In-Network Dynamic Pathlet Switching with VIRO for SDN Networks

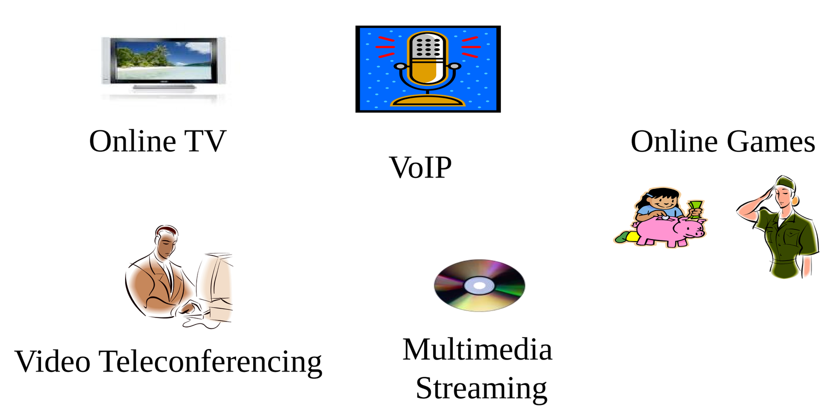
Braulio Dumba, Hesham Mekky, Guobao Sun, Zhi-Li Zhang

Department of Computer Science, University of Minnesota-Twin Cities, MN, 55455

This project is supported in part by a Raytheon/NSF subcontract 9500012169/CNS-1346688, the NSF grants CNS-1017092, CNS-1117536 and CRI-1305237, and the DTRA grant HDTRA1- 09-1-0050.

MOTIVATION

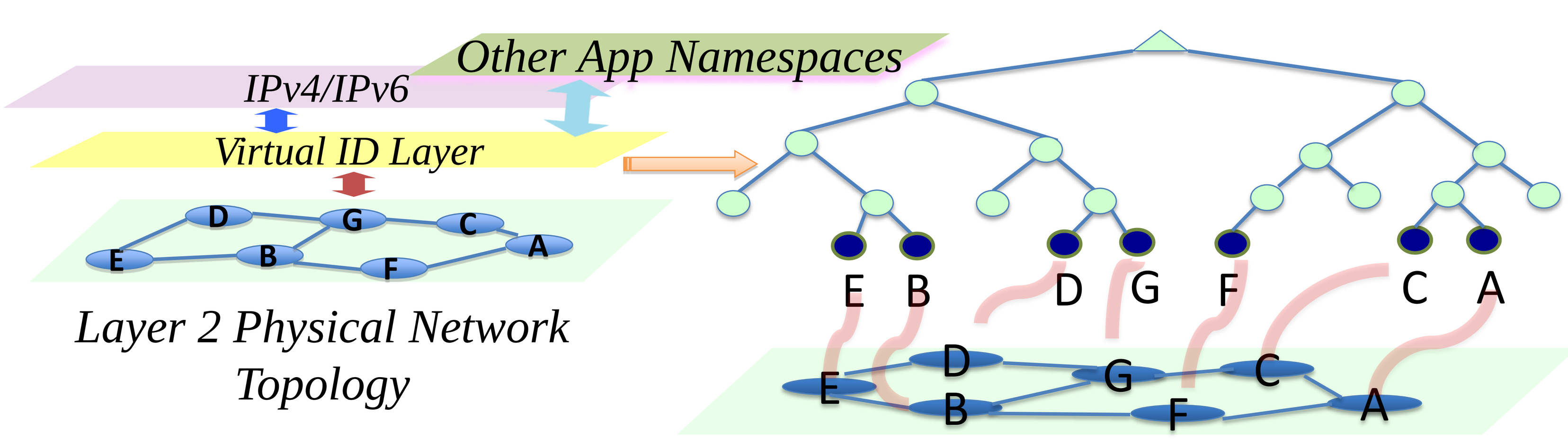
- The conventional **best-effort IP** protocol cannot readily provide the bandwidth and other service guarantees required by many of today's multimedia applications



- **Project Goal:** we propose a novel **in-network pathlet switching** framework for SDN networks using the VIRO routing protocol to fully exploit the path diversity available in the networks

- **Current Solutions:**
  - **QoS mechanisms**
  - **Overlay networks or stub networks**
  - **Multi-path TCP (mpTCP)**
- **Limitations of current solutions:**
  - **QoS mechanisms are difficult to deploy widely and add extra complexity in the network**
  - **All the other solutions are end-system (or stub network)-based and therefore do not have explicit information regarding the path diversity available in the network**

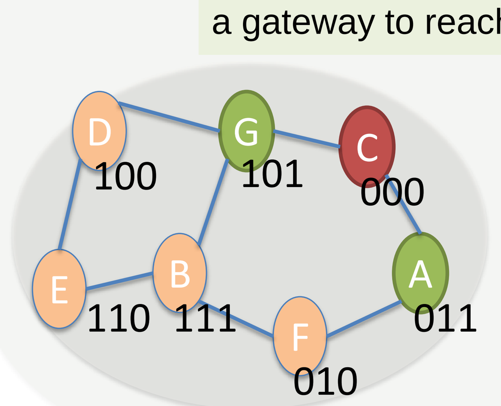
VIRO: VIRTUAL ID ROUTING PROTOCOL



VIRO Routing

- Inspired by Kademlia DHT
  - but need to build end-to-end connectivity/routes!
- Bottom-up, round-by-round process
  - round 0: neighbor discovery
    - discover and find directly/locally connected neighbors
  - round k (1<= k <=L):
    - build routing entry to reach one's own level-k sub-tree
      - a list of one or more (gateway, next-hops)
    - use "publish-query" (rendezvous) mechanisms

Logical Distance: Height of the nearest common parent in the binary tree.
Sub-tree: Set of nodes with in a given logical distance.
Rendezvous Point: a node that store gateway information to reach specific levels in the vid space.
Gateway: A node in SubTree(k-1,x) which is connected to a node in Bucket(k,x) is a gateway to reach Bucket(k) for node x.



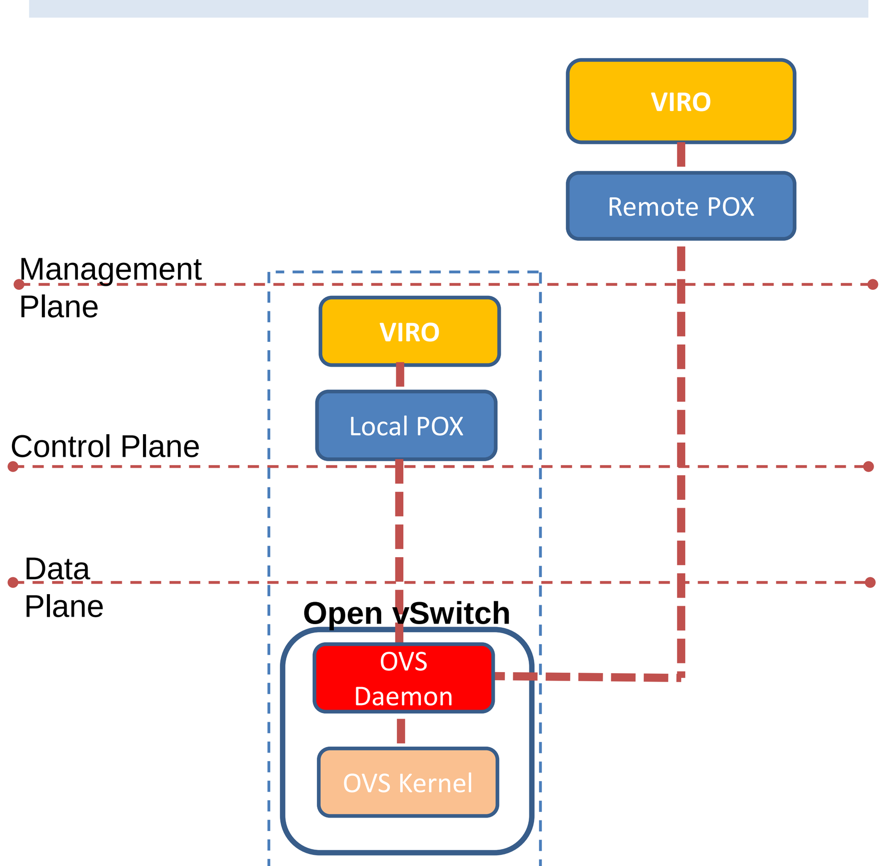
Bucket Distance	Next hop	Gateway
1	D	C
2	B	C
3	D	D

VIRO Routing Table for **node A** (Round 3)

- The connectivity information store at a rendezvous point (rdv): [**level, gateways list**]
- Each rdv also maintains a list of the nodes using its list of gateways:**{(GWx: nodex, nodey); (Gwy: nodeK, nodez); ..}**

IMPLEMENTATION OF VIRO IN GENI

VIRO-GENI NODE



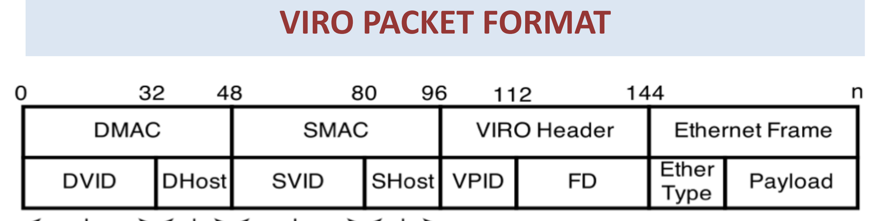
CONTROL PLANES

- **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



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
- To achieve in-network pathlet switching the following **conditions** need to be met:
  a) Performance information of the current path and all the alternative paths in the network
  b) Mechanism and/or component responsible for making the path switching decision inside the network

Pathlet 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

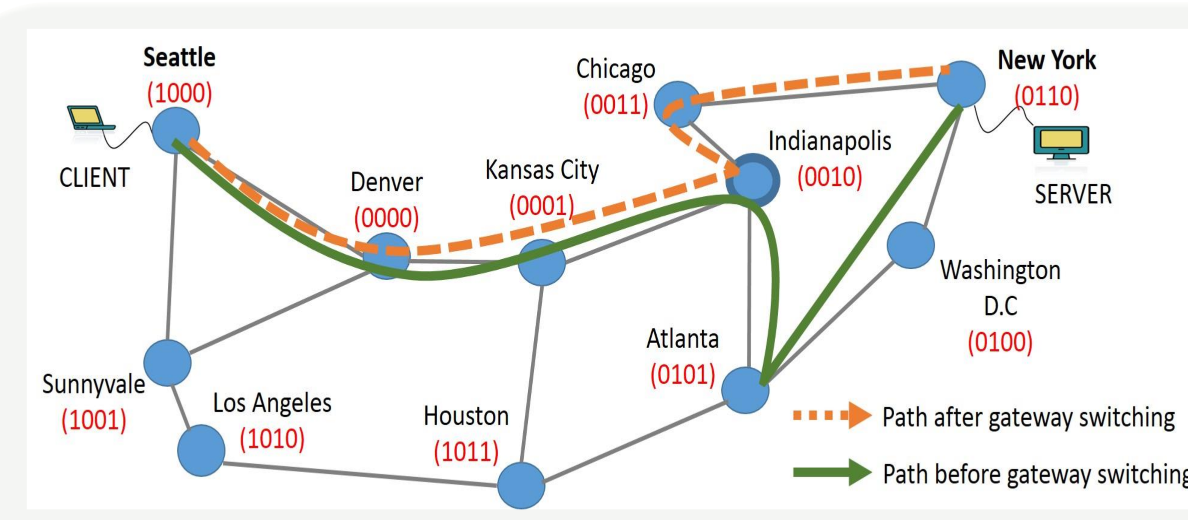
VIRO Rendezvous Point

- 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

EXPERIMENTS

- We carry out experiments to investigate the potential benefits of in-network pathlet switching with VIRO and compared it against MPTCP:
  - Using the **Abilene** network a client in Seattle communicates with a server in New York
  - The network tool **iperf** is used to generate traffic from client to sever for 150 seconds
  - We use openFlow rules into **OVS switches** to set-up all the paths in our experiments

