



NANOG 74 HACKATHON RECAP



Sponsored By: **ORACLE®**
Cloud Infrastructure

Lab Sponsor: **TESUTO**
network emulation at scale

Day of Hackathon

- Tutorial on network model, graph representation, graph theory, shortest path algorithms, SR and BGP-LU
- 50+ registered attendees for Sunday Hackathon, self-organizing into teams
- Remote participants were able to take part on the hack
- Teams worked during the day (10:30am - 6pm) to prototype in the challenge and prepare a short presentation on the results of their work

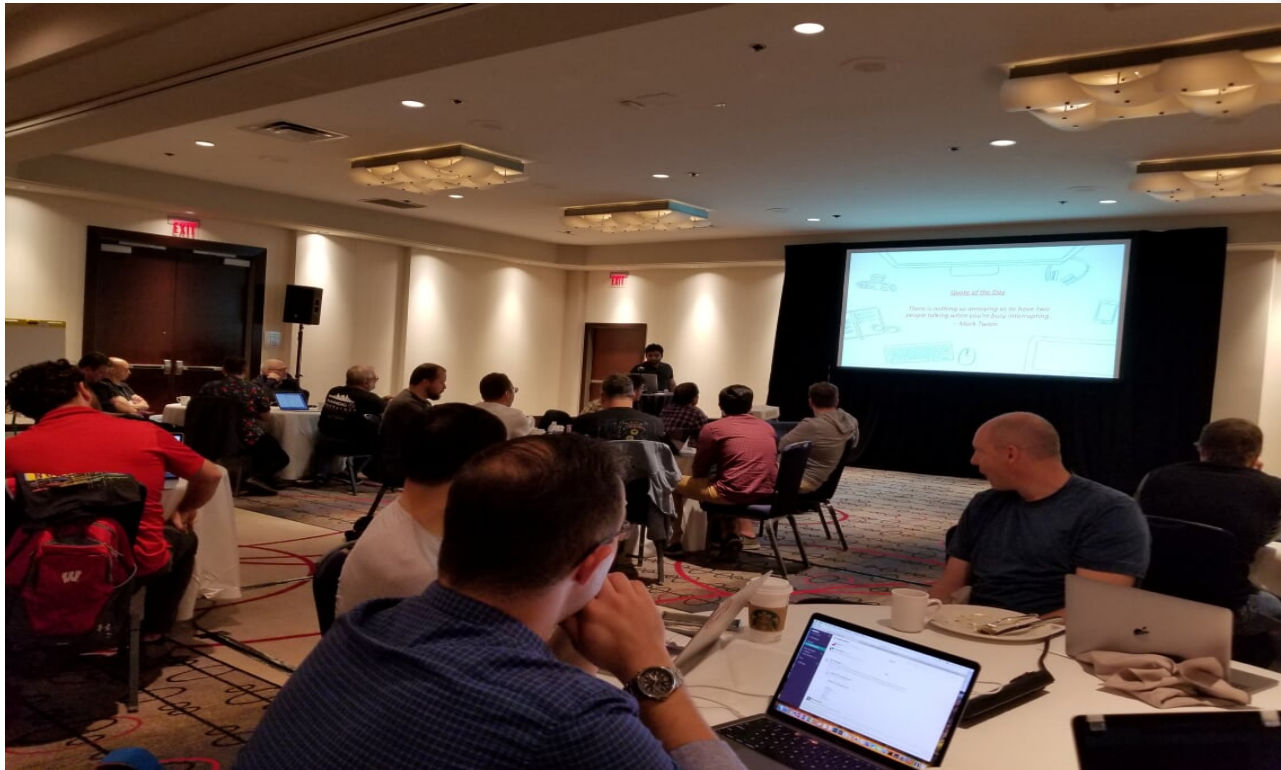


Problem Statement

- Create model of the network topology
- Solve some of the TE goals
- Program the network accordingly



Hackathon Tutorial



Hackathon Tutorial



Hackathon Teams

- Team 1: Buffs
- Team 2: Missing Semicolon
- Team 3: The_Lost_Pod
- Team 4: Newbies
- Team 5: Cat5e Hurricane
- Team 6: Failed to Route
- Team 7: 5S
- Team 8: OABC
- Team 9: RPKI Fixers



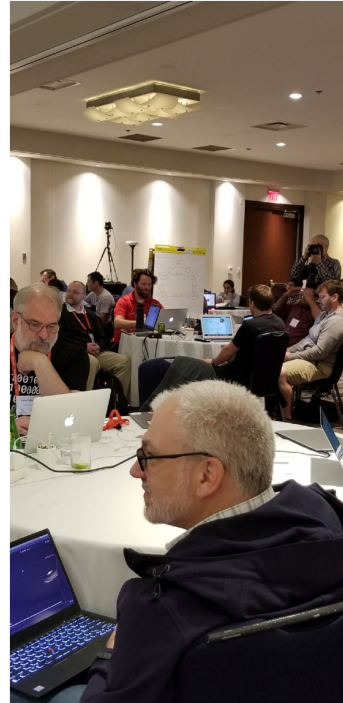
Hackathon Teams



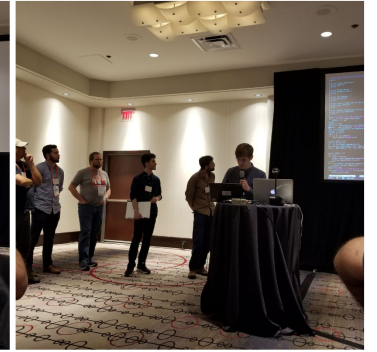
Intense Hacking



Good Learning Experience



Presentations and Sharing





People's Choice: Winning Hacks

- Team CAT5e Hurricane
- Team Buffs
- Team The Lost Pod

Dynamic Network Costing for TE

Cat5e Hurricane

Product Overview

- **Open-Source Tool:**

NAPALM (Network Automation and Programmability Abstraction):

Freely available

- **Supports a wide range of platforms:**

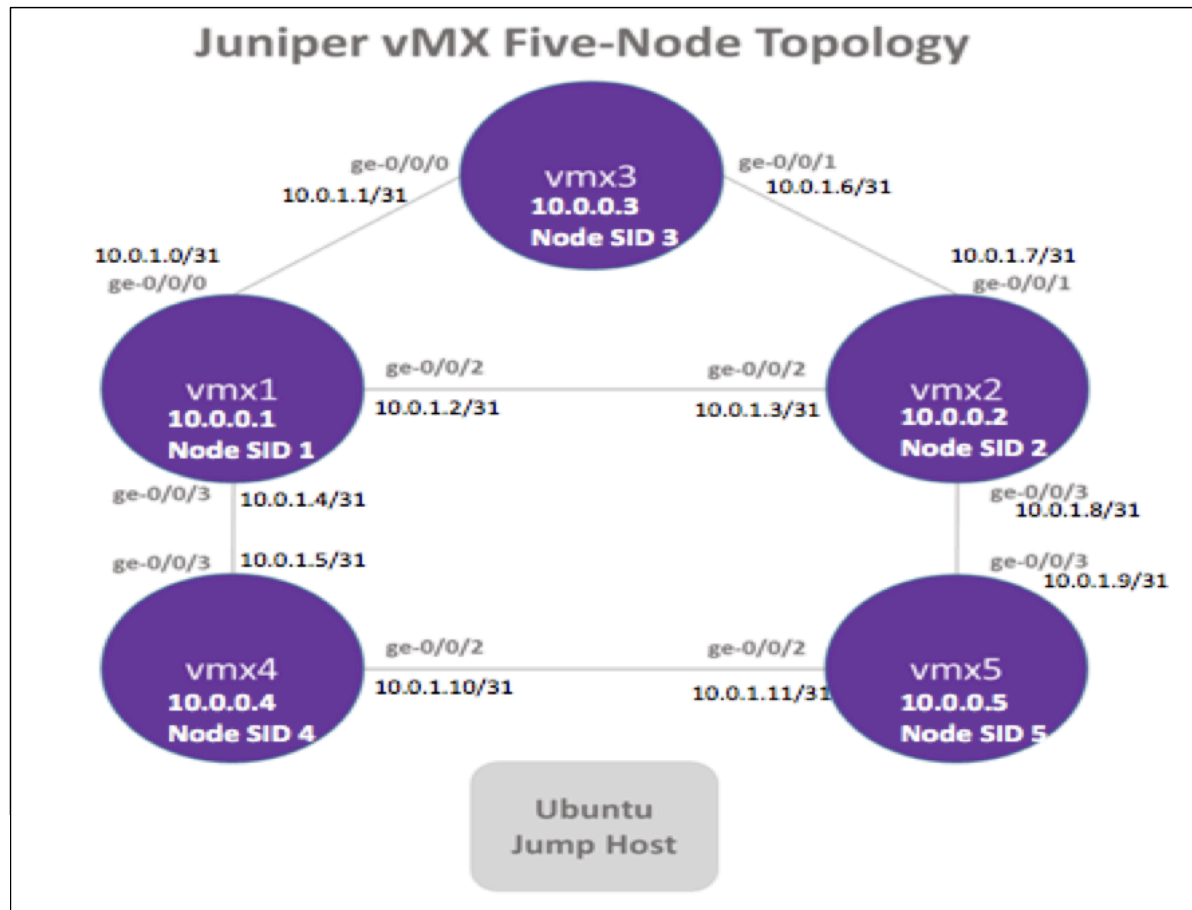
- **networkx** - Utilize traffic data to calculate best path ***graph it!***

- **BGP-LU:** ***Southbound for SR***

- **Controller:** Provision tunnels on the fly ***Easy to deploy!***

Does this impact your MPLS cloud?

