Telemetry Study Notebook

For: August 29-September 4	LATEST UPDATES	*Last updated: August 28*
Status	Description	Comments
Last completed *Due date:*	1. Added perfSONAR info to background section of paper 2.	
Working on currently/next	Currently working on : Adding citations to research paper Next to work on: Finding additional supplemental materials for the paper Next milestones: Access to DMZ? Running experiments with perfSONAR Problems: perfSONAR nodes blocked by LEO firewall	

Plan & Timeline

Aug 29 - Sept 4	Work on paper (decide more measurements to run?); RIPE Atlas measurements
Sept 5 - Sept 11	Ping Measurements

Experiment 1:

- Title: perfSONAR Node Ping
- Aim: I am going to start small by gathering ping data between perfSONAR nodes and LEO, then I will progress to ping data on ScienceDMZ. This data is going to be compared to see differences between the networks
- Data: Link and description of the data you are using
- Code: Where is this code? Every bit of code should be in GitHub. If you need a new repository, come talk to me.
- Methods: Please include as much detail as possible. These include any references, methods, calculations.
    - Possible perfSONAR nodes:
        - 136.142.202.118 (University of Pittsburgh)
        - 141.216.99.254 (University of Michigan-Flint)
        - 192.111.110.77 (Vanderbilt University)
        - 66.99.43.226 (Illinois Century Network)
        - 206.71.76.62 (Salt Lake City, UT)
        - 207.189.117.10 (Hillsboro, OR)
        - 209.170.192.2 (Las Vegas, NV)
        - 216.58.152.198 (Richardson, TX)

        - 72.253.66.3 (Honolulu, HI)
        - 67.58.50.74 (La Jolla, CA)
- Results: Both Raw data and graphs/figures are important. Make sure we have a link or the actual data in the notebook.
- Conclusions: What did we learn? Does it match with the aim?
- Caveats: Is there a specific condition where things don't work? Note those.
- ESMOD
-

For: September 19-September 23 LATEST UPDATES *Last updated: September 21*		
Status	Description	Comments
Last completed *Due date:*	3. 4.	
Working on currently/next	Currently working on : Testing pSchdeduler commands on Google Cloud VM to decide how to do measurements on Leo Next to work on: Running test on Leo Next milestones: Graphing test results Problems: perfSONAR not installed, installation blocked by LEO firewall	

Plan & Timeline

Sept 19-23	Experiment 1, planning + troubleshooting
Sept 26-30	Install perfSONAR, run experiment
Oct 3-7	Graph Results

For: September 26-September 30 LATEST UPDATES *Last updated: September 27*		
Status	Description	Comments
Last completed *Due date:*	5. Installed perfSONAR on Leo 6. Tested IP Addresses for Pscheduler Activity	
Working on currently/next	Currently working on : Setting up measurements to run on Leo Next to work on: Setting up program to graph perfSONAR json files; Run tests on DMZ? Next milestones: Graphing test results Problems: None of the sample perfSONAR nodes are functional (some have no activity, some time out)	

Plan & Timeline

Sept 26-30	Install perfSONAR, run experiment
Oct 3-7	Graph Results
Oct 8-13	

- perfSONAR nodes:
  - 136.142.202.118 (University of Pittsburgh)
  - 141.216.99.254 (University of Michigan-Flint)
  - 192.111.110.77 (Vanderbilt University)
  - 66.99.43.226 (Illinois Century Network)
  - 206.71.76.62 (Salt Lake City, UT)
  - 207.189.117.10 (Hillsboro, OR)
  - 209.170.192.2 (Las Vegas, NV)
  - 216.58.152.198 (Richardson, TX)
  - 72.253.66.3 (Honolulu, HI) - returned no data with normal ping
  - 67.58.50.74 (La Jolla, CA)

(bright red) - pscheduler not active

(blue) - timed out

(dark red) - error /screenshot below/

```
root@leo:/etc/apt/sources.list.d# pscheduler ping 67.58.50.74
67.58.50.74: Returned status 500: <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>500 Internal Server Error</title>
</head><body>
<h1>Internal Server Error</h1>
<p>The server encountered an internal error or
misconfiguration and was unable to complete
your request.</p>
<p>Please contact the server administrator,
 root@localhost and inform them of the time the error occurred,
and anything you might have done that may have
caused the error.</p>
<p>More information about this error may be available
in the server error log.</p>
<hr>
<address>Apache/2.2.15 (CentOS) Server at 67.58.50.74 Port 443</address>
</body></html>
```

For: October 17-October 21 LATEST UPDATES		*Last updated: October 17*
Status	Description	Comments
Last completed *Due date:*	7. Created Ping Script 8. Ran Brief Sample Test with 3 IP Addresses	
Working on currently/next	Currently working on : Converting to JSON Next to work on: Graphing results; committing script to github Next milestones: Problems: Leo automatically disconnects after ~1 hour and it appears that tests stop running; results are added to log file out of order (can, most likely, be easily fixed in the JSON conversion)	

Plan & Timeline

Oct 17-21	Convert to JSON
Oct 24-28	Graph Results
Oct 31- Nov 4	

Ping results sent by the script to a log file are shown below. They are out of order now but can be fixed when converted to JSON. This particular test only took place over the course of a few minutes and with only a few of the selected IP addresses in order to receive sample data, other tests will be longer and use all of the IP addresses.

```
141.216.99.254 206.71.76.62 207.189.117.10
Start Time: 6:07pm Oct 15
PING 141.216.99.254 (141.216.99.254) 56(84) bytes of data.
64 bytes from 141.216.99.254: icmp_seq=1 ttl=42 time=47.3 ms
PING 206.71.76.62 (206.71.76.62) 56(84) bytes of data.
64 bytes from 206.71.76.62: icmp_seq=1 ttl=48 time=62.7 ms
PING 207.189.117.10 (207.189.117.10) 56(84) bytes of data.
64 bytes from 207.189.117.10: icmp_seq=1 ttl=47 time=82.1 ms
64 bytes from 141.216.99.254: icmp_seq=2 ttl=42 time=46.8 ms
64 bytes from 206.71.76.62: icmp_seq=2 ttl=48 time=62.4 ms
64 bytes from 207.189.117.10: icmp_seq=2 ttl=47 time=81.6 ms
64 bytes from 141.216.99.254: icmp_seq=3 ttl=42 time=46.4 ms
64 bytes from 206.71.76.62: icmp_seq=3 ttl=48 time=62.2 ms
64 bytes from 207.189.117.10: icmp_seq=3 ttl=47 time=81.5 ms
64 bytes from 141.216.99.254: icmp_seq=4 ttl=42 time=46.8 ms
64 bytes from 206.71.76.62: icmp_seq=4 ttl=48 time=62.1 ms
64 bytes from 207.189.117.10: icmp_seq=4 ttl=47 time=81.4 ms
64 bytes from 141.216.99.254: icmp_seq=5 ttl=42 time=50.5 ms
64 bytes from 206.71.76.62: icmp_seq=5 ttl=48 time=66.1 ms
64 bytes from 207.189.117.10: icmp_seq=5 ttl=47 time=85.3 ms
64 bytes from 141.216.99.254: icmp_seq=6 ttl=42 time=46.6 ms
64 bytes from 206.71.76.62: icmp_seq=6 ttl=48 time=62.3 ms
64 bytes from 207.189.117.10: icmp_seq=6 ttl=47 time=81.7 ms
64 bytes from 141.216.99.254: icmp_seq=7 ttl=42 time=47.1 ms
64 bytes from 206.71.76.62: icmp_seq=7 ttl=48 time=62.5 ms
64 bytes from 207.189.117.10: icmp_seq=7 ttl=47 time=81.7 ms
64 bytes from 141.216.99.254: icmp_seq=8 ttl=42 time=49.4 ms
64 bytes from 206.71.76.62: icmp_seq=8 ttl=48 time=65.2 ms
64 bytes from 207.189.117.10: icmp_seq=8 ttl=47 time=83.8 ms
64 bytes from 141.216.99.254: icmp_seq=9 ttl=42 time=98.0 ms
64 bytes from 206.71.76.62: icmp_seq=9 ttl=48 time=99.9 ms
64 bytes from 207.189.117.10: icmp_seq=9 ttl=47 time=121 ms
64 bytes from 141.216.99.254: icmp_seq=10 ttl=42 time=46.6 ms

--- 141.216.99.254 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 90095ms
rtt min/avg/max/mdev = 46.478/52.599/98.071/15.213 ms
64 bytes from 206.71.76.62: icmp_seq=10 ttl=48 time=62.4 ms

--- 206.71.76.62 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 90093ms
rtt min/avg/max/mdev = 62.164/66.844/99.917/11.107 ms
64 bytes from 207.189.117.10: icmp_seq=10 ttl=47 time=81.3 ms

--- 207.189.117.10 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 90095ms
rtt min/avg/max/mdev = 81.389/86.204/121.181/11.728 ms
```

For: November 6-November 13 LATEST UPDATES *Last updated: October 17*		
Status	Description	Comments
Last completed *Due date:*	9. Created Graph Script	
Working on currently/next	Currently working on : Creating script that shows data from more than one measurement (bar graph for averages of individual ip addresses, cdf displayed above) Next to work on: Create traceroute measurement script Next milestones: Problems:	

Plan & Timeline

Nov 6 - 13	Graph Script
Nov 14 - 20	Traceroute
Nov 21 - 27	

For: January 23-29	LATEST UPDATES	*Last updated:*
Status	Description	Comments
Last completed *Due date:*	10. Switched focus from diy measurements and database to perfSONAR	
Working on currently/next	Currently working on : figuring out how to use perfSONAR data archive feature Next to work on: run measurements and create graph scripts to collaborate with perfSONAR data Next milestones: run measurements in the DMZ Problems: can't find a video tutorial for data archive feature, but did find some documents that may help	

Plan & Timeline

Jan 23 - 29	Figure out perfSONAR
Jan 30 - Feb 5	Graphs?
Feb 6 - 12	Could work on analysis

For: January 23-29 LATEST UPDATES		*Last updated: January 23, 2023*
Status	Description	Comments
Last completed *Due date:*	11. Figured out how to archive measurements with perfSONAR	
Working on currently/next	Currently working on : run measurements for ping and traceroute with the IP address list Next to work on: create graph scripts to collaborate with perfSONAR data Next milestones: run measurements in the DMZ Problems: can’t find a video tutorial for data archive feature, but did find some documents that may help	

Plan & Timeline

Jan 23 - 29	Figure out perfSONAR
Jan 30 - Feb 5	Graphs?
Feb 6 - 12	Could work on analysis

Created API Key with username:

/usr/sbin/esmond_manage add_ps_metadata_post_user example_user
/usr/sbin/esmond_manage add_timeseries_post_user example_user

Use resulting key when using archive command in command line

Example Archive command:

pscheduler task --archive '{"archiver": "esmond","data":{"url":"http://localhost/esmond/perfsonar/archive/","_auth-token": "abc123"}}' trace --dest www.perfsonar.net

Use curl to display different measurement results, different uris display different sections of data

command | json_pp : formats json data command-line

Measurement requirements:

- ping should run every 10 minutes
- traceroute should run every hour
- list of ip addresses:
    - 136.142.202.118 (University of Pittsburgh)
    - 141.216.99.254 (University of Michigan-Flint)
    - 192.111.110.77 (Vanderbilt University)
    - 66.99.43.226 (Illinois Century Network)
    - 206.71.76.62 (Salt Lake City, UT)
    - 207.189.117.10 (Hillsboro, OR)
    - 209.170.192.2 (Las Vegas, NV)
    - 216.58.152.198 (Richardson, TX)
    - 72.253.66.3 (Honolulu, HI) - returned no data with normal ping
    - 67.58.50.74 (La Jolla, CA)

For: January 30-February 5 LATEST UPDATES *Last updated: February 01, 2023*		
Status	Description	Comments
Last completed *Due date:*	12. Started one set of ping and traceroute measurements	
Working on currently/next	Currently working on : run measurements for ping and traceroute with the IP address list Next to work on: create graph scripts to collaborate with perfSONAR data Next milestones: run measurements in the DMZ Problems:	

Plan & Timeline

Jan 30 - Feb 5	Gather data, figure out how to access data externally?
Feb 6 - 12	Graphs?
Feb 13 - 20	TBD

```
curl "http://localhost/esmond/perfsonar/archive/" | json_pp
```

^^^ command to access archive in LEO

For: February 6-February 12 LATEST UPDATES *Last updated: February 02, 2023*		
Status	Description	Comments
Last completed *Due date:*	13. Started measurements for entire IP address list	
Working on currently/next	Currently working on : run measurements for ping and traceroute with the IP address list Next to work on: create graph scripts to collaborate with perfSONAR data Next milestones: run measurements in the DMZ Problems: all of the trace measurements return no response. I think it may be the same issue that I had last semester with normal traceroutes running UDP instead of ICMP, but I haven't been able to find anything to fix that when using perfsonar	

Plan & Timeline

Feb 6 - 12	Gather data, ping graphs
Feb 13 - 19	Traceroute graphs?
Feb 20 - 26	TBD

Example commands for perfSONAR tests:

```
pscheduler task --archive '{"archiver": "esmond","data":{"url":"http://localhost/esmond/perfsonar/archive/",
"_auth-token":"789ab4ef06de7bf4a1e2dd1d79ad2cec3e8f81c6"}}' --repeat PT10M --quiet rtt --dest
67.58.50.74
```

Runs rtt task every 10 minutes to IP address 67.58.50.74. Archives in given url using esmond

```
pscheduler task --archive '{"archiver": "esmond","data":{"url":"http://localhost/esmond/perfsonar/archive/",
"_auth-token":"789ab4ef06de7bf4a1e2dd1d79ad2cec3e8f81c6"}}' --repeat PT60M –probe-type icmp
--quiet trace --dest 67.58.50.74
```

Runs trace task every hour to IP address 67.58.50.74. Archives in given url using esmond

Access the archived results using curl and the specified url, archive lists all of the archived measurements and lists extended urls with more information about each measurements.

Each user must create a unique API key to archive with esmond

RTT archive has a histogram statistics feature that lists mean,median,and standard deviation of rtt (could be used for the CDF graphs)

For: February 13-February 19 LATEST UPDATES *Last updated: February 07, 2023*		
Status	Description	Comments
Last completed *Due date:*	14. Gathering significant rtt data, trace still not working (trying to find a solution) 15. Started working on graphs 16. Completed script to graph CDF of multiple ping measurements	
Working on currently/next	Currently working on : run measurements for ping and traceroute with the IP address list; creating graphs for single measurements, packet loss, and traceroute graphs (after I get trace working on perfsonar) Next to work on: Next milestones: run measurements in the DMZ Problems:	

Plan & Timeline

Feb 13 - 19	Gather data, ping graphs
Feb 20 - 26	TBD
Feb 27 - Mar 5	TBD

useful archive extensions (tacked onto curl urls mentioned previously):

/?time-range=86400 : results from past 24 hours

=3600 : results from past hour

histogram-rtt/statistics : mean,median,standard deviation,variance,etc

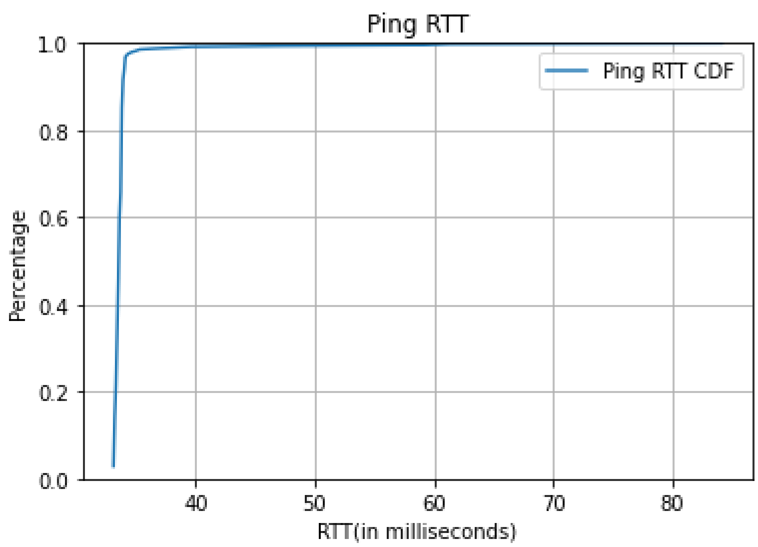
histogram-rtt/aggregation :counts of rtt times

(both can be grouped by 24-hour intervals and 1-hour intervals)

I am going to use 24-hour intervals for now

I believe I will need to use both histogram-rtt/statistics and histogram-rtt/aggregation to create a CDF graph for the RTT, will need to find more information on calculating CDF since this data is in a different format from what was used in past graphs

^ actually only needed aggregation



Sample Ping CDF from IP address 66.99.43.226 for 24-hour span on February 12, 2023

It appears that there is 100% packet loss for IP Address 72.253.66.3, but that is not unusual as this IP Address has been recorded to have this issue in the past

Traceroute on Leo (Susmit)

Question: Is there a way to do this with perfsonar?

```
susmit@leo:~$ traceroute 66.99.43.226
```

```
traceroute to 66.99.43.226 (66.99.43.226), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  *^C
```

```
susmit@leo:~$ sudo traceroute -I 66.99.43.226
[sudo] password for susmit:
traceroute to 66.99.43.226 (66.99.43.226), 30 hops max, 60 byte packets
 1  * * *
 2  149.149.2.1 (149.149.2.1)  2.786 ms  2.552 ms  2.975 ms
 3  63.135.176.205 (63.135.176.205)  3.204 ms  3.011 ms  2.975 ms
 4  host-63-135-177-7.twlakes.net (63.135.177.7)  3.973 ms  3.776 ms  3.923 ms
 5  * * *
 6  be2848.ccr42.atl01.atlas.cogentco.com (154.54.6.117)  12.787 ms  12.568 ms  12.398 ms
 7  be3365.rcr21.ind01.atlas.cogentco.com (154.54.27.18)  37.987 ms  37.995 ms  38.001 ms
 8  be3181.ccr42.ord01.atlas.cogentco.com (154.54.46.162)  29.797 ms  29.662 ms  29.777 ms
 9  be2522.agr21.ord01.atlas.cogentco.com (154.54.81.62)  29.798 ms  29.902 ms  29.607 ms
10  te0-0-2-1.nr12.b010917-1.ord01.atlas.cogentco.com (154.24.64.26)  30.281 ms  30.424 ms  29.967 ms
11  38.88.49.110 (38.88.49.110)  34.303 ms  34.346 ms  34.597 ms
12  206.166.35.25 (206.166.35.25)  34.778 ms  34.801 ms  34.938 ms
13  206.166.35.13 (206.166.35.13)  35.250 ms  35.272 ms  34.868 ms
14  66.99.43.226 (66.99.43.226)  33.481 ms  33.393 ms  33.249 ms
susmit@leo:~$
```

For: February 26-March 4 LATEST UPDATES		*Last updated: March 2, 2023*
Status	Description	Comments
Last completed *Due date:*	17. Gathering significant rtt data 18. Solved trace issue 19. Working on stopping all currently faulty trace tests and replacing with trace tests using –probe-type	
Working on currently/next	Currently working on : run measurements for ping and traceroute with the IP address list Next to work on: traceroute & packet loss graphs Next milestones: run measurements in the DMZ Problems:	

Plan & Timeline

Feb 27 - Mar 5	Gather data, ping graphs, restart trace tests
Mar 6 - Mar 11	Trace graphs
Mar 12 - Mar 19	Trace Graphs (multi-measurement plotting)

perfSONAR Tools Ports		
Tool	**TCP ports**	**UDP Ports**
owamp (control)	861	
owamp (test)		8760-9960
twamp (control)	862	
twamp (test)		18760-19960
pscheduler	443	
traceroute		33434-33634
simplestream	5890-5900	
nuttcp	5000, 5101	
iperf3	5201	
iperf2	5001	
ntp		123

Ports used for perfSONAR

The `trace` test has a `probe-type` parameter (and a matching switch for the CLI) that can be `icmp`, `udp` or `tcp`:

```
pscheduler task trace --probe-type icmp --dest www.perfsonar.net
```

The `traceroute` tool supports all three, but I did just notice a bug in the pScheduler plugin that doesn't set the right permissions to do TCP. I've opened a ticket on that (https://github.com/perfsonar/pscheduler/issues/1324

(from perfSONAR customer service)

```
ebmutter42@leo:~$ pscheduler task trace --probe-type icmp --dest 206.71.76.62
Submitting task...
Task URL:
https://localhost/pscheduler/tasks/d6008e84-272d-464e-8f49-21f43920eb03
Running with tool 'traceroute'
Fetching first run...

Next scheduled run:
https://localhost/pscheduler/tasks/d6008e84-272d-464e-8f49-21f43920eb03/runs/c0fd83e2-af40-4909-8c1c-056b83d84ccf
Starts 2023-03-02T08:50:03-06 (~2 seconds)
Ends   2023-03-02T08:50:11-06 (~7 seconds)
Waiting for result...

1       149.149.10.252 AS19331 0.4 ms
          TWIN-LAKES, US
2       149.149.2.1 AS19331 2.6 ms
          TWIN-LAKES, US
3       63.135.176.205 AS19331 2.8 ms
          TWIN-LAKES, US
4       ce-2-1-0.a02.atlnga05.us.bb.gin.ntt.net (128.241.219.93) AS2914 13.3 ms
          NTT-LTD-2914, US
5       ae10.cr2-atl2.ip4.gtt.net (199.229.229.201) AS3257 11.4 ms
          GTT-BACKBONE GTT, US
6       ae22.cr4-atl2.ip4.gtt.net (89.149.141.94) AS3257 14.1 ms
          GTT-BACKBONE GTT, US
7       72.29.215.51 AS3257 11.6 ms
          GTT-BACKBONE GTT, US
8       be21.bbrt01.atl10.flexential.net (148.66.237.104) AS13649 64.1 ms
          ASN-VINS, US
9       be136.bbrt01.dal01.flexential.net (148.66.237.92) AS13649 64.5 ms
          ASN-VINS, US
10      be10.bbrt02.dal01.flexential.net (66.51.7.42) AS13649 64.5 ms
          ASN-VINS, US
11      be106.bbrt01.las03.flexential.net (66.51.7.70) AS13649 64.4 ms
          ASN-VINS, US
12      be10.bbrt02.las03.flexential.net (66.51.7.90) AS13649 63.5 ms
          ASN-VINS, US
13      be102.bbrt02.slc09.flexential.net (66.51.5.41) AS13649 63.5 ms
          ASN-VINS, US
14      be150.bbrt01.slc04.flexential.net (66.51.9.125) AS13649 63.7 ms
          ASN-VINS, US
15      be26.crrt01.slc04.flexential.net (66.51.4.242) AS13649 63.5 ms
          ASN-VINS, US
16      vl62.aggm01.slc04.viawest.net (66.51.1.94) AS13649 63.3 ms
          ASN-VINS, US
17      speedster.slc04.viawest.net (206.71.76.62) AS13649 62.8 ms
          ASN-VINS, US

No further runs scheduled.
```

Results from trace using –probe-type

Paper plan (In addition to what we have in overleaf):

To measure the performance of a ScienceDMZ, we need to deploy multiple measurement instruments at strategic points across the campus. These measurement instruments should include PerfSonar nodes, docker containers with necessary software (iperf3, ping, traceroute, Wireshark, etc.), and packet capture instruments (both optical taps and mirrored ports as required).

