

Motivation

Network engineers on campuses: troubleshoot SDN devices and find paths for specific “flows”

FLOW: can be defined with L2-L7

Inspired by the requirements listed at I2

TechExchange meeting:

<https://spaces.internet2.edu/display/sdn/2014-10-28+-+BoF+notes+-+SDNTrace>

Find multiple paths based on flow rules in switches

Experiment setup

RYU Controller

SDNTrace NB app (modified simple_switch)

Scapy to generate probe at originator

VTS: create topology on GENI

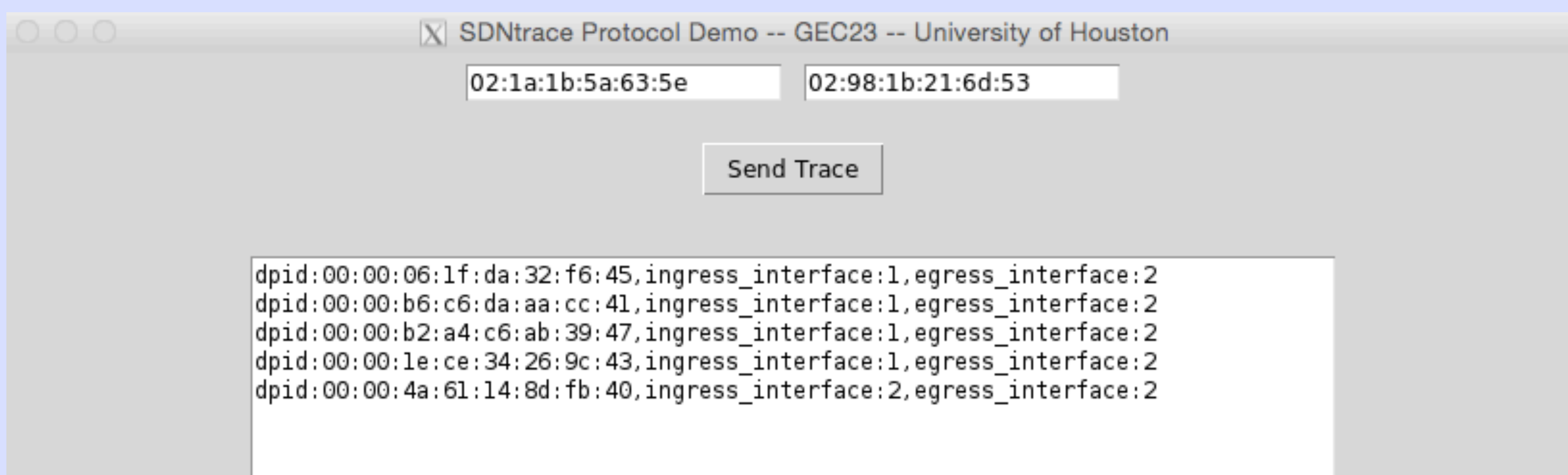
VTS features port UP/DOWN: manually create a directed acyclic graph to remove loops

Ping to discover the switches and attachment points

VTS feature dumpFlows: verify flows on datapath

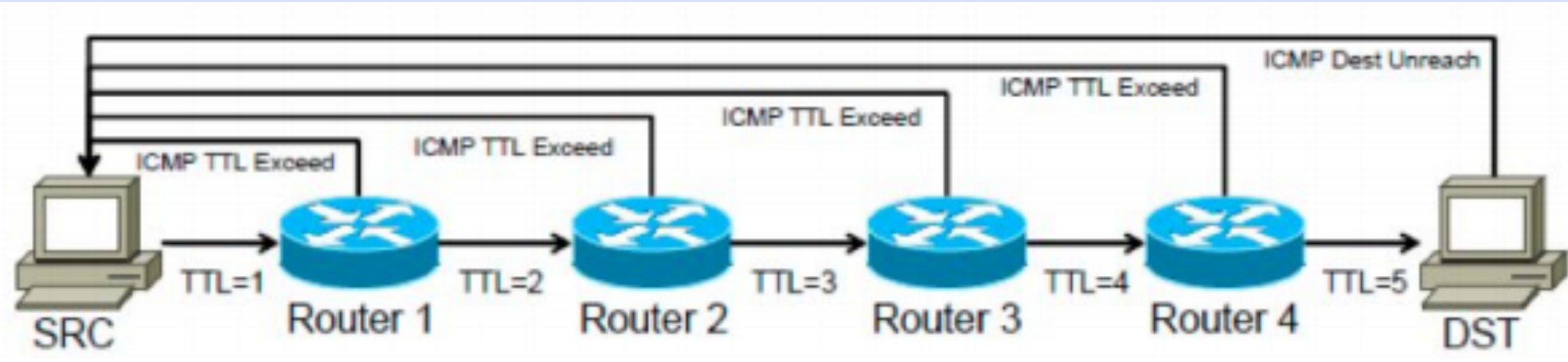
RUN SDNTrace to send a probe packet from source to destination with the “to-be-traced” flow packet

PG: Controller with SDNTrace app → **VTS:** Hosts and Topology → **VTS:** Port DOWN (acyclic graph)
 → **CLI@hosts:** PING to learn nodes
 → **CLI@host:** RUN SDNTrace to trace paths



```

dpid:00:00:06:1f:da:32:f6:45,ingress_interface:1,egress_interface:2
dpid:00:00:b6:c6:da:aa:cc:41,ingress_interface:1,egress_interface:2
dpid:00:00:b2:a4:c6:ab:39:47,ingress_interface:1,egress_interface:2
dpid:00:00:1e:ce:34:26:9c:43,ingress_interface:1,egress_interface:2
dpid:00:00:4a:61:14:8d:fb:40,ingress_interface:2,egress_interface:2
    
```



SDNTrace Protocol Message Types

TraceRequest – probe packet with an embedded “to-be-traced” packet for flow definition

TraceReply – reply packet carrying all node information on the path of “to-be-traced” flow

Traditional Traceroute

Based on ICMP TTL_EXCEEDED messages

Tests L3 connectivity

New approach – SDNTrace Protocol

Implemented as a NB application

Carries all path information in a reply message

Uses existing datapath state