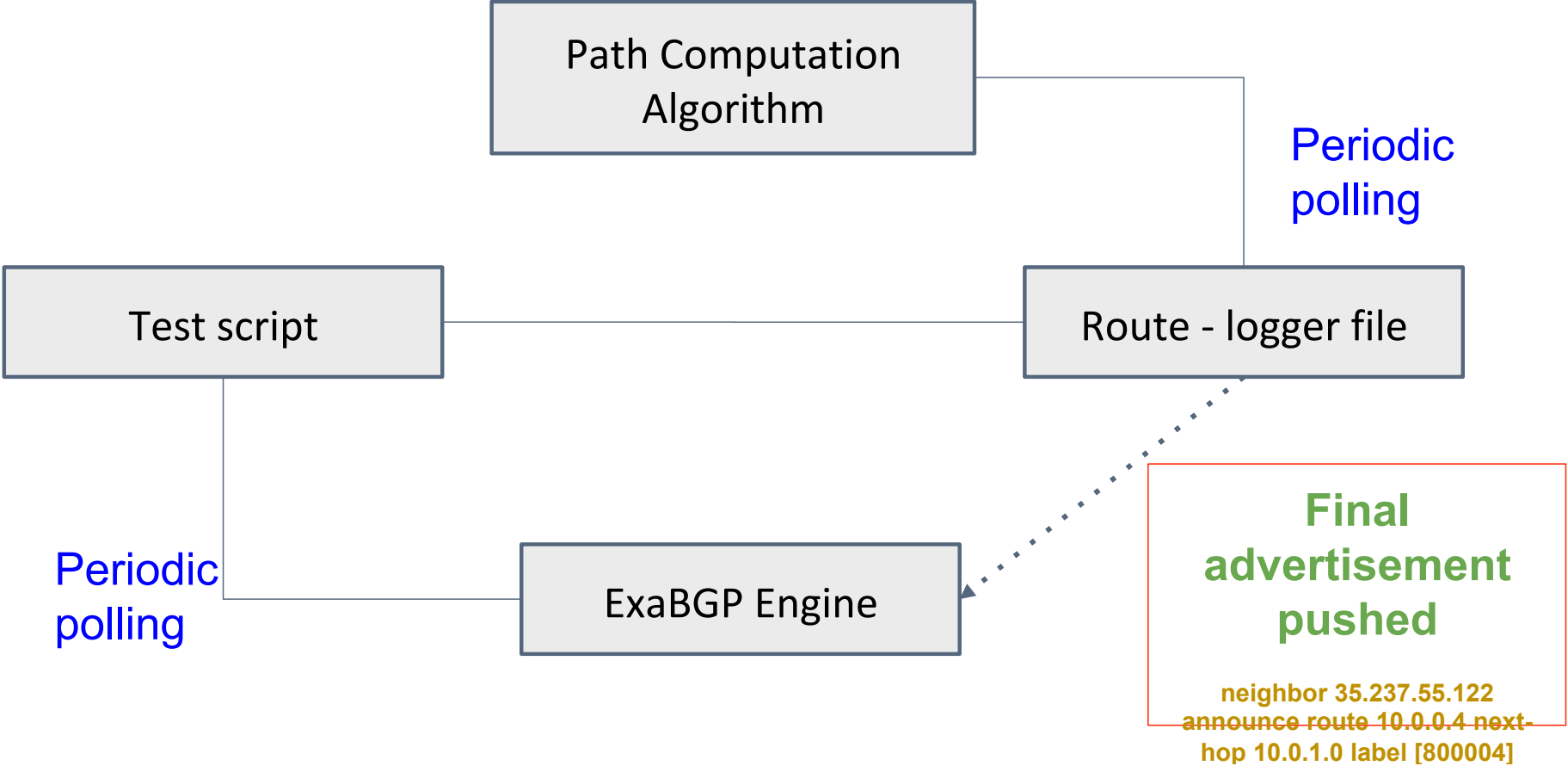
Topology



Packet Based Traffic Shift

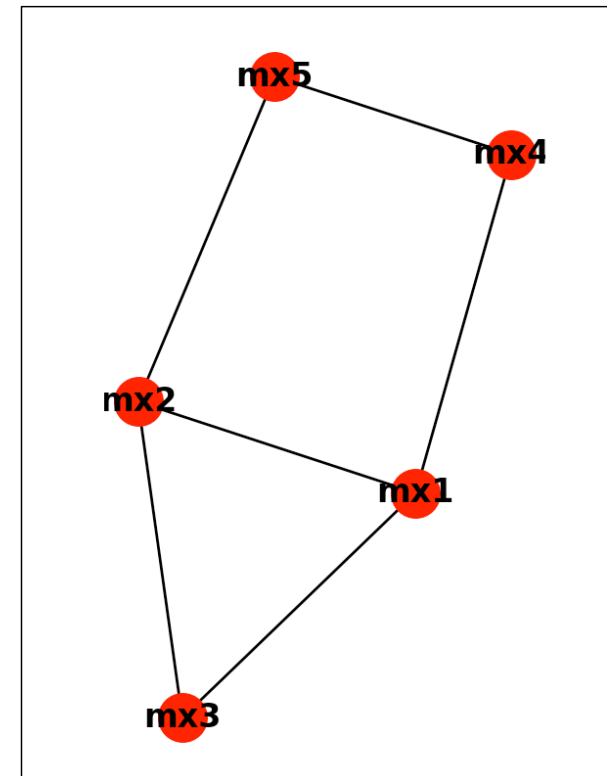
- Programmatically modify routes in a network based on the **aggregate flow** of traffic throughout time
- Ensure varied traffic patterns are considered: **Compute SPF accordingly**
- As interfaces take more traffic, they become less preferred: **dynamic approach**