We should perform the following experiments to quantify the improvements brought forward by the new ScienceDMZ:

1. LAN performance measurements: We should measure the network performance between the ScienceDMZ gateway, the data transfer node (DTN), and the individual research boxes. This measurement should include parameters such as throughput, RTT (round-trip time) between nodes, jitter, packet loss, bottlenecks, and congestion, and error rates (checksum errors, CRC).

2. Data transfer performance: We should also measure data transfer performance between research boxes, including duration, data size, type of traffic (video, web, file transfer), packet loss sensitivity, per-flow throughput, and number of flows.

3. WAN performance measurements: We should measure both inbound and outbound traffic parameters, including BGP (Border Gateway Protocol) routes to and from external vantage points, path length between campus and external vantage points, IPv6 reachability, RTT between node and the gateway, and packet loss.
    a. Populate a list of destinations
    b. IP address (top 5 external IP addresses on campus)

4. DMZ traffic characteristics: We should measure the characteristics of DMZ traffic, including duration, data size, type of traffic, packet loss sensitivity, per-flow throughput, and number of flows.
    a. S-flow data <- Juniper

5. Data transfer software evaluation: We should evaluate different data transfer software, including FTP, Globus, ASPERA, FDT, and more, to determine which software is most appropriate for specific user requirements.

6. User feedback: We should perform a campus-wide survey and interview the PIs of the 13 identified use cases to learn about their experience after the ScienceDMZ

deployment. We should also adjust our design and deployment based on the user experience and feedback.

7. Trade-offs between security policies and performance: We should measure the trade-offs between security policies and performance, specifically how VLANs are allocated and their overall utilization. We should explore the possibility of dynamic resource allocation to various VLANs, and intelligent algorithms to expedite security checks for large data transfers.

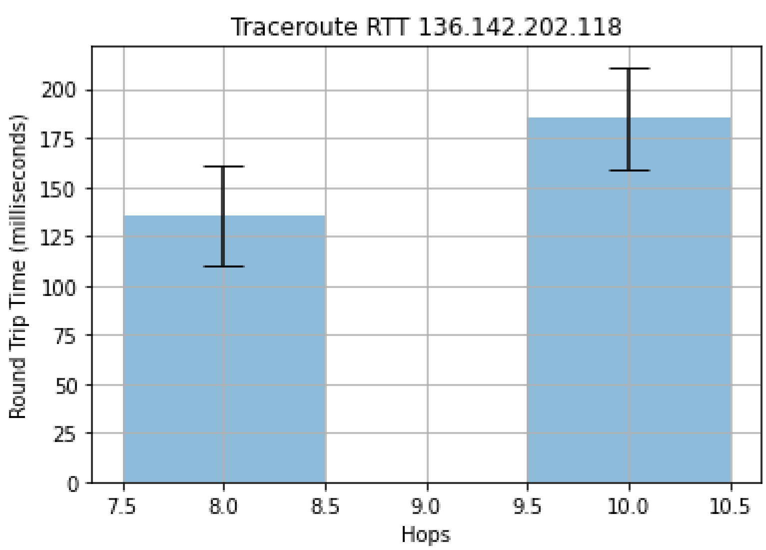
We should also perform experiments externally by deploying virtual machines on three large cloud platforms (AWS, Google Cloud, and Azure) as well as across the world using the RIPE Atlas platform. We should run continuous measurements over a few months (3-6 months minimum) and compare the results to those obtained after the ScienceDMZ deployment.

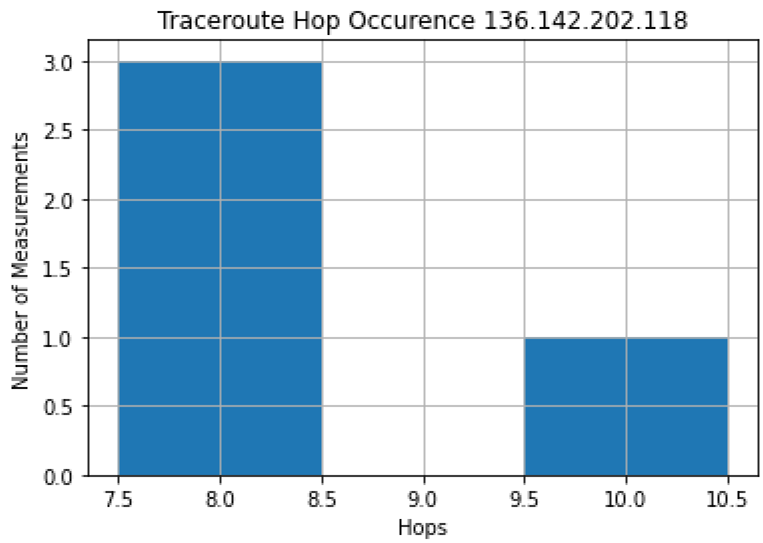
For: March 20-March 27 LATEST UPDATES		*Last updated: March 21, 2023*
Status	Description	Comments
Last completed *Due date:*	20. Gathered significant trace data 21. Completed traceroute graphs, need to error check	
Working on currently/next	Currently working on : traceroute graphs Next to work on: traceroute & packet loss graphs Next milestones: run measurements in the DMZ Problems:	

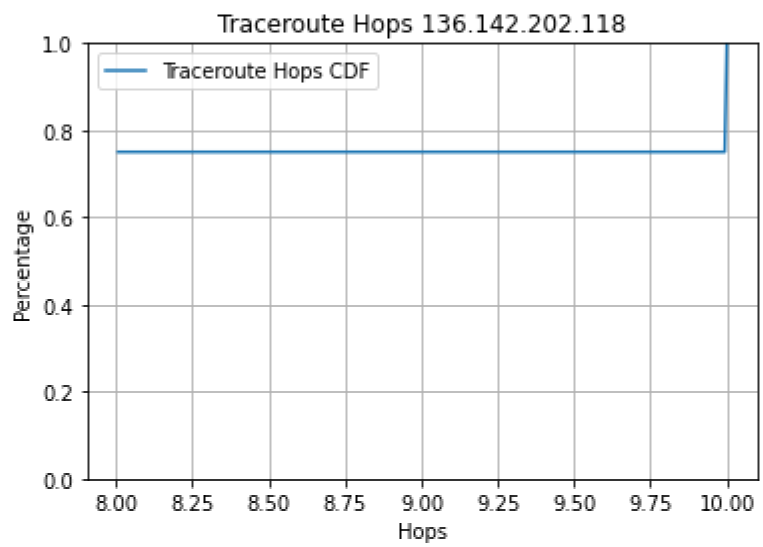
Plan & Timeline

Mar 21 - Mar 27	Trace graphs
Mar 28 - Apr 1	Altering trace graph
Apr - Apr	TBD

Traceroute graphs for 4 hours of data; destination 136.142.202.118



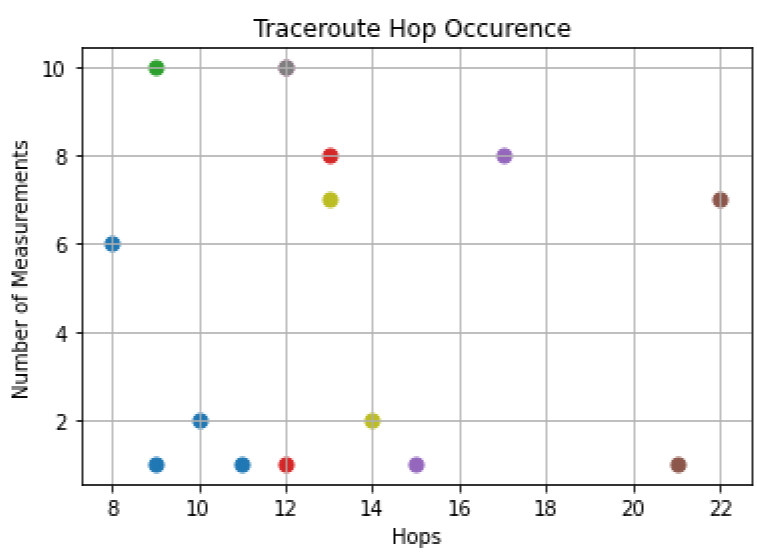

Traceroute Hop Occurence 136.142.202.118
Number of Measurements
Hops
0.0
0.5
1.0
1.5
2.0
2.5
3.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5


Traceroute Hops 136.142.202.118
Traceroute Hops CDF
Percentage
Hops
0.0
0.2
0.4
0.6
0.8
1.0
8.00
8.25
8.50
8.75
9.00
9.25
9.50
9.75
10.00

For: March 20-March 27 LATEST UPDATES		*Last updated: March 21, 2023*
Status	Description	Comments
Last completed *Due date:*	22. Altering trace graph to better portray data	
Working on currently/next	Currently working on : traceroute graphs Next to work on: traceroute & packet loss graphs Next milestones: run measurements in the DMZ Problems:	

Plan & Timeline

Mar 28 - Apr 1	Altering trace graphs
Apr 2 - Apr 8	TBD
Apr 9 - Apr 15	Updating Graph Scripts on Github; Reviewing the Paper on Overleaf to see if there is anything able to be added



Trace of all IP addresses, altered to be a scatterplot to avoid overlap of data. It is slightly more readable.

Accessing perfSONAR data from esmond archive

curl "http://localhost/esmond/perfsonar/archive/?destination=216.58.152.198" | json_pp
Example of how to find archived data for IP 216.58.152.198 (traceroute and ping data)(also includes data from previous traceroute measurements before switching to ICMP protocol, so must be sure to look for the data that is specified as "ip-transport-protocol" : "icmp")

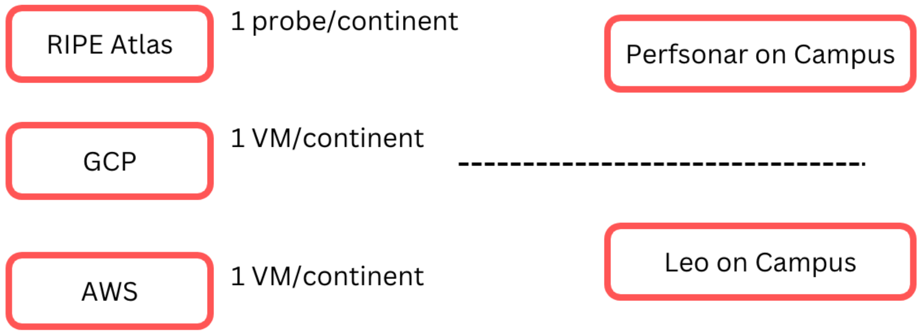
curl "http://localhost/esmond/perfsonar/archive/6370013a1c9b4070adaf05e0cc0bc188/packet-trace/base?time-range=36000" | json_pp
Example of how to find archived data of the packet traces (what you need to graph traceroute data, shows hops, rtt, and node addresses) collected in the past 36000 seconds for the measurement with a metadata key of 6370013a1c9b4070adaf05e0cc0bc188. The highlighted section can be found within the general archive, similar to the earlier curl command for destination 216.58.152.198, command would be *curl "http://localhost/esmond/perfsonar/archive" | json_pp* to see all results of all IP addresses.

Since the graph scripts must be executed externally to Leo, the data must be extracted and put into a json file. I have only worked with the smaller ends of the data (3600 second - 86400 second range) so I have been copying data from Leo and pasting it into a file on my local machine manually. This may not be the best method once we start examining larger time ranges.

For: April 17 - May 2 LATEST UPDATES		*Last updated: May 2, 2023*
Status	Description	Comments
Last completed *Due date:*	23. Updated Github 24. Revisiting Overleaf Paper 25. Tested traceroute, ping, and iperf3 26. Verified functionality of perfsonar1	
Working on currently/next	Currently working on : testing tools from the paper Next to work on: Meeting with Mike Renfro this week to verify that perfsonar1 is working, then I should be able to start running measurements; finish testing tools Next milestones: run measurements in the DMZ Problems:	

Plan & Timeline

Apr 17 - Apr 23	Update Overleaf Paper (did not accomplish much more this week due to exams)
Apr 24 - Apr 30	Tested some of the tools from the paper
May 1 - May 7	Verify that perfsonar1 is functioning correctly



(1) Start with 1 VM on each (RIPE, GCP, AWS)

(2) 2.1 - Ping <- RTT/loss/jitter
2.2 - Traceroute <- RTT/hop count/ bottleneck
2.3 - Transfer Large Files <- throughput
<- loss/retransmission
<- errors/checksum
2.4 - BGP (?) ITS (?) - look up ASN Tech/GCP
look up in the database (DGPMon)
measure over time

IPerf3: Leo should be server, cloud should be client

Run “iperf3 -s” on Leo; Run “iperf3 -c 149.149.2.70” on cloud/local machine

Standard Tools:

- Traceroute - currently working with approved IP addresses
- Ping - currently working with all IP addresses
- IPerf3 - not working with Leo as server and local machine as client; does not work with GCP, IP addresses of GCP VMs are blocked on firewall
- Wireshark - not installed on Leo yet

GCP VMs currently being used:

instance-1-em	IP address: 10.142.15.236
instance-2-em	IP address: 10.142.15.237

For: May 1 - 7 LATEST UPDATES		*Last updated: May 8, 2023*
Status	Description	Comments
Last completed *Due date:*	27. Verified functionality of perfsonar1 28. Upgraded to perfsonar 5.0 on Leo	
Working on currently/next	Currently working on : testing tools from the paper Next to work on: finish testing tools; start running measurements on DMZ Next milestones: running measurements on VMs; graphing data Problems: Esmond was replaced with OpenSearch in the version of perfsonar installed on perfsonar1, so I will have to learn how to use that. I can only ssh into perfsonar1 if I am using the campus network, so I cannot do anything with the measurements on there until I am back on campus. The password for perfsonar1 is not working, I have already emailed Dr. Renfro to get that fixed.	

Plan & Timeline

May 1 - May 7	Verify that perfsonar1 is functioning correctly
May 8 - May 14	TBD
May 15 - 21	Perfsonar1 measurements w/ OpenSearch

- 136.142.202.118 (University of Pittsburgh)
- 141.216.99.254 (University of Michigan-Flint) Removed from Perfsonar Lookup Directory Service
- 192.111.110.77 (Vanderbilt University)
- 66.99.43.226 (Illinois Century Network)
- 206.71.76.62 (Salt Lake City, UT)
- 207.189.117.10 (Hillsboro, OR)
- 209.170.192.2 (Las Vegas, NV)
- 216.58.152.198 (Richardson, TX)
- 72.253.66.3 (Honolulu, HI) - returned no data with normal ping
- 67.58.50.74 (La Jolla, CA)

149.149.2.70 (Leo IP)

For: May 22 - 28	LATEST UPDATES	*Last updated: May 31, 2023*
Status	Description	Comments
Last completed *Due date:*	29.	
Working on currently/next	Currently working on : getting measurements running with OpenSearch archiver on Leo and PerfSONAR1; jitter graphs Next to work on: finish testing tools; debug Ripe Atlas graph scripts Next milestones: graphing data Problems: rtt tasks are not working on Leo machine, but trace tasks are All measurements are failing to archive on Leo due to server errors. None of the pscheduler tasks are running on Leo. Ripe Atlas graph scripts only work for some measurement IDs, not all. Will need to debug	

Plan & Timeline

May 22 - 28	Work out issues with PerfSONAR1 and Leo
May 29 - June 4	TBD
June 5 - June 11	TBD

No server errors on perfsonar1, but when trying to see measurement results w/ “curl “https://perfsonar1.rcd.tntech.edu/logstash” with authentication, “ok” is the only thing returned in the terminal. But through the perfsonar web service it appears that all of the measurements that have been run on perfsonar1 are currently being archived through esmond without explicitly stating that they should be.

It is also possible that the server issues with leo are due to an inability to access the perfsonar web service with the hostname.

