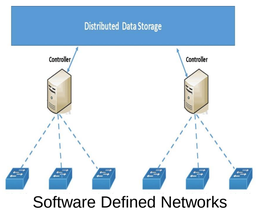
VIRO-GENI: Deployment of a plug & play, scalable, robust virtual Id routing in GENI

Braulio Dumba, Guobao Sun, Hesham Mekky, Zhi-Li Zhang
Department of Computer Science, University of Minnesota-Twin Cities, MN, 55455

MOTIVATION

- **IP Known limitations:**
  - Poor support for mobility
  - Need of careful and extensive network configurations
- **VIRO-GENI project goal:** to implement VIRO as a non-IP service in GENI using the SDN platform
- **VIRO:** three main components
  - Virtual Id space construction
  - Routing tables computation
  - Data forwarding using virtual Ids

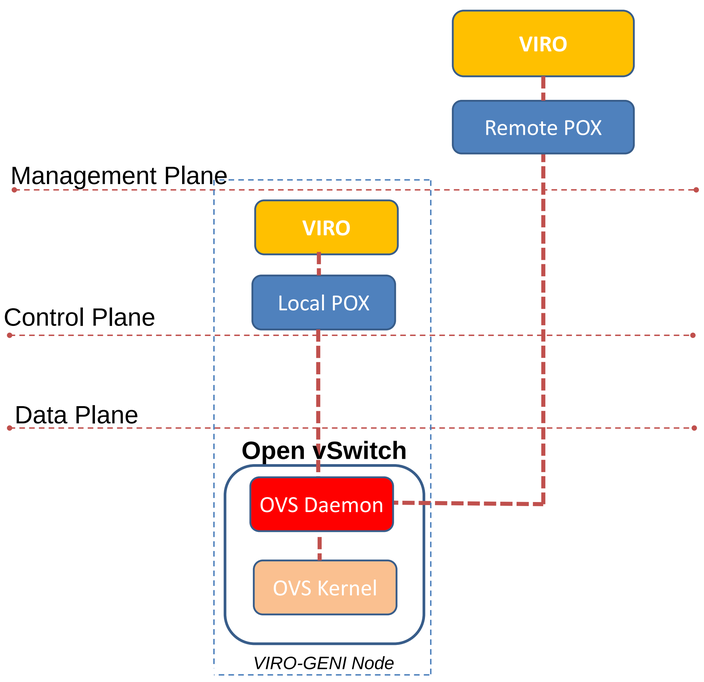


Software Defined Networks

- **CHALLENGES:**
  - OpenFlow protocol flexible forwarding behavior is still tied to the standard **Ethernet/IP/TCP** protocol task
  - VIRO has its own "topology-aware" addressing and forwarding behavior, where forwarding is based on the destination **vid** and the **forwarding directive**

IMPLEMENTATION OF VIRO IN GENI

VIRO-GENI NODE



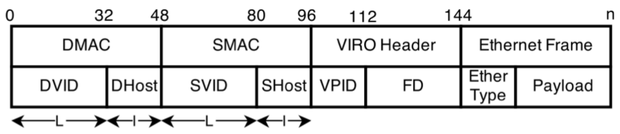
CONTROL PLANES

- **Management Plane :** VIRO remote controller is responsible for the following tasks:
  - topology discovery/maintenance (host/switch added/removed)
  - Vid assignment
  - ARP and DHCP Requests
  - IP/VID Mapping (Global View)
- **Control Plane:** VIRO local controllers are responsible for the following tasks:
  - MAC/VID Mapping (Local View)
  - Populate Routing Table
  - Insert forwarding rules for the first packet of any flow