Overview

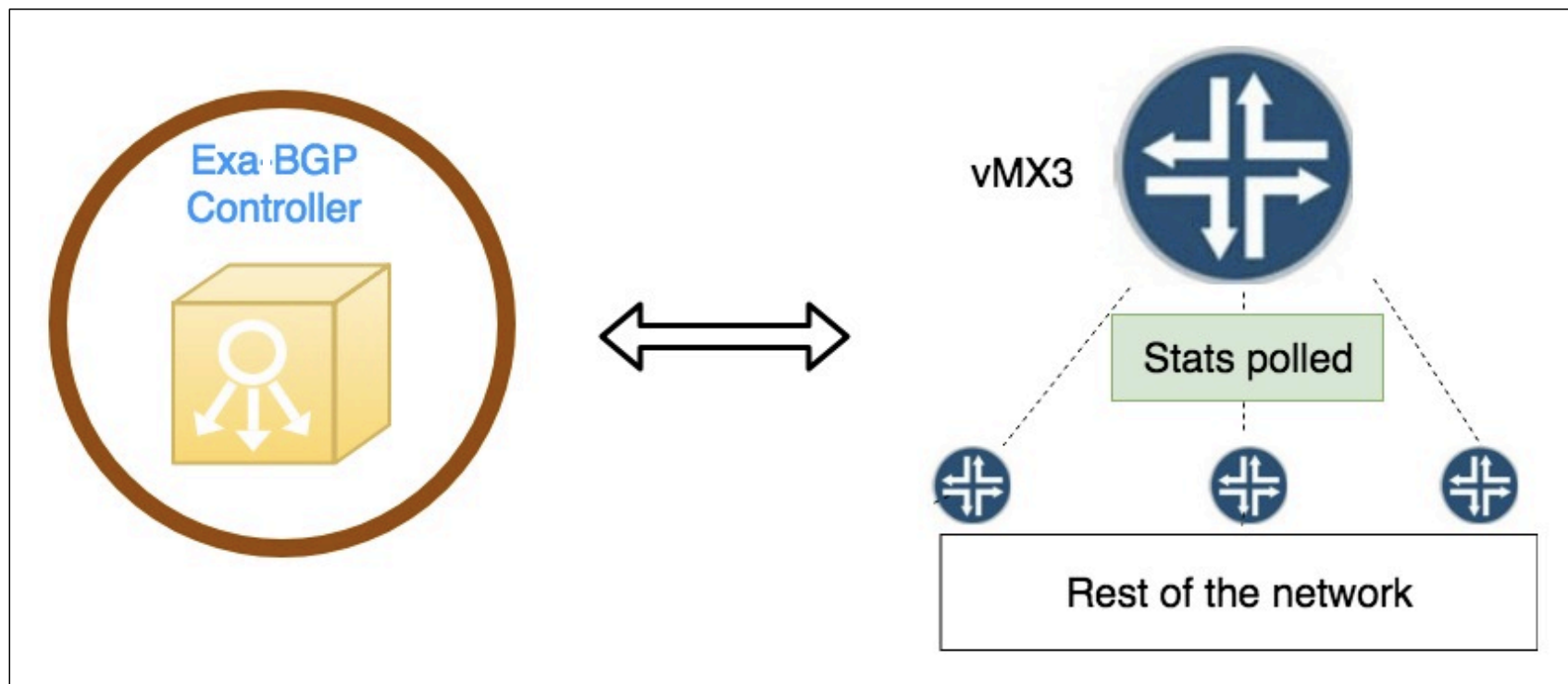


Networkx

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 g = nx.Graph()
5
6 g.add_node('mx1')
7 g.add_node('mx2')
8 g.add_node('mx3')
9 g.add_node('mx4')
10 g.add_node('mx5')
11 g.add_edge('mx3', 'mx1', weight=10)
12 g.add_edge('mx3', 'mx2', weight=10)
13 g.add_edge('mx4', 'mx5', weight=20)
14 g.add_edge('mx1', 'mx2', weight=10)
15 g.add_edge('mx1', 'mx4', weight=10)
16 g.add_edge('mx2', 'mx5', weight=10)
17
18
19 topology = {'mx1': {'mx2': 'ge-0/0/2', 'mx3': 'ge-0/0/0', 'mx4': 'ge-0/0/3'},
20             'mx2': {'mx3': 'ge-0/0/1', 'mx1': 'ge-0/0/2', 'mx5': 'ge-0/0/3'},
21             'mx3': {'mx1': 'ge-0/0/0', 'mx2': 'ge-0/0/1'},
22             'mx4': {'mx1': 'ge-0/0/3', 'mx5': 'ge-0/0/2'},
23             'mx5': {'mx4': 'ge-0/0/2', 'mx2': 'ge-0/0/3'}}
24
25
26 G = nx.petersen_graph()
27 plt.subplot(121)
28 nx.draw(g, with_labels=True, font_weight='bold')
29 plt.show()
30
31
32
```



Controller -- Device Overview

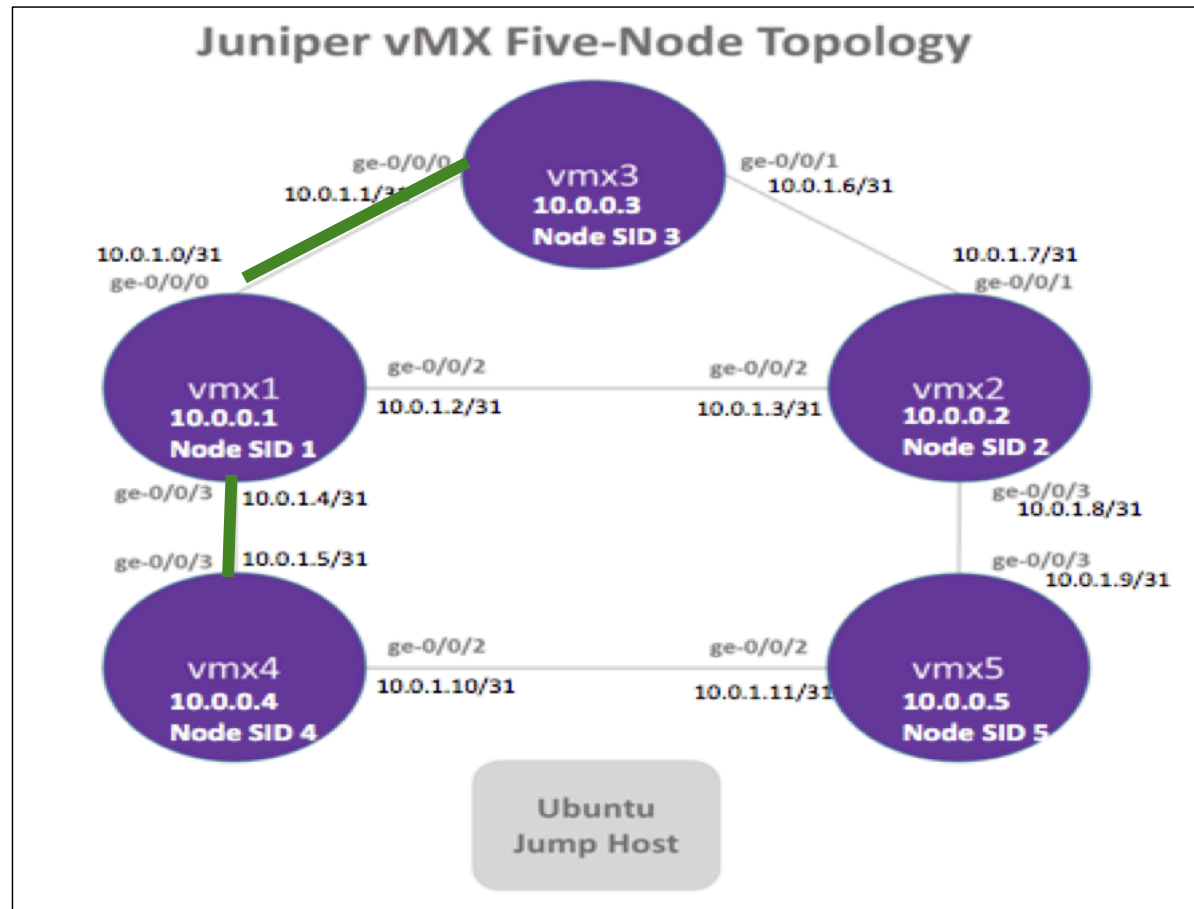


The ExaBGP operation

```
sohshah exabgp-master $ ./sbin/exabgp exabgptest.conf
14:07:20 | 56183 | welcome      | Thank you for using ExaBGP
14:07:20 | 56183 | version      | 4.0.8-793a2931
14:07:20 | 56183 | interpreter   | 3.6.5 (default, Apr 25 2018, 14:23:58) [GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.1)]
14:07:20 | 56183 | os           | Darwin sohshah-mac 17.7.0 Darwin Kernel Version 17.7.0: Thu Jun 21 22:53:14 PDT 2018; root:xnu-4570.71
.2~1/RELEASE_ARM_T8020 x86_64
14:07:20 | 56183 | installation  | /Users/sohshah/Downloads/exabgp-master
14:07:20 | 56183 | cli control   | named pipes for the cli are:
14:07:20 | 56183 | cli control   | to send commands /Users/sohshah/Downloads/exabgp-master/run/exabgp.in
14:07:20 | 56183 | cli control   | to read responses /Users/sohshah/Downloads/exabgp-master/run/exabgp.out
14:07:20 | 56183 | configuration | performing reload of exabgp 4.0.8-793a2931
14:07:20 | 56183 | reactor      | loaded new configuration successfully
14:07:20 | 56183 | api          | route added to neighbor 35.237.55.122 local-ip None local-as 65535 peer-as 65535 router-id 172.16.2.1
family-allowed-in-open : 10.0.0.4/32 label [ 800001 800004 ] next-hop 10.0.1.7
14:07:20 | 56183 | reactor      | connected to peer-2 with outgoing-1 199.187.220.199-35.231.211.240
14:07:20 | 56183 | reactor      | connected to peer-1 with outgoing-2 199.187.220.199-35.237.55.122
```