Possible IP Addresses:

66.133.111.78 (Salt Lake City, UT)

66.133.96.78 (West Jordan, UT)

68.142.131.6 (Millcreek, UT)

```
May 23 11:16:47 leo archiver WARNING  169571: Failed to archive https://leo/pscheduler/tasks/d943f80b-b537-4fae-9cd7-bca46eeaa6a3/runs/917ae6cb-457e-4267-ab73-7775ef629970 to h
ttp: Failed to post result: 503: <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
May 23 11:16:47 leo archiver WARNING  <html><head>
May 23 11:16:47 leo archiver WARNING  <title>503 Service Unavailable</title>
May 23 11:16:47 leo archiver WARNING  </head><body>
May 23 11:16:47 leo archiver WARNING  <h1>Service Unavailable</h1>
May 23 11:16:47 leo archiver WARNING  <p>The server is temporarily unable to service your
May 23 11:16:47 leo archiver WARNING  request due to maintenance downtime or capacity
May 23 11:16:47 leo archiver WARNING  problems. Please try again later.</p>
May 23 11:16:47 leo archiver WARNING  </body></html>
May 23 11:16:47 leo archiver WARNING  169571: Gave up archiving https://leo/pscheduler/tasks/d943f80b-b537-4fae-9cd7-bca46eeaa6a3/runs/917ae6cb-457e-4267-ab73-7775ef629970 to h
ttp
```

Should be able to access archived measurement results using the following command:
(LEO)
curl -X GET -H "Authorization: Basic cGVyZnNvbmFyOlFIVHV5aDI5d1JmMmU1OUhRdUFV="
https://leo/logstash/
But not currently archiving nor able to access due to server errors

Example Commands for running tests on perfsonar1 and leo with OpenSearch:

pscheduler task ---archive @local_archive.json --repeat PT10M --quiet rtt --dest 67.58.50.74

Runs rtt task every 10 minutes to IP address 67.58.50.74.

pscheduler task --archive @local_archive.json--repeat PT60M --quiet trace --dest 67.58.50.74
pscheduler task --archive @local_archive.json--repeat PT60M --quiet trace –probe-type icmp --dest 67.58.50.74

Runs trace task every hour to IP address 67.58.50.74. Uses –probe-type on leo to bypass firewall

/etc/perfsonar/opensearch/logstash_login (should have authentication information)

pscheduler task --archive '{"archiver": "http","data":{ "_url":"http://localhost/logstash", "verify-ssl": false, "op": "put", "_headers": { "x-ps-observer": "{% scheduled_by_address %}", "content-type": "application/json", "Authorization":"Basic eXamPleTokEn" }}}' trace --dest www.perfsonar.net
(example command without @local_archive.json)

https://docs.perfsonar.net/install_debian.html

https://docs.perfsonar.net/pscheduler_client_tasks.html

https://docs.perfsonar.net/multi_ma_install.html

https://docs.perfsonar.net/legacy_archive_esmond.html

For: June 5 - 11 LATEST UPDATES		*Last updated: June 5, 2023*
Status	Description	Comments
Last completed *Due date:*	30. Started ping/traceroute runs 31. Debugged RIPE Atlas ping graph script	
Working on currently/next	Currently working on : running ping/traceroute on Leo and perfsonar1; updating notebook; replicate RIPE measurements to perfsonar Next to work on: finish testing tools; debug Ripe Atlas graph scripts; experiment with cloud host; iperf3 tests Next milestones: graphing data Problems: rtt tasks are not working on Leo machine, but trace tasks are All measurements are failing to archive on Leo due to server errors. None of the pscheduler tasks are running on Leo. Not all RIPE Atlas probes are accessible from perfsonar1, need to make a list of IP addresses.	

Plan & Timeline

June 5 - June 11	Update Notebook for Annual Report, run ping/traceroute on leo and perfsonar1, replicate RIPE Atlas measurements on perfsonar1
June 12 -	TBD
June	TBD

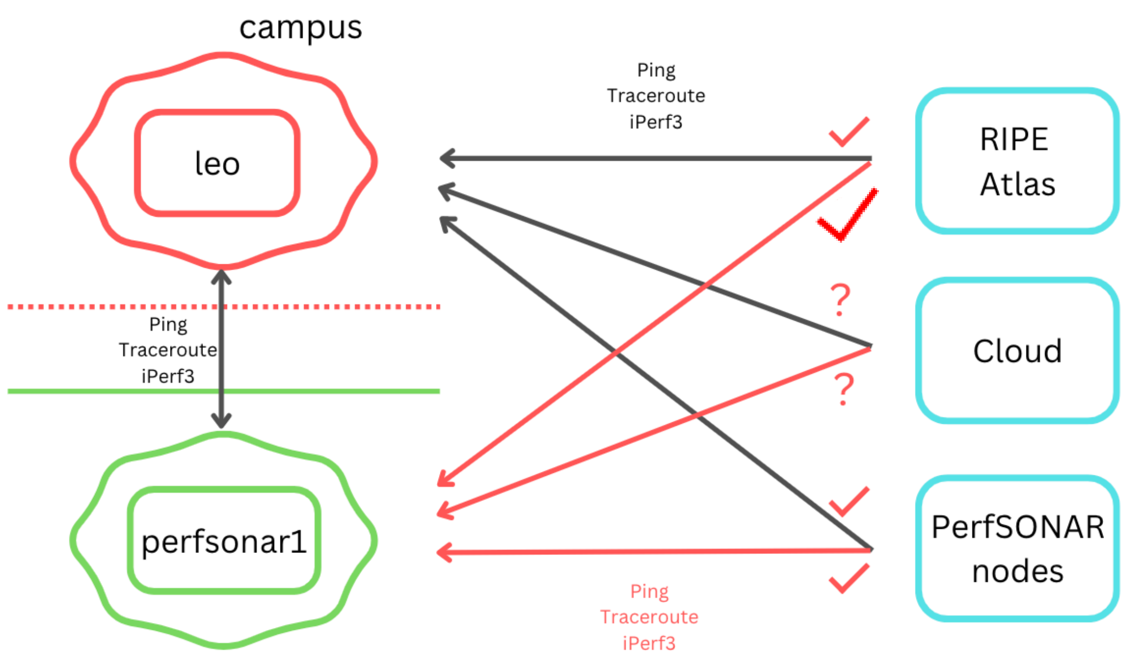
Command-line RIPE Atlas scheduling script works when run on Leo but not on perfsonar1; can run the script on Leo but change the target IP to perfsonar1's IP.

Updated IP list:

- 66.133.111.78 (Salt Lake City, UT)
- 66.133.96.78 (West Jordan, UT)
- 68.142.131.6 (Millcreek, UT)
- 136.142.202.118 (University of Pittsburgh)
- 192.111.110.77 (Vanderbilt University)

- 66.99.43.226 (Illinois Century Network)
- 206.71.76.62 (Salt Lake City, UT)
- 207.189.117.10 (Hillsboro, OR)
- 209.170.192.2 (Las Vegas, NV)
- 216.58.152.198 (Richardson, TX)
- 67.58.50.74 (La Jolla, CA)
- 149.149.2.70 (Leo IP)
- 149.149.248.20 (Perfsonar1 IP)

- (   ) used in recent test set (June 2 - June 6)



Experiments:

- RIPE Atlas:
  - 5 sets of 5 probes worldwide
  - Probe sets ping Leo every 3600 seconds (1 hour)
  - Probe sets traceroute to Leo every 21600 seconds (6 hours)
- PerfSONAR:
  - 10 Nodes within the U.S.
  - Leo pings the 10 nodes every 10 minutes
  - Leo traceroutes to the 10 nodes every hour
- Basic Ping:
  - 3 Nodes within the U.S.
  - Leo pings the 3 nodes every 30 minutes
  - Perfsonar1 pings the 3 nodes every 30 minutes
  - (On previous tests, Leo pinged a set of 5 nodes every 10 minutes)
- Basic Traceroute:

- 3 Nodes within the U.S.
- Leo traceroutes to the 3 nodes every 30 minutes
- Perfsonar1 traceroutes to the 3 nodes every 30 minutes
- (On previous tests, Leo traceroute tested a set of 5 nodes every hour)

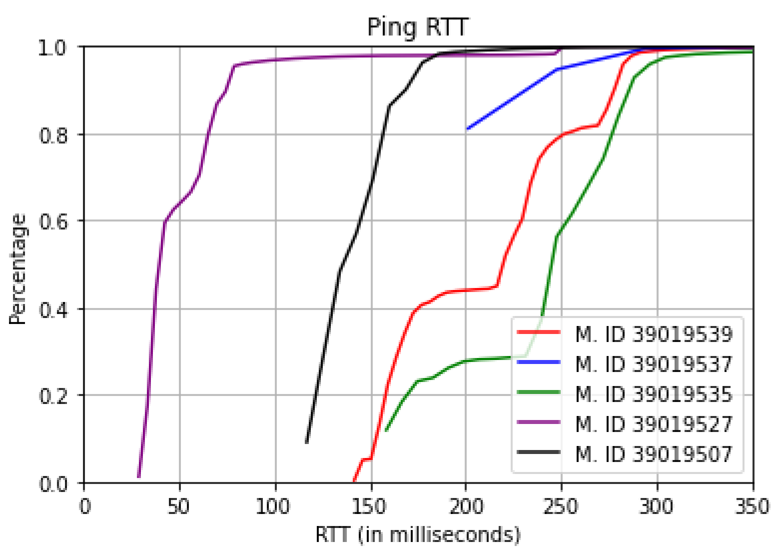
Ripe Atlas measurement updates:

Traceroute -

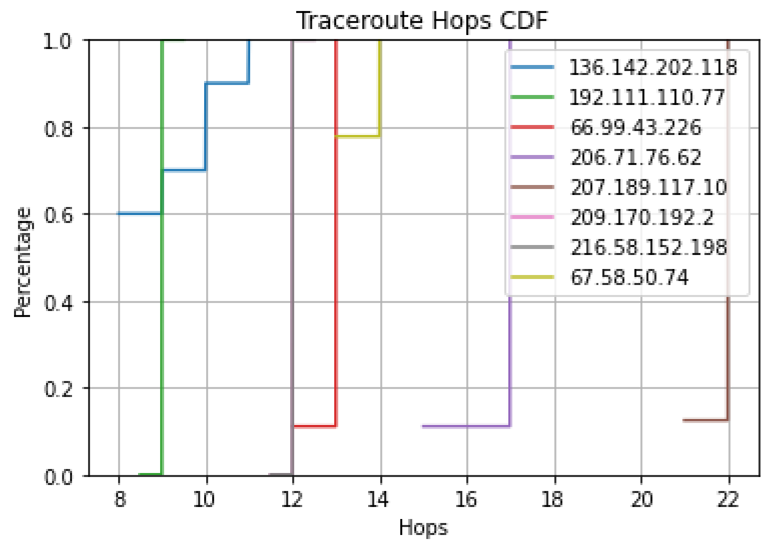
- Tests sent from probes within North America tend to have significantly lower RTT rates.
- Tests across all probes tend to be within 14 - 25 hops.
- Tests sent from probes outside North America tend to have RTT rates higher than 160 ms.

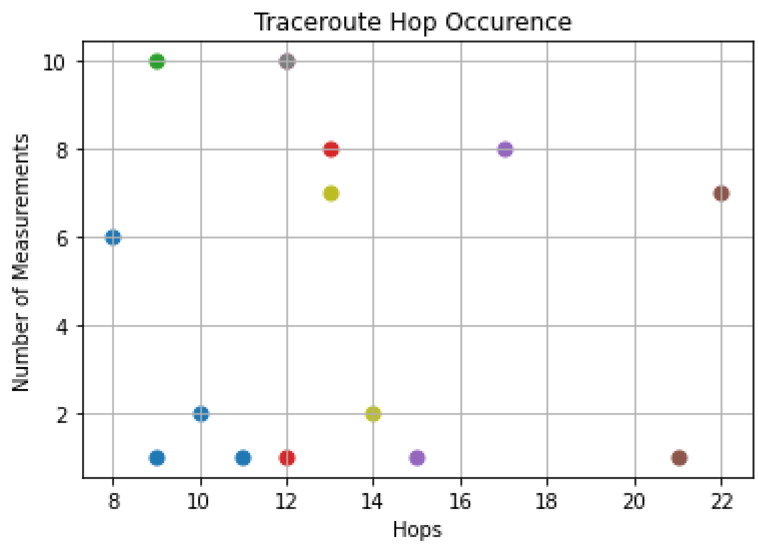
Ping -

- Tests sent from probes within North America tend to have low RTT rates (US probes tend around 34 ms).
- Tests sent from probes outside North America tend to have RTT rates between 120 ms and 250 ms.
- There has been very little packet loss over the past year of measurement.
- There is a high concentration of RTT values near 150-250 ms.

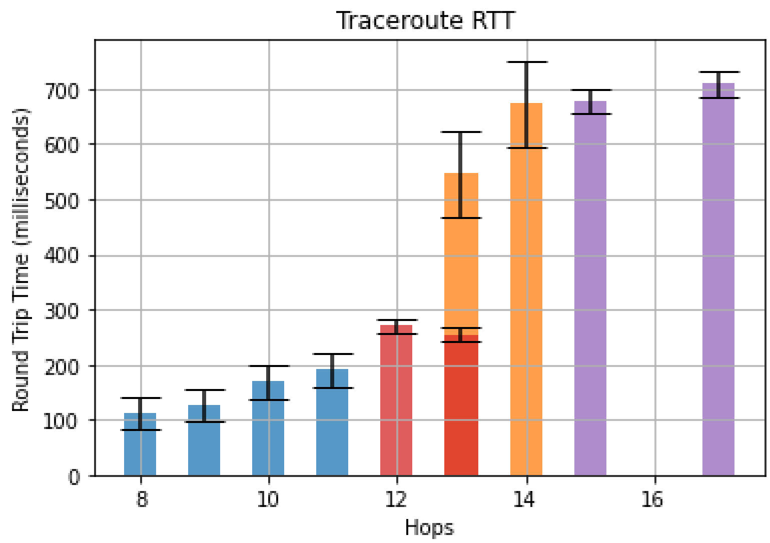


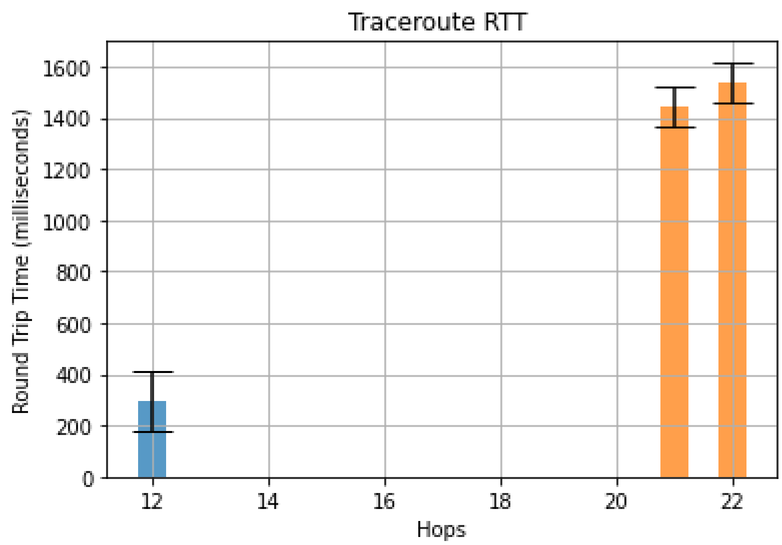
Leo Data (perfsonar):

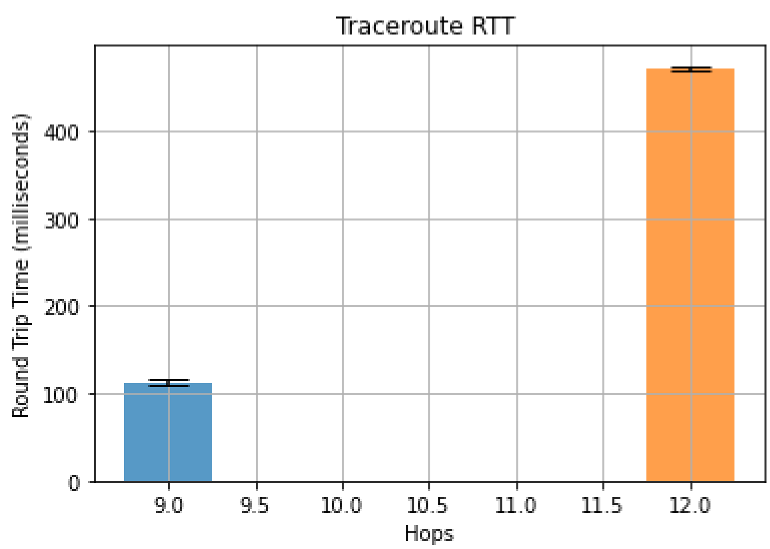




- Based on data collected over a 10 hour period, there is a fairly consistent network path amongst all 9 IP addresses; there is occasional variation. Address 136.142.202.118 has the most frequent variation in hops.
- There is a wide range of hops amongst all 9 IP addresses as a whole.

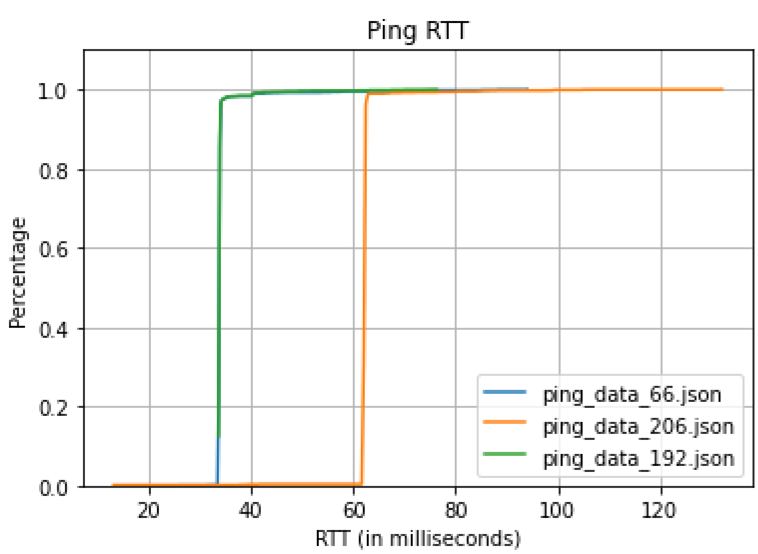

Traceroute RTT
Round Trip Time (milliseconds)
0
100
200
300
400
500
600
700
8
10
12
14
16
Hops


Traceroute RTT
Round Trip Time (milliseconds)
0
200
400
600
800
1000
1200
1400
1600
12
14
16
18
20
22
Hops


Traceroute RTT
Round Trip Time (milliseconds)
0
100
200
300
400
9.0
9.5
10.0
10.5
11.0
11.5
12.0
Hops

- These three graphs represent the rtt/hops (w/ stddev) of 9 different IP addresses (separated into 3 graphs for simpler visual analysis).
- In the first graph, the blue address has paths ranging from 8-11 hops and average RTTs ranging 100-200 ms with ± 25 ms variation; the red address tends around 12-13 hops with an average RTT near 250 ms on both with little variation; the orange address has a path of 13-14 hops and average RTTs of 550 and 650 ms with ±50 ms variation on each; the purple address ranges 15-17 hops and averages near 700 ms RTT with low variation.
- In the second graph, the blue address has a consistent network path of 12 hops and averages 300 ms RTT with a variation of ± 100 ms; the orange address tends around 21-22 hops and has average RTTs nearing 1500ms with variations of ± 100 ms.
- In the third graph, the blue address has a consistent network path of 9 hops and averages 110 ms RTT with little variation; the orange address has a consistent network path of 12 hops and averages nearly 500 ms RTT with little variation.
- The orange address from graph two is significantly farther away compared to the other addresses.
- The network appears to be stable and consistent based on these graphs.

Leo Data (basic ping recent):



- Over 3 days, all three addresses maintain relatively constant RTT with few outliers. Two of the addresses overlap on the graph and have similar RTTs.