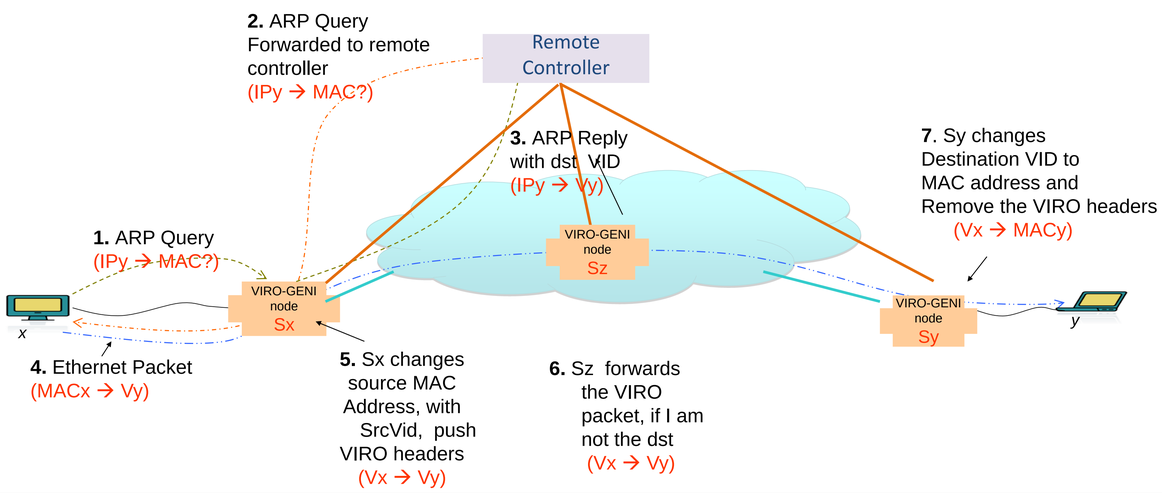
DATA PLANE

- **OVS Daemon:**
  - Translation between IP packets/VIRO packets (EtherType, Forwarding Directive)
  - Insert rules for routing at Kernel
- **OVS Kernel:**
  - Translation between IP packets/VIRO packets (End-Host)
  - Forwarding IP packets among local machines
  - Forwarding VIRO packets

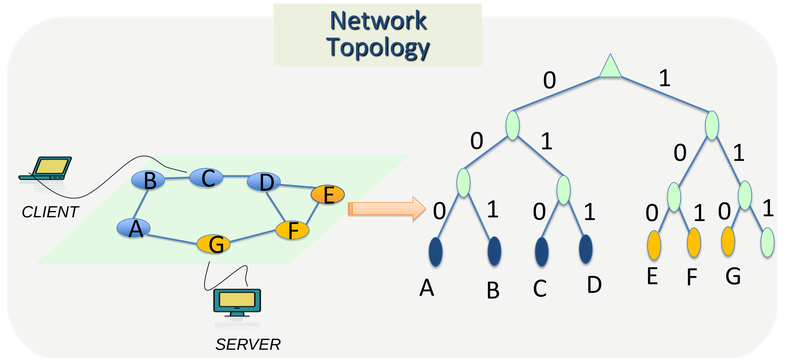
VIRO PACKET FORMAT



ADDRESS /VID RESOLUTION AND DATA FORWARDING IN VIRO-GENI



DEMO OF INITIAL PROTOTYPE

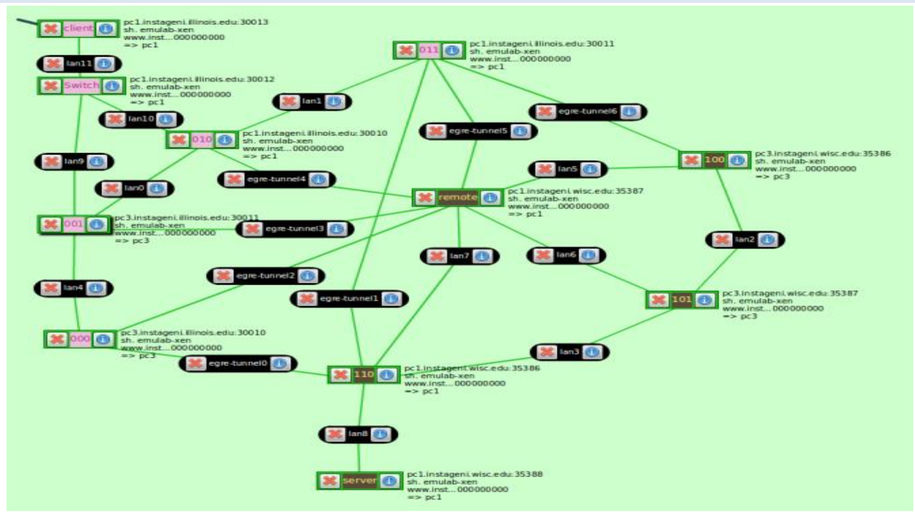


- Our **goal** is to show how **VIRO** handles **links/switch failures** and **host mobility**.
- To pursue this goal, we carried out experiments in GENI using a network topology with **7 nodes**

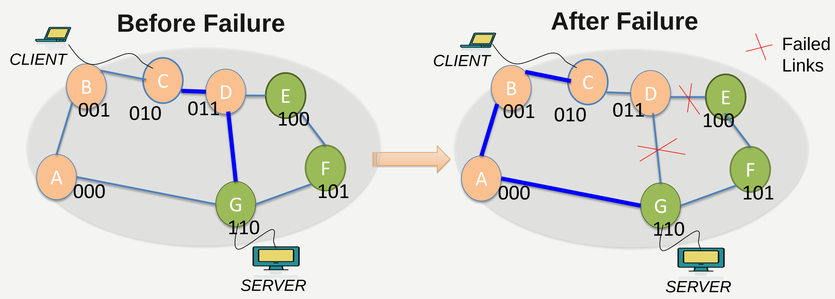
- The leaf nodes in the binary tree represent VIRO switches
- The color of the nodes represent GENI Aggregate Managers (AM)
- We attached a client host at node **C** and an Apache server at node **G**

DEMO SET-UP IN GENI

We use two GENI AMs (**Wisconsin and Illinois**) , 11 XenVMs and 4 PCS in our experiment. EGRE tunnels were used to connect nodes at different GENI AMs.



Link Failure Experiment



Bucket Distance	Next hop	Gateway
1	D	C
2	B	C
3	D	D

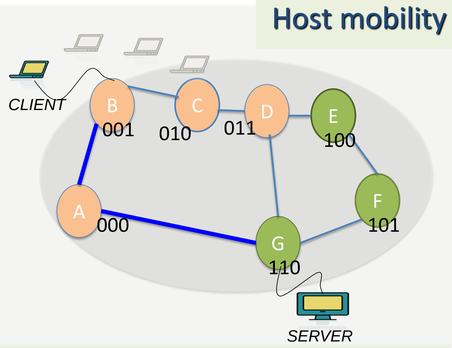
Bucket Distance	Next hop	Gateway
1	D	C
2	B	C
3	B	A

Routing Table for node C (Round 3)

- **In this experiment the client at node C downloads a large image from the server at node G**
  - **Before Failure:** Node **C** uses its level-3 gateway (node **D**) to communicate with the server
  - **After Failure:**
    - Node **D** updates its routing table and sends a gw withdraw message to level-3 rdv (node **A**)
    - Node **A** updates its rdv store and sends a gw removal message to node **C**
    - Node **C** updates its routing table and queries for level-3 gw
    - Level-3 rdv (node **A**) returns itself as the new level-3 gw

- VIRO handles node/link failures , without resorting to flooding of failure notifications (as used in OSPF). Instead, it utilizes a withdraw & update mechanism.

Host mobility Experiment



- **In this experiment the client host moves from node C to B while downloading a file from the server at node G**
  - The client is assigned a new vid, after moving to node B
  - The server issues ARP request to the remote controller to find the client new vid
  - The client TCP connection is unaffected during this process

VIRO topology-aware, structure virtual id (vid) space offers support for host mobility