

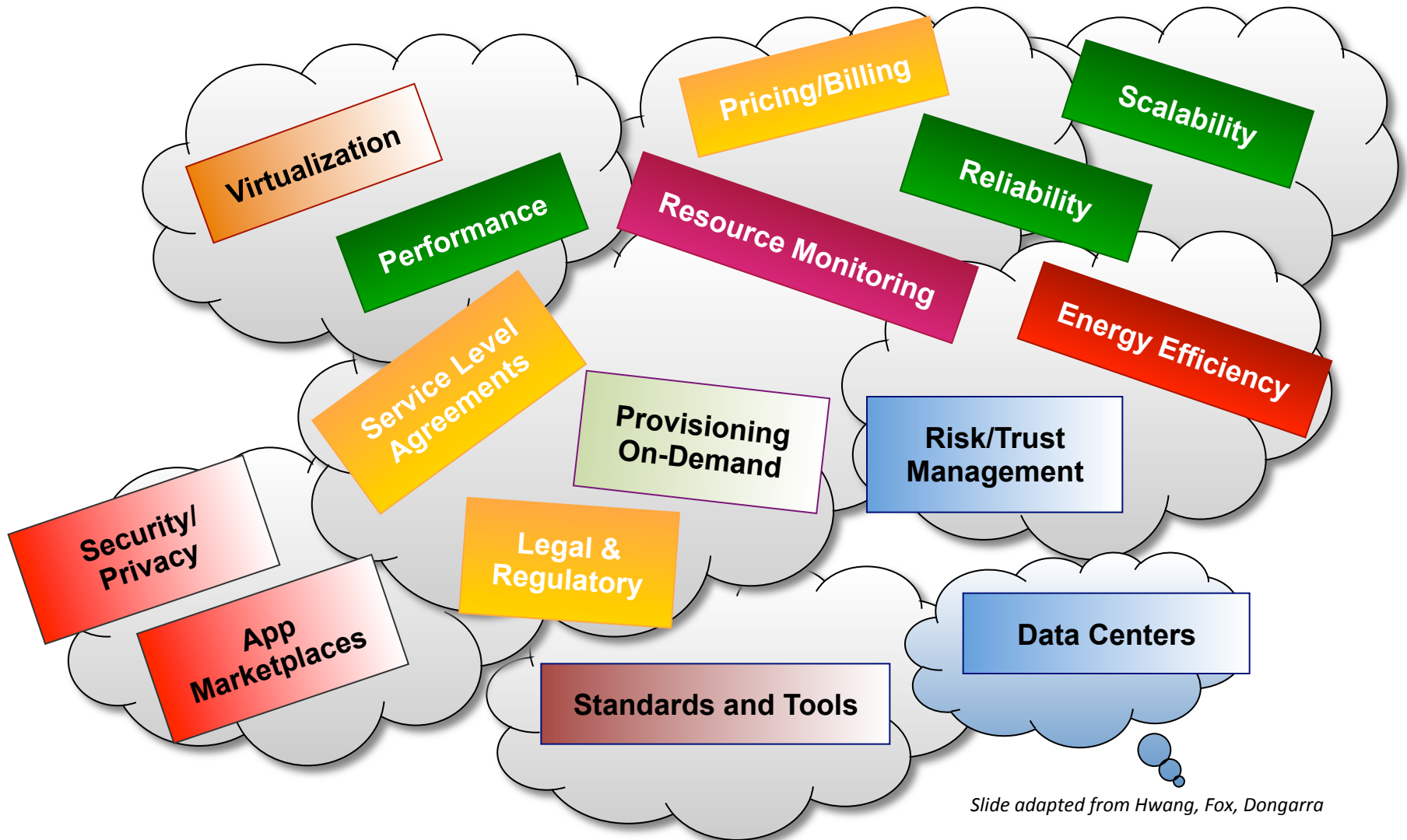
GENI Laboratory Exercises for a Cloud Computing course

Prasad Calyam, Ph.D.

Assistant Professor, Department of Computer Science

GEC19, March 2014

Cloud Computing Challenges?