```
tesuto@mx3> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State  Pending
inet.0
1
AS            InPkt    OutPkt    OutQ   Flaps  Last Up/Dwn  Statel#Active/Received/Accepted/Damped...
199.187.220.199 65535    250      274     0     1    2:03:21 Establ
inet.0: 1/1/1/0
```

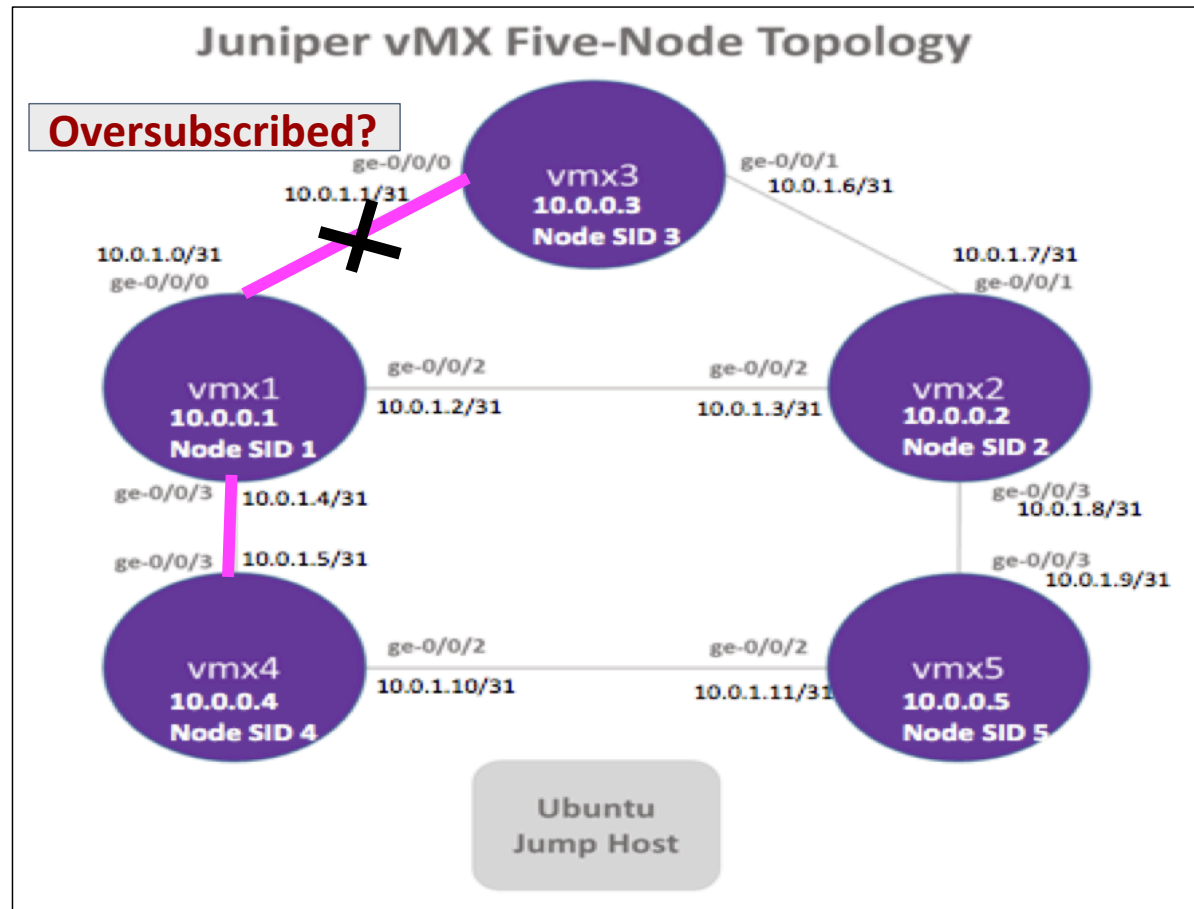
Path Taken: Shortest!



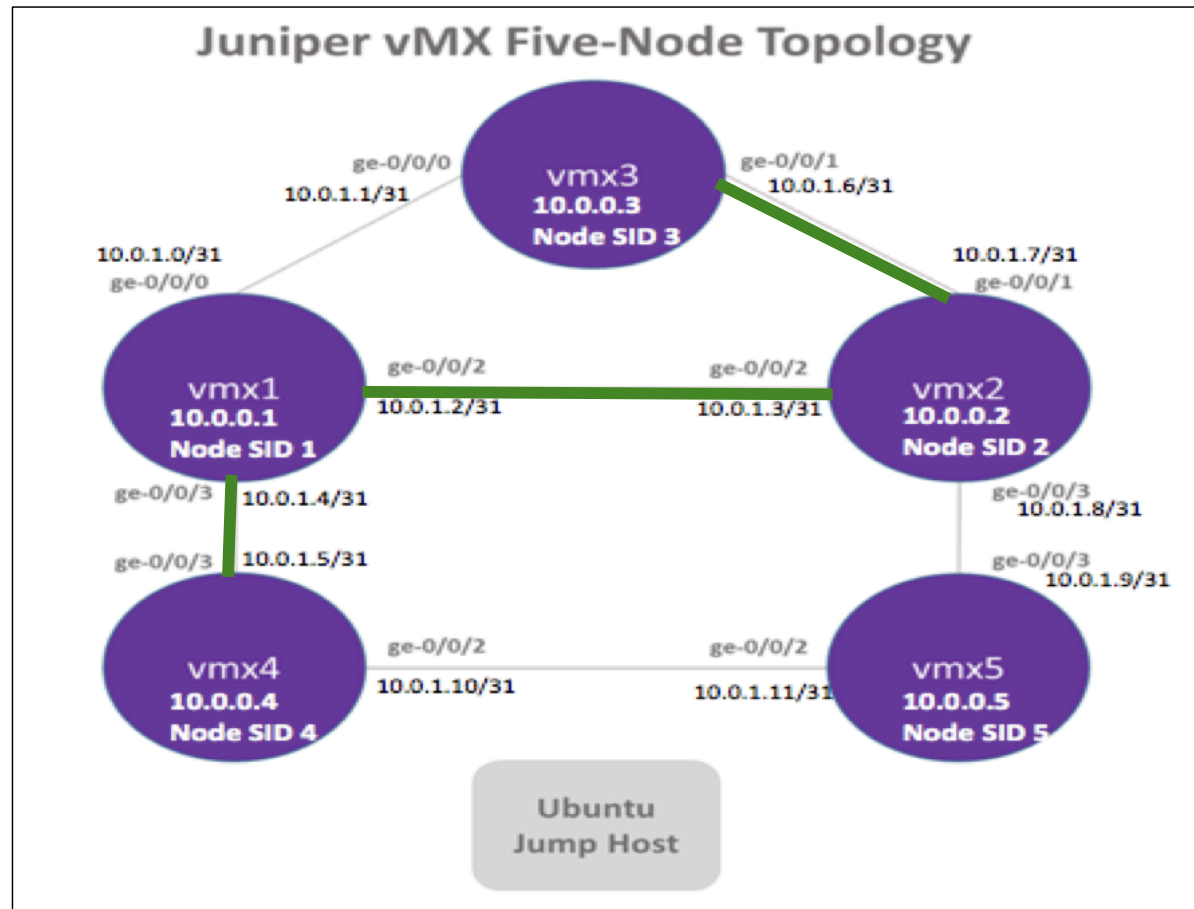
Controller Run #1

```
sohshah src $ ./exa_controller.py
Calculated weight on mx1:ge-0/0/0 to be 0
Calculated weight on mx1:ge-0/0/2 to be 0
Calculated weight on mx1:ge-0/0/3 to be 0
Calculated weight on mx2:ge-0/0/1 to be 1
Calculated weight on mx2:ge-0/0/2 to be 0
Calculated weight on mx2:ge-0/0/3 to be 1
Calculated weight on mx3:ge-0/0/0 to be 2
Calculated weight on mx3:ge-0/0/1 to be 1
Calculated weight on mx4:ge-0/0/2 to be 2
Calculated weight on mx4:ge-0/0/3 to be 1
Calculated weight on mx5:ge-0/0/2 to be 2
Calculated weight on mx5:ge-0/0/3 to be 1
Best path from mx3 to mx4 is ['mx3', 'mx1', 'mx4']
Labels for best path are: ['800004']
```

Change in flow



Path Taken



Controller Run #2

Best path from mx3 to mx4 is ['mx3', 'mx2', 'mx1', 'mx4']

Labels for best path are ['800001', '800004']

Calculated weight on mx1:ge-0/0/0 to be 110

Calculated weight on mx1:ge-0/0/2 to be 15

Calculated weight on mx1:ge-0/0/3 to be 10

Calculated weight on mx2:ge-0/0/1 to be 11

Calculated weight on mx2:ge-0/0/2 to be 49

Calculated weight on mx2:ge-0/0/3 to be 10

Calculated weight on mx3:ge-0/0/0 to be 110

Calculated weight on mx3:ge-0/0/1 to be 10

Calculated weight on mx4:ge-0/0/2 to be 11

Calculated weight on mx4:ge-0/0/3 to be 10

Calculated weight on mx5:ge-0/0/2 to be 11

Calculated weight on mx5:ge-0/0/3 to be 10

Route on MX3

```
tesuto@mx3> show route 10.0.0.4

inet.0: 21 destinations, 23 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/32      *[BGP/13] 02:03:29, localpref 100, from 199.187.220.199
                  AS path: I, validation-state: unverified
                  > to 10.0.1.7 via ge-0/0/1.0, Push 800004, Push 800001(top)
                  [IS-IS/18] 16:20:50, metric 20
                  > to 10.0.1.0 via ge-0/0/0.0

