

FINAL COURSE PROJECT

Final Course Project

- **Objective:**

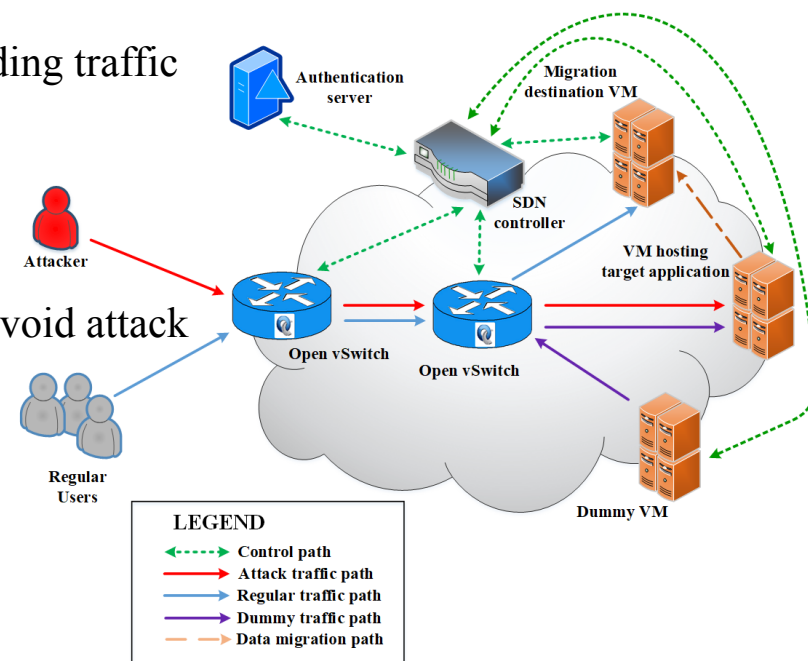
- To develop programming and performance evaluation skills
- To reinforce the understanding of major concepts in the course

- **Approach:**

- Solve a “real” problem on a GENI testbed
 - **Tasks:** Core Logic, Testbed, User Interface
 - Must use cloud resources, implement novel algorithm, show design tradeoffs
 - Build upon cloud systems knowledge developed in class
- In-Class “Circle Time” to provide on-going guidance
- Initial Idea, Mid-Progress, and Final Project Presentations/Demos In-Class
- At least **20%** of the grade (undergrad and grad) based on overall team performance

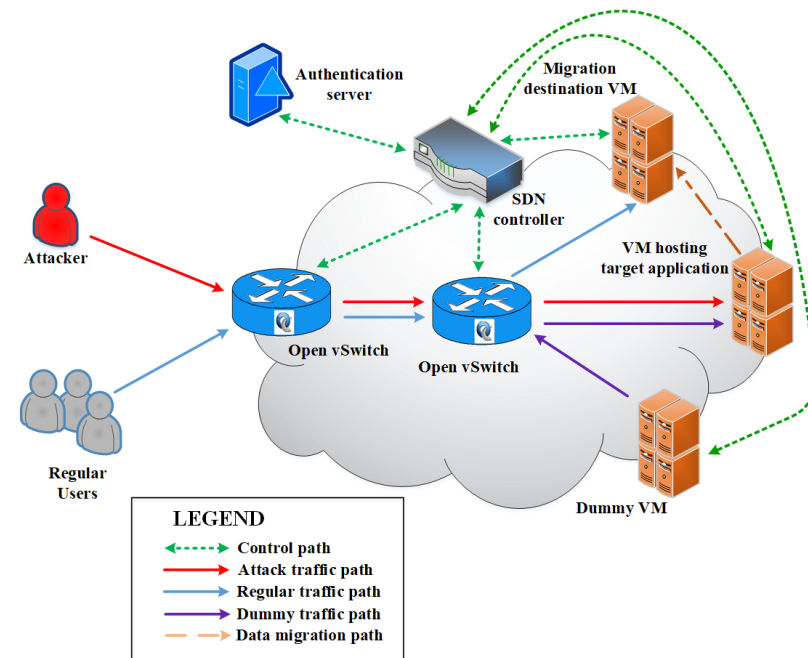
False Reality for Resilient Cloud Management

- Create a cloud topology in GENI
 - Cloud VM hosting a target video streaming service
 - OpenFlow switches through SDN controller forwarding traffic
 - Attacker and regular user traffic patterns
 - Dummy VMs on standby
- Implement migration and false reality
 - Proactive service migration to a new decoy VM to avoid attack
 - When to migrate?
 - Where to migrate?
 - Use reactive when attack is detected
 - Where to migrate?
 - Redirect only regular users to new VM using SDN
 - Create false reality on target VM with dummy VMs
 - How does the attacker understand success of a DDoS attack? What indicators?
 - How can those indicators be manipulated to make the attacker think that the attack is successful where actually it is not?