a) In-Network Pathlet Switching with VIRO



Bucket Distance	Next hop	Gateway
1	Chicago	Indianapolis
2	Kansas City	Indianapolis
3	**Atlanta**	**Indianapolis**
4	Kansas City	Kansas City

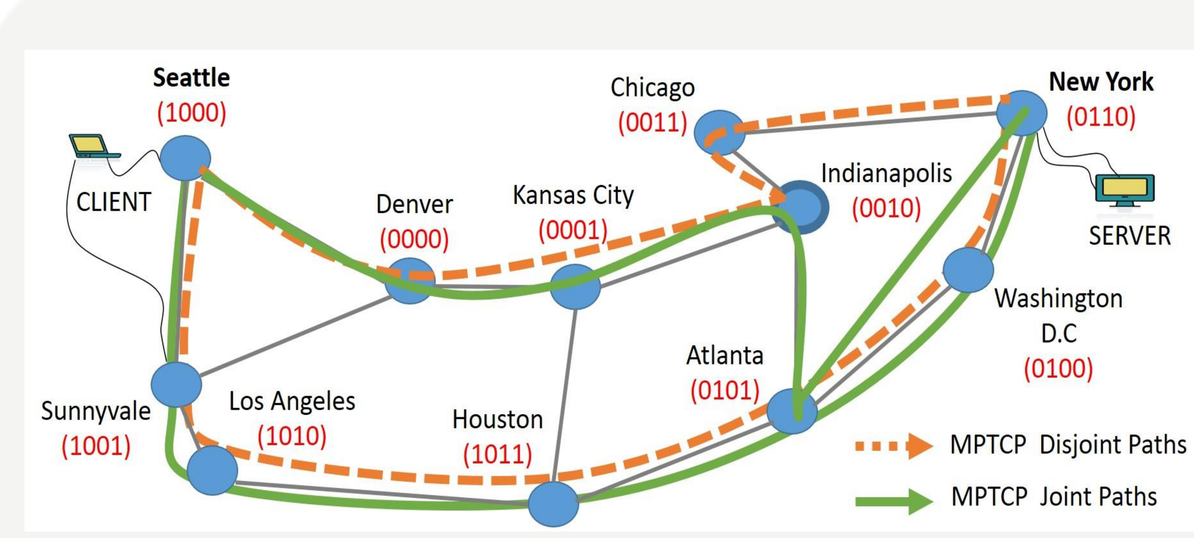
Routing Table for **node Indianapolis** (Before path-switching)

Bucket Distance	Next hop	Gateway
1	Chicago	Indianapolis
2	Kansas City	Indianapolis
3	**Chicago**	**Chicago**
4	Kansas City	Kansas City

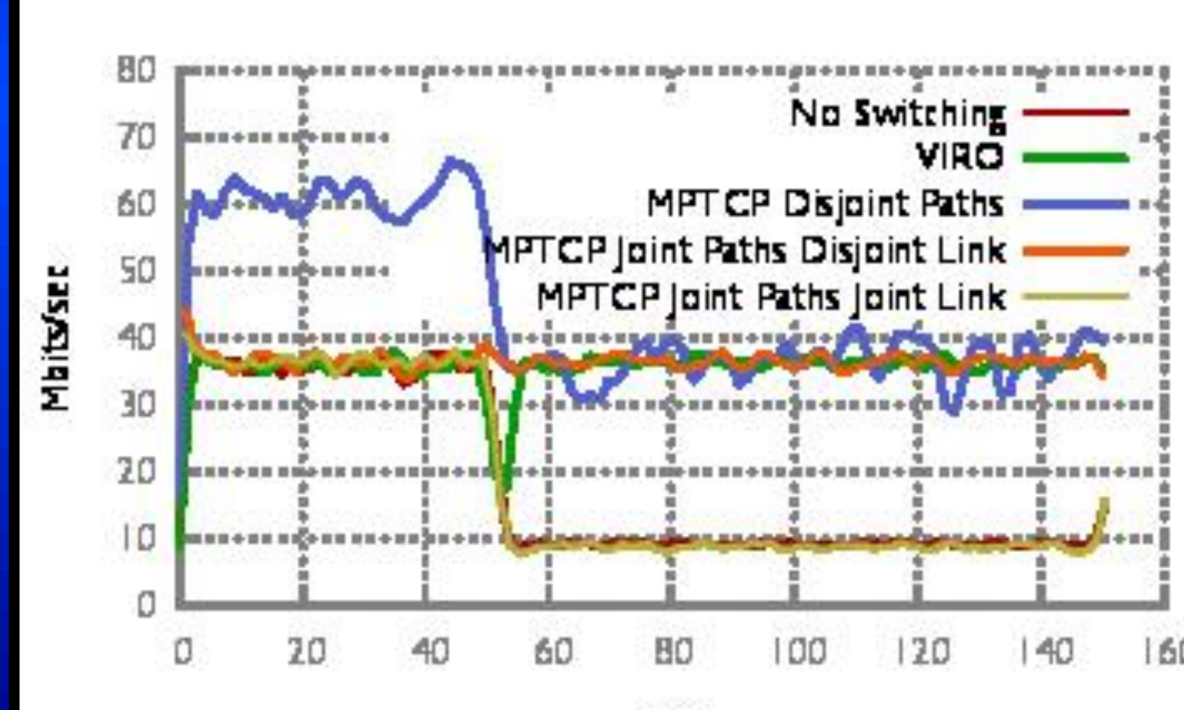
Routing Table for **node Indianapolis** (After path-switching)

- **In this experiment the client at Seattle communicates with a server in New York**
- **The link Indianapolis-Chicago is throttle from 40Mbits/sec to 10Mbits/sec**
  - **Path before switching:**
    Seattle -> Denver -> Kansas City -> Indianapolis -> Atlanta -> Washington D.C -> New York
  - **Path after switching:**
    Seattle -> Denver -> Kansas City -> Indianapolis -> Chicago -> New York (**Pathlet Switching**)

b) Path Switching with MPTCP



- **In this experiment the client at Seattle communicates with a server in New York using MPTCP, we throttle the links from 40Mbits/sec to 10Mbits/sec**
  - **MPTCP Disjoint Paths: throttle the link Indianapolis-Chicago**
    a) Seattle-> Denver -> Kansas City -> Indianapolis -> Chicago -> -> New York
    b) Seattle-> Sunnyvale -> Los Angeles -> Houston -> Atlanta -> Washington D.C -> New York
  - **MPTCP Joint Paths Disjoint Link: : throttle the link Indianapolis-Atlanta**
    a) Seattle-> Denver -> Kansas City -> Indianapolis -> Atlanta -> Washington D.C -> New York
    b) Seattle-> Sunnyvale -> Los Angeles-> Houston -> Atlanta -> Washington D.C -> New York
  - **MPTCP Joint Paths joint Link: throttle the link Atlanta-Washington D.C**
    a) Seattle-> Denver -> Kansas City -> Indianapolis -> Atlanta -> Washington D.C -> New York
    b) Seattle-> Sunnyvale -> Los Angeles-> Houston -> Atlanta -> Washington D.C -> New York



- **MPTCP** with disjoint TCP sub-flows paths provides the **best throughput**
- **MPTCP** performs poorly and provides results similar to the traditional TCP, when the TCP sub-flows share a **congested link**
- **VIRO** provides comparable results to **MPTCP** when the TCP sub-flows share congested links