inet.3: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

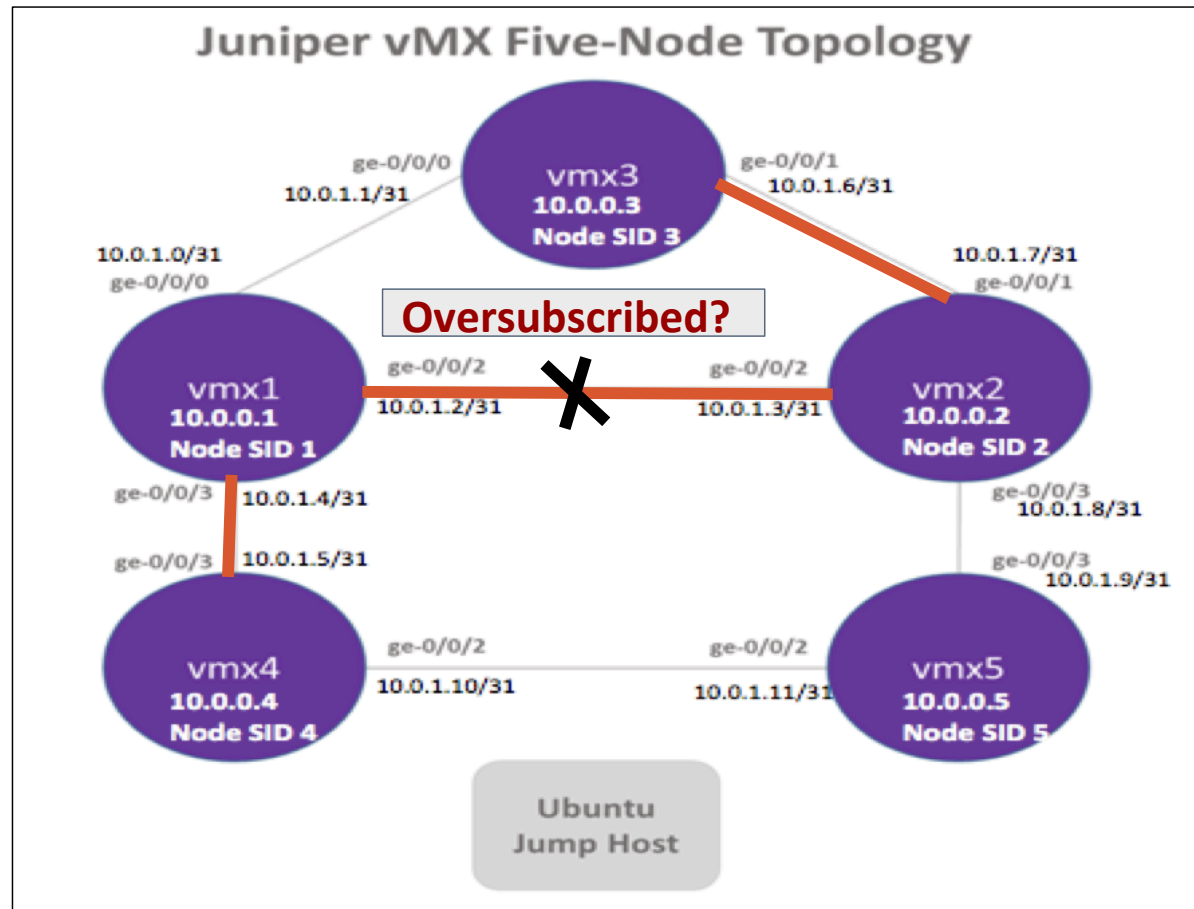
10.0.0.4/32      *[L-ISIS/14] 16:20:50, metric 20
                  > to 10.0.1.0 via ge-0/0/0.0, Push 800004
```

Traceroute from src:dest

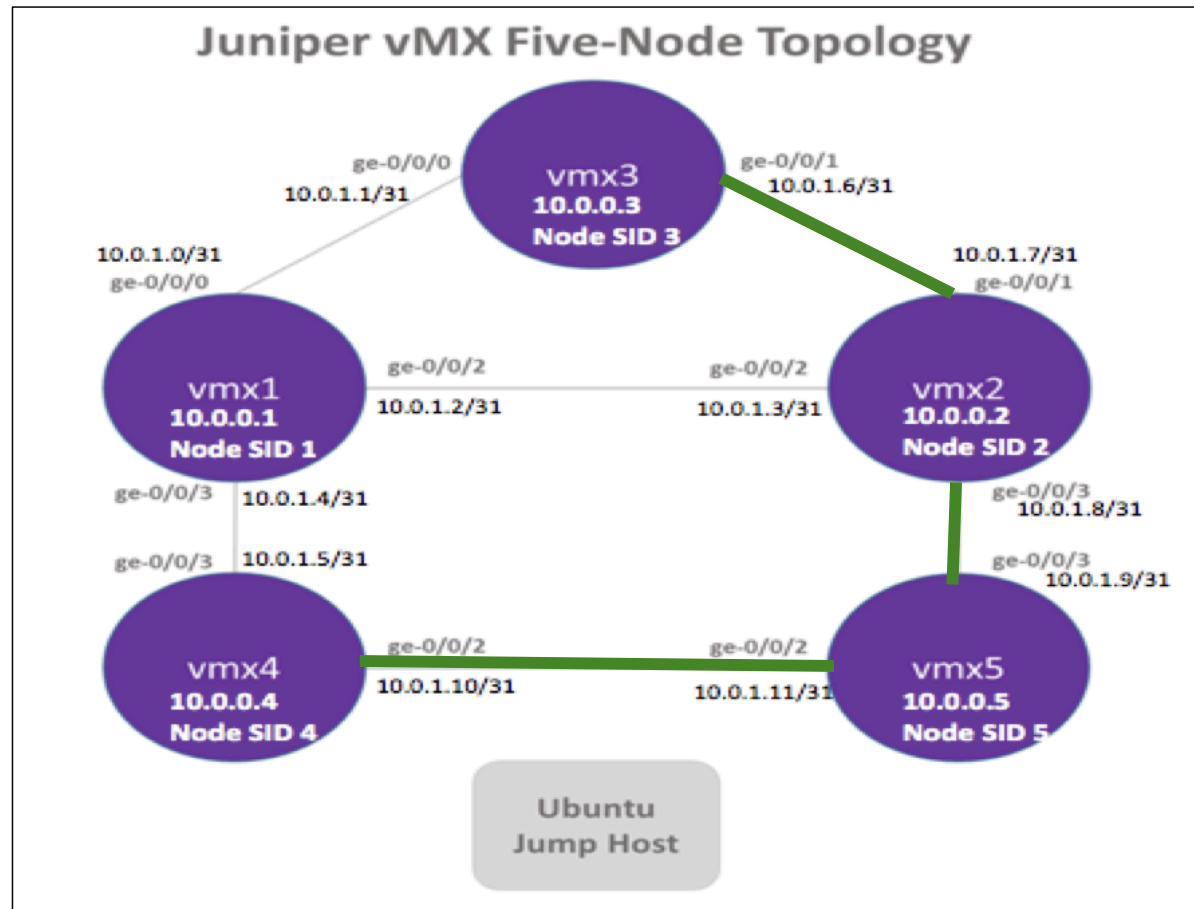
```
tesuto@mx3> traceroute 10.0.0.4
traceroute to 10.0.0.4 (10.0.0.4), 30 hops max, 52 byte packets
 1  10.0.1.7 (10.0.1.7)  19.455 ms  10.579 ms  16.087 ms
    MPLS Label=8000001 CoS=0 TTL=1 S=0
    MPLS Label=8000004 CoS=0 TTL=1 S=1
 2  10.0.1.2 (10.0.1.2)  23.681 ms  22.347 ms  24.021 ms
    MPLS Label=8000004 CoS=0 TTL=1 S=1
 3  10.0.0.4 (10.0.0.4)  39.828 ms  38.614 ms  39.621 ms

tesuto@mx3> █
```


Change in flow



Path Taken



Controller Run #3

```
Calculated weight on mx3:ge-0/0/3 to be 18  
Best path from mx3 to mx4 is ['mx3', 'mx2', 'mx5', 'mx4']  
Labels for best path are: ['800005', '800004']  
Calculated weight on mx1:ge-0/0/0 to be 121  
Calculated weight on mx1:ge-0/0/2 to be 117  
Calculated weight on mx1:ge-0/0/3 to be 20  
Calculated weight on mx2:ge-0/0/1 to be 19  
Calculated weight on mx2:ge-0/0/2 to be 117  
Calculated weight on mx2:ge-0/0/3 to be 18  
Calculated weight on mx3:ge-0/0/0 to be 120  
Calculated weight on mx3:ge-0/0/1 to be 18  
Calculated weight on mx4:ge-0/0/2 to be 19  
Calculated weight on mx4:ge-0/0/3 to be 20  
Calculated weight on mx5:ge-0/0/2 to be 19  
Calculated weight on mx5:ge-0/0/3 to be 18  
Best path from mx3 to mx4 is ['mx3', 'mx2', 'mx5', 'mx4']  
Labels for best path are: ['800005', '800004']
```

