Project Charter Draft

OpenFlow Traceroute (DRAFT)

(possible names: pathtrace? Slicetrace? Sdntrace? – sdn more than openflow? Can we abstract Framework to fit other types?)

Introduction

Modern traceroute utilities provide "hop-by-hop" feedback to network operators. This feedback enables diagnostic troubleshooting and the ability to isolate issues along a path between two endpoints. Most of these utilities require protocols from the TCP/IP protocol stack to perform the diagnostics and to transmit the messages back to the user. In a network utilizing OpenFlow, the paths may be layer 2 or layer 3 and do not always support an IP stack. These virtual paths, ideally, do not expose any underlying physical path boundaries. Therefore, the need exists for a new type of tool which is able to give "hop-by-hop" feedback along a seamless OpenFlow path which might span across multiple switches and across multiple interdomain segments.

Proposed Work

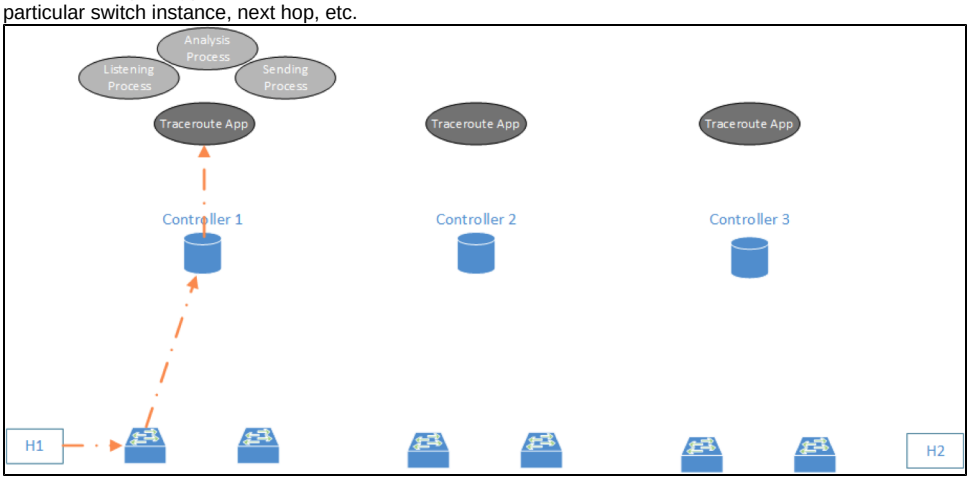
Members of the network operations community and the network research community propose to collaborate on an OpenFlow traceroute tool which will inject probes into an OpenFlow path, either from an end host, or via an application that sits on top of a local controller. This proposal will create a definition of a reference ethernet frame, a reference controller plug-in, and a reference command line interface (CLI) application for an end host.

Two key elements of this proposal are the assumptions that every controller knows its own virtual topology, and every controller can match on a specific frame and modify it appropriately. These key elements lead to the more general description: If every controller along a path establishes a specialized rule matching the reference probes this proposal defines, then a reference application talking to the controller can accomplish such tasks as inserting a timestamp, decrementing a hop counter, inserting a switch Data Path Identifier (DPID), etc. The application can then send back a message to the originating tool, as well as send on further frames to discover the full path, and possible attributes of the full path. This approach would allow diagnostics within an OpenFlow path, without regard to higher layer protocols that a specific OpenFlow path may carry during its normal operations.

Scope

The scope of the proposed project is to create a simple traceroute architecture as shown in Figure 1. The architecture comprises the following:

- prototype test probe
- prototype CLI tool that can reside on an end host and send prototype test probe from one end host towards the ultimate end host along the virtual path
- three discrete domains, each with a respective controller
- prototype mechanism which will listen for the probe, respond to the probe with pertinent information as to its particular hop and decrement a hop counter. This prototype mechanism should be able to communicate with a local domain controller in order to glean information about the particular switch instance, next hop, etc.



Exclusions:

- optical implementation

Team

For the work on this proposal, a team comprising operations engineers from the Research and Education community and network researchers will assemble to investigate the various aspects to create a prototype. The strength of this team is the pairing of the experience of many years of troubleshooting with tools such as the original traceroute with the new and innovative ideas that are coming out of the network research community.

The expectation of the team members are:

- .5hr-1hr weekly (or bi-weekly) commitments by one or more community operational engineers with students – suggested meeting venue: Google Hangouts? MS Lync? -- need shared whiteboard or at least shared desktop drawing application
- Definition and documentation of initial requirements by community team
- Prototype implementation by students
- Review and iterate requirements by community team and students
- Iterate prototype by students based on review

Requirements

- Tool must work across multiple domains
- Must be agnostic of arbitrary upper level protocols, i.e. IP, NDN, IPv6, etc.
- Must support the capability to inject frames from tool into the SDN path, without being on an end host that might not support the tool yet, i.e. a router.
    - How to deal with a layer 3 path with one or more routers?
- Frame definitions must be able to at least the following:
    - hop count
    - timestamp
    - DPID/switch name
    - ingress interface
    - egress interface
    - next hop DPID, if known from LLDP or some other mechanism
        - must be able to identify gaps - path A-B-C-D (A-.-B-D C is not participating in the protocol with ABD)
- Must be able to provide portion of 12-tuple header upon which control plane will execute action
- Frames must be scalable to allow more definitions of things to track and provide feedback, i.e. topological feedback, etc.
- Tool must provide raw text feedback.
- Tool may provide JSON feedback.
- Each controller must respect defined path route frame, process appropriately and pass it onwards for more processing
- Frames are not dependent on vlan on ingress or egress – will be able to travel a full data path
- Frames may provide local organizational identifier: i.e. AS number, unique ID

Tasks and Timeline

- Define requirements list
- Identify key interested operations community leaders who are able to commit an hour each week to meet with prospective students
- Define basic probe description
- Setup simple mininet environment
- Create simple CLI tool to send out reference frame across mininet environment
- Create very basic concept controller application with reference controller like OpenDaylight or Ryu
    - application should setup match for reference probe on the switch
    - application should perform basic decrement
    - application should send back message to originating application/CLI tool
- Setup environment on multiple GENI racks over a stitched together infrastructure – check visibility at each local rack
- Setup test environment with three stand-alone domains and test

Future work

- Does this approach scale?
- Does this approach generalize to other non-OpenFlow areas?
- Does this approach generalize to optical networks controlled by OpenFlow?

References:

1. Traceroute -- https://www.freebsd.org/cgi/man.cgi?query=traceroute
2. Tracert --
3. TCPtraceroute
4. pathping
5. mrt
6. layer four traceroute -- http://pwhois.org/lft/index.who
7. A Resource Delegation Framework for Software Defined Networks -- Ilya Baldin, Shu Huang, Rajesh Gopidi
8. EtherPIPE: http://conferences.sigcomm.org/sigcomm/2013/papers/hotsdn/p61.pdf
9. Cisco Layer 2 Traceroute Utility: http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/l2trace.html
10. Juniper "traceroute ethernet" utility: http://www.juniper.net/techpubs/en_US/junos11.4/topics/reference/command-summary/traceroute-ethernet-command.html
11. Brocade Ethernet Fabric traceroute utility: http://www.brocade.com/downloads/documents/html_product_manuals/NA_SAN_IP_1200/wwhelp/wwhimpl/common/html/wwhelp.htm#href=Ch_VCS.20.7.html&single=true
12. GEANT eduPERT layer 2 traceroute: http://kb.pert.geant.net/PERTKB/Layer2traceroute
13. IEEE 802.1ag - IEEE Stanadard for Local and Metropoltan Area Networks Virt
14. IEEE 802.1ag - Ethernet Connectivity Faluth Management (CFM) protocol
15. http://standards.ieee.org/getieee802/download/802.1ag-2007.pdf
16. ITU-T Recommendation Y.1731
17. IETF Draft: Ipflow-pathtrace -- http://tools.ietf.org/html/draft-yin-tsvwg-ipflow-pathtrace-ps-00
18. Other references:
19. http://en.wikipedia.org/wiki/Traceroute
20. http://en.wikipedia.org/wiki/MTR_%28software%29
21. http://en.wikipedia.org/wiki/PathPing
22. http://en.wikipedia.org/wiki/Layer_four_traceroute