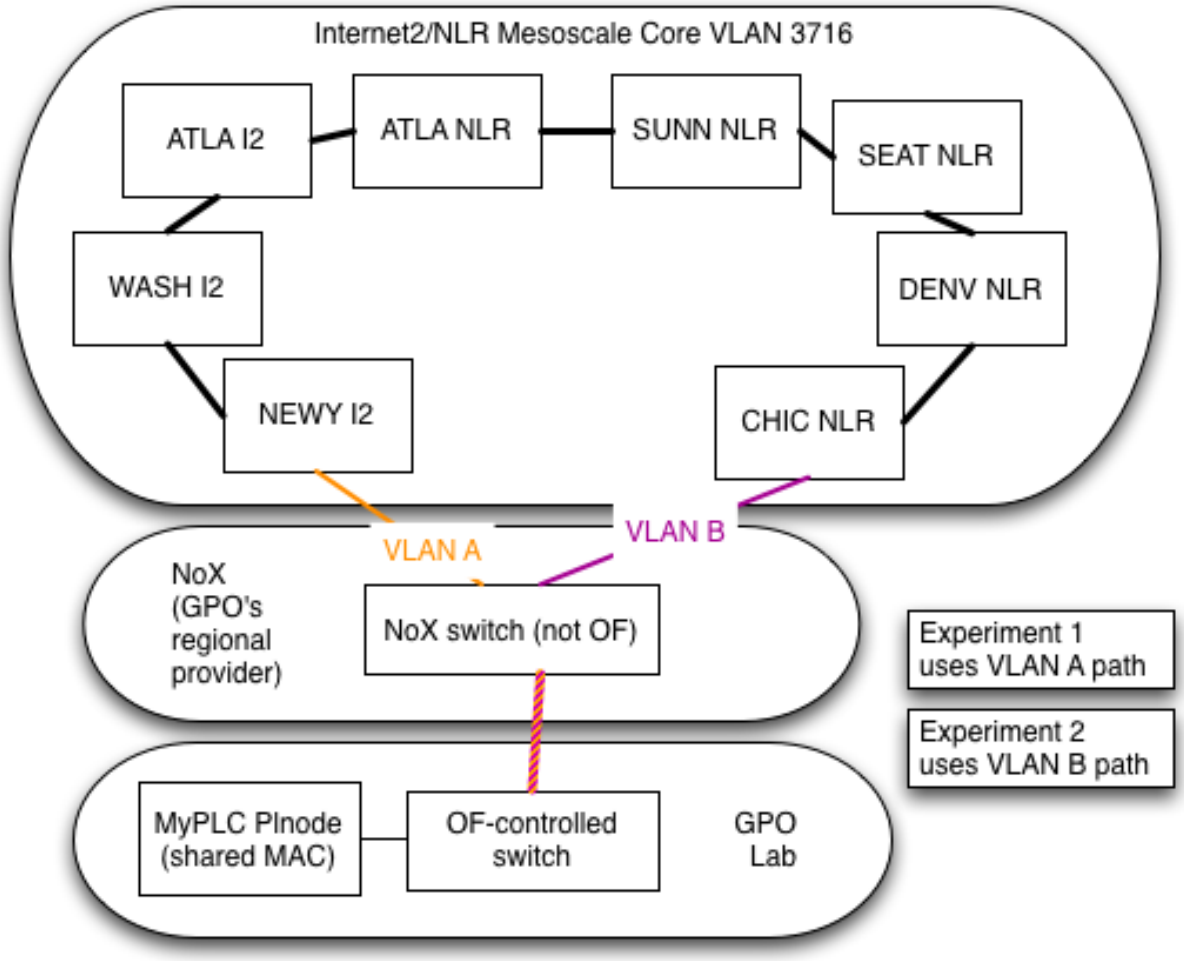
Detecting uncontrolled VLAN bridging

- Detection of severe failures is easy:
  - Define traffic which should never be seen on each VLAN (IP monitoring subnet of other VLAN)
  - Alert if you see that traffic
  - http://groups.geni.net/geni/wiki/NetworkCore/TrafficLeaks
- Tracking down failures so they can be resolved:
  - GMOC and GPO have tracked down many severe failures using suspicious MAC address table patterns
  - But: the leaking traffic is not the traffic causing the leak
  - Intermittent failures are much harder to find
- How do we find leaks affecting only a subset of (non-monitoring) traffic?

