In-Network Dynamic Pathlet Switching with VIRO for SDN Networks

Braulio Dumba, Hesham Mekky, Guobao Sun, Zhi-Li Zhang

Motivation

- The conventional **best-effort** IP protocol cannot readily provide the bandwidth and other service guarantees required by many of today's multimedia applications

Multimedia Streaming

Video Teleconferencing

Games

VoIP

Online TV

Motivation

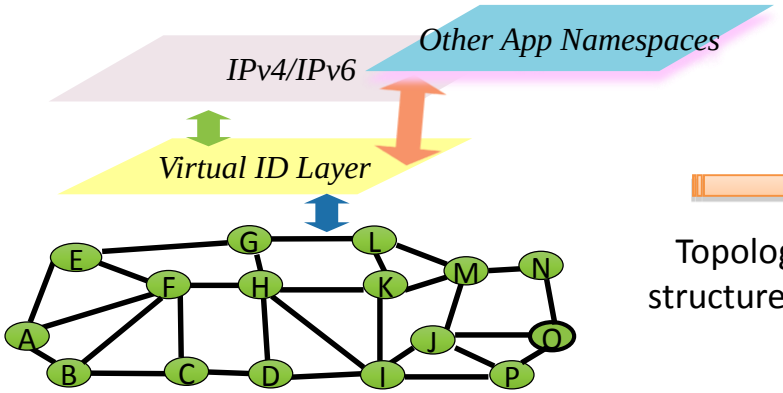
- Current Solutions:
  - QoS mechanisms
  - Overlay networks or stub networks
  - Multi-path TCP (**mpTCP**)
- Limitations of current solutions:
  - QoS mechanisms:
    - difficult to deploy widely and add extra **complexity** in the network
  - All other solutions:
    - **end-systems** (or **stub networks**)-based
    - No **explicit** information regarding the path **diversity** available in the network
    - Require end-systems (or stub-networks) to be **multi-homed**

Project Goal:

- We propose a novel **in-network pathlet switching** framework for SDN networks using **VIRO**, a non-IP routing protocol, to fully exploit the path diversity available in the networks

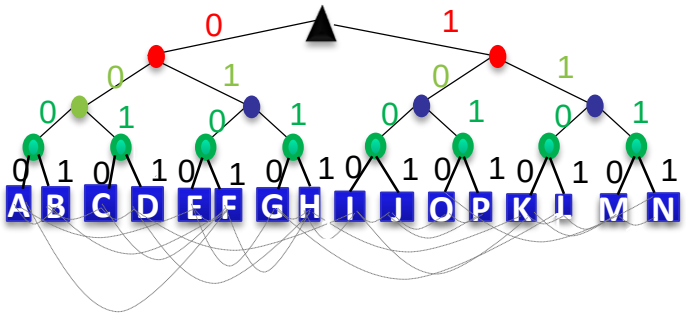
VIRO: Virtual Id Routing

- Non-IP protocol designed to address IP limitations
- Key Components: (**Vid Id space construction**)

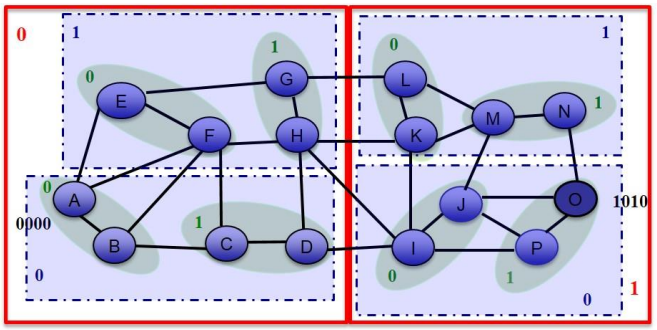


Layer 2 Physical Network Topology

Topology-aware, structured vid space



A node vid encodes its topological location in the network



Network hierarchically partitioned into subtrees of varying levels

VIRO: Virtual Id Routing

- Key Components: (**Routing Tables and Forwarding**)