Slide adapted from Hwang, Fox, Dongarra

Cloud Computing Course Objectives

- Be competent with principles and architectures that integrate computing theories and information technologies
- Be competent with distributed system models, computer virtualization concepts, network virtualization concepts
- Be competent with the design, programming and application of *distributed* and *cloud computing* systems

and...

- Be familiar with cyber-security and service-oriented concepts relating to use/design of cloud computing systems
- Be familiar with App customization of cloud computing infrastructures, and standards through hands-on experience

Cloud Computing Course Information

- **Pre-requisites:**
 - Object Oriented Programming
 - (Operating Systems-I), (Computer Networks-I), (Cyber Security-I)
- **Students:**
 - 23 Graduate, 7 Undergraduate, 5 visitors! 😊
- **Textbook:**
 - Distributed and Cloud Computing - by K. Hwang, J. Dongarra, and G. Fox; First Edition [ISBN: 9780123858801] (required)
- **Reference books:**
 - Programming Amazon EC2 - by J. Vliet and F. Paganelli [Online copy available] (optional)
 - Cloud Computing: Principles and Paradigms - by R. Buyya, J. Broberg, A. Goscinski (optional)
- **Teaching Assistant:**
 - Sripriya Seetharam – attended GENI Summer Camp, GECs, Thesis Research that uses GENI infrastructure
 - Worked closely with GPO (Thanks! Niky, Sarah and Vic)
 - Monitored and answered student issues in Blackboard Discussion Forum

Blackboard Discussion Forum


Discussion Board

Forums are made up of individual discussion threads that can be organized around a particular subject. Create Forums to organize discussions. [More Help](#)

Create Forum

Delete

| <input type="checkbox"/> Forum | Description | Total Posts |
|---|--|-------------|
| <input type="checkbox"/> AWS Lab-1 Discussion | | 7 |
| <input type="checkbox"/> GENI Lab-1 Discussion | | 15 |
| <input type="checkbox"/> Share Cool Tools for Labs/Projects | Please share any cool tools for AWS or GENI that you feel other students can use to more effectively do their labs and the final project. | 2 |
| <input type="checkbox"/> AWS Lab-2 Discussion | | 5 |
| <input type="checkbox"/> GENI Lab-2 Discussion | | 8 |
| <input type="checkbox"/> Register for MU Data Center Tour | <p>Hi All,</p> <p>As mentioned in class, I am arranging a MU Data Center Tour for you to see a "real data center"! The tour will be on November 6th at 1:30p. I will send data center address and meet-up details later.</p> <p>Please put a "Yes, I will attend the tour" note into this forum if you can come to the tour. This will help me in getting a count of students, and who will come to the tour.</p> <p>Thanks</p> <p>-Prasad</p> | 15 |

 Missouri GENI Rack Tour!

Course Lab Assignments

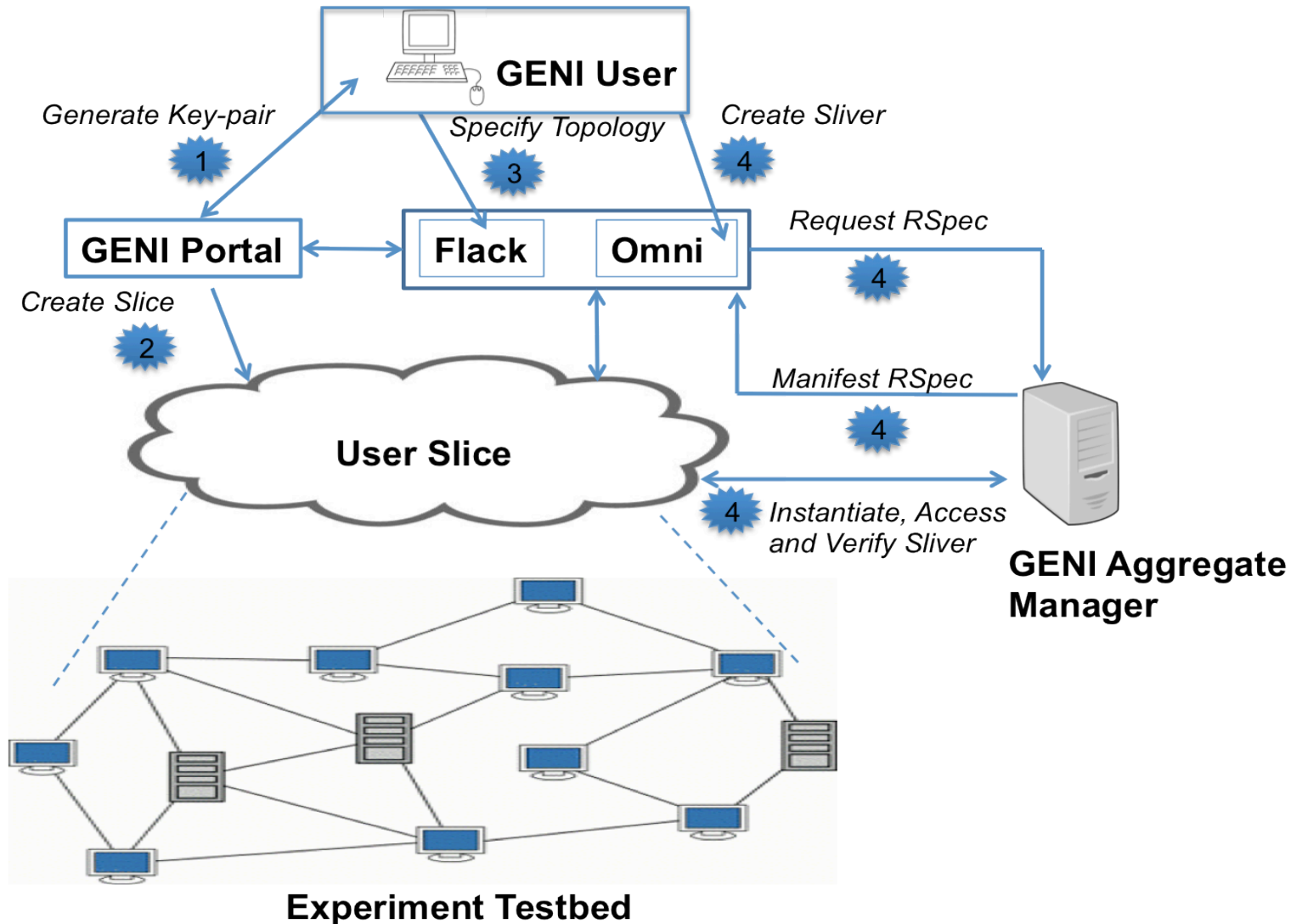
- 6 labs to develop technical background and skills for working with distributed system and related software environments
- 3 Amazon Web Services (AWS) labs - <http://aws.amazon.com>
 - Received \$100 usage credit per student through their academic program
 - Focus areas:
 - Instance setup with Cloud Watch alarm for billing alerts
 - Web App setup with EC2, S3 and Cloud Watch
 - Autoscaling to handle user load bursts in a Web App
- **3 GENI labs** - <http://groups.geni.net/geni/wiki>
 - “Calyam UMissouri Fall2013” Project in GENI portal (<http://portal.geni.net>)
 - **Focus areas:**
 - **Slice setup for “Hello GENI” experiment**
 - **Instrumentation & Measurement Web App setup with GENI Rack VMs and OnTimeMeasure software**
 - **OpenFlow controller programming for 2 traffic engineering applications: QoS Control based on enterprise policy, and Load Balancing**

GENI Lab-1 Exercise



- **Lab # 1 – GENI Account Setup and Services Overview**
- **Purpose of the Lab**
 - Setup your GENI account, and use it to reserve an experiment environment a.k.a. “slice” that features 2 compute nodes communicating over a Layer 2 connection
 - Run a “Hello GENI” experiment with GUI (Flack) and Command-line (Omni) tools in your reserved slice by automatic installation and execution of software to perform basic bandwidth measurement tests between the nodes

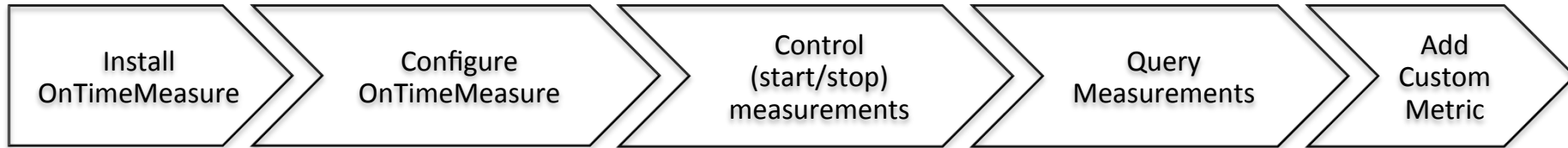
GENI Experiment Workflow



GENI Lab-1 Q&A

1. Provide a screenshot of the reserved slice resources in your GENI portal account
2. What would be the added capabilities and benefits in performing an experiment on GENI Future Internet infrastructure versus the commercial Internet?
3. Define: (i) slice, (ii) sliver, (iii) aggregate manager and (iv) Rspec
4. How does Federated Identity and Access Management in the GENI portal allow you to use your Pawprint and password to login? What are the benefits of this approach?
5. Explain the role of “Experiment Control Tools” such as Omni, Gush and Flack in GENI. What are the advantages of using the Omni command line tool in comparison to using the Flack GUI tool?
6. What is the significance of generating the SSH and SSL keys in your .ssh folder?

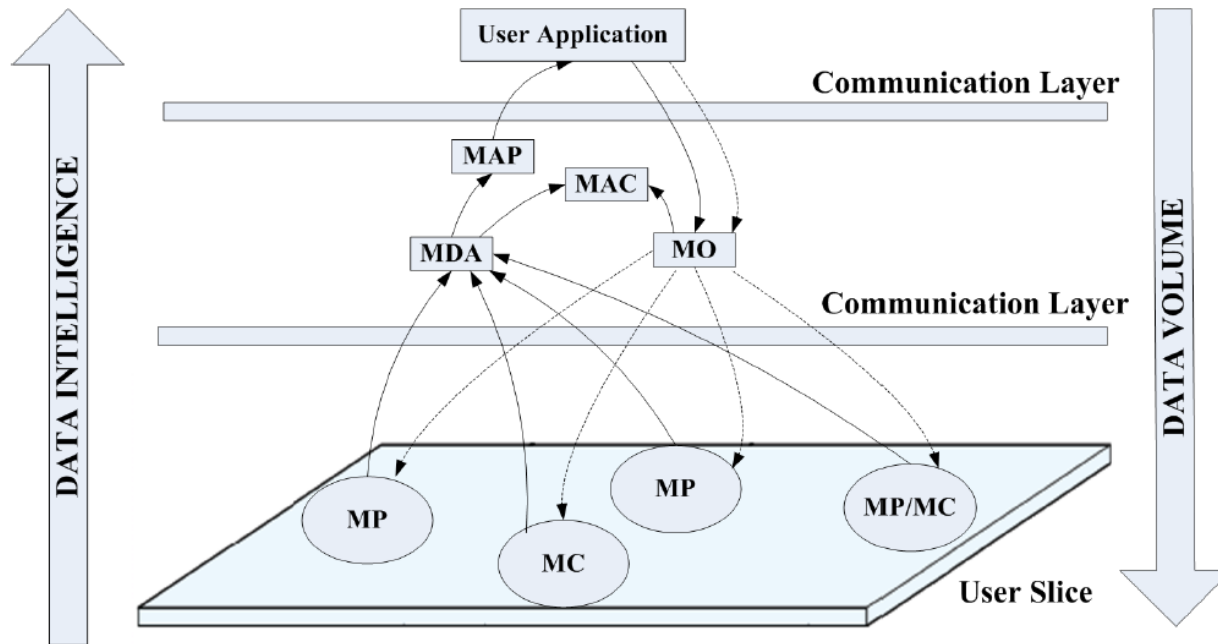
GENI Lab-2 Exercise



- **Lab # 2 – Instrumentation and Measurement of GENI Slice**
- **Purpose of the Lab**
 - Install and configure a distributed measurement Web App viz., “OnTimeMeasure” within a slice
 - Configuration involves ‘Node Beacons’ at measurement points, and a ‘Root Beacon’ that acts as a central intelligence module
 - Schedule measurements from the Root Beacon and query/visualize performance between two compute nodes that host Node Beacons
 - Performance data collection comprises of network health metrics such as round trip delay, loss, jitter and throughput

To administer this lab exercise on GENI Racks (versus using raw PCs as done previously), we had to make a substantial effort to make OnTimeMeasure run on VMs

GENI I&M Overview



Legend:

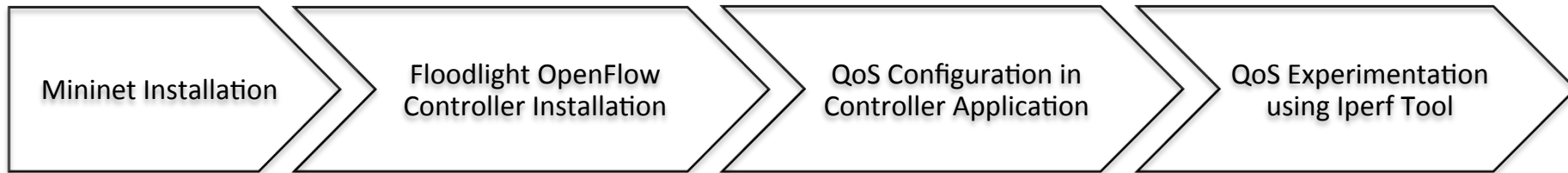
- | | | | |
|--------|-----------------------|-------|---------------------------------------|
| —▶ | Data Flow | MDA - | Measurement Data Archive |
| - - -▶ | Control Flow | MO - | Measurement Orchestration |
| MP - | Measurement Point | MAC - | Measurement Access Control |
| MC - | Measurement Collector | MAP - | Measurement Analysis and Presentation |

Reference:

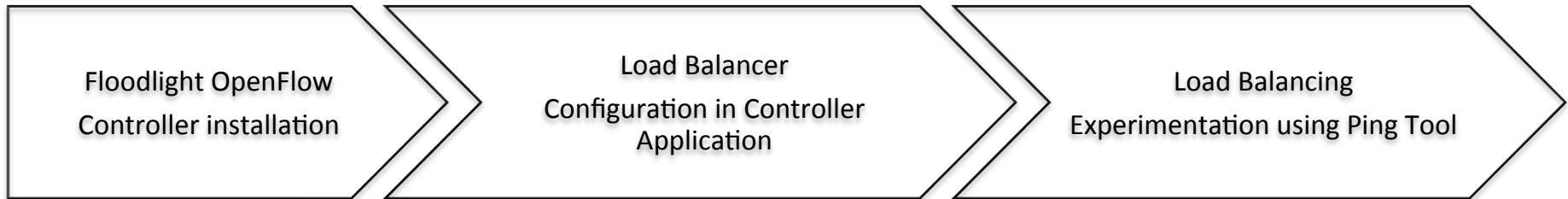
P. Calyam, M. Sridharan, Y. Xiao, K. Zhu, A. Berryman, R. Patali, "Enabling Performance Intelligence for Application Adaptation in the Future Internet", *Journal of Communications and Networks (JCN)*, 2011.

http://groups.geni.net/geni/attachment/wiki/FirstGenCalyam/fi-ontimemeasure-vdcloud_jcn11.pdf

GENI/SDN Lab-3 Exercise



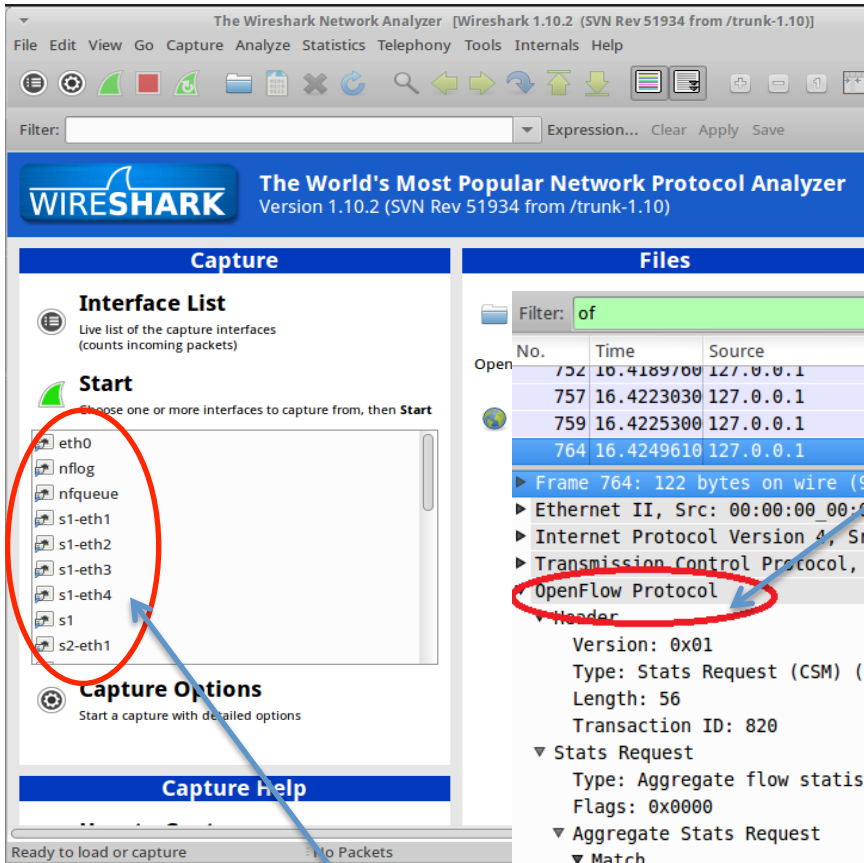
Lab Experiment #1 (QoS Control through Network-Edge Rate Limiting) Steps Overview



Lab Experiment #2 (Load Balancer for Scalable Handling of Traffic Flows) Steps Overview

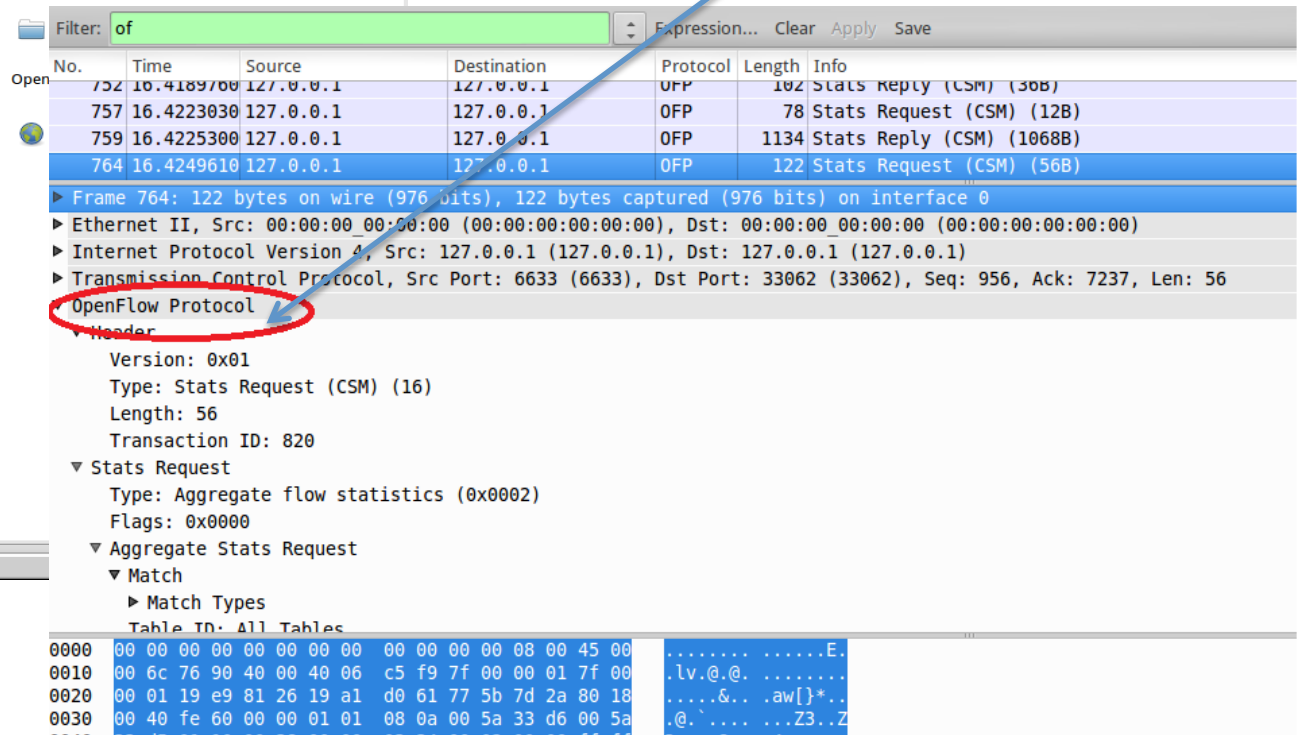
- **Lab # 3 – QoS Configuration and Load Balancing using Software Defined Networking/OpenFlow**
- **Purpose of the Lab**
 - Install and configure Mininet SDN emulator with 2 traffic engineering experiment applications to understand how to program ‘flow spaces’ within networks to: (i) comply with enterprise network capacity provisioning policies, and (ii) balance the utilization of network resources
 - Use Iperf and Ping Tools to verify your SDN functionality

Wireshark for Debugging your OpenFlow Controller!

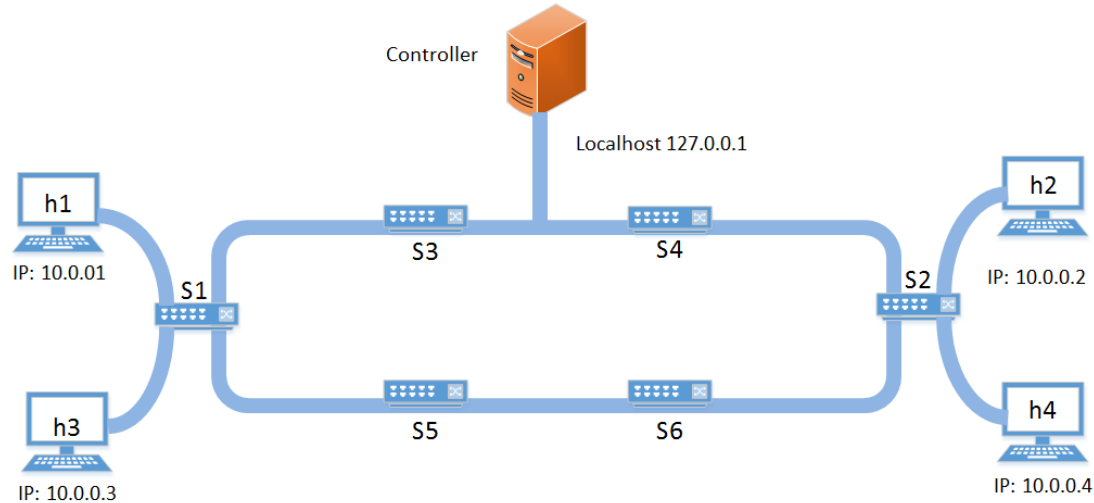


Virtual Ethernet ports for each switch

OpenFlow Protocol packet analysis

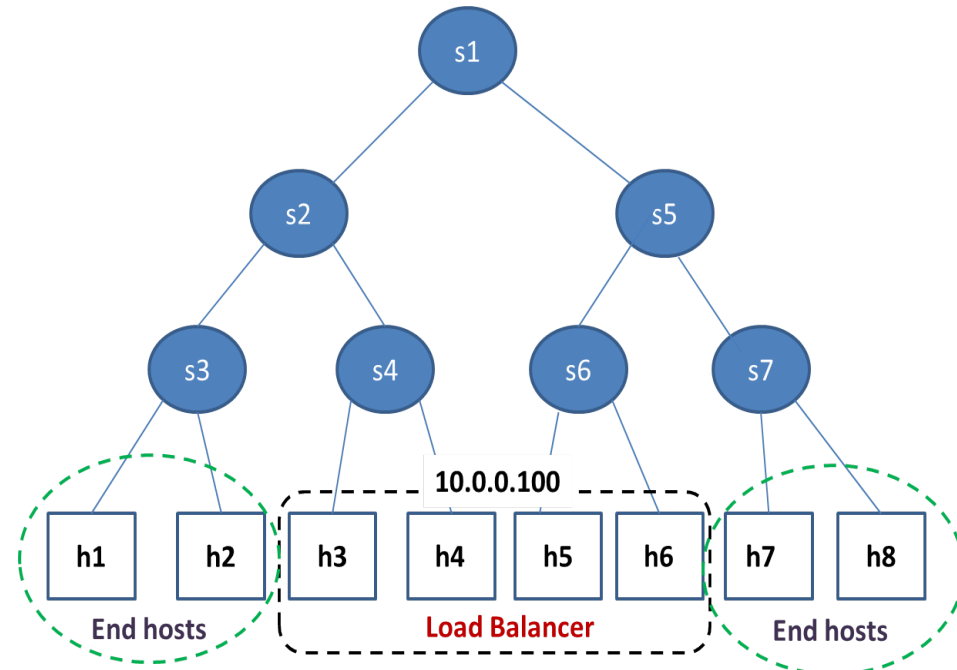


SDN/GENI Lab-3 Experiment #1



- Use the OpenvSwitch commands to set the network policies
- Setup 3 queues (Q0, Q1 and Q2) on every switch and configure network-edge bandwidth capacity using the 'ovs-vsctl' commands
 - Q0 – default queue
 - Q1 – queue 1 rate limiting bandwidth to 50 Mbps
 - Q2 – queue 2 rate limiting bandwidth to 40 Mbps

SDN/GENI Lab-3 Experiment #2



- Use a Load Balancing experiment topology with pools of end-hosts and load balancers
 - Test load balancing functionality with Ping requests from end-hosts
- Extend the 'Load Balancing' module in your Floodlight Controller
 - Scale the load balancer to handle more Ping requests by adding two new hosts to the load balancer pool
 - Examine the response patterns from end-host Ping responses

Final Course Project

- **Objective:**
 - To develop programming and performance evaluation skills
 - To reinforce the understanding of major concepts in the course
 - *Graduate Student team projects* require use of either C/C++ or Java, as well as cloud-platform specific tools and applications
 - *Undergraduate student participation is optional (extra credit)*
- **Approach:**
 - Solve a “real” problem in a GENI/AWS testbed
 - **Teams:** Core Logic Team, Testbed Team, User Interface Team
 - Must use web services, show design tradeoffs (e.g., *how VM obfuscation for ‘security’ complicates ‘manageability’ of infrastructure*)
 - Build upon Cloud Computing material covered in class
 - Student domain expert for each project
 - In-Class “Circle Time” to provide on-going guidance
 - Final Project Presentations/Demos
 - 20% Grade based on overall team performance, peer-reviews

Final Course Projects

- 1. Platform-as-a-Service for City-supported Hybrid Cloud**
 - Based on requirements from “ElderCare-as-a-Service” testbed in GENI
 - Concepts of Desktop-to-Cloud, Recommendation Engine
- 2. Software-defined Networking for Multi-tenancy**
 - Based on requirements from a MU-OSU “Science DMZ” testbed
 - Concepts of Network Virtualization, RoCE 10 Gbps Data Transfer Performance
- 3. Data Integrity Security Protection and Monitoring**
 - Based on requirements from “Thomson Reuters” testbed in GENI/AWS
 - Concepts of Distributed Trust, Anomaly Detection for Just-in-time News Feeds
- 4. App Marketplace for Advanced Manufacturing**
 - Based on requirements from “Simulation-as-a-Service” testbed in GENI/AWS
 - Concepts of App Workflow in Cloud, Freemium and Pricing Models

Other thoughts...

- Detailed instructions were needed to complete the labs
 - Using distributed system/network resources and working with tools such as those in GENI/AWS was a new experience
- AWS and GENI combination benefits
 - Many similarities in tools, best practices – reinforced learning!
- GENI Project Office co-ordination was important
 - To determine timing of labs, ensuring resource availability
- Open challenges in the course use of GENI
 - OpenFlow controller exercise on GENI infrastructure versus just using Mininet environment
 - Need to have two part course (Cloud Computing I & II) in subsequent semesters to have more in-depth GENI experiments

Thank you for your attention!