False Reality for Resilient Cloud Management

- UI for the cloud provider
 - Dials/knobs for system inputs
 - Visualization for outputs
- UI for the attacker
- Video playback for regular user(s)
- Run a complete experiment during the demo
 - Initiate operation through regular traffic
 - Initiate proactive migration
 - Launch the attack
 - Initiate reactive migration
 - Initiate false reality
 - Show system admin, attacker and user perspectives during the entire process
- Prepare a report on experiment evaluations
 - Changing the system parameters with knobs
 - Visualizing the performance metrics



Project Lifecycle

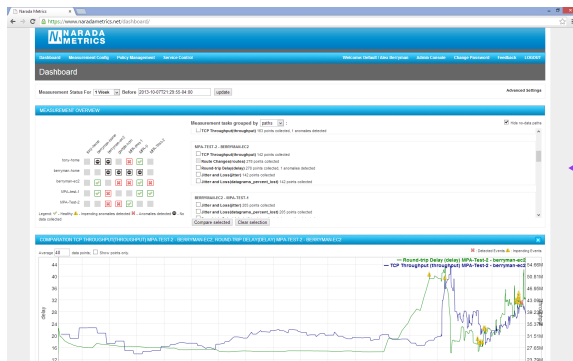
User



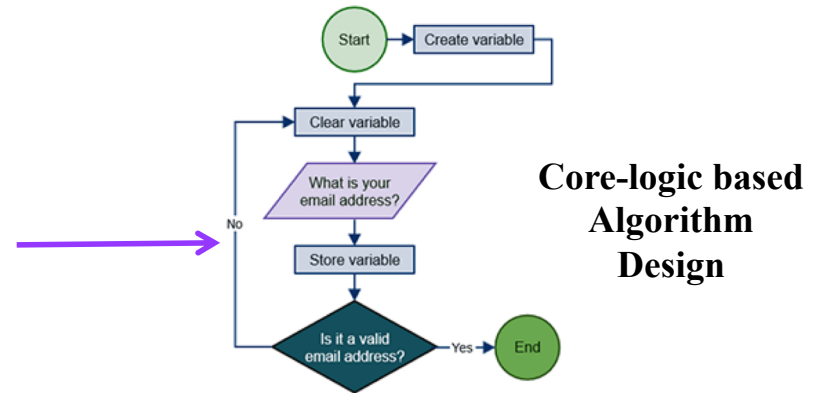
Science DMZ Application Dashboard

- Enter your university
- Enter the destination university
- Select the multimedia type of the data you would like to transfer:
- Is this a priority data transfer?
- Bandwidth specifications

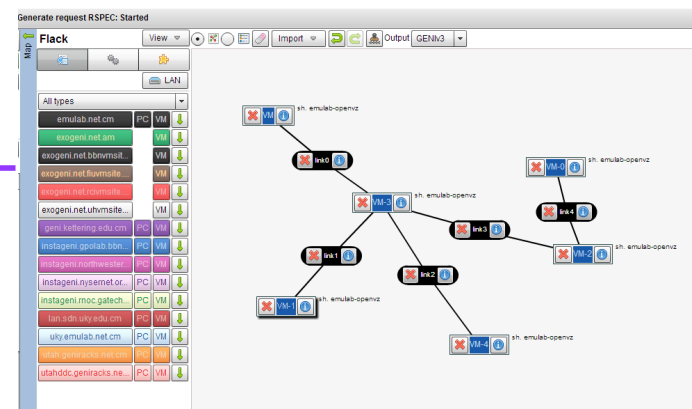
Application Dashboard



Performance Variance due to Design Tradeoffs



Core-logic based
Algorithm
Design



GENI Resource Reservation and Algorithm Implementation

Project Work Schedule

- **April 24th**
 - 2-Page project plan due via Blackboard
 - In-class UI mock-ups and core logic discussion
- **May 1st**
 - Mid-term Progress Presentation and Initial Demo in class (20 mins)
 - In-class testbed setup plan discussion
- **May 15th**
 - In-class discussion on final presentation and demo
 - Q&A
- **May 22nd**
 - Final presentation and demo
 - Final project report submission through Blackboard

Words of wisdom!!

- Start early and start aggressively!!
 - *A lot of work* to do
 - Imagine this to be a group capstone project
 - But only *4 weeks* left
- Ask a lot of questions!!
 - Clear any confusion or clarifications
 - Ask early and in details
 - I am here to guide you through the process
- Be modular and incremental at the same time!!
 - Divide and conquer
 - Start with smaller model and add more features gradually

Additional Information

Mid-progress and Final report and slides from each project team should organize information based on below sections checklist:

I. Project Title

II. Project Scope Overview

What are you trying to do? Articulate your objectives using absolutely no jargon.

How is it done today, and what are the limits of current practice?

What's new in your approach and why do you think it will be successful?

Who cares? *Think from both user side and service provider sides...*

If you're successful, what difference will it make?

What major concepts related to the course were used and reinforced in the project?

III. Team Members and their specific Contributions

Who did what? Who did a lot of what?

IV. User Interface Layout

Have you added the important implementation screen shots, either concepts or actual screens?

V. Core Logic Design

Have you added design diagram, described logic in detail and discussed tradeoffs?

VI. Testbed Setup

Have you added GENI testbed diagram, described details of testbed configuration and included all of the code written?

VII. Experiment results

How did you set up the experiments? What were the goals? How do you think your system performed? Discussion of results.