Node A routing table

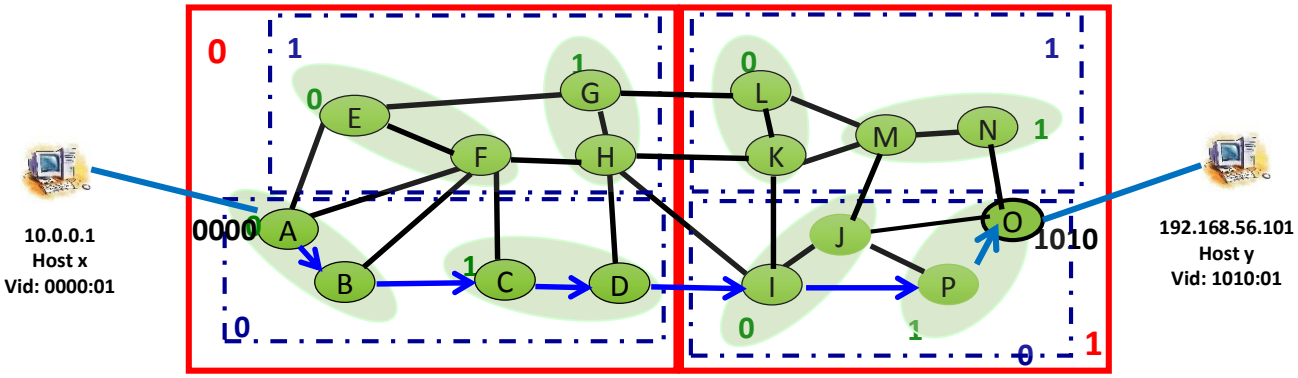
Bucket Distance	Prefix	Next hop	Gateway
1	0001	B	A
2	001*	B	B
3	01**	E, F	A
4	1***	E, F, B	G, H, D

Nodes publish-&-query gateways information to reach specific level of the vid space to rendezvous point

Gateway: is a node that has a direct (physical) edge to a node in a neighboring subtree

Rendezvous Point: nodes that store gateways information to reach specific levels in the vid space:

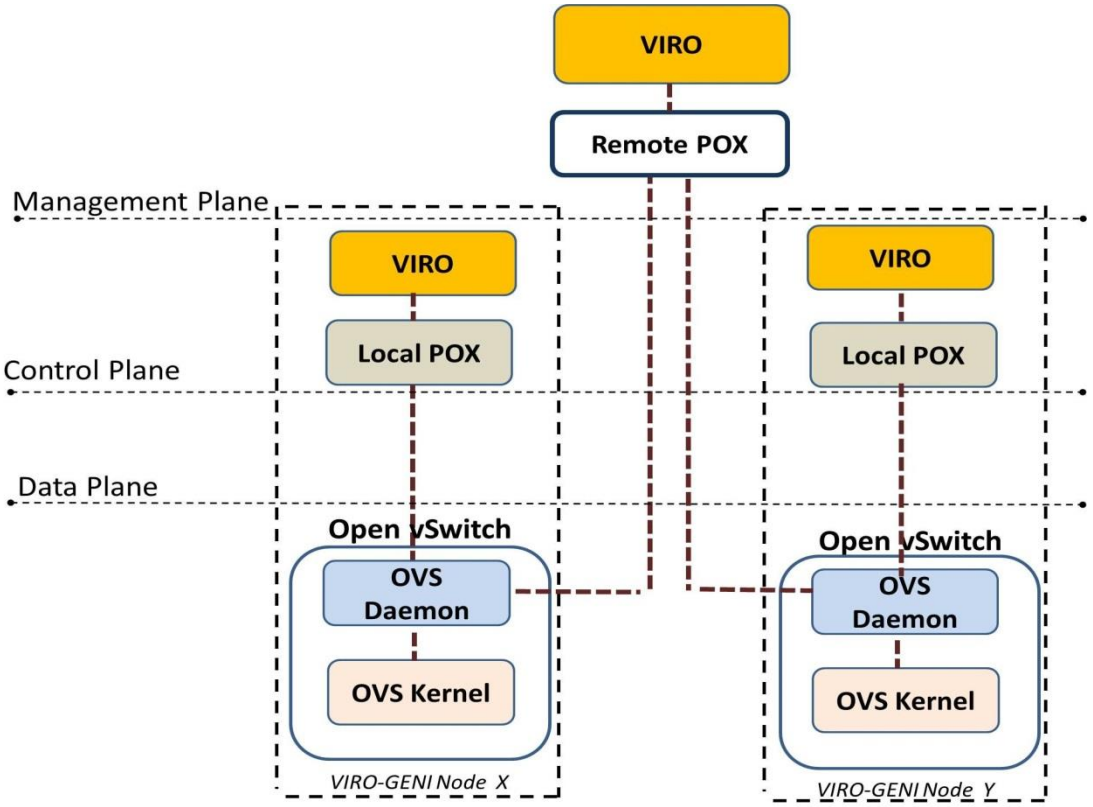
- **[Level, gateway list]**
- **{(GWx: nodex, nodey); (Gwy: nodeK, nodez); ..}**