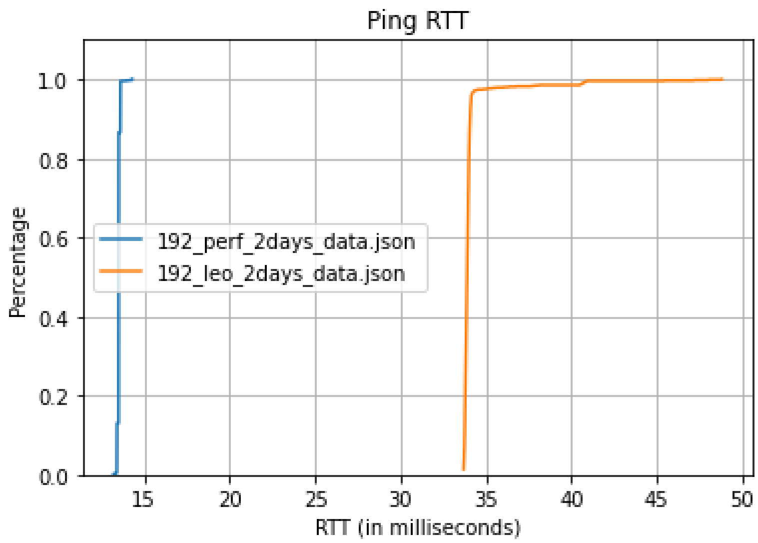
Leo Data (basic ping last semester):

date-time	destination	rtt_avg	rtt_min	rtt_max	rtt_mdev	packet_loss_rate
Oct 20 9:17 PM	141.216.99.254	49.421	46.98	119.035	6.953	0.0
Oct 24 11:15 AM	141.216.99.254	53.975	35.926	162.583	18.867	0.0
Oct 27 8:00 PM	141.216.99.254	50.489	47.004	129.827	12.625	0.0
Oct 20 9:17 PM	206.71.76.62	66.307	62.336	121.674	8.842	0.0
Oct 24 11:15 AM	206.71.76.62	66.269	62.277	144.226	12.865	0.0
Oct 27 8:00 PM	206.71.76.62	66.649	62.217	170.149	14.92	0.0
Oct 20 9:17 PM	207.189.117.10	84.417	81.708	170.999	8.08	0.0
Oct 24 11:15 AM	207.189.117.10	86.523	81.602	180.884	15.322	0.0
Oct 27 8:00 PM	207.189.117.10	86.178	81.667	252.275	17.266	0.0
Oct 20 9:17 PM	209.170.192.2	62.771	60.476	91.82	4.825	0.0
Oct 24 11:15 AM	209.170.192.2	63.171	60.519	122.536	7.621	0.0
Oct 27 8:00 PM	209.170.192.2	64.78	58.603	165.595	15.839	0.0
Oct 20 9:17 PM	216.58.152.198	34.902	30.194	131.227	13.453	0.0
Oct 24 11:15 AM	216.58.152.198	34.257	30.304	99.387	10.922	0.0
Oct 27 8:00 PM	216.58.152.198	34.828	28.086	124.701	15.628	0.0

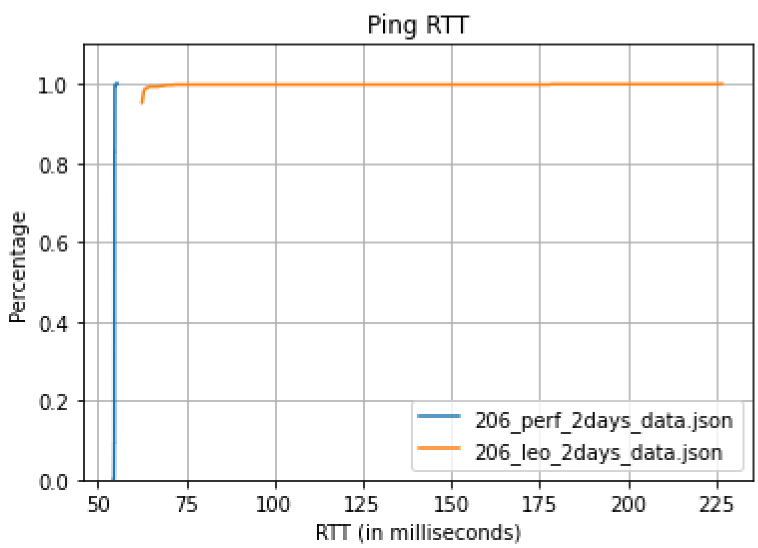
- There is a consistent rate of 0% packet loss; all test addresses have consistent average rtts over the test period.

Leo vs Perfsonar1:

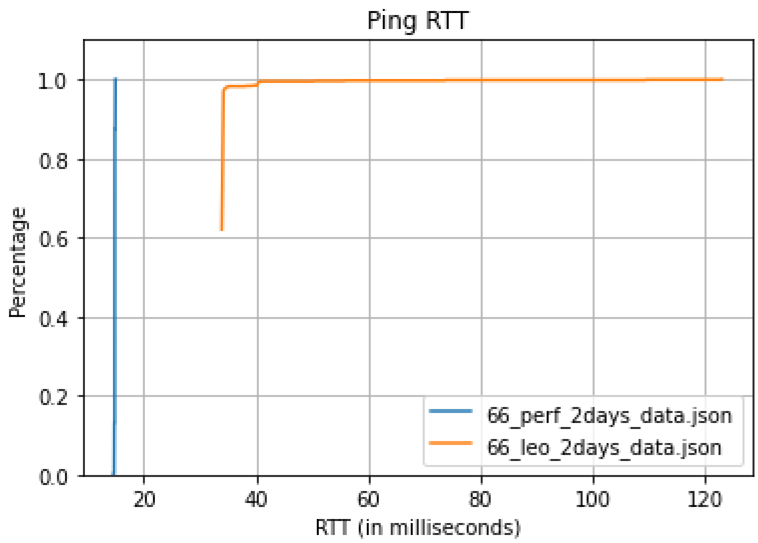
- 0% packet loss on all three IPs on both machines



- Both machines have fairly consistent RTTs, Leo has greater outliers than perfsonar1. Perfsonar1 RTT tends to be half the size as Leo's for this IP



- Both machines have fairly consistent RTTs, Leo has greater outliers than perfsonar1. Perfsonar1 RTT tends to be lower than Leo's.



- Both machines have fairly consistent RTTs, Leo has greater outliers than perfsonar1. Perfsonar1 RTT tends to be half the size as Leo's for this IP

For: June 12 - 18 LATEST UPDATES		*Last updated: June 20, 2023*
Status	Description	Comments
Last completed *Due date:*	32. Scheduled RIPE Atlas measurements to Perfsonar1 33. Added timestamps to CLI ping/traceroute scripts 34. Moved first CLI measurement logs to github 35. Started CLI measurement period for complete list of IPs 36. Worked on RIPE Atlas packet loss graph script	
Working on currently/next	Currently working on : running ping/traceroute on Leo and perfsonar1; experimenting with cloud host; moving all working graph scripts into a singular file; IP/port table Next to work on: finish testing tools; debug Ripe Atlas graph scripts; iperf3 tests Next milestones: graphing data Problems: rtt tasks are not working on Leo machine, but trace tasks are All measurements are failing to archive on Leo due to server errors. None of the pscheduler tasks are running on Leo. There are quite a few probes on RIPE Atlas that are unreachable to perfsonar1 but return good results for Leo, not sure if this is a firewall issue. This is causing significant packet loss on some measurements.	

Plan & Timeline

June 12 - 18	run ping/traceroute on leo and perfsonar1, replicate RIPE Atlas measurements on perfsonar1
June 19 - 25	Move ping/traceroute logs to github; update graph scripts
June	TBD

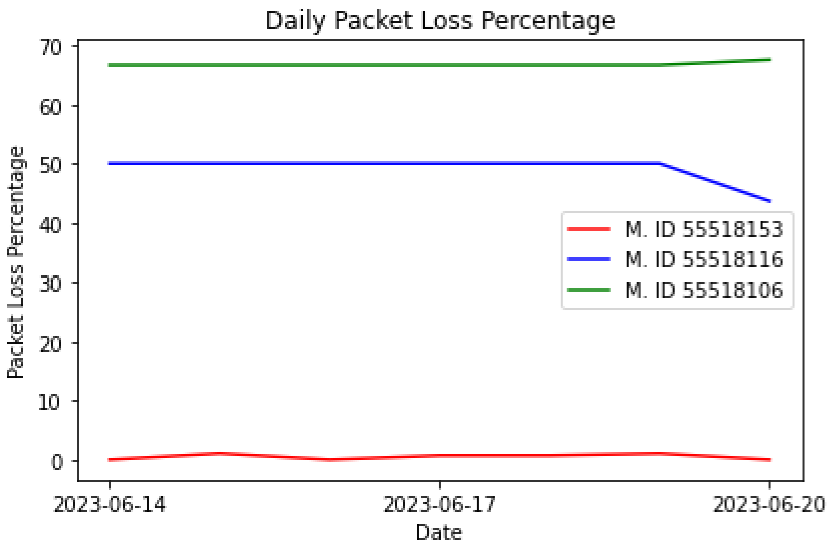
CLI IP Input for Leo
66.133.111.78 66.133.96.78 68.142.131.6 136.142.202.118 192.111.110.77 66.99.43.226 206.71.76.62 207.189.117.10 209.170.192.2 216.58.152.198 67.58.50.74 149.149.248.20

CLI IP Input for PF1
66.133.111.78 66.133.96.78 68.142.131.6 136.142.202.118 192.111.110.77 66.99.43.226 206.71.76.62 207.189.117.10 209.170.192.2 216.58.152.198 67.58.50.74 149.149.2.70

CLI IP Input for DTN1

66.133.111.78 66.133.96.78 68.142.131.6 136.142.202.118 192.111.110.77 66.99.43.226 206.71.76.62 207.189.117.10 209.170.192.2 216.58.152.198 67.58.50.74 149.149.248.20 149.149.248.53

Packet Loss:

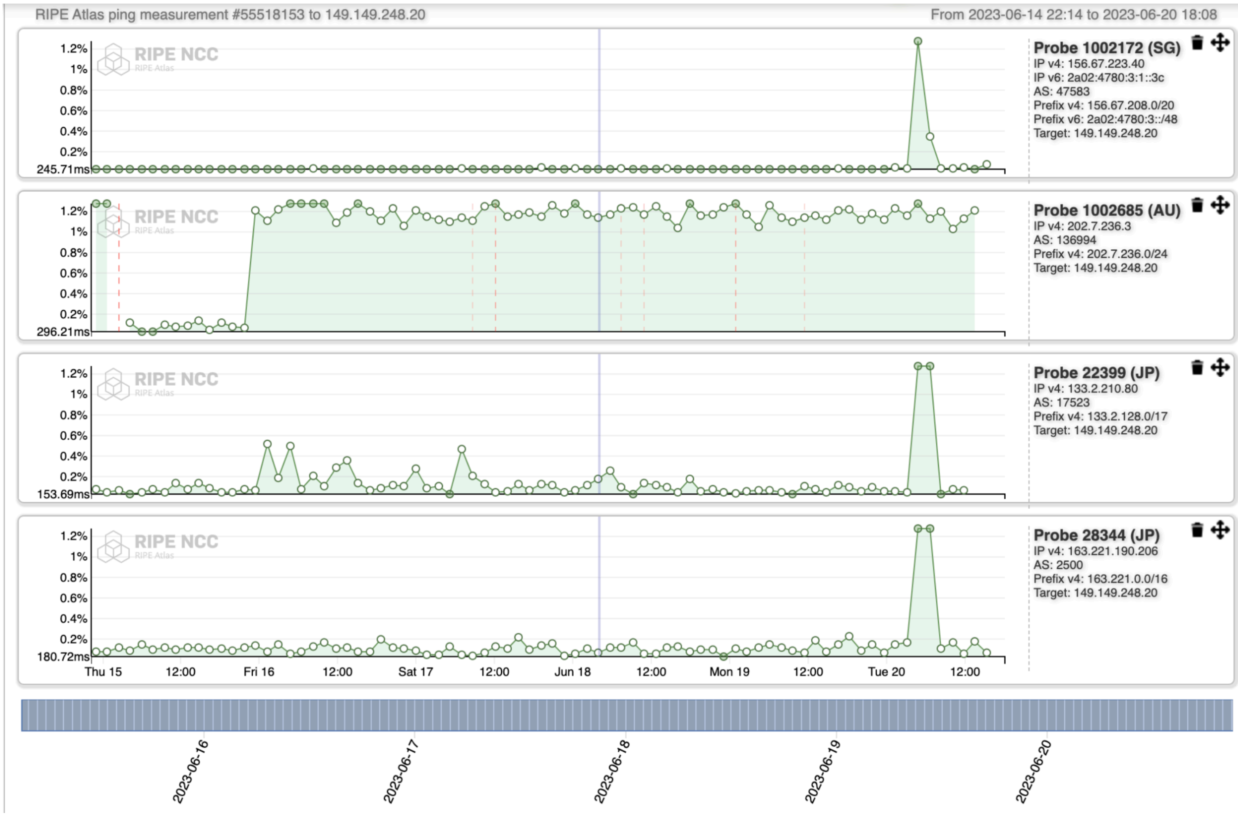


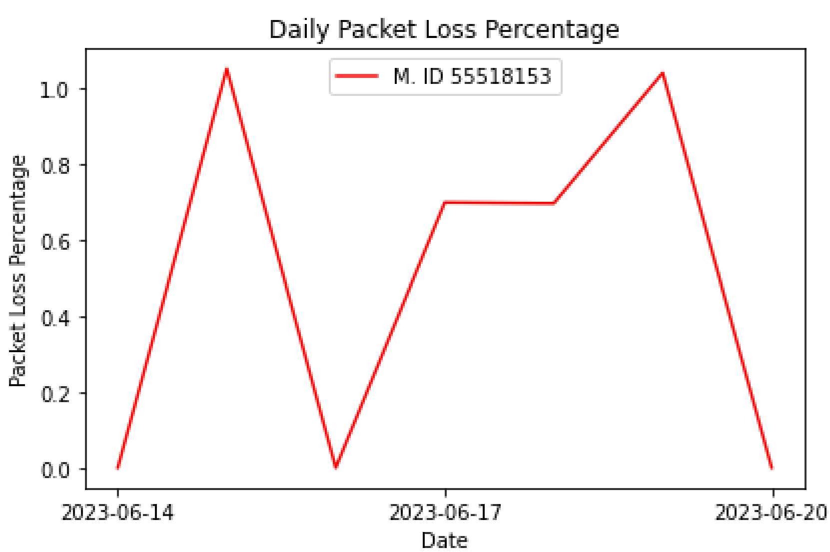
Measurement #55518153 to Perfsonar1 Node (below):

LatencyMON Data from RIPE Atlas Interface compared to Packet Loss Graph calculated in Python Script - at a glance the packet loss data appears to be accurately represented by the Python Graph script

Red Dotted Lines on LatencyMON snippet represent packet loss, by the faintness of the lines it can be assumed that packet loss exists but is low at points of the measurement. Packet loss is between 0.0 and 1.0% on the Packet Loss Graph; it appears to be correct.

The graph currently displays the percentage of packet loss per day so that it daily trends can be observed and graph data can be more easily compared to LatencyMON, but it can be changed to CDF if need be.





Leo IP/port Table

149.149.2.70

IP Address	Ping	Traceroute	RIPE Ping	RIPE Traceroute
133.2.210.80			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
163.221.190.206			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
156.67.223.40			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
202.7.250.126			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
212.164.54.90			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
185.54.40.8			-Not Currently Working -Traffic Inbound to Leo -Port ??	-Not Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
212.193.78.192			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
2.78.43.132			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
197.149.149.17			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
71.255.254.218			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
24.109.199.146			-Currently Working -Traffic Inbound to	-Currently Working -Traffic Inbound to

			Leo -Port ??	Leo -Port 80 (ICMP)
71.168.145.110			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
34.136.31.64			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
213.165.161.164			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
84.192.164.13			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
213.166.194.116			-Currently Working -Traffic Inbound to Leo -Port ??	-Currently Working -Traffic Inbound to Leo -Port 80 (ICMP)
66.133.111.78	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
66.133.96.78	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
68.142.131.6	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
136.142.202.118	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
192.111.110.77	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
66.99.43.226	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
206.71.76.62	-Currently Working -Traffic Outbound	-Currently Working -Traffic Outbound		

	From Leo -Port ?? (ICMP)	From Leo -Port ?? (ICMP)		
207.189.117.10	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
209.170.192.2	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
216.58.152.198	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
67.58.50.74	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		
149.149.248.20	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)	-Currently Working -Traffic Outbound From Leo -Port ?? (ICMP)		

Perfsonar1 IP/port Table

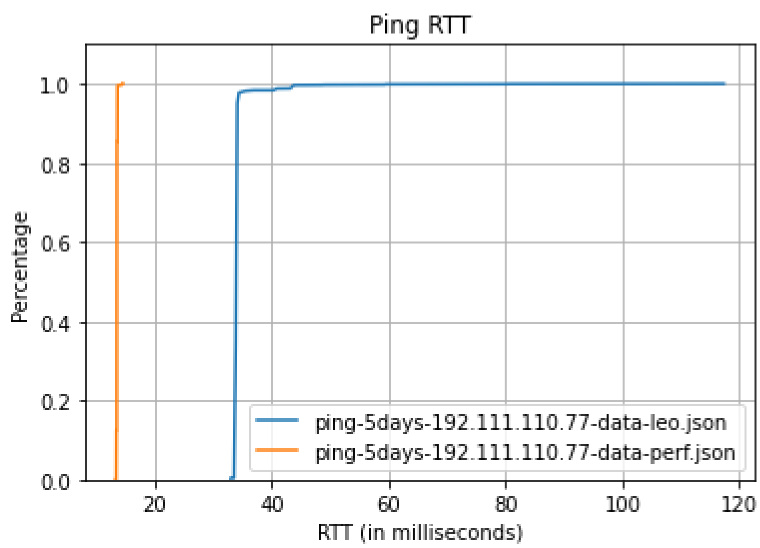
149.149.248.20

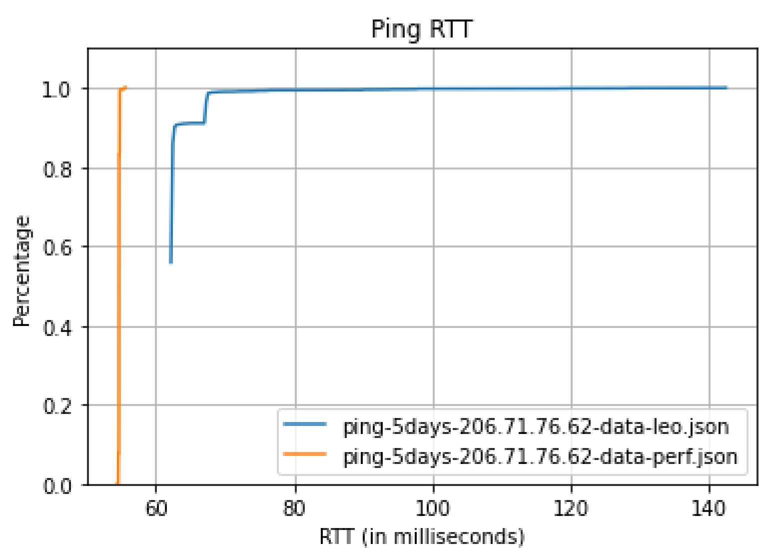
IP Address	Ping	Traceroute	RIPE Ping	RIPE Traceroute
133.2.210.80			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
163.221.190.206			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
156.67.223.40			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
202.7.250.126			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
212.164.54.90			-Not Currently	-Not Currently

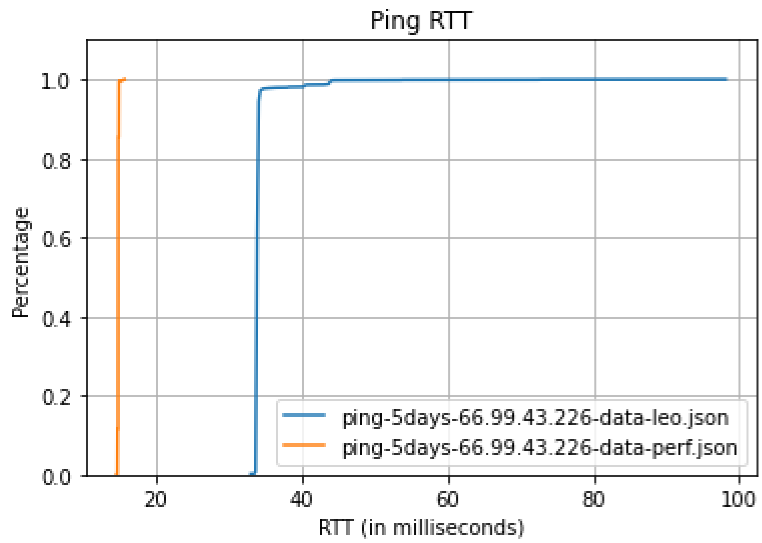
			Working -Traffic Inbound to perfsonar1 -Port ??	Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
185.54.40.8			-Not Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Not Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
212.193.78.192			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
2.78.43.132			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
197.149.149.17			-Not Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Not Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
71.255.254.218			-Not Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Not Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
24.109.199.146			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
71.168.145.110			-Not Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Not Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
34.136.31.64			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
213.165.161.164			-Not Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Not Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
84.192.164.13			-Not Currently Working -Traffic Inbound to perfsonar1	-Not Currently Working -Traffic Inbound to perfsonar1

			-Port ??	-Port 80 (ICMP)
213.166.194.116			-Currently Working -Traffic Inbound to perfsonar1 -Port ??	-Currently Working -Traffic Inbound to perfsonar1 -Port 80 (ICMP)
66.133.111.78	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
66.133.96.78	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
68.142.131.6	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
136.142.202.118	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
192.111.110.77	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
66.99.43.226	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
206.71.76.62	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
207.189.117.10	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
209.170.192.2	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
216.58.152.198	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Currently Working -Traffic Outbound from perfsonar1 -Port ??		
67.58.50.74	-Currently Working -Traffic Outbound from perfsonar1	-Currently Working -Traffic Outbound from perfsonar1		

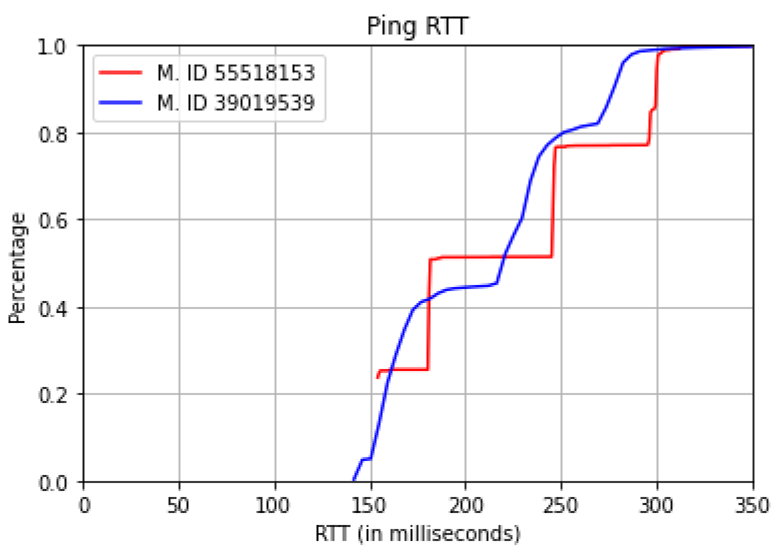
	-Port ?? (ICMP)	-Port ??		
149.149.2.70	-Currently Working -Traffic Outbound from perfsonar1 -Port ?? (ICMP)	-Not Currently Working -Traffic Outbound from perfsonar1 -Port ??		

