

Cloud-Based Autonomic Service Monitoring in GENI Network

Muhammad Anan and Levi Ilyes
Purdue Univ. Calumet, IN



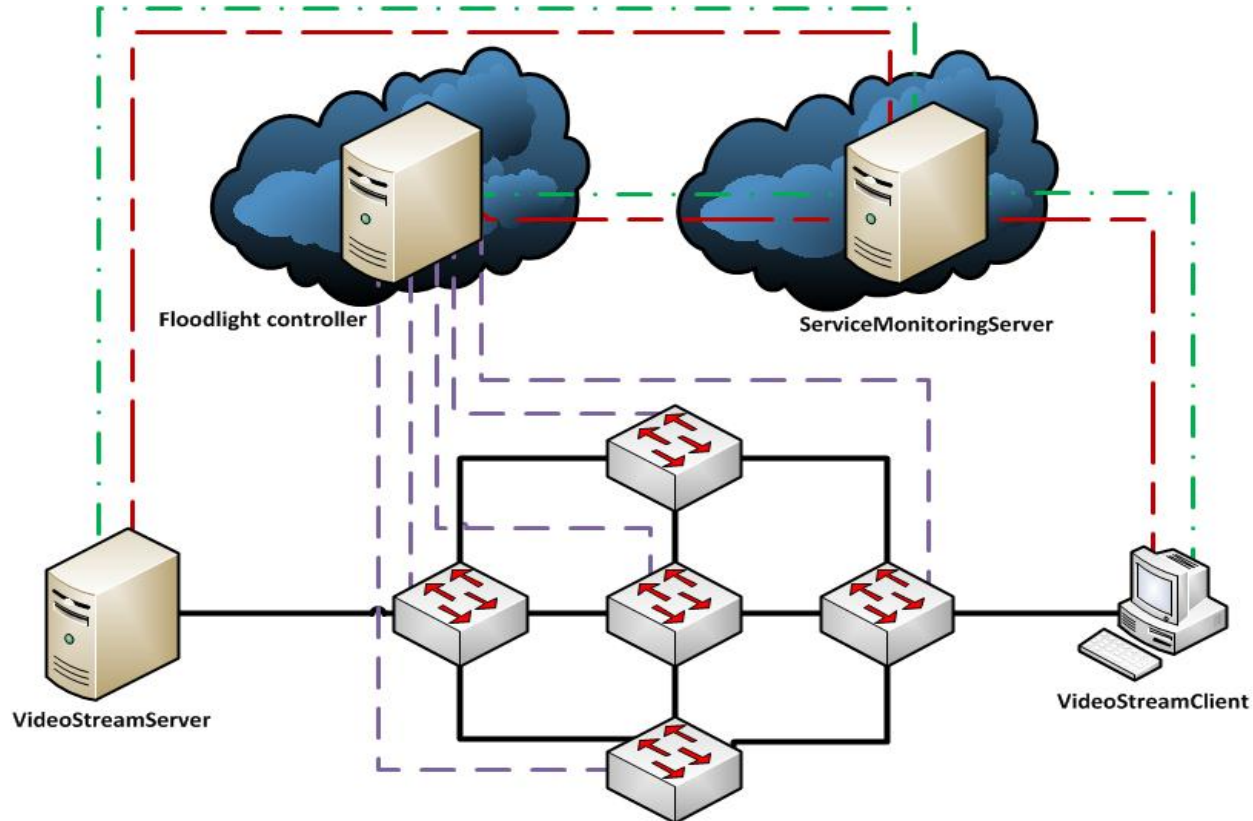
Musa Ayyash
Chicago State University, IL



Research Goals

- ▶ Develop a complete system, designed with the new requirements of the Future Internet in mind
- ▶ Provide a well known service and actively monitor its behavior
- ▶ Prove how combining Future Internet Technologies can improve TCP performance under unstable, changing network conditions
- ▶ Highlight the advantages of Software Defined Networking