```
sohshah exabgp-master $  
sohshah exabgp-master $ cat route-logger.log  
neighbor 35.237.55.122 announce route 10.0.0.4 next-hop 10.0.1.7 label [800005 800004]  
sohshah exabgp-master $
```

Learning and Challenges

- Fetching **requisite data** from the testbed to take decisions
- Integrate **graph theory** with existing algorithm to traffic engineer effectively
- Provision **SR tunnels using BGP-LU** as per inputs received

Conclusion

- **Software driven approach** for scaling MPLS infrastructures
- Harnesses BGP-LU, SR, and controller based approach for **speeding up** provisioning
- Takes care of **link failure**, and **oversubscription detection**





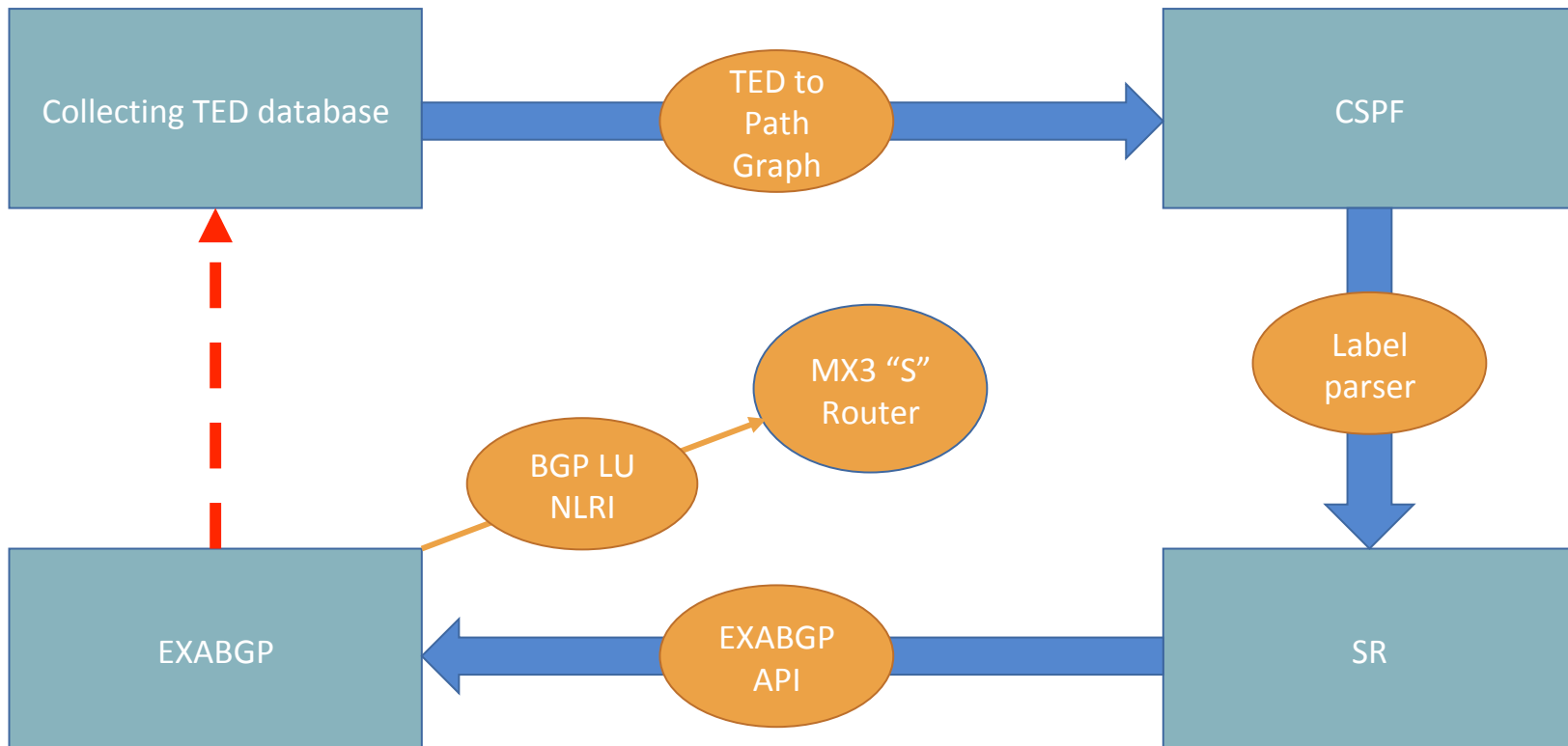
SDN Segment Routing in 1 Day

'The Buffs'
University of Colorado Boulder

Our approach to the given problem

- Extract the topology information
- Determine best path using Constrained Shortest Path First (CSPF) algorithm
- Construct the Segment Routing Traffic Engineering routes
- Push configuration to the network

Data Flow

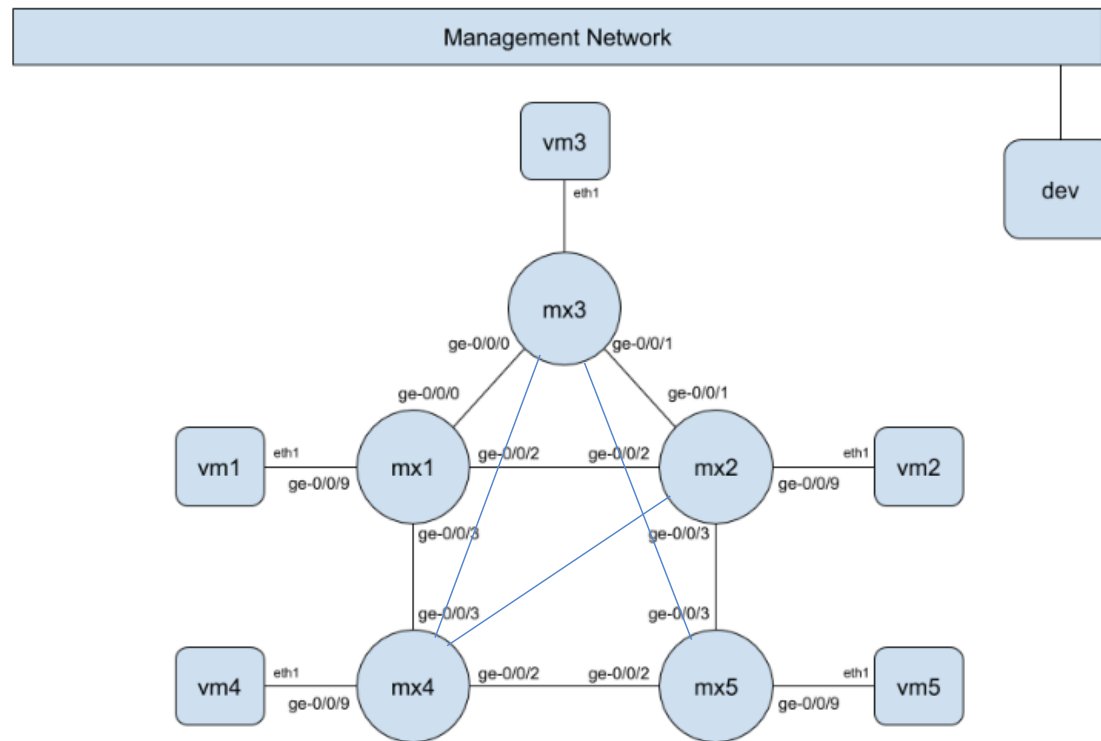


Topology Discovery TPI

- PyEZ was used to extract information from the TED database for the MX routers
- Information included: source node, next hop router, cost and reservable bandwidth
- Example: 'mx1', 'mx4' , '10' , '100'

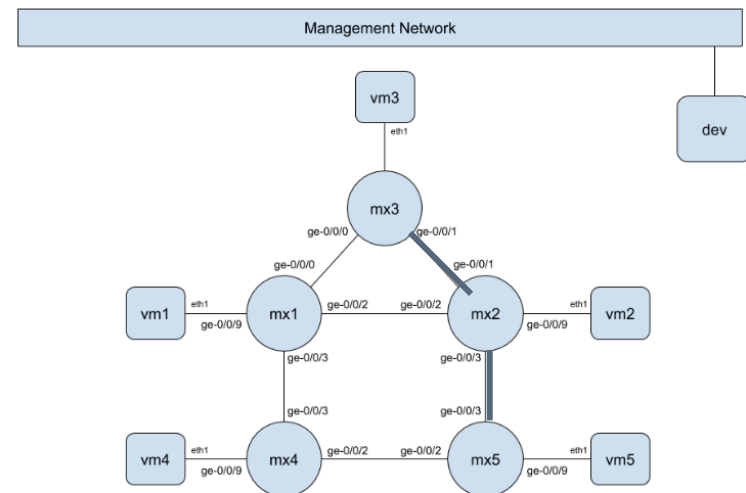
```
nelson@nelson-VirtualBox:~$ sudo python3 nelson.py
Connecting
mx1
Connecting
mx2
Connecting
mx3
Connecting
mx4
Connecting
mx5
Superlinks are:
[('mx1', 'mx4', '10', '300'), ('mx1', 'mx2', '10', '200'), ('mx1', 'mx3', '10', '100'), ('mx2', 'mx4', '10', '300'), ('mx2', 'mx3', '10', '100'), ('mx3', 'mx2', '10', '200'), ('mx3', 'mx4', '10', '300'), ('mx4', 'mx2', '10', '200'), ('mx4', 'mx3', '10', '100'), ('mx5', 'mx4', '10', '300'), ('mx5', 'mx2', '10', '200'), ('mx5', 'mx3', '10', '100')]
SHORTEST
['mx1', 'mx4', 'mx5']
```

Network Diagram



Shortest Path in whole Topology based on SPF: For MX3 to MX5

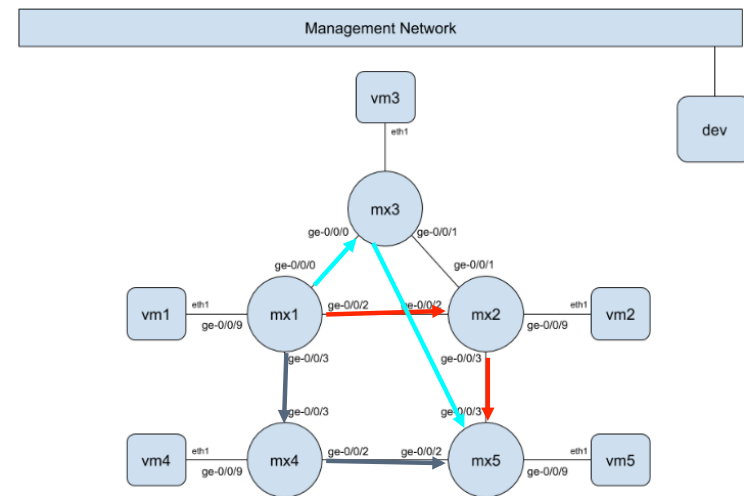
- Utilize available bandwidth
- Reduce the available link bandwidth when a possible path utilizes it
- Automated route generation



```
SHORTEST
['mx3', 'mx2', 'mx5']
neighbor 35.237.127.235 announce route 10.100.5.1 next-hop 10.0.1.6 label[ 800002 800005 ];
nelson@nelson-VirtualBox:~$
```

Disjoint Path Links

- Computed the possible disjointed links using the constraint based algorithm



```
DISJOINT
['mx1', 'mx4', 'mx5']
['mx1', 'mx2', 'mx5']
['mx1', 'mx3', 'mx5']
Route Not Possible
```

All possible paths between nodes after reducing bandwidth

- Removed links with no available bandwidth from the computation

```
SPF
['mx1', 'mx4', 'mx5']
['mx1', 'mx2', 'mx5']
['mx1', 'mx3', 'mx5']
['mx1', 'mx4', 'mx5']
['mx1', 'mx2', 'mx5']
['mx1', 'mx3', 'mx5']
['mx1', 'mx4', 'mx5']
['mx1', 'mx2', 'mx5']
nelson@nelson-VirtualBox:~$
```

ExaBGP

- Routing rules were obtained from CSPF
- We pushed the SR-TE rules using ExaBGP to the MX3 router

```
SHORTEST
['mx3', 'mx2', 'mx5']
neighbor 35.237.127.235 announce route 10.100.5.1 next-hop 10.0.1.6 label[ 800002 800005 ];
nelson@nelson-VirtualBox:~$

netman@netman:~$ exabgp conf.ini
18:53:20 | 4797 | welcome | Thank you for using ExaBGP
18:53:20 | 4797 | version | 4.0.8-793a2931
18:53:20 | 4797 | interpreter | 2.7.3 (default, Oct 26 2016, 21:04:23) [GCC 4.6.3]
18:53:20 | 4797 | os | Linux netman 3.2.0-58-generic-pae #88-Ubuntu SMP Tue Dec 3 18:00:02 UTC 2013 i686
18:53:20 | 4797 | installation | /usr/local
18:53:20 | 4797 | cli | could not find the named pipes (exabgp.in and exabgp.out) required for the cli
18:53:20 | 4797 | cli | we scanned the following folders (the number is your PID):
18:53:20 | 4797 | cli control | - /run/exabgp/
18:53:20 | 4797 | cli control | - /run/1000/
18:53:20 | 4797 | cli control | - /run/
18:53:20 | 4797 | cli control | - /var/run/exabgp/
18:53:20 | 4797 | cli control | - /var/run/1000/
18:53:20 | 4797 | cli control | - /var/run/
18:53:20 | 4797 | cli control | - /usr/local/run/exabgp/
18:53:20 | 4797 | cli control | - /usr/local/run/1000/
18:53:20 | 4797 | cli control | - /usr/local/run/
18:53:20 | 4797 | cli control | - /usr/local/var/run/exabgp/
18:53:20 | 4797 | cli control | - /usr/local/var/run/1000/
18:53:20 | 4797 | cli control | - /usr/local/var/run/
18:53:20 | 4797 | cli control | please make them in one of the folder with the following commands:
18:53:20 | 4797 | cli control | > mkfifo /home/netman/run/exabgp.{in,out}
18:53:20 | 4797 | cli control | > chmod 600 /home/netman/run/exabgp.{in,out}
18:53:20 | 4797 | cli control | > chown 1000:1000 /home/netman/run/exabgp.{in,out}
18:53:20 | 4797 | configuration | performing reload of exabgp 4.0.8-793a2931
18:53:20 | 4797 | reactor | loaded new configuration successfully
18:53:20 | 4797 | api | route added to neighbor 35.237.127.235 local-ip 198.51.100.2 local-as 65535 peer-as 65535 router-id 198.51.100.2 family-allowed in-open : 10.100.5.1/32 la
18:53:20 | 4797 | reactor | bel [ 800004 800005 ] next-hop 10.0.1.0
18:53:20 | 4797 | reactor | connected to peer-1 with outgoing-1 198.51.100.2-35.237.127.235
```

Verifying the route configured by ExaBGP

Route MX3-MX1-MX4-MX5 on MX1

```
800004(S=0)          *[L-ISIS/14] 00:01:53, metric 10  
                    > to 10.0.1.5 via ge-0/0/3.0, Pop
```

Route MX3-MX1-MX4-MX5 on MX4

```
800005(S=0)          *[L-ISIS/14] 00:07:30, metric 20  
                    > to 10.0.1.11 via ge-0/0/2.0, Pop
```

Route MX3-MX1-MX4-MX5 on MX3

```
tesuto@mx3> show route exact 10.100.5.1  
  
inet.0: 22 destinations, 23 routes (22 active, 0 holddown, 0 hidden)  
+ = Active Route, - = Last Active, * = Both  
  
10.100.5.1/32        *[BGP/13] 00:01:27, localpref 100, from 199.187.219.98  
                    AS path: I, validation-state: unverified  
                    > to 10.0.1.0 via ge-0/0/0.0, Push 800005, Push 800004(top)
```


Summary

- Learning about:
 - . Junos-eznc (PyEZ)
 - . Constrained shortest path first (CSPF)
 - . ExaBGP
- Worked on the three objectives and integrating the hack

Takeaway

- Planning, collaborating and hacking!

Contact

Srinidhi Vajapeyam Srivatsa

sva7028@colorado.edu

Nelson Valerian Agera

neag0709@colorado.edu

Simran Kaur Kohli

siko0339@colorado.edu

Hariharakumar Narasimhakumar

hana8349@colorado.edu

Jaimeen Parmar

japa2615@colorado.edu

The Lost Pod

NANOG 74 Hackathon

Clinton Work
Joshua McBeth
Adrian Schmidt
Cesar Cordero
TJ Trask

Breakdown and Attack

- Ncclient/netconf used to pull topology and xmlltodict to parse returned data structure
- Networkx graph and digraph to store topology information and run spf algorithm
- Split up solvers per question, pass graph to each solver function
- Pull loopback and SID details from digraph using the solved path to generate BGP-LU routes
- Exabgp used to advertise solved bgp label stack to the Juniper vmx routers



Default state

```
tesuto@mx3> show route 10.0.0.4

inet.0: 21 destinations, 22 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/32          *[IS-IS/18] 19:39:55, metric 20
                    > to 10.0.1.0 via ge-0/0/0.0

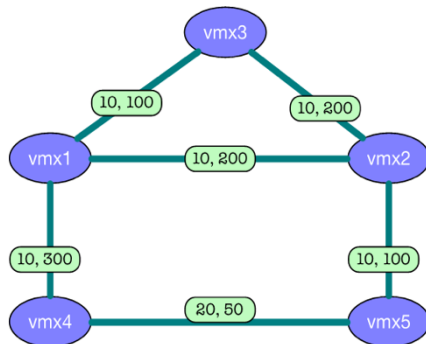
inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/32          *[L-ISIS/14] 19:39:55, metric 20
                    > to 10.0.1.0 via ge-0/0/0.0, Push 800004
```

```
tesuto@mx3> traceroute 10.0.0.4
traceroute to 10.0.0.4 (10.0.0.4), 30 hops max, 52 byte packets
 1  10.0.1.0 (10.0.1.0)  10.400 ms  14.238 ms  15.999 ms
 2  10.0.0.4 (10.0.0.4)  23.912 ms  18.408 ms  15.831 ms
```

Problem 1 - shorted path mx3 to mx4

- Parsed topology information shown from getted-database-information netconf RPC call.
- Load graph and use SPF to calculate mx3 -> mx4 shortest path



```
tesuto@dev:~/pod3$ ./main.py --problem 1
mx1 10.0.0.1 sid 800001
  4 cost 10 capacity 300 link_sid 299776
  2 cost 10 capacity 200 link_sid 299792
  3 cost 10 capacity 100 link_sid 299808
mx2 10.0.0.2 sid 800002
  1 cost 10 capacity 200 link_sid 299776
  5 cost 10 capacity 100 link_sid 299792
  3 cost 10 capacity 200 link_sid 299808
mx3 10.0.0.3 sid 800003
  1 cost 10 capacity 100 link_sid 299776
  2 cost 10 capacity 200 link_sid 299792
mx4 10.0.0.4 sid 800004
  1 cost 10 capacity 300 link_sid 299776
  5 cost 20 capacity 50 link_sid 299792
mx5 10.0.0.5 sid 800005
  4 cost 20 capacity 50 link_sid 299776
  2 cost 10 capacity 100 link_sid 299792
[3, 1, 4]
Wrote 1 routes
```

Problem 1 - route and traceroute results

```
tesuto@mx3> show route 10.0.0.4

inet.0: 21 destinations, 23 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/32          *[BGP/13] 00:00:54, localpref 100, from 100.96.0.24
                    AS path: I, validation-state: unverified
                    > to 10.0.1.0 via ge-0/0/0.0, Push 800004
                    [IS-IS/18] 19:42:19, metric 20
                    > to 10.0.1.0 via ge-0/0/0.0

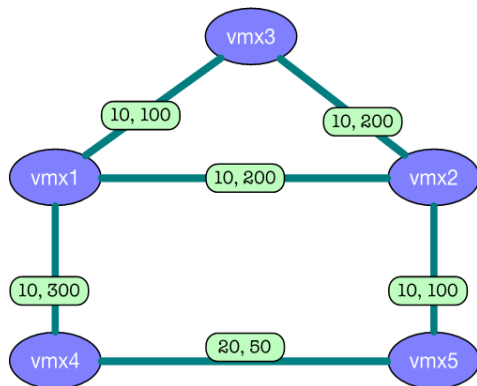
inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/32          *[L-ISIS/14] 19:42:19, metric 20
                    > to 10.0.1.0 via ge-0/0/0.0, Push 800004
```

```
tesuto@mx3> traceroute 10.0.0.4
traceroute to 10.0.0.4 (10.0.0.4), 30 hops max, 52 byte packets
 1  10.0.1.0 (10.0.1.0)  13.400 ms  42.641 ms  11.416 ms
    MPLS Label=800004 CoS=0 TTL=1 S=1
 2  10.0.0.4 (10.0.0.4)  16.215 ms  14.271 ms  23.993 ms
```

Problem 2 - mx3 to mx4 200Mbps constrained

- Remove edges with less than 200Mbps of bandwidth from graph
- Use SPF to compute shortest path using remaining edges



```
tesuto@dev:~/pod3$ ./main.py --problem 2
mx1 10.0.0.1 sid 800001
  4 cost 10 capacity 300 link_sid 299776
  2 cost 10 capacity 200 link_sid 299792
  3 cost 10 capacity 100 link_sid 299808
mx2 10.0.0.2 sid 800002
  1 cost 10 capacity 200 link_sid 299776
  5 cost 10 capacity 100 link_sid 299792
  3 cost 10 capacity 200 link_sid 299808
mx3 10.0.0.3 sid 800003
  1 cost 10 capacity 100 link_sid 299776
  2 cost 10 capacity 200 link_sid 299792
mx4 10.0.0.4 sid 800004
  1 cost 10 capacity 300 link_sid 299776
  5 cost 20 capacity 50 link_sid 299792
mx5 10.0.0.5 sid 800005
  4 cost 20 capacity 50 link_sid 299776
  2 cost 10 capacity 100 link_sid 299792
[3, 2, 1, 4]
Wrote 1 routes
```


Problem 2 - route and traceroute results

```
tesuto@mx3> show route 10.0.0.4

inet.0: 21 destinations, 23 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/32      *[BGP/13] 00:00:00, localpref 100, from 100.96.0.24
                  AS path: I, validation-state: unverified
                  > to 10.0.1.7 via ge-0/0/1.0, Push 800004, Push 800001(top)
                  [IS-IS/18] 19:44:00, metric 20
                  > to 10.0.1.0 via ge-0/0/0.0

inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

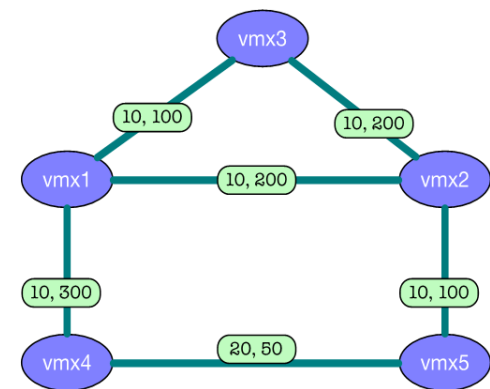
10.0.0.4/32      *[L-ISIS/14] 19:44:00, metric 20
                  > to 10.0.1.0 via ge-0/0/0.0, Push 800004

tesuto@mx3> traceroute 10.0.0.4
traceroute to 10.0.0.4 (10.0.0.4), 30 hops max, 52 byte packets
 1  10.0.1.7 (10.0.1.7)  15.133 ms  14.110 ms  16.202 ms
    MPLS Label=800001 CoS=0 TTL=1 S=0
    MPLS Label=800004 CoS=0 TTL=1 S=1
 2  10.0.1.2 (10.0.1.2)  25.852 ms  20.032 ms  24.052 ms
    MPLS Label=800004 CoS=0 TTL=1 S=1
 3  10.0.0.4 (10.0.0.4)  103.814 ms  38.933 ms  39.423 ms
```

Problem 3 - path1 200Mbps, path2 60Mbps

- Path1 200Mbps from mx3 -> mx4, path2 60Mbps from mx5 -> mx4
- Create a directional graph of the network and subtract the bandwidth used by the 200Mbps path1
- Find the shortest path with at least 60Mbps of remaining bandwidth for path2

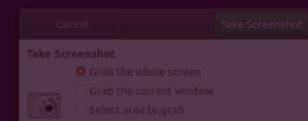
```
tesuto@dev:~/pod3$ ./problem3.py
path1 200Mbps: ['3', '2', '1', '4']
path2 60Mbps: ['5', '2', '3', '1', '4']
tesuto@dev:~/pod3$
```



Problem 3 - 2nd solution using path splitting

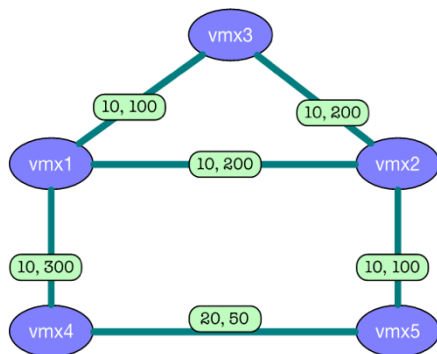
- Find shortest path for path1 (mx3 -> mx4) and determine how much bandwidth it can carry
- Subtract used bandwidth from edges for path1 and repeat SPF for remaining path1 bandwidth
- Repeat same steps for path2 from mx5 -> mx4 with 60Mbps

```
tesuto@dev:~/pod3$ ./extract_topo_compute_paths.py | grep Using
Using path [3, 1, 4] with rate 100
Using path [3, 2, 1, 4] with rate 100
Using path [5, 4] with rate 50
Using path [5, 2, 1, 4] with rate 10
tesuto@dev:~/pod3$
```



Problem 4 - disjoint paths

- Compute the shortest path for path1
- Remove all the edges from the graph used by path1
- Compute the shortest path for path2 using the remaining edges



```
tesuto@dev:~/pod3$ ./main.py --problem 4
mx1 10.0.0.1 sid 800001
  4 cost 10 capacity 300 link_sid 299776
  2 cost 10 capacity 200 link_sid 299792
  3 cost 10 capacity 100 link_sid 299808
mx2 10.0.0.2 sid 800002
  1 cost 10 capacity 200 link_sid 299776
  5 cost 10 capacity 100 link_sid 299792
  3 cost 10 capacity 200 link_sid 299808
mx3 10.0.0.3 sid 800003
  1 cost 10 capacity 100 link_sid 299776
  2 cost 10 capacity 200 link_sid 299792
mx4 10.0.0.4 sid 800004
  1 cost 10 capacity 300 link_sid 299776
  5 cost 20 capacity 50 link_sid 299792
mx5 10.0.0.5 sid 800005
  4 cost 20 capacity 50 link_sid 299776
  2 cost 10 capacity 100 link_sid 299792
path1: [3, 1, 4]
path2: [3, 2, 5, 4]
Wrote 2 routes
```

Learnings from the Hack

- The initial problem was daunting, but breaking into smaller steps really helped
- Great way to network with other professionals and learn something new
- Practical experience with the python ncclient/netconf, networkx, and exabgp modules
- When working with ncclient rpc calls, the junos “show <command> | display xml rpc” command is your friend.
- The networkx python module provides very useful graph data structures and algorithms
- Problem 4 - Don't forget path identifiers / path-information in exaBGP advertised routes :-)



Extra Images - Exabgp Output

```
01:36:31 | 17546 | api | route added to neighbor 100.96.0.8 local-ip 100.96.0.24 local-as 65535 peer-as 65535 router-id 9.9.9.9 family-allowed in-open : 10.0.0.4/32 label 800004 next-hop 10.0.1.0
01:36:32 | 17546 | api | route added to neighbor 100.96.0.8 local-ip 100.96.0.24 local-as 65535 peer-as 65535 router-id 9.9.9.9 family-allowed in-open : 10.0.0.4/32 label 800004 next-hop 10.0.1.0
```

<- 1

```
01:38:12 | 17546 | api | route added to neighbor 100.96.0.8 local-ip 100.96.0.24 local-as 65535 peer-as 65535 router-id 9.9.9.9 family-allowed in-open : 10.0.0.4/32 label [ 800001 800004 ] next-hop 10.0.1.7
01:38:12 | 17546 | api | route added to neighbor 100.96.0.8 local-ip 100.96.0.24 local-as 65535 peer-as 65535 router-id 9.9.9.9 family-allowed in-open : 10.0.0.4/32 label [ 800001 800004 ] next-hop 10.0.1.7
```

<- 2

```
01:39:48 | 17546 | api | route added to neighbor 100.96.0.8 local-ip 100.96.0.24 local-as 65535 peer-as 65535 router-id 9.9.9.9 family-allowed in-open : 10.0.0.4/32 label 800004 next-hop 10.0.1.0
01:39:48 | 17546 | api | route added to neighbor 100.96.0.8 local-ip 100.96.0.24 local-as 65535 peer-as 65535 router-id 9.9.9.9 family-allowed in-open : 10.0.0.4/32 label [ 800005 800004 ] next-hop 10.0.1.7
```

<- 4 - two
routes