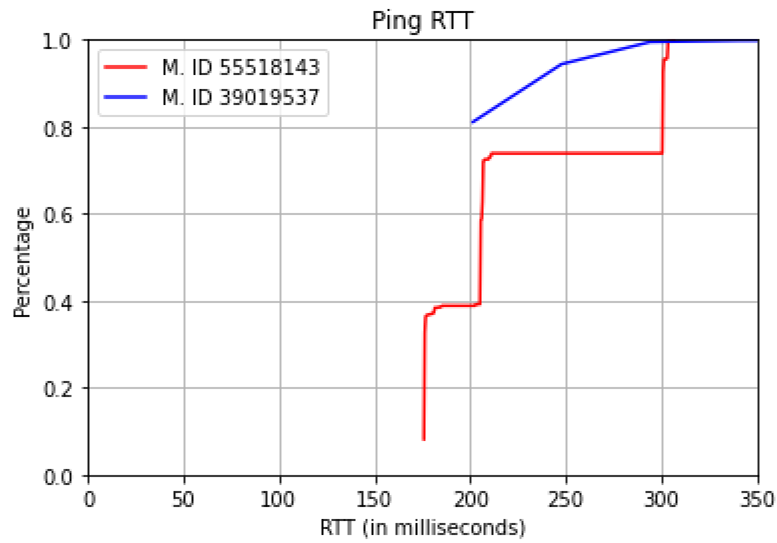


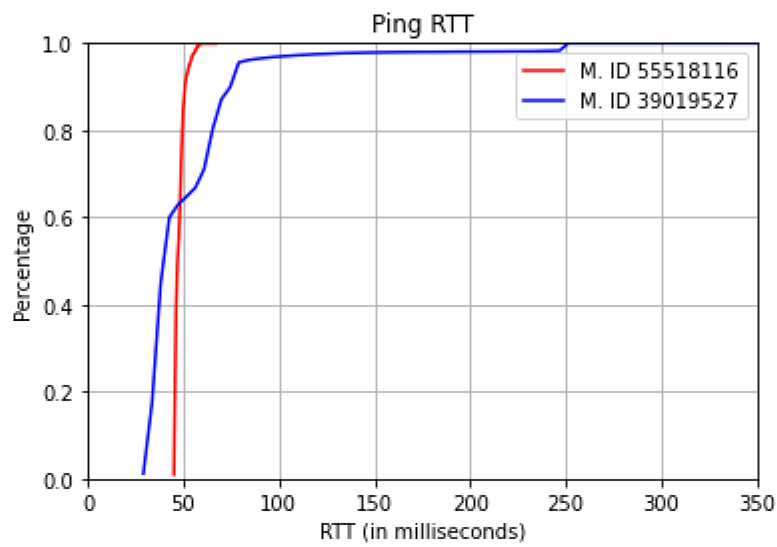


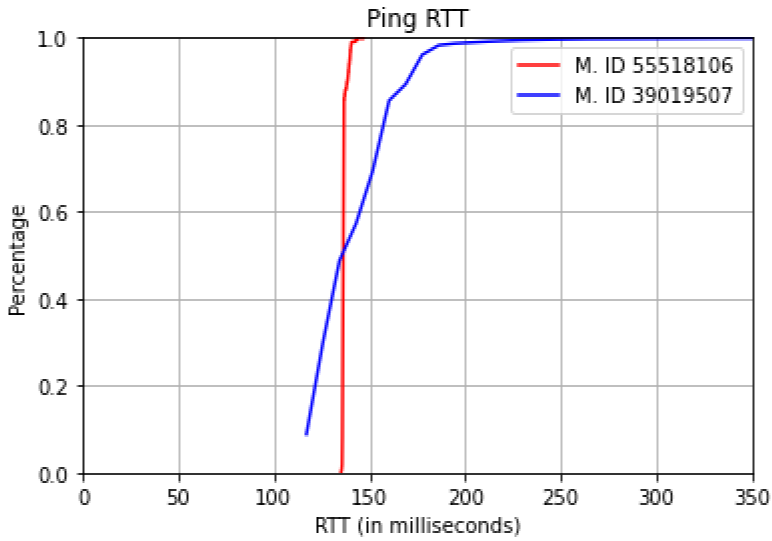


RIPE Atlas (Leo : prefix 5 ; Perfsonar1 : prefix 3)




Ping RTT
M. ID 55518143
M. ID 39019537
Percentage
RTT (in milliseconds)
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
350


Ping RTT
M. ID 55518116
M. ID 39019527
Percentage
RTT (in milliseconds)
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
350

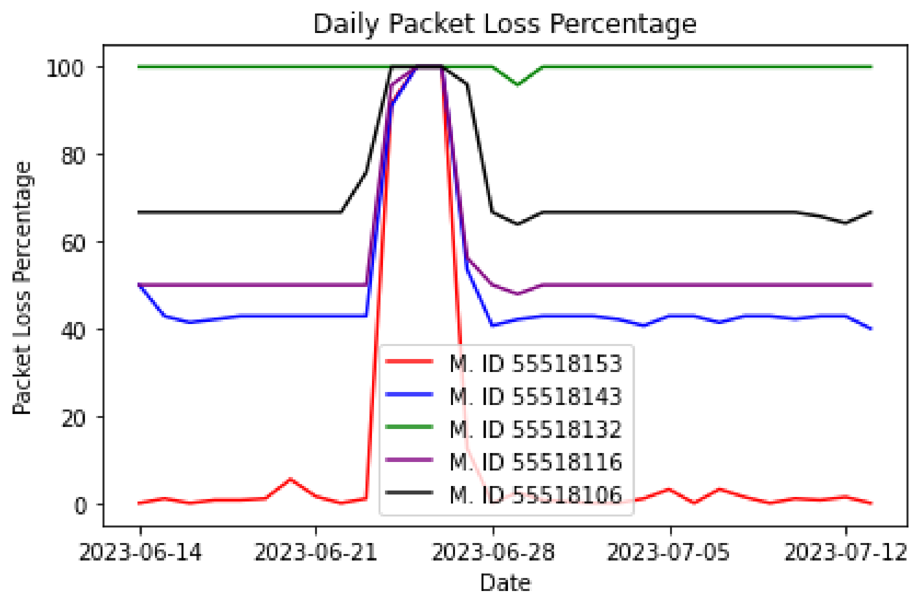

Ping RTT
M. ID 55518106
M. ID 39019507
Percentage
RTT (in milliseconds)
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
350

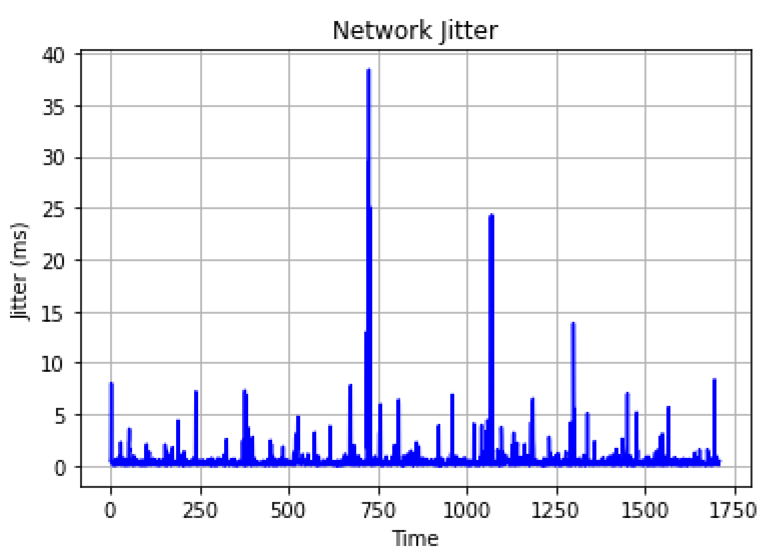
For: July 10 - 17 LATEST UPDATES		*Last updated: July 12, 2023*
Status	Description	Comments
Last completed *Due date:*	37. Working jitter graph for RIPE Atlas data 38. JSON conversion files updated with timestamp	
Working on currently/next	Currently working on : running ping/traceroute on Leo and perfsonar1; experimenting with cloud host; moving all working graph scripts into a singular file; jitter graph for command-line data Next to work on: finish testing tools; debug Ripe Atlas graph scripts; iperf3 tests Next milestones: graphing data Problems: rtt tasks are not working on Leo machine, but trace tasks are All measurements are failing to archive on Leo due to server errors. None of the pscheduler tasks are running on Leo. There are quite a few probes on RIPE Atlas that are unreachable to perfsonar1 but return good results for Leo, not sure if this is a firewall issue. This is causing significant packet loss on some measurements. Cannot log in to Leo, so no measurements are being run aside from RIPE Atlas	

Plan & Timeline

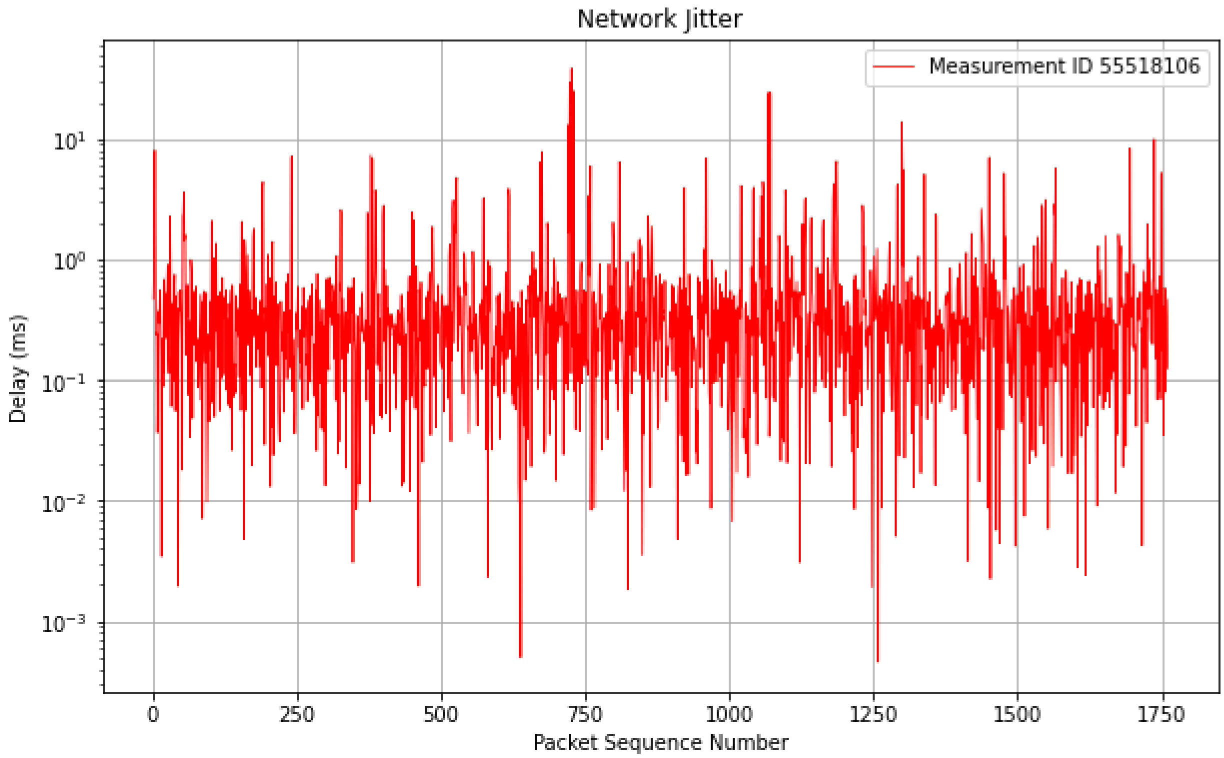
July 10 -16	Graph scripts for jitter
July 17 -	TBD
June	TBD

Most recent packet-loss graph



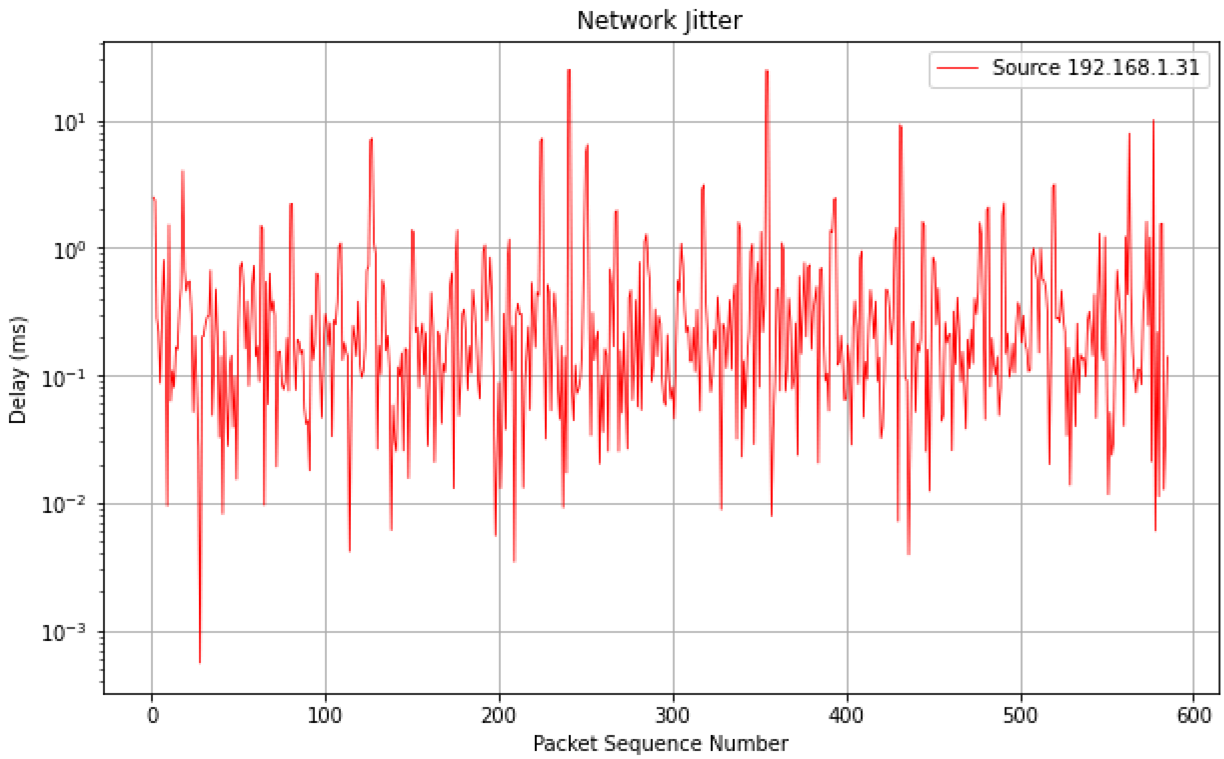


Start to jitter graph, x-axis needs to be fixed to show dates. RIPE Atlas data.



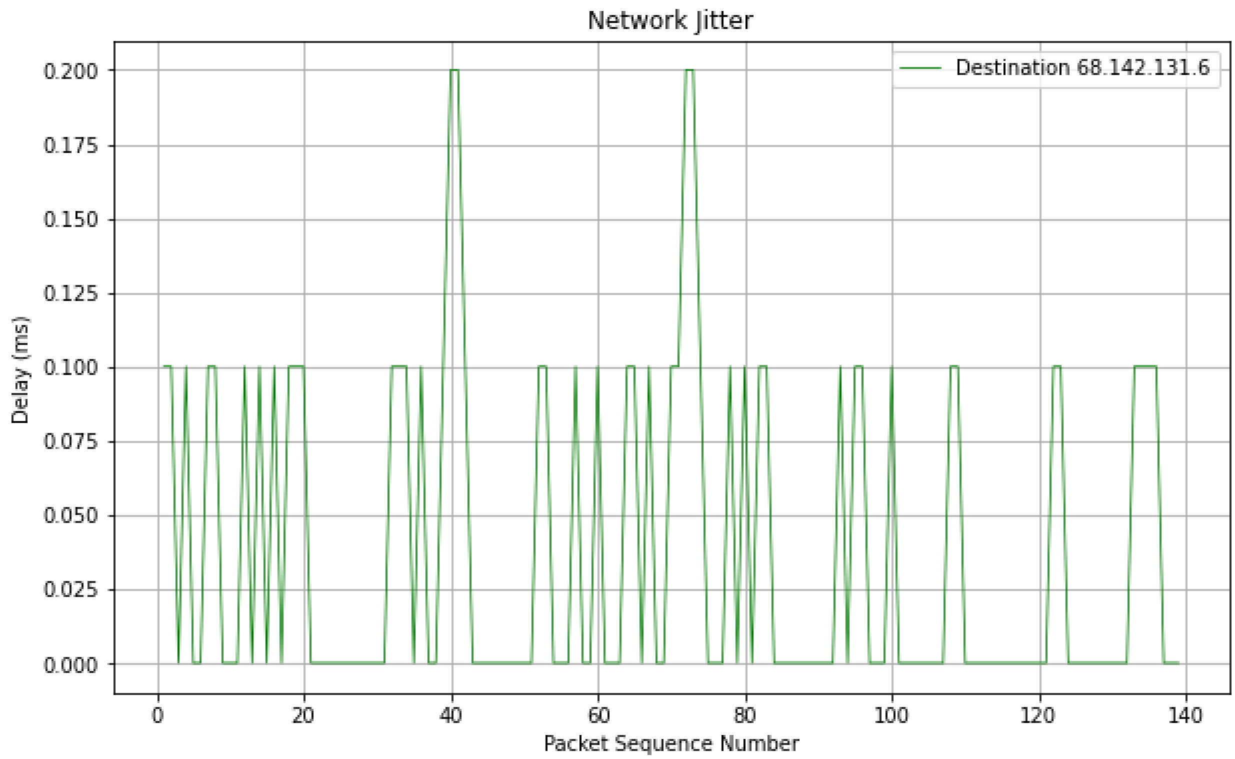
Jitter graph with same data set, but the y-axis is on a logarithmic scale; decided to keep the x-axis as it was but change the label. This may show the data better

These graphs are using the RIPE Atlas data as a whole set, but the data should probably be separated out by source address so that there is not misconceptions.



This graph shows jitter on the first source address of the data set, there is a bug causing every source to appear in a separate graph, but it may be more readable this way.

When graphing the RIPE Atlas measurements that are run to Leo, there are more unique source addresses than there are probes used, the IP address on the probes may be changing. I may change the graph to separate by probe instead of source address.



Jitter graph using Command-line ping data from perfsonar1, not using logarithmic y-scale on this one because there are no large delay values to make the graph hard to read with a normal y-scale.