Forwarding using vids only and at the network edges the vids are mapped to host identifiers (e.g. MAC/IP)

Implementation & Deployment of VIRO in GENI

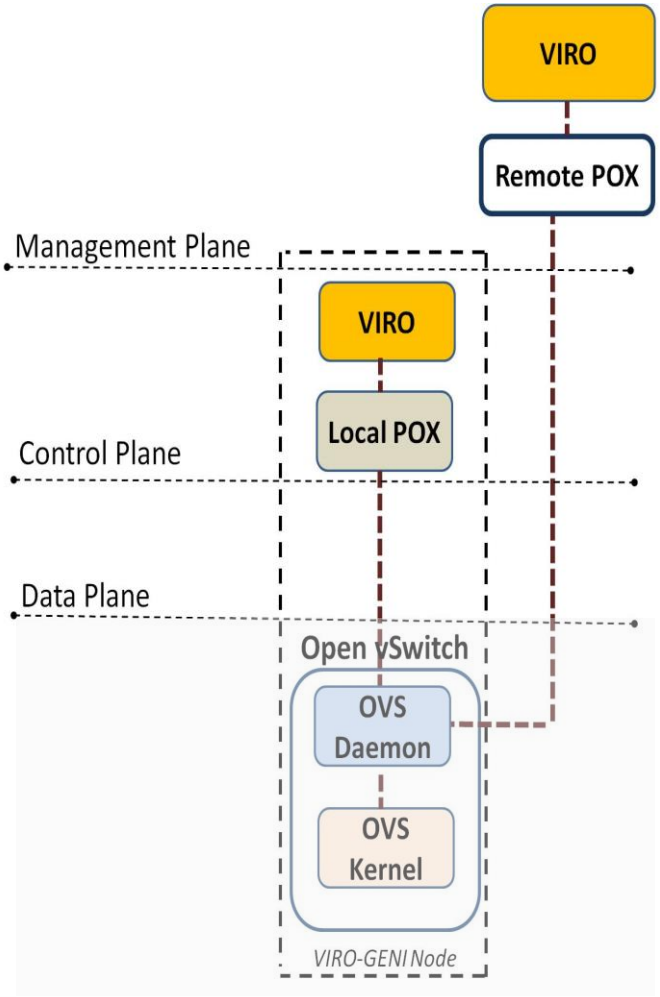
- **VIRO-GENI node architecture**



Implementation & Deployment of VIRO in GENI

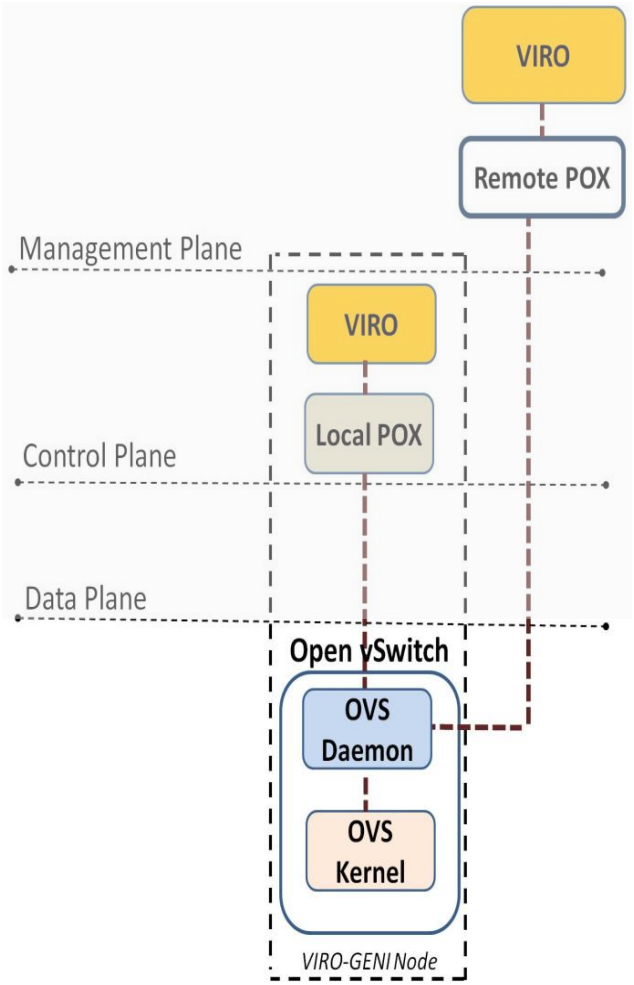
- **Management Plane**
  - Topology discovery/maintenance
  - (host/switch added/remove)
  - Vid Assignment
  - ARP and DHCP Requests
  - IP/VID Mapping (Global View)

- **Control Plane**
  - MAC/IP/VID Mapping (Local View)
  - Populate Routing Table
  - Insert forwarding rules for the first packet of any flow



Implementation & Deployment of VIRO in GENI

- **Data Plane**
  - OVS Daemon
    - Translation between IP packets/**VIRO packets** (Ether Type, Forwarding Directive)
    - Insert rules routing at Kernel
  - OVS Kernel
    - Translation between IP packets/**VIRO packets** (End-Host)
    - Forwarding VIRO packets
    - Forwarding IP packets among local machines



Dumba et. al, "Experience in Implementing & Deploying a Non-IP Routing Protocol VIRO in GENI.'', CNERT'14, co-located with IEEE ICNP'14

We are **extending** our VIRO framework in GENI with mechanism for **in-network pathlet switching**

In-Network Dynamic Pathlet Switching Framework

- **In-network pathlet switching** is a mechanism that allows network devices (e.g. routers, switches) to dynamically switch among several paths to a destination based on their performance

- Two **conditions** need to be met:

1) **Performance** information of the current path and all the alternative paths in the network

2) **Mechanism** and/or **component** responsible for making the path switching decision inside the network

In-Network Dynamic Pathlet Switching Framework

- **Performance Information:** **Latency** and **Throughput**
- **Latency** information is collected by adding additional mechanism to measure the latency to gateway per node in VIRO
- We use the statistics reported by OVS to get the **throughput** information per gateway or flow throughput

In-Network Dynamic Pathlet Switching Framework

Path Switching Components

VIRO Local Controller

- Estimate per level gateway throughput
- Periodically report gateway throughput information to the Remote controller and Rendezvous Point
- Query upper-level gateway's throughput information from the rendezvous point

VIRO Remote Controller

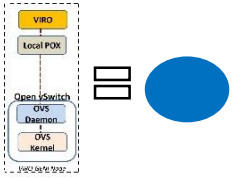
- Maintains a global view of the network
- Receives the list of gateways from the rendezvous point in the network
- Coordinate with the local controller and rendezvous point in order to initiate path switching in the network

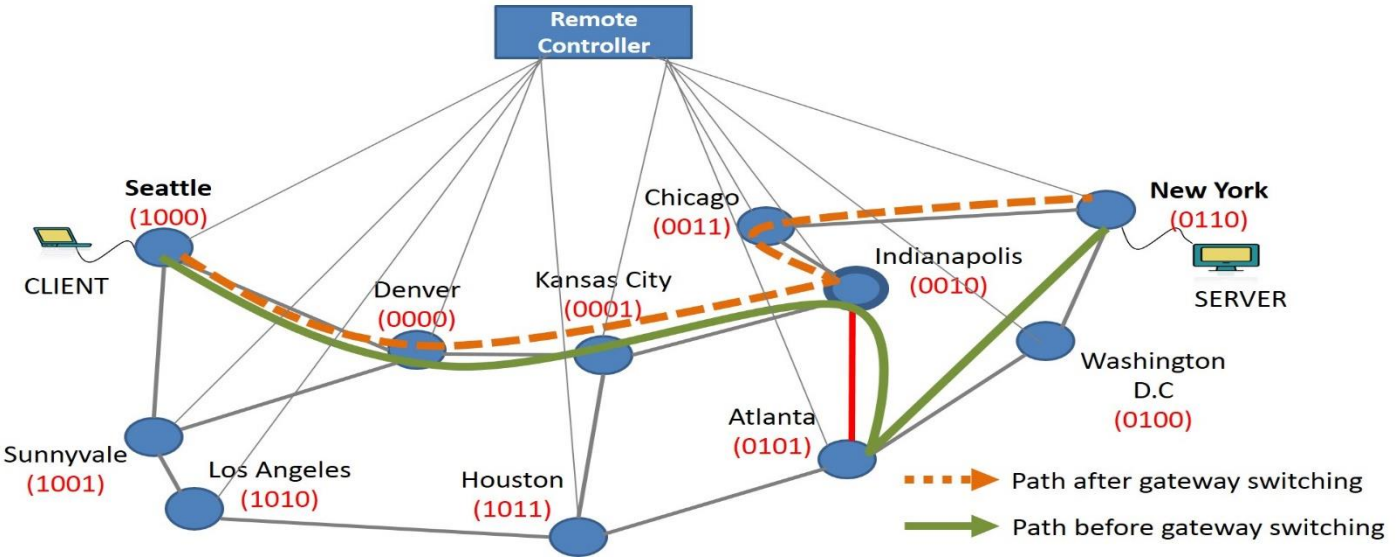
VIRO Rendezvous Point

- Maintains a list of gateways and their throughput information for some levels in the binary tree
- Send gateway failure notifications for the nodes using the gateways in its rendezvous store
- Reply to gateway query message from the local controller

In-Network Dynamic Pathlet Switching Framework

- **End-to-End Example**





Event: congestion in link **Indianapolis – Atlanta**

1. **RC** notifies **Rdv** Denver about the poor performance of level-3 gateway (GW) Indianapolis
2. **Rdv** Denver notifies node Kansas City about GW Indianapolis poor performance
3. Kansas City **LC** updates its level-3 gateway from Indianapolis to Chicago (**Path-switching**)

Key Points

- Path switching is performed **inside** the networks

- Path switching decisions can be made at different **levels** in the network:
  - VIRO's routing capabilities and OVS's statistics used to make **local** and **global** path switching decisions

- Our **solution** can potentially react to network performance degradation **faster** than the traditional methods used for path switching at the **end-points**

Conclusion

- In computer networks is feasible to avoid paths of low **performance** when other alternatives are available

- We propose an In-network dynamic pathlet switching framework with **VIRO** for SDN Networks

- On-going Work:

  - Developing algorithms that take advantage of the different levels of **decisions** for path switching

  - Evaluate the **scalability** and **performance** of our framework in GENI over larger topologies

THANK YOU!
QUESTIONS???