System Components



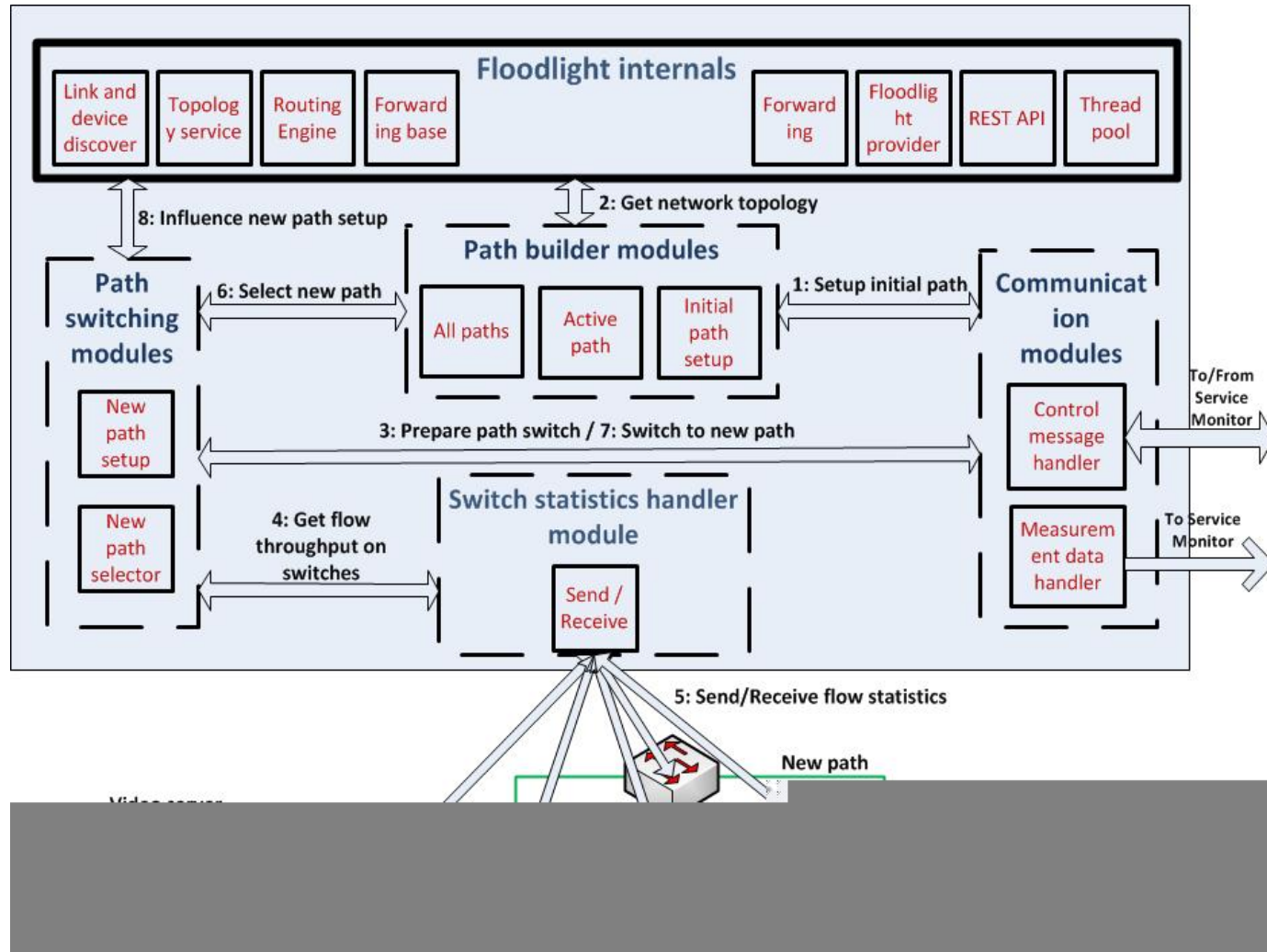
VideoStreamServer <-> VideoStreamClient TCP connection for Video Data Transfer

OFSwitch <-> OFController SSL connection for OpenFlow control traffic

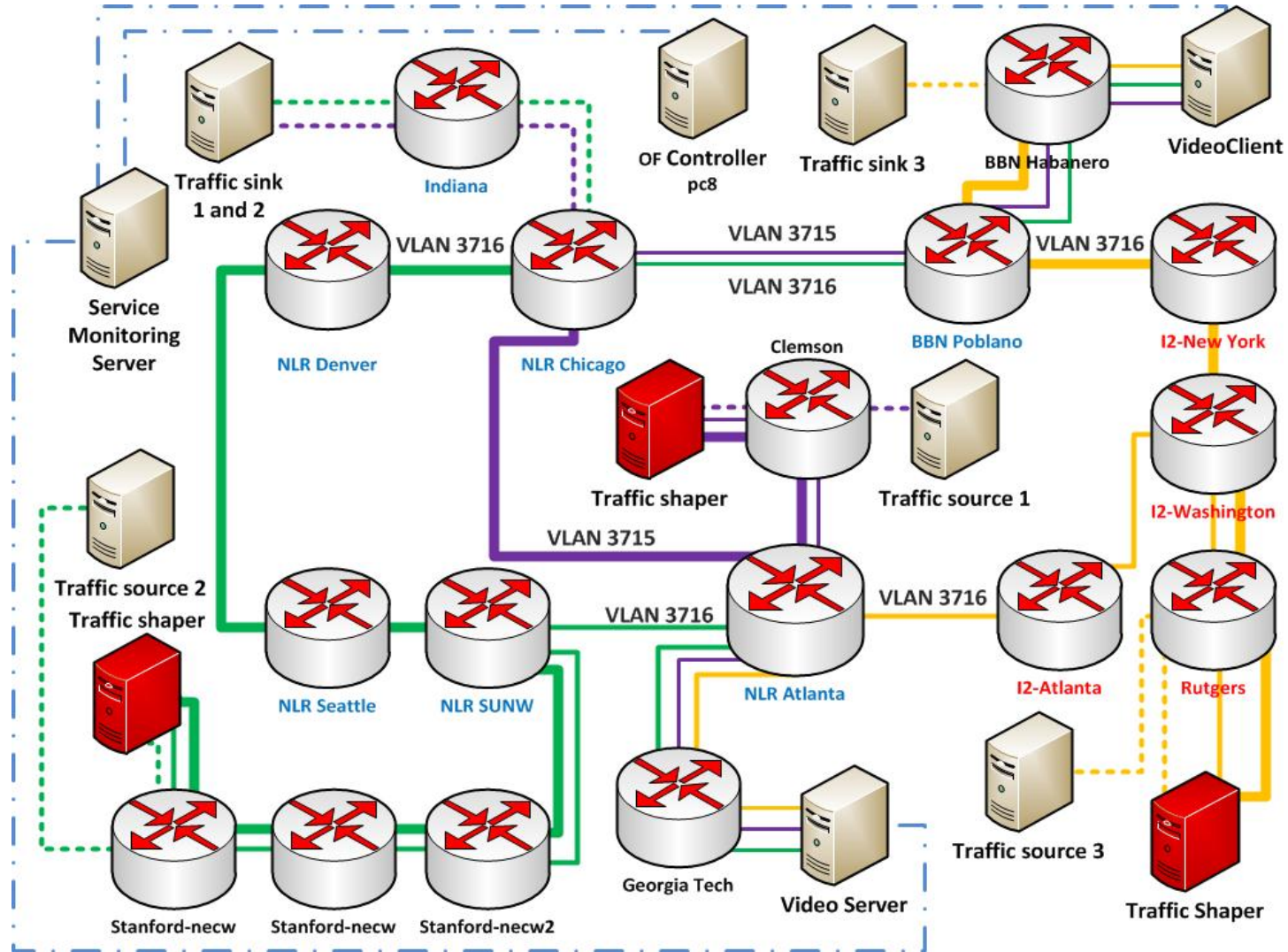
VideoServer <-> MonitoringServer; VideoClient <-> MonitoringServer; OFController <-> MonitoringServer TCP connection for data message exchange

VideoServer <-> MonitoringServer; VideoClient <-> MonitoringServer; OFController <-> MonitoringServer TCP connection for measurement data

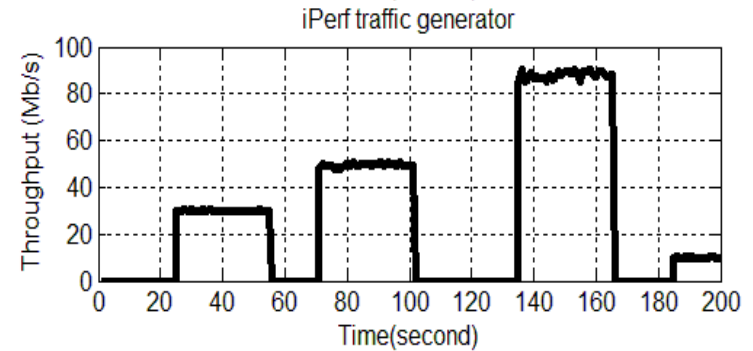
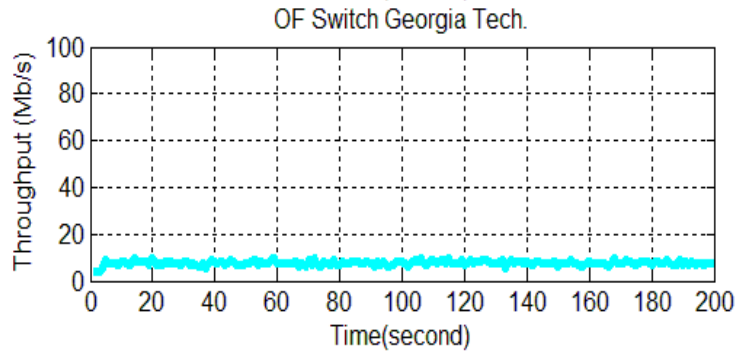
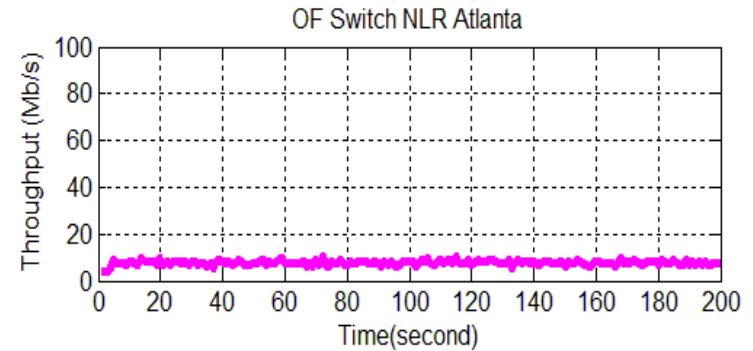
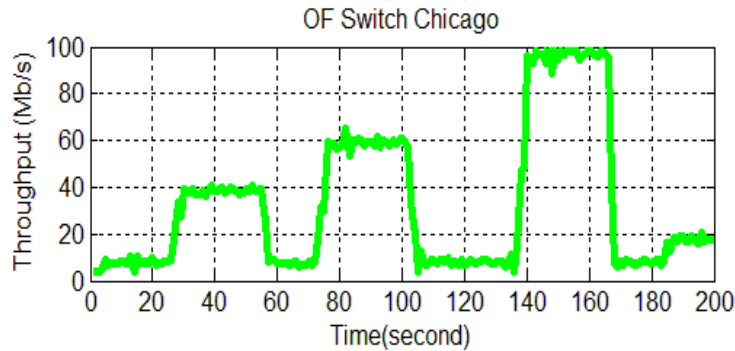
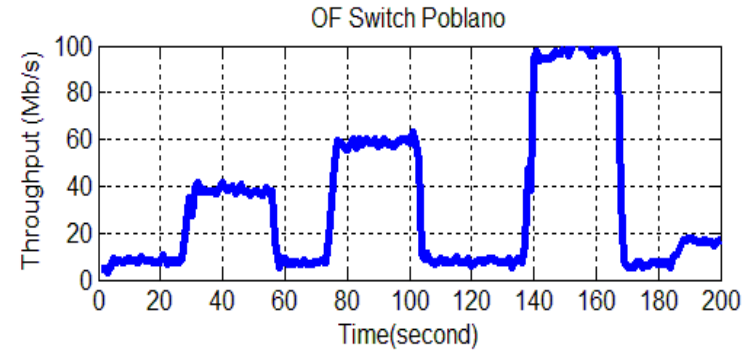
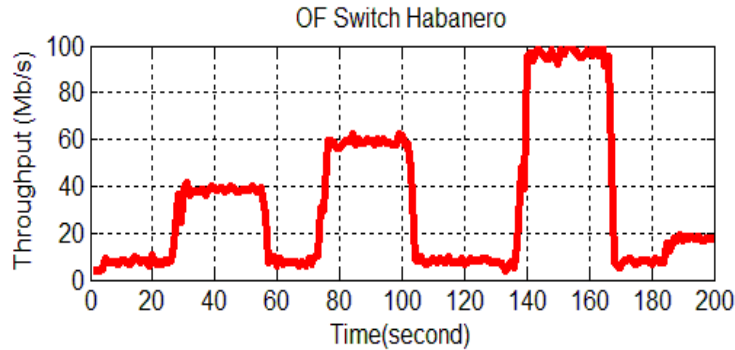
OpenFlow Controller



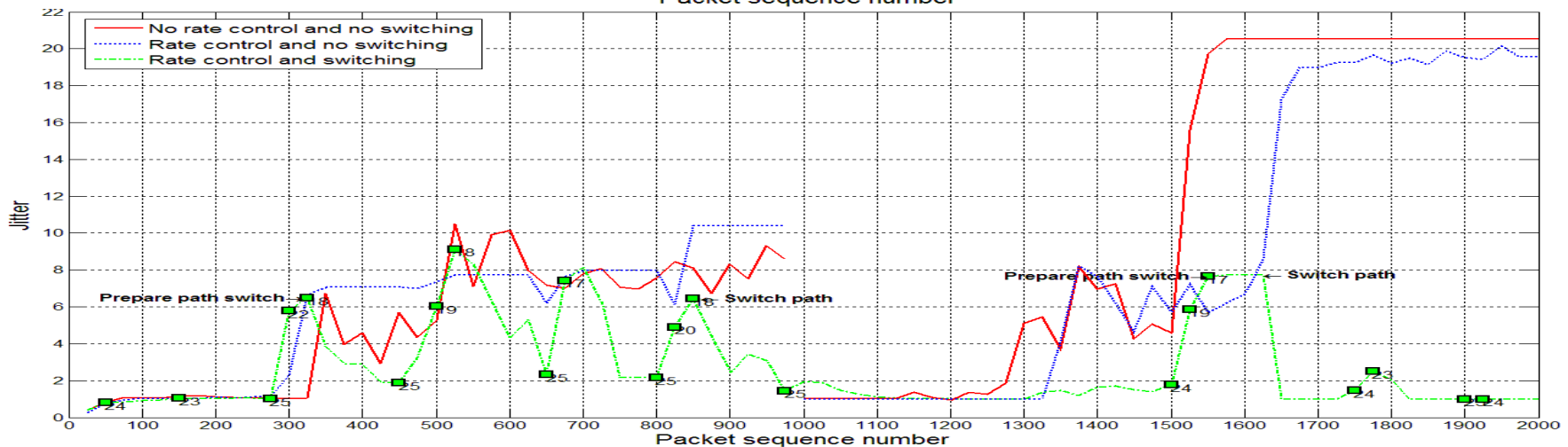
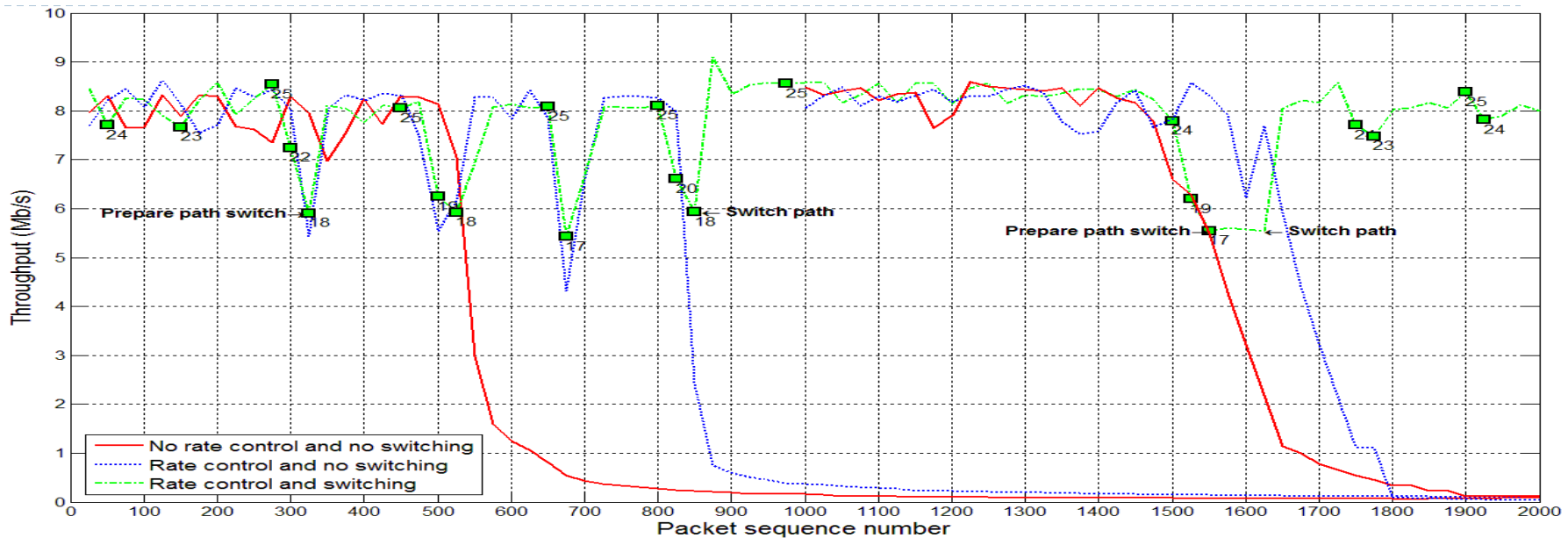
Testing Env. using GENI Testbed



Recognizing Network Load



Dynamic Path Switching



Experimenter Report

▶ Resources Used:

- ▶ GENI mesoscale deployment (ProtoGENI and MyPLC)
- ▶ OpenFlow topology (17 OF switches)
- ▶ Used VLANs (3715, 3716)

Experimenter Report (cont.)

▶ **Tools Used:**

- ▶ **Omni** command line tool to reserve and manage all resources
 - ▶ Great support, documentation, clear messages
 - ▶ Lots of commands involved but it can be easily scripted
 - ▶ Need to keep track of correct RSPECS for future use
 - ▶ Good number of RSPECS examples are available
- ▶ **Flack GUI** to generate RSPECS for the hosts
 - ▶ OF RSPECS still need to be written in a text editor
- ▶ **Click** software router to shape the bandwidth of each path (traffic shaper)
 - ▶ Very handy, easy to use
 - ▶ Good documentation
- ▶ **Floodlight** for the OpenFlow controller
 - ▶ Modular, easy to use and understand
 - ▶ Great examples and walkthroughs on the website
 - ▶ Portable and reliable
 - ▶ Highly recommended for those who want to learn about OpenFlow and see controller in action
- ▶ **Ipref** for congesting paths

Experimenter Report (cont.)

▶ Challenges:

- ▶ Network configuration and software related to reserve all resources and set up a reliable (redundant) topology
 - ▶ Identifying nodes and writing RSPECS are time consuming
 - ▶ All OpenFlow resources need to be manually approved by the FlowVisor support engineer -> delay
 - ▶ To modify the topology, you may need to start over because almost all slivers must be deleted
 - ▶ Must be done carefully to avoid broadcast storms
 - ▶ Not all hosts have the same OS -> can't be configured uniformly
 - ▶ Internal VLANS: not all the aggregate webpages have an up to date diagram and sometimes -> hard to know which VLANS to configure on the host interfaces
- ▶ Frequency in querying the OP switches for statistic updates

Experimenter Report (cont.)

▶ Suggested Improvements:

- ▶ Better organize the information currently available and make as up to date as possible
- ▶ Provide more examples for all host reservation RESPECS
- ▶ Automate RSPECS creation for OpenFlow and non-OpenFlow resources
- ▶ A website that shows resource availability dynamically
- ▶ The option of selecting the OS flavor to booted on ALL used hosts in the experiment -> make the configuration more uniform
- ▶ Having different types of OF controllers readily available and automating their configuration based on experimenter's needs
- ▶ A standalone, dedicated portal

Thank you!