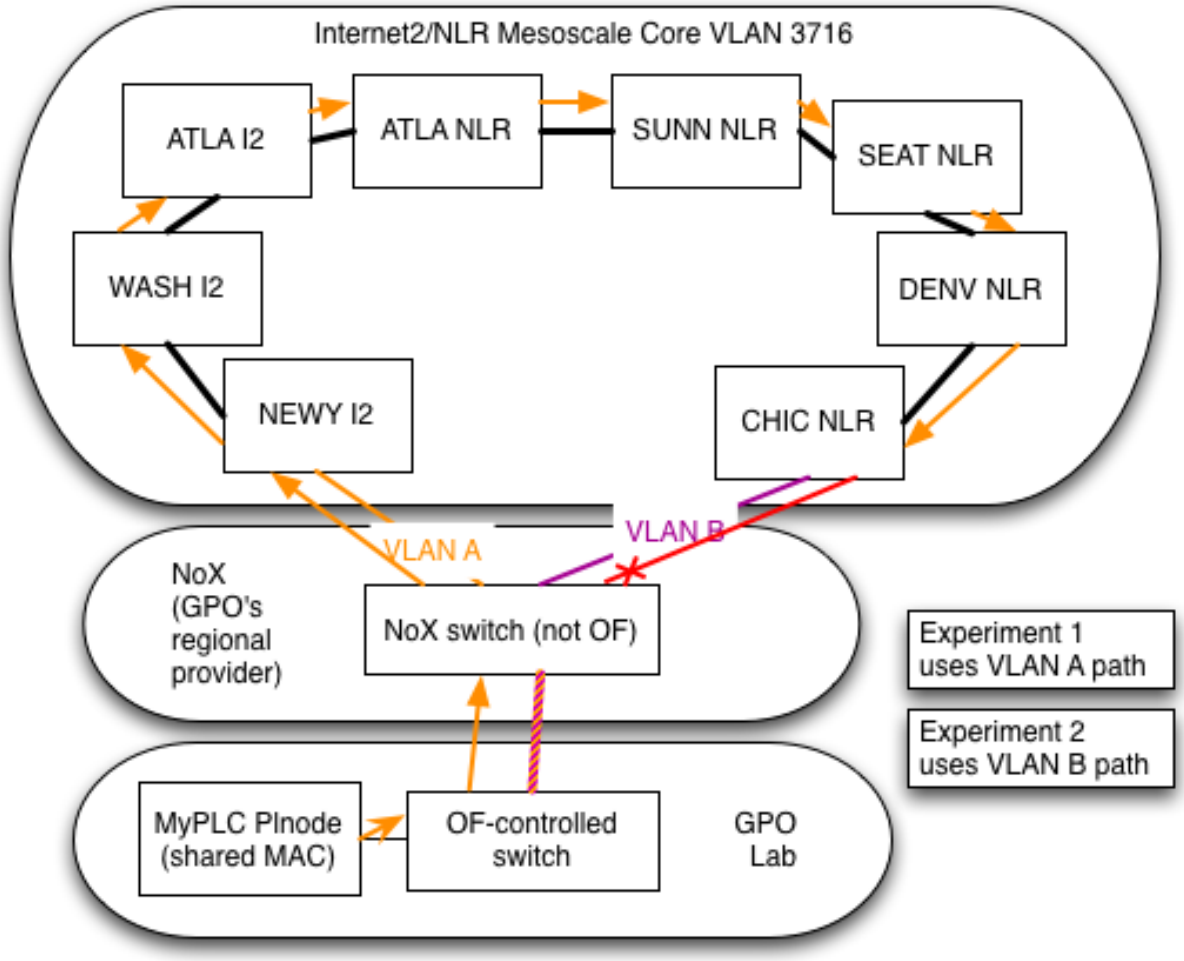
Shared MAC Problems

- Why MACs get shared between experiments:
  - Physical interfaces are expensive
  - Subinterfacing puts one interface on many networks
  - Default (Linux) subinterfaces share MAC address
  - Examples: mesoscale MyPLC, monitoring nodes
- When this causes problems:
  - Different topologies using the same MACs on the same devices
  - Depending on bridging, maybe even the same VLANs
  - Complicated by non-OF devices in an OF path

Shared MAC Problems: GPO Example (1)



Shared MAC Problems: GPO Example (2)



Shared MAC Problems: Detection

- NoX detected this problem via syslog:

```
Sep  8 11:46:18 EDT: %SW_MATM-4-MACFLAP_NOTIF: Host 001b.215f.8fe1
in vlan 3745 is flapping between port Gi0/9 and port Te0/1
Sep  8 11:47:16 EDT: %SW_MATM-4-MACFLAP_NOTIF: Host 001b.214b.3fad
in vlan 3745 is flapping between port Gi0/9 and port Te0/1
```

- General solutions feasible in the short term:
  - Track MAC addresses of known hosts, and alert if you see them somewhere unexpected
  - Collect problematic log messages which might be seen on non-OpenFlow devices connecting OpenFlow networks; ask regionals to watch for them

What these problems have in common

- They are ways the topology can fail to match experimenter expectations
  - …which could lead to experimental results mismatching expectations
- These problems have occurred “in the wild” in the early mesoscale deployment
- We want to accurately detect problems like these
  - GENI software will help in the long term
  - Some simple diagnostics have been useful so far
  - We can do more in the near term

Long-term GENI solutions

- Stitching:
  - Design a custom end-to-end per-experiment topology
  - Virtualize resources which we manually provision, and often share, right now (e.g. VLANs, MACs)
  - Programmatic verification of topology may be part of topology reservation
- Instrumentation and Measurement:
  - Richer capture of activity on intermediate networks to help experimenters validate the topology they expect
  - Within-slice on-demand measurement to verify complex experimental configurations

Diagnostics in Current Use

- Define something you expect not to see, and alert if you see it. Examples:
  - Bridged VLAN detection ("Monitoring subnets A and B are distinct. If they see each other's traffic, something is wrong")
  - Shared MAC address ("MAC address M belongs to site B. If it is learned on an interface facing away from site B, something is wrong")
- These checks are easy to do
- With reasonable definitions, false positives are low

Future directions

- Tools to improve detection and response:
  - "Known bad symptom" tests which can run at sites, regionals, and backbones
    - Then share code for those tests, and run them
    - Simple things for regionals to be aware of are especially useful
  - Monitor MAC addresses learned by experimental switches, and alert on unexpected paths when feasible
- Other suggestions?
- Thank you for helping operate the mesoscale!