For: July 24 - 30 LATEST UPDATES		*Last updated: Aug 2, 2023*
Status	Description	Comments
Last completed *Due date:*	39. Measurements running on leo and perfsonar1 40. Compressed all functional graph scripts into a single file 41. Fixed broken graph script 42. Reworked json conversion file to include packets sent/received and rtt statistics 43. Graph script for packet loss in Leo/Perfsonar1 measurements	
Working on currently/next	Currently working on : running ping/traceroute on Leo and perfsonar1; experimenting with cloud host; figure out how to simplify the process of moving data from leo and perfsonar1 to local machine to graph Next to work on: finish testing tools; iperf3 tests Next milestones: Problems: There are quite a few probes on RIPE Atlas that are unreachable to perfsonar1 but return good results for Leo, not sure if this is a firewall issue. This is causing significant packet loss on some measurements.	

Plan & Timeline

July 24 - 30	Get measurements running again on leo and perfsonar1
July 30 - Aug 6	Work on broken graph scripts
Aug 7 - 13	TBD

Ideas for simplifying data transfer for graphing:

- SCP (Secure Copy): SCP is a secure and efficient way to transfer files between a local machine and remote servers. We can use SCP in the command-line measurement scripts to copy the data from Leo and Perfsonar1 to the local machine automatically.
- SFTP (Secure File Transfer Protocol): Similar to SCP, SFTP provides secure file transfer between local and remote machines. We can use SFTP in the scripts to automate the transfer of measurement data.

- rsync: Rsync is a powerful tool that allows incremental file transfers, which means it can efficiently sync only the changes made in files between the source and destination. It can be helpful for syncing measurement data between Leo and Perfsonar1 and the local machine.

Graphs Scripts that We Have:

- Ping RTT CDF graph for command-line ping (outdated, used for sample data from last year and likely will not be used for the paper)
    - RTT Percentages
- Ping RTT CDF graph for RIPE Atlas Data
    - RTT Percentages
- RIPE Atlas Packet Loss Percentage Graph (not currently in CDF format but could be changed)
    - Percentage Packet Loss Per Day
- RIPE Atlas Traceroute Hops CDF
    - Hop Count Percentages
- RIPE Atlas Traceroute Hops Bar Graph
    - Number of Measurements with specific Hop Counts
- RIPE Atlas Traceroute RTT and Traceroute Hops Graphs
    - Bar Graph Average RTT for a Hop Count + Standard Deviation
    - Number of Measurements with Specific Hop Counts
- Ping RTT CDF command-line (Leo specific)
    - RTT Percentages
- Command-Line Ping RTT CDF (works for Perfsonar1 and Leo data)
    - RTT Percentages
- Ping RTT CDF and normal graph Perfsonar (perfsonar software not machine)
    - Occurrences of specific RTTs
    - RTT Percentages
- Traceroute Hops Bar and Scatter Plot Perfsonar (perfsonar software not machine)
    - Average RTT of Hop Counts (with Standard Deviation)
    - Number of Measurements with Specific Hop Counts
- Traceroute Hops CDF Perfsonar (perfsonar software not machine)
    - Hop Count Percentages
- Command-line Traceroute CDF and Scatter Plot (Perfsonar1 and Leo)
    - Hop Count Percentages
    - Number of Measurements with Specific Hop Counts
- RIPE Atlas Jitter Graph
    - Delay (difference) between sequential packets
- Command-line Jitter Graph (Perfsonar1 and Leo)
    - Delay (difference) between sequential packets

For: August 7-13 LATEST UPDATES		*Last updated: Aug 14, 2023*
Status	Description	Comments
Last completed *Due date:*	44. Tested that GCP could ping Perfsonar1 45. Checked access to DTN and checked ping/traceroute 46. Started ping measurements on DTN1 47. Started traceroute measurements on DTN1 48. Rescheduled RIPE measurements to DTN 49. Started CL ping/trace measurements from GCP to Leo/Perf1/DTN1	
Working on currently/next	Currently working on : experimenting with cloud host; figure out how to simplify the process of moving data from leo and perfsonar1 to local machine to graph; graph comparison models Next to work on: finish testing tools; iperf3 tests; packet loss graph for CLI Next milestones: Problems: There are quite a few probes on RIPE Atlas that are unreachable to perfsonar1 but return good results for Leo, not sure if this is a firewall issue. This is causing significant packet loss on some measurements. Iperf3 does not work with all measurement routes	

Plan & Timeline

Aug 7 - 13	Organize tasks for paper, check machine/tool access
Aug 14 - 20	Start LAN1 and LAN2 Measurements, experiment with iperf3, start RIPE-DTN measurements, start DTN traceroute measurements
Aug 21 - 27	Add GCP destination to ping/trace measurements, start RIPE measurements for DTN

RIPE Atlas data comparison from Aug 23, 2023 - now

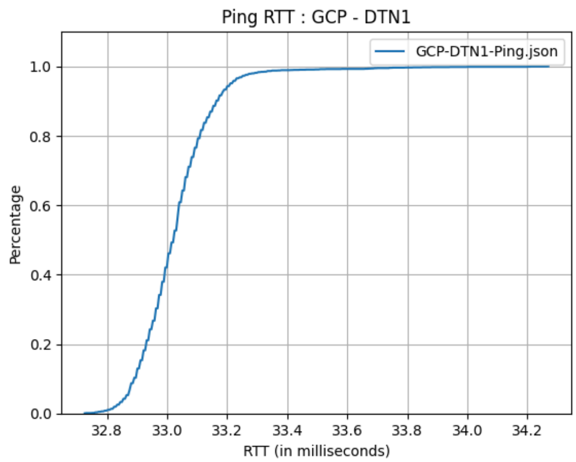
For: August 28- September 3 LATEST UPDATES *Last updated: Sept 6, 2023*		
Status	Description	Comments
Last completed *Due date:* NEW	50. Updated CLI loss graph 51. Traceroute RTT graph now working	
Working on currently/next	Currently working on : experimenting with cloud host; figure out how to simplify the process of moving data from leo and perfsonar1 to local machine to graph; graph comparison models; convert all graphs to use timestamp restrictions to observe data from common time ranges Next to work on: finish testing tools; iperf3 tests Next milestones: Problems: There are quite a few probes on RIPE Atlas that are unreachable to perfsonar1 but return good results for Leo, not sure if this is a firewall issue. This is causing significant packet loss on some measurements. Iperf3 does not work with all measurement routes	

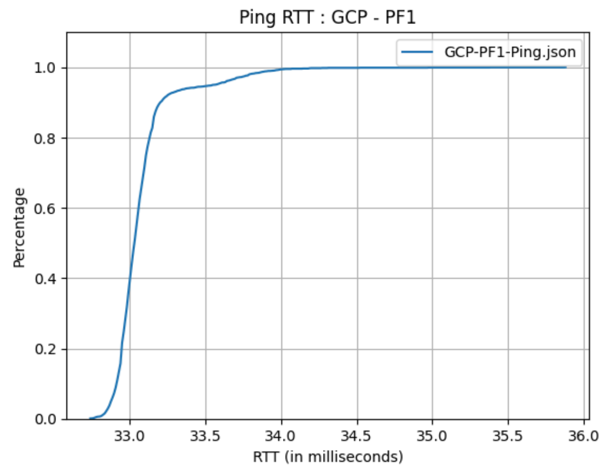
Plan & Timeline

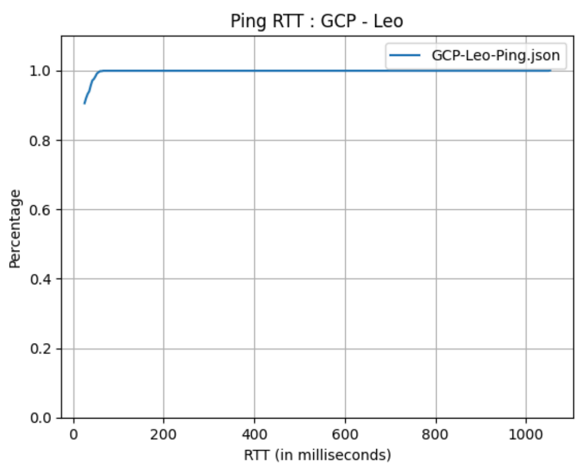
Aug 28 - Sep 3	Compare current datasets
Sep 4 - Sept 10	Convert remaining graphs to use seaborn, add avg and std dev to jitter graphs
Sep 11 - 18	Restart Leo measurements, correct broken graphs

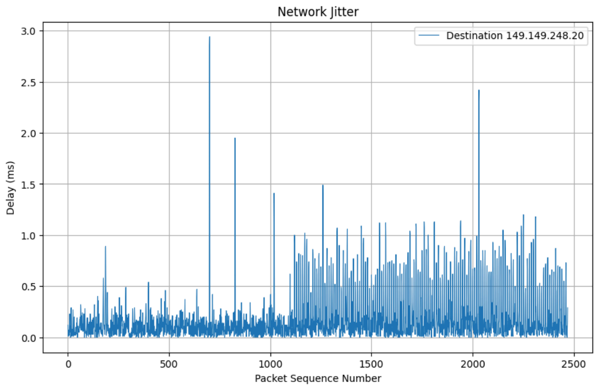
WAN Routes

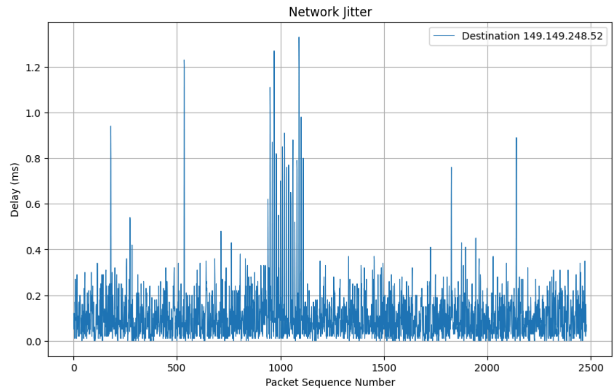
GCP

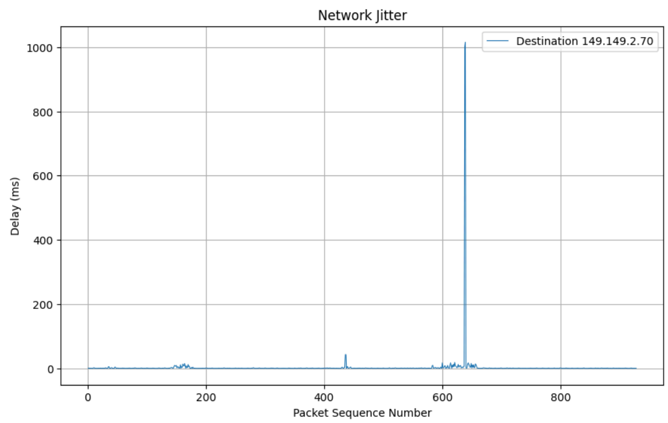


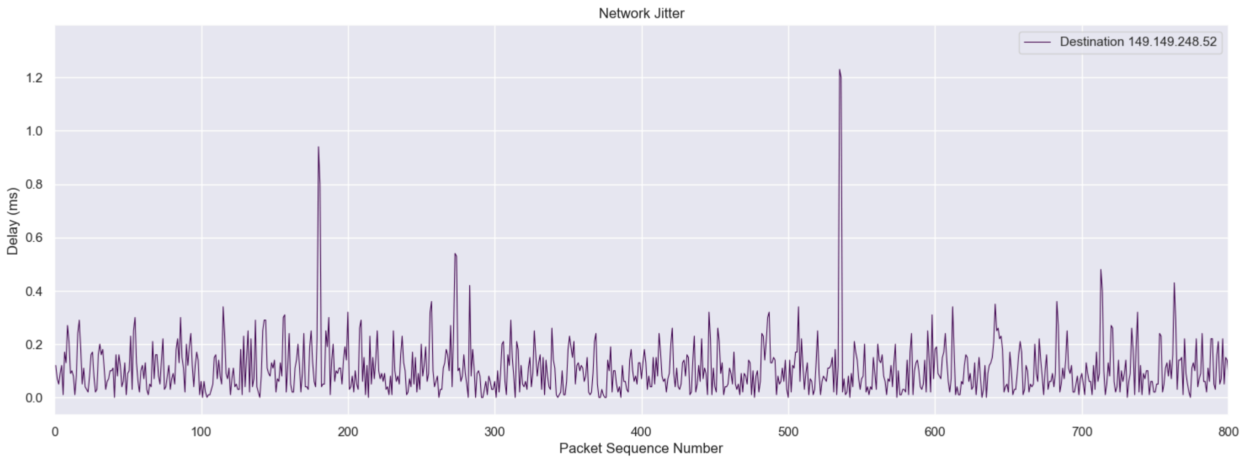


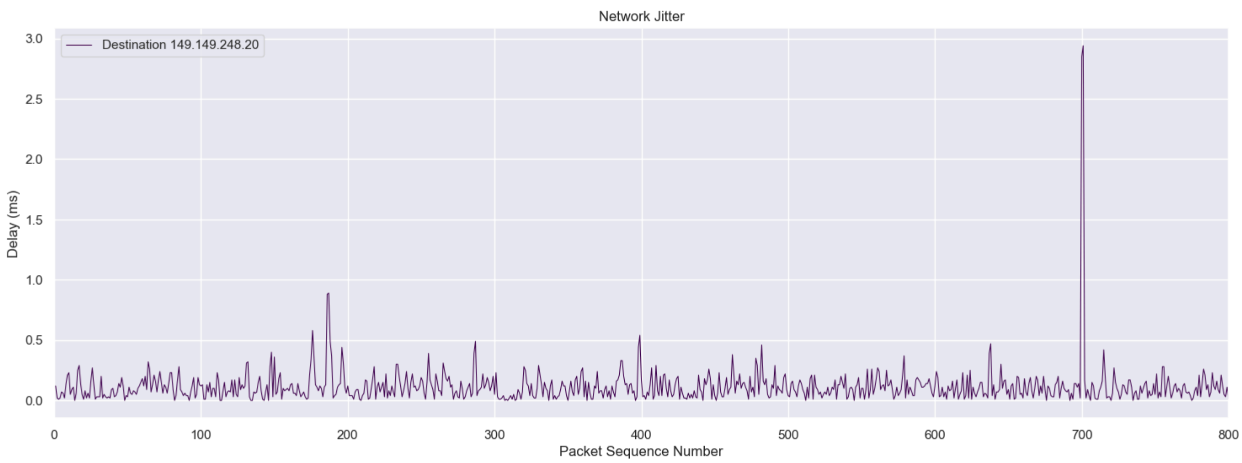


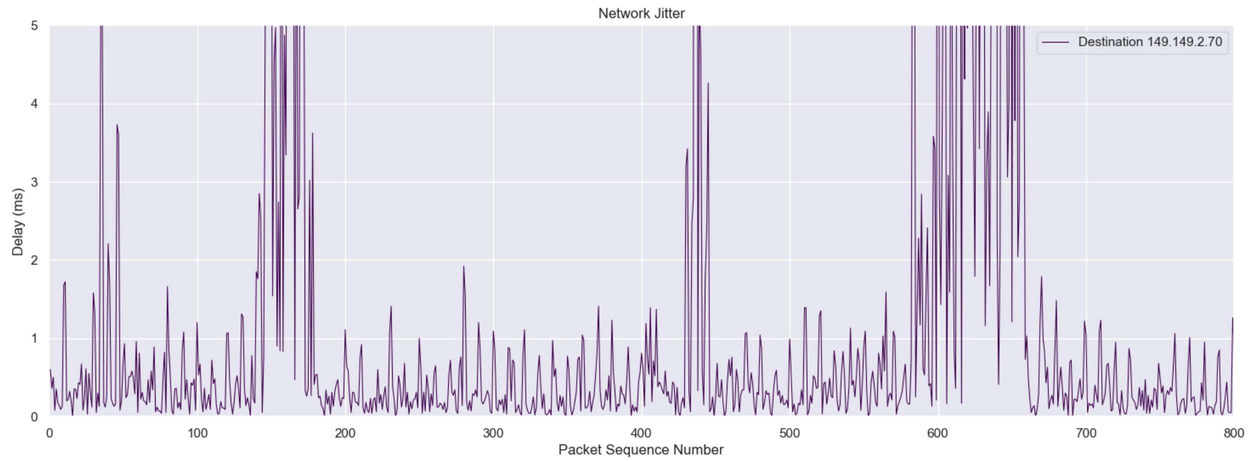

Network Jitter
Destination 149.149.248.20
Delay (ms)
Packet Sequence Number
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
500
1000
1500
2000
2500

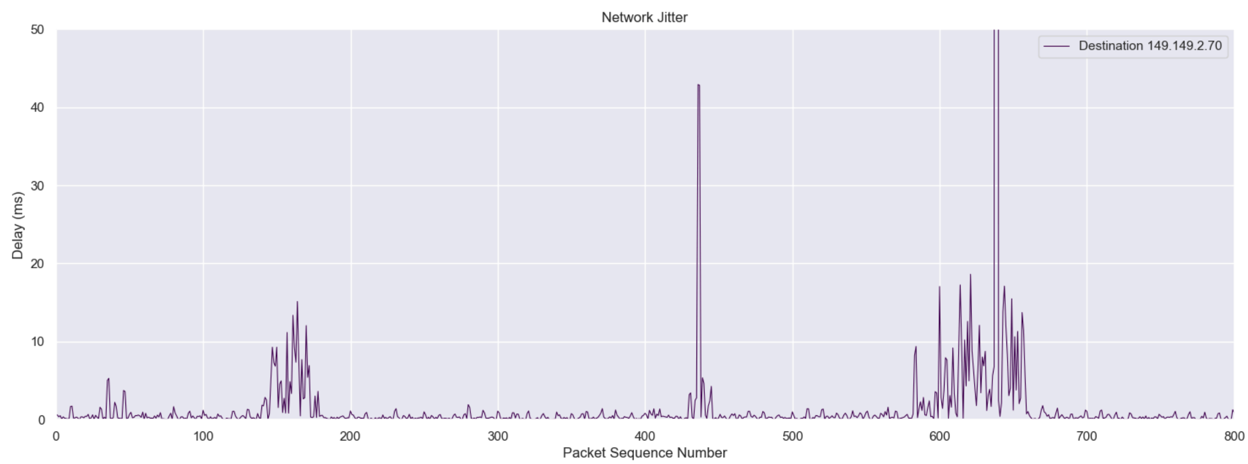

Network Jitter
Destination 149.149.248.52
Delay (ms)
Packet Sequence Number
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
500
1000
1500
2000
2500


Network Jitter
Destination 149.149.2.70
Delay (ms)
Packet Sequence Number
0
200
400
600
800
1000
0
200
400
600
800

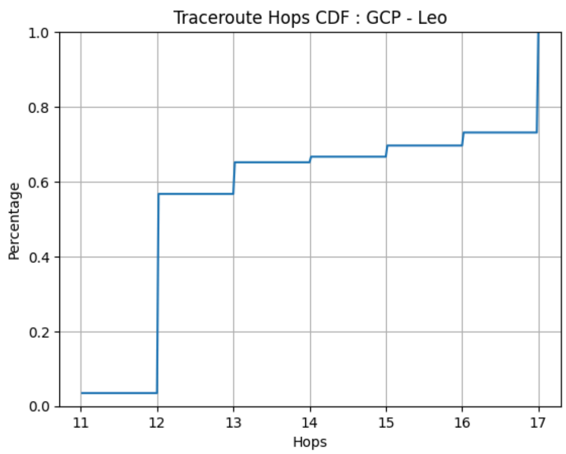

Network Jitter
Destination 149.149.248.52
Delay (ms)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
100
200
300
400
500
600
700
800
Packet Sequence Number


Network Jitter
Destination 149.149.248.20
Delay (ms)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
100
200
300
400
500
600
700
800
Packet Sequence Number

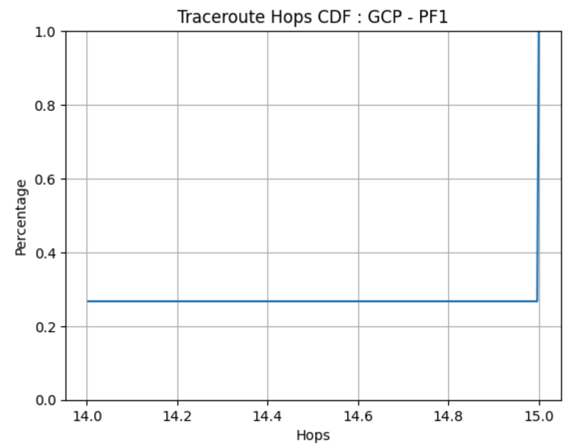




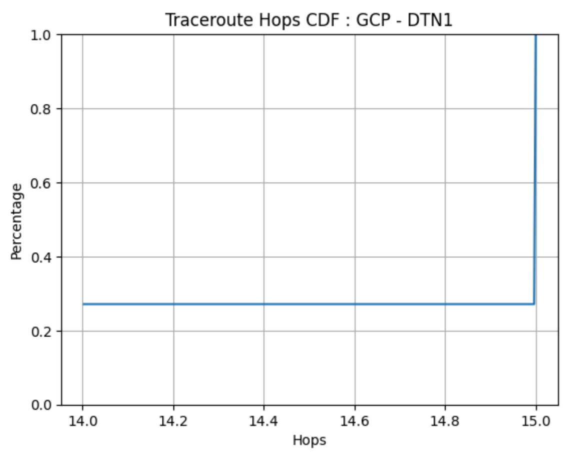
*Leo has high levels of jitter and two major outliers or ~1000 ms and ~50 ms delay. There are two graphs displayed, one cut to 50 ms delay and one cut to 5 ms delay to more accurately depict the jitter in comparison to PF1 and DTN1. Leo also has significantly less data points than PF1 and DTN1 so the x-axis has been limited to 800 on all graphs for accurate comparison



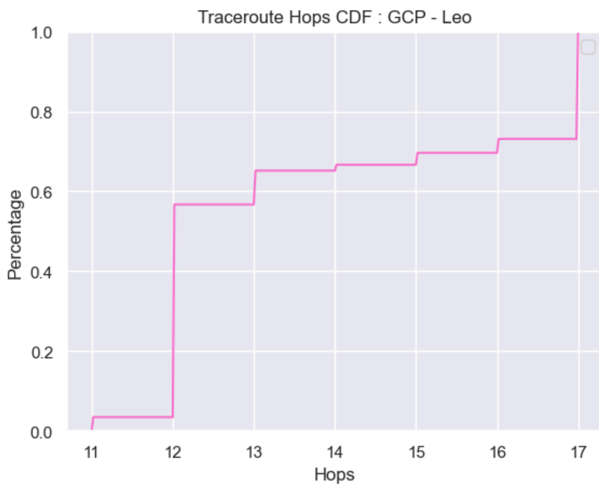
Traceroute Hops CDF : GCP - Leo
Percentage
Hops

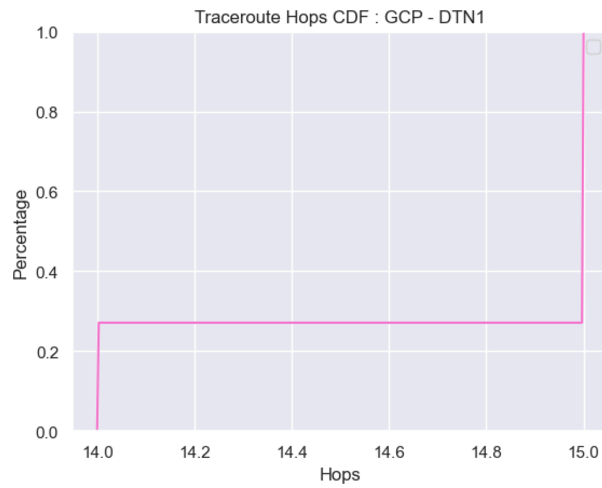


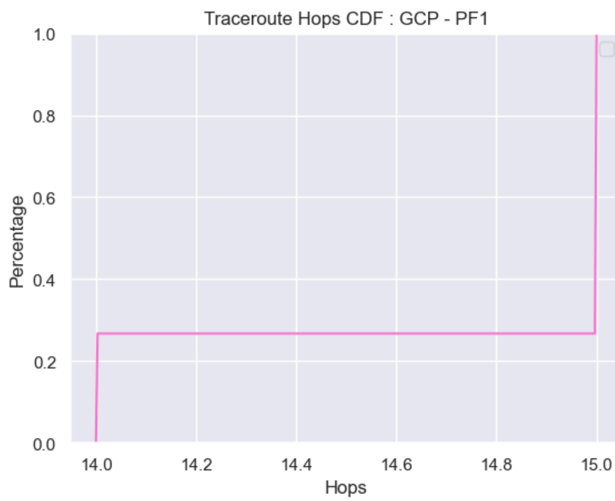
Traceroute Hops CDF : GCP - PF1
Percentage
Hops

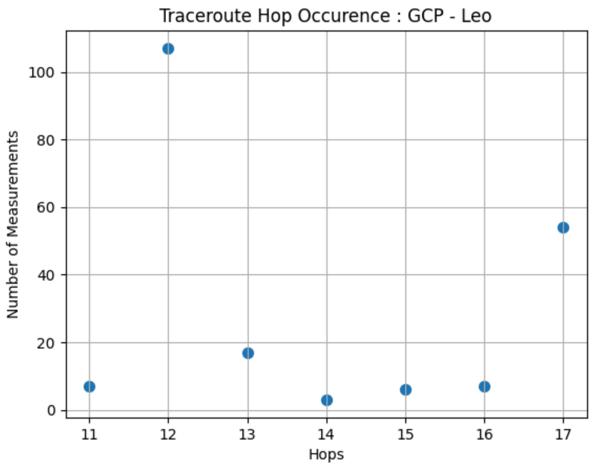


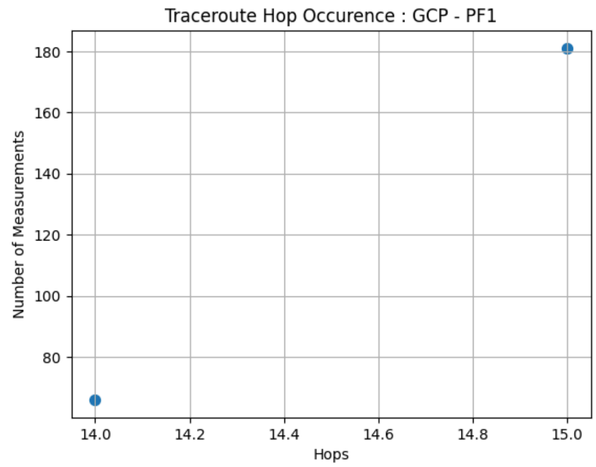
Traceroute Hops CDF : GCP - DTN1
Percentage
Hops

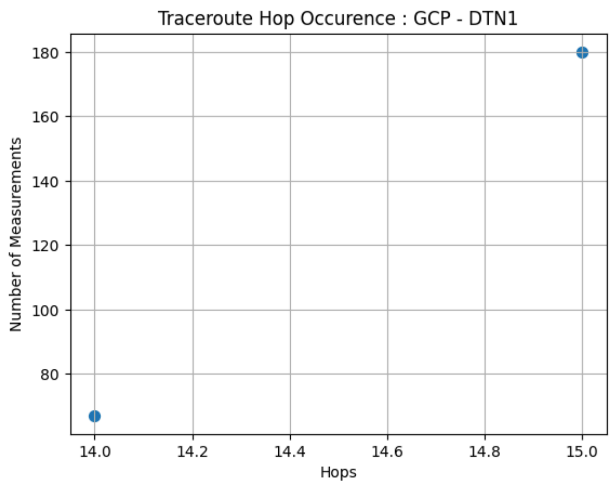

Traceroute Hops CDF : GCP - Leo
Percentage
Hops
1.0
0.8
0.6
0.4
0.2
0.0
11
12
13
14
15
16
17

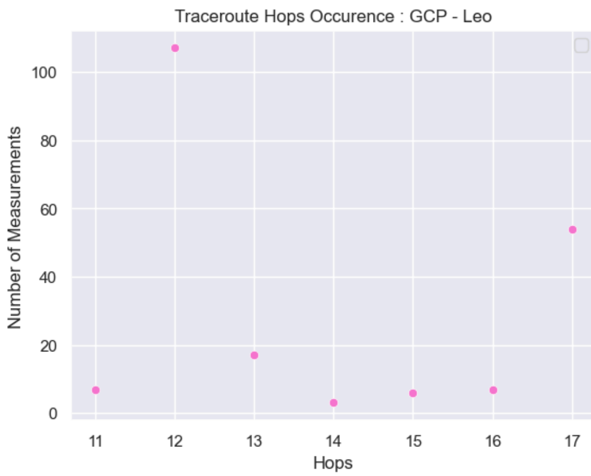

Traceroute Hops CDF : GCP - DTN1
Percentage
Hops
1.0
0.8
0.6
0.4
0.2
0.0
14.0
14.2
14.4
14.6
14.8
15.0

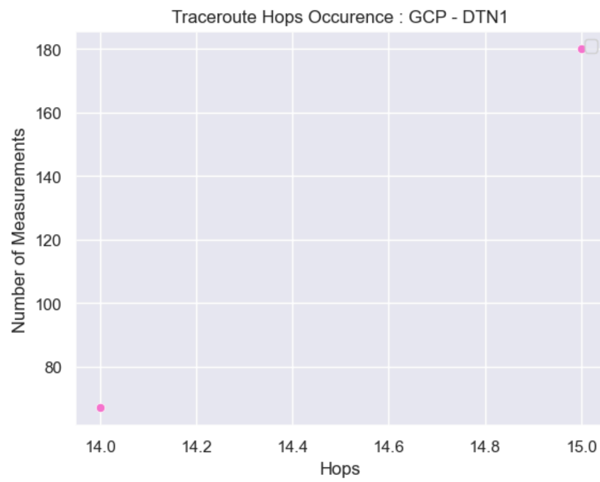

Traceroute Hops CDF : GCP - PF1
Percentage
Hops
1.0
0.8
0.6
0.4
0.2
0.0
14.0
14.2
14.4
14.6
14.8
15.0

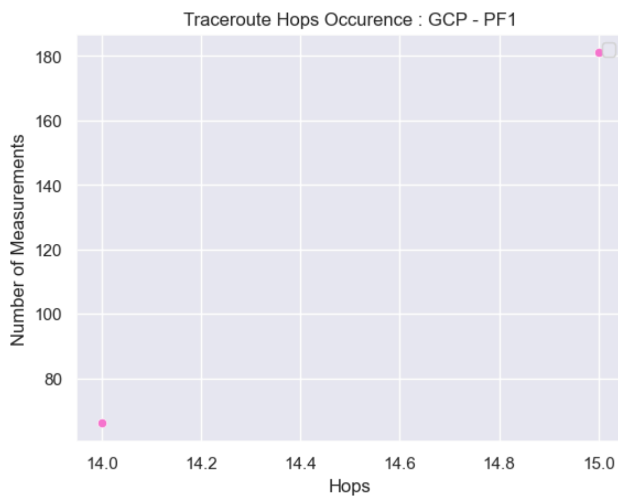

Traceroute Hop Occurence : GCP - Leo
Number of Measurements
Hops
0
20
40
60
80
100
11
12
13
14
15
16
17


Traceroute Hop Occurence : GCP - PF1
Number of Measurements
Hops
80
100
120
140
160
180
14.0
14.2
14.4
14.6
14.8
15.0

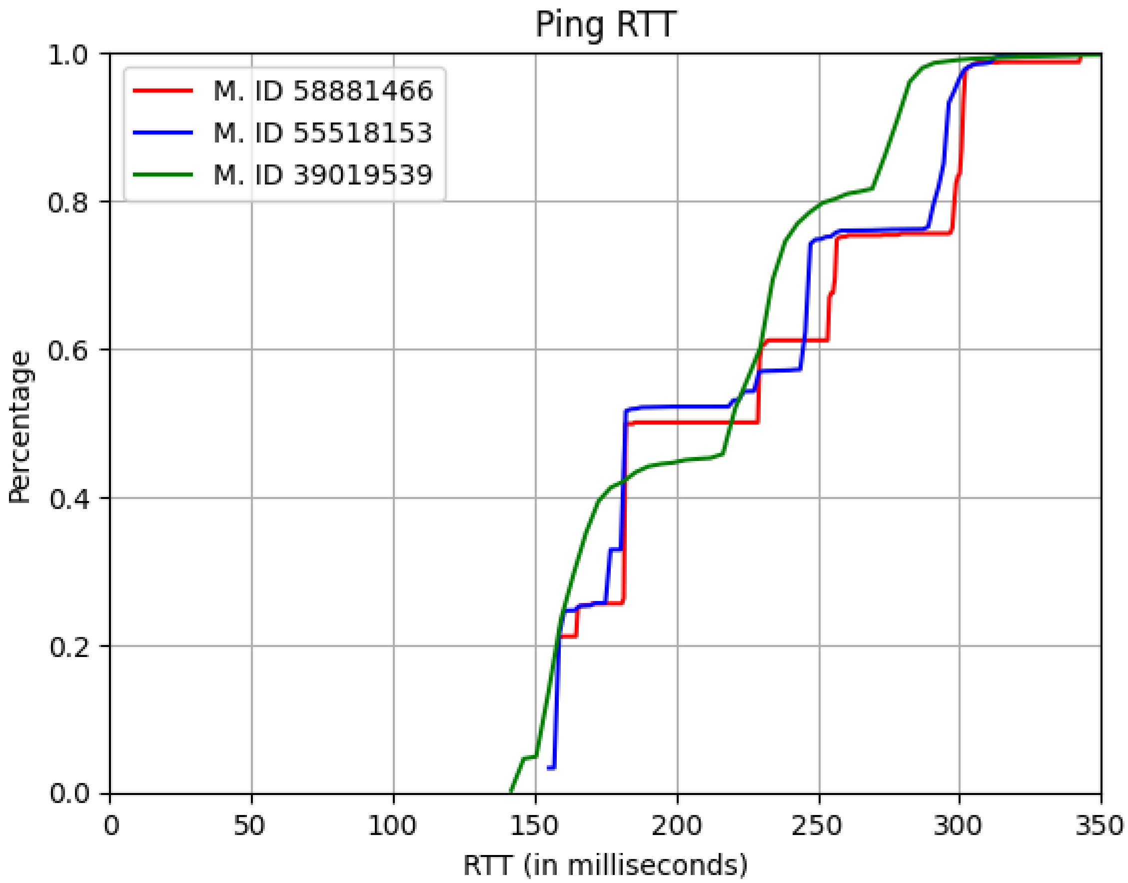

Traceroute Hop Occurence : GCP - DTN1
Number of Measurements
Hops
80
100
120
140
160
180
14.0
14.2
14.4
14.6
14.8
15.0


Traceroute Hops Occurence : GCP - Leo
Number of Measurements
Hops
0
20
40
60
80
100
11
12
13
14
15
16
17


Traceroute Hops Occurence : GCP - DTN1
Number of Measurements
Hops
80
100
120
140
160
180
14.0
14.2
14.4
14.6
14.8
15.0

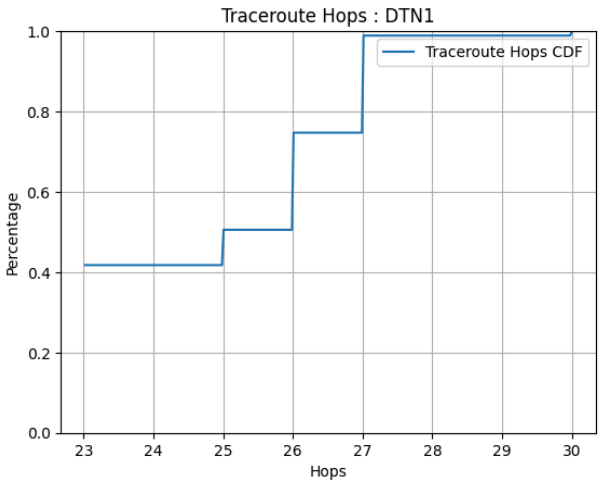

Traceroute Hops Occurence : GCP - PF1
Number of Measurements
Hops
80
100
120
140
160
180
14.0
14.2
14.4
14.6
14.8
15.0

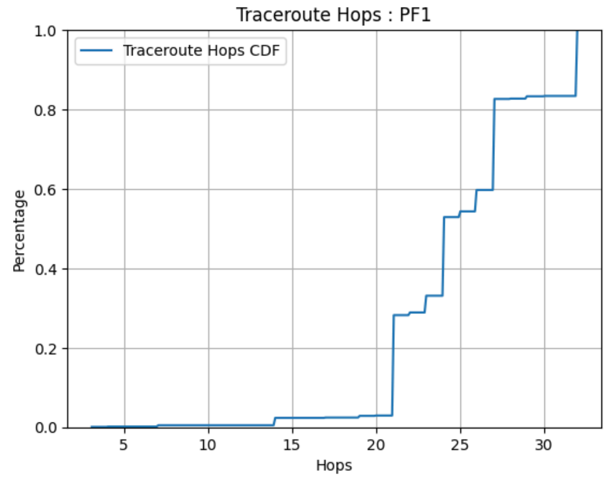
RIPE

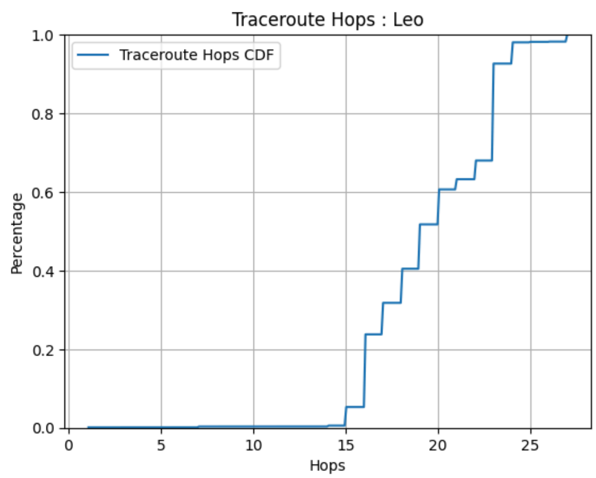


Red (DTN1), Blue (PF1), Green (Leo)

Plotted RIPE measurements that use the most consistently functioning probes


Traceroute Hops : DTN1
Traceroute Hops CDF
Percentage
Hops


Traceroute Hops : PF1
Traceroute Hops CDF
Percentage
Hops

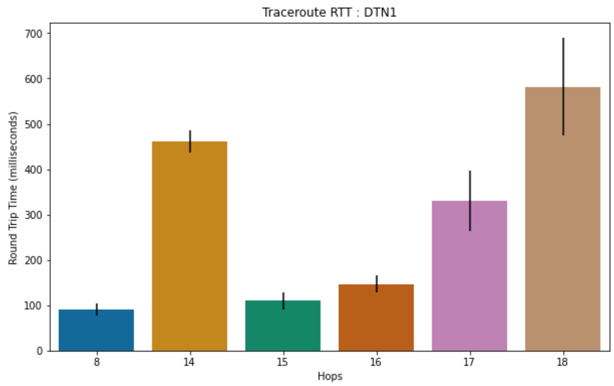

Traceroute Hops : Leo
Traceroute Hops CDF
Percentage
Hops

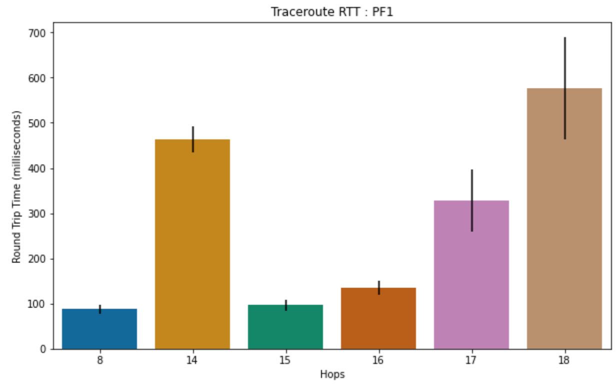
For: September 18 - September 24 LATEST UPDATES *Last updated: Oct 3, 2023*		
Status	Description	Comments
Last completed *Due date:*	52. avg and stddev on jitter graphs 53. Selected urls for data transfer experiment 54. made diagram to show WAN/LAN routes 55. Created a script to periodically download data and capture packets on Leo 56. Data Transfer Experiment running on DTN 57. Restarted GCP measurements	
Working on currently/next	Currently working on : experimenting with cloud host; graph comparison models; convert all graphs to use timestamp restrictions to observe data from common time ranges Next to work on: finish testing tools; iperf3 tests Next milestones: Problems: iperf3 does not work with all measurement routes: (sever-client) "Leo - PF1" and "DTN1 - GCP" still do not work correctly	

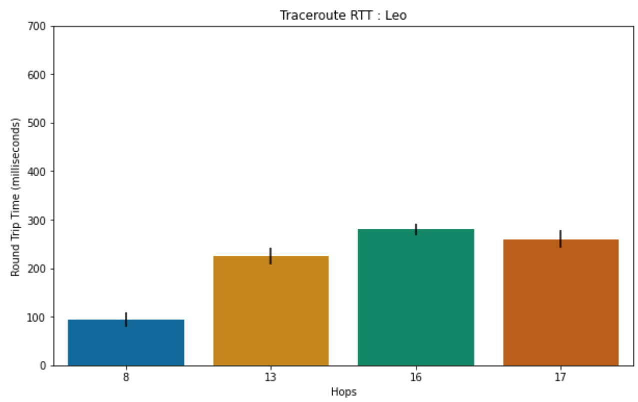
Plan & Timeline

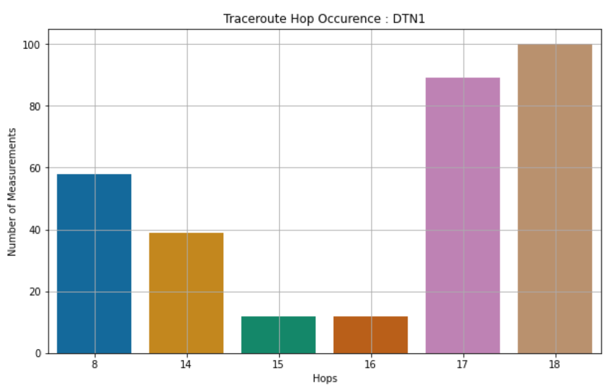
Sept 18 - 24	Get all graphs working; add figures to Overleaf; gather sources for data transfer experiments
Sep 25 - Oct 1	add figures to Overleaf; start on a script to download and gather packets for data transfer experiment; think about a script to run iperf3 experiments; make WAN/LAN diagram
Oct 2 - 8	Get data transfer experiments running regularly; add figures to Overleaf; restart GCP experiments

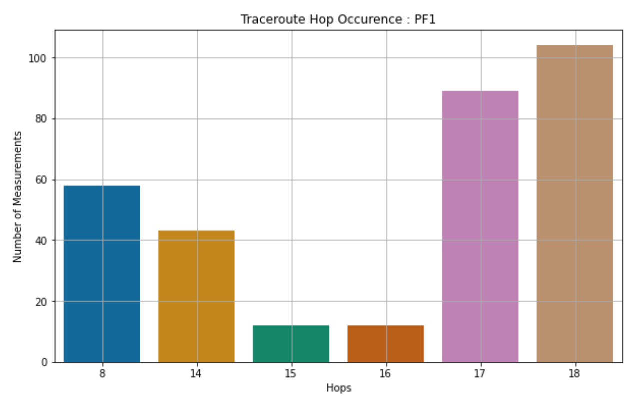
Can use figma for route diagrams

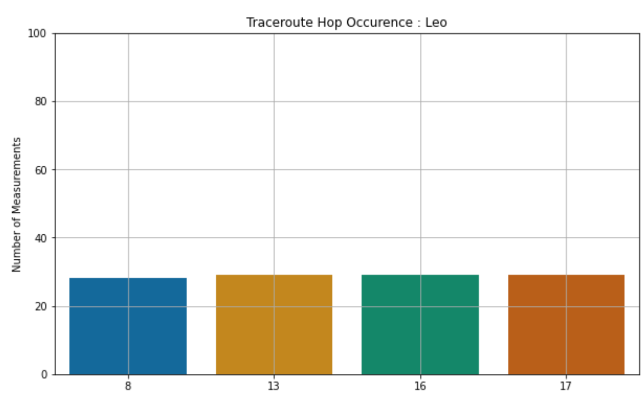

Traceroute RTT : DTN1
Round Trip Time (milliseconds)
0
100
200
300
400
500
600
700
8
14
15
16
17
18
Hops

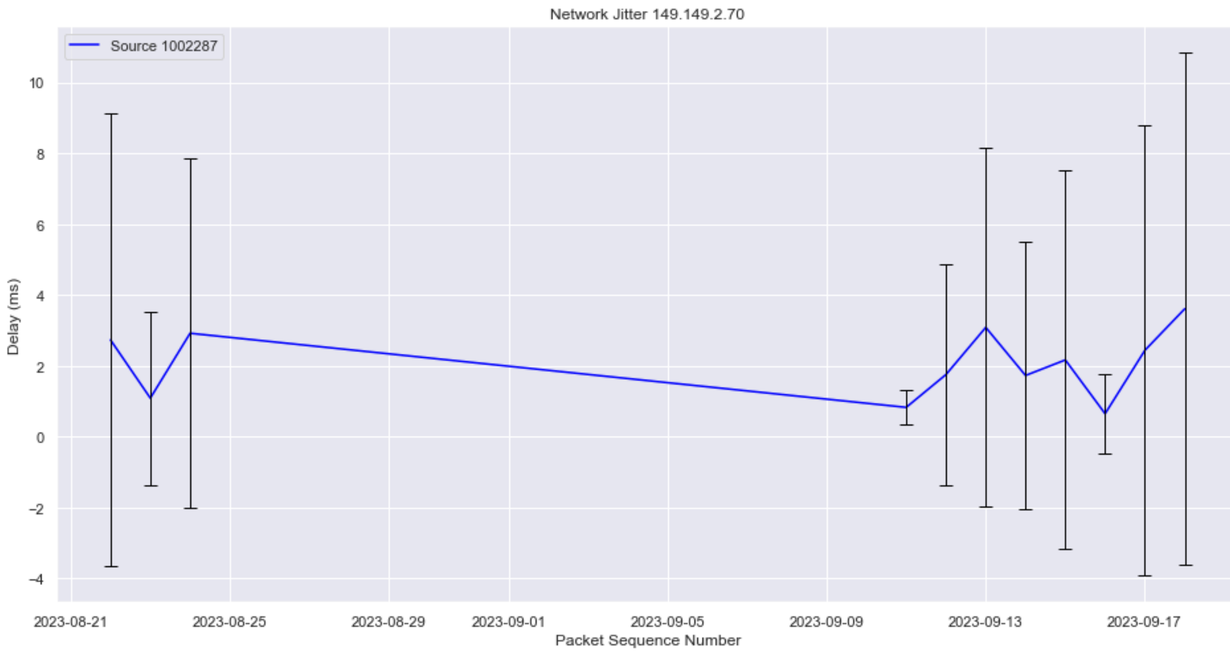

Traceroute RTT : PF1
Round Trip Time (milliseconds)
0
100
200
300
400
500
600
700
8
14
15
16
17
18
Hops


Traceroute RTT : Leo
Round Trip Time (milliseconds)
0
100
200
300
400
500
600
700
8
13
16
17
Hops

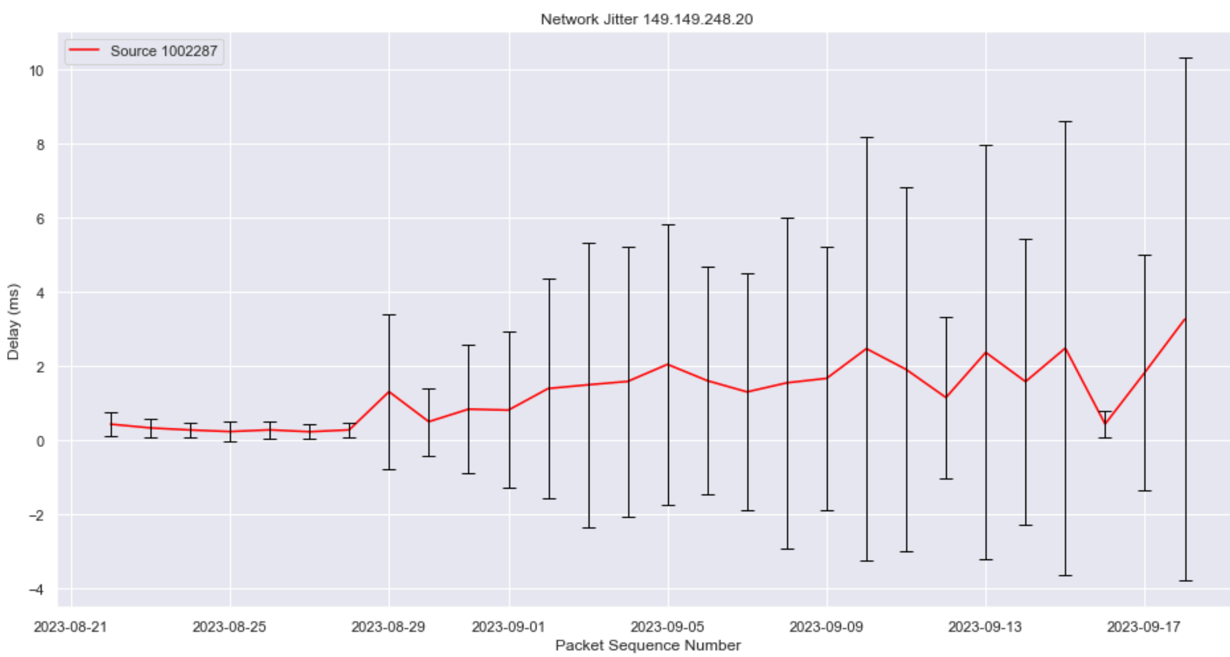

Traceroute Hop Occurence : DTN1
Number of Measurements
0
20
40
60
80
100
8
14
15
16
17
18
Hops


Traceroute Hop Occurence : PF1
Number of Measurements
0
20
40
60
80
100
8
14
15
16
17
18
Hops

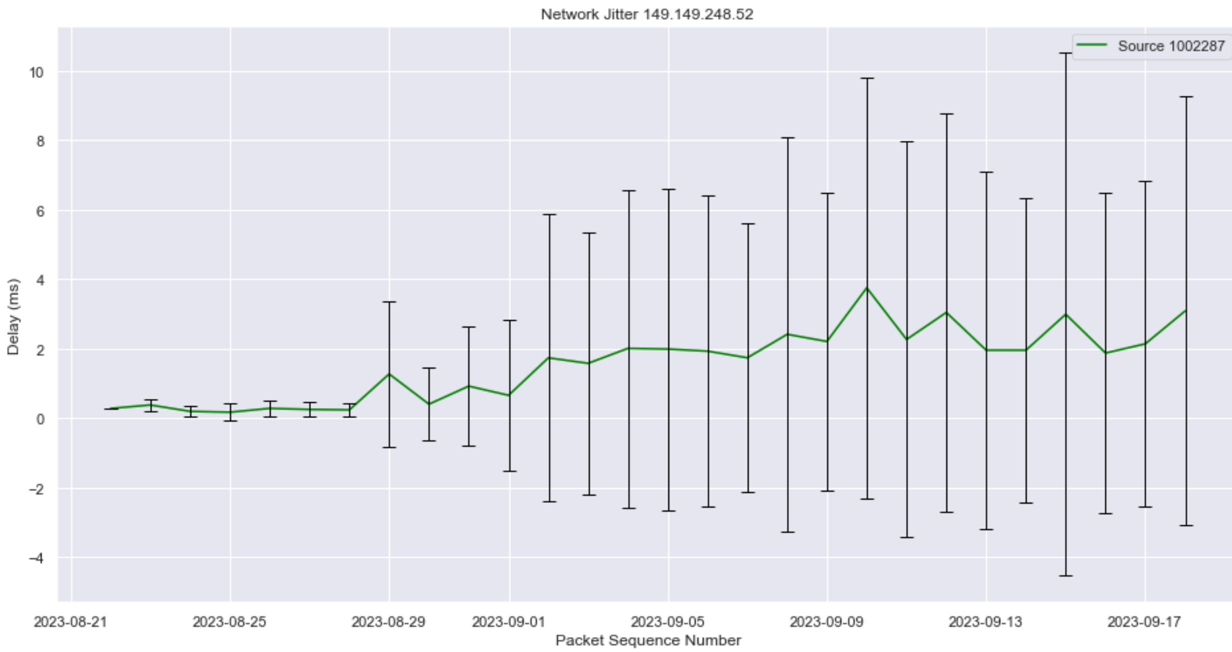

Traceroute Hop Occurence : Leo
Number of Measurements
0
20
40
60
80
100
8
13
16
17
Hops



Network Jitter 149.149.2.70
Source 1002287
Delay (ms)
10
8
6
4
2
0
-2
-4
2023-08-21
2023-08-25
2023-08-29
2023-09-01
2023-09-05
2023-09-09
2023-09-13
2023-09-17
Packet Sequence Number



Network Jitter 149.149.248.20
Source 1002287
Delay (ms)
10
8
6
4
2
0
-2
-4
2023-08-21
2023-08-25
2023-08-29
2023-09-01
2023-09-05
2023-09-09
2023-09-13
2023-09-17
Packet Sequence Number



Verified for a small sample that the jitter graphs are correct

IDs for RIPE Atlas Measurements that are currently being used for graphs:

Trace:
- 58881290 - DTN1
- 40016357 - Leo
- 55517771 - PF1

Ping:
- 39019527 - Leo
- 55518116 - PF1
- 58881309 - DTN1

URLs for Data Transfer Experiments:

https://mirror.umd.edu/fedora/linux/releases/38/Workstation/x86_64/iso/

http://ftp.usf.edu/pub/fedora/linux/releases/38/Workstation/x86_64/iso/

http://mirrors.maine.edu/Fedora/releases/38/Workstation/x86_64/iso/

https://mirrors.mit.edu/fedora/linux/releases/38/Workstation/x86_64/iso/

https://archive.linux.duke.edu/fedora/pub/fedora/linux/releases/38/Workstation/x86_64/iso/

Sudo curl <url> > /tmp/test1.iso

Ex. sudo curl
https://archive.linux.duke.edu/fedora/pub/fedora/linux/releases/38/Workstation/x86_64/iso/

> /tmp/test.iso

Sudo tcpdump -i eno2 port 443 or port 80 > /tmp/test1.pcap

Tcpdump -i p1p1 ‘(host systems-mr.tntech.edu or host _____) and (port 80 and port 443)’

Experiment	Source	Destination	Added?
Jitter	RIPE Atlas	Leo	✓
Jitter	RIPE Atlas	Perfsonar1	✓
Jitter	RIPE Atlas	DTN1	✓
Jitter	GCP	Leo	✓
Jitter	GCP	Perfsonar1	✓
Jitter	GCP	DTN1	✓
Jitter	Leo	Extern	
Jitter	Perfsonar1	Extern	
Jitter	DTN1	Extern	
Jitter	DTN1	DTN2	✓
Jitter	DTN1	Perfsonar1	✓
Jitter	Leo	Perfsonar1	✓
RTT (Ping)	RIPE Atlas	Leo	✓
RTT (Ping)	RIPE Atlas	PF1	✓
RTT (Ping)	RIPE Atlas	DTN1	✓
RTT (Ping)	GCP	Leo	✓
RTT (Ping)	GCP	Perfsonar1	✓
RTT (Ping)	GCP	DTN1	✓
RTT (Ping)	Leo	Extern	
RTT (Ping)	Perfsonar1	Extern	
RTT (Ping)	DTN1	Extern	
RTT (Ping)	DTN1	DTN2	✓

RTT (Ping)	DTN1	Perfsonar1	✓
RTT (Ping)	Leo	Perfsonar1	✓
Packet Loss	RIPE Atlas	Leo	✓
Packet Loss	RIPE Atlas	PF1	✓
Packet Loss	RIPE Atlas	DTN1	✓
Packet Loss	GCP	Leo	✓
Packet Loss	GCP	Perfsonar1	✓
Packet Loss	GCP	DTN1	✓
Packet Loss	Leo	Extern	
Packet Loss	Perfsonar1	Extern	
Packet Loss	DTN1	Extern	
Packet Loss	DTN1	DTN2	
Packet Loss	DTN1	Perfsonar1	
Packet Loss	Leo	Perfsonar1	
Hops	RIPE Atlas	Leo	✓
Hops	RIPE Atlas	PF1	✓
Hops	RIPE Atlas	DTN1	✓
Hops	GCP	Leo	✓
Hops	GCP	Perfsonar1	✓
Hops	GCP	DTN1	✓
Hops	Leo	Extern	
Hops	Perfsonar1	Extern	
Hops	DTN1	Extern	
Hops	DTN1	DTN2	
Hops	DTN1	Perfsonar1	
Hops	Leo	Perfsonar1	
RTT (Trace)	RIPE Atlas	Leo	✓
RTT (Trace)	RIPE Atlas	PF1	✓
RTT (Trace)	RIPE Atlas	DTN1	✓
RTT (Trace)	GCP	Leo	✓
RTT (Trace)	GCP	Perfsonar1	✓

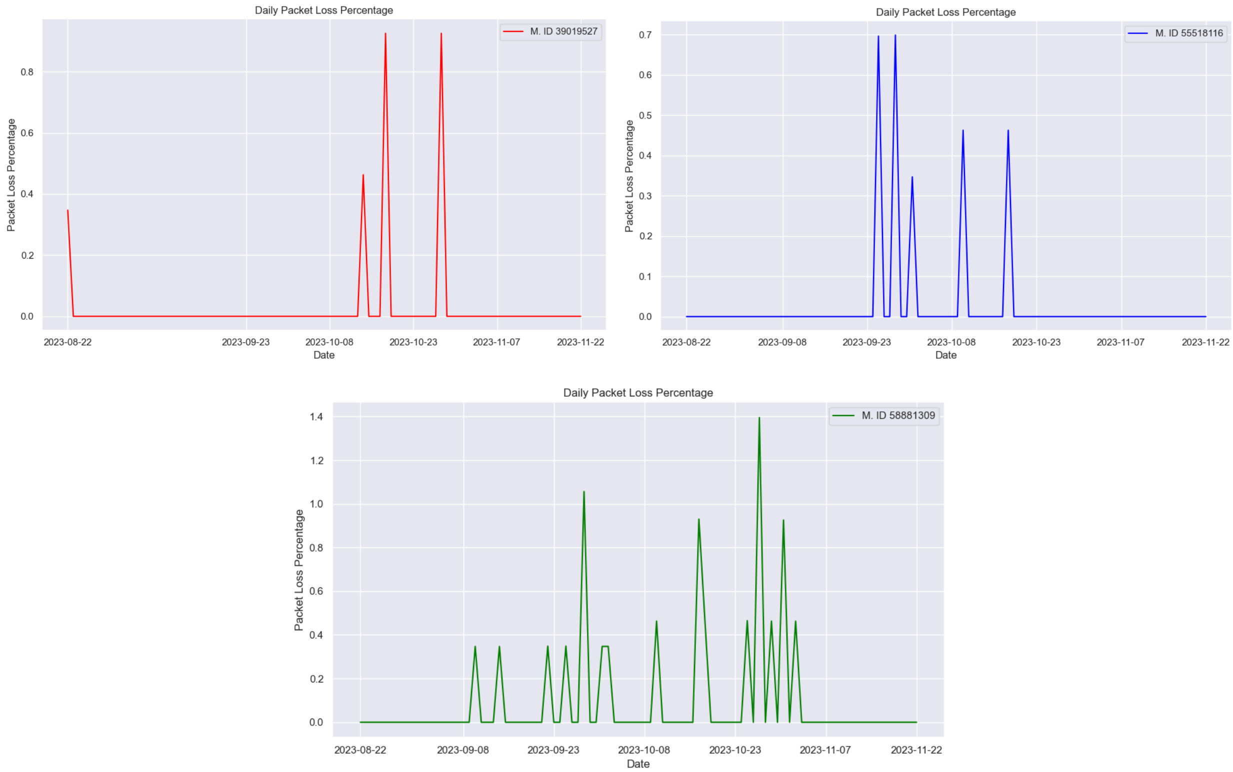
RTT (Trace)	GCP	DTN1	✓
RTT (Trace)	Leo	Extern	
RTT (Trace)	Perfsonar1	Extern	
RTT (Trace)	DTN1	Extern	
RTT (Trace)	DTN1	DTN2	
RTT (Trace)	DTN1	Perfsonar1	
RTT (Trace)	Leo	Perfsonar1	

For: October 15 - October 21 LATEST UPDATES		*Last updated: Oct 23, 2023*
Status	Description	Comments
Last completed *Due date:*	58. Data Transfer Running on Leo (for real this time) 59. Route diagrams 60. Reevaluated packet loss graphs, need to double check accuracy	
Working on currently/next	Currently working on : experimenting with cloud host; graph comparison models; convert all graphs to use timestamp restrictions to observe data from common time ranges Next to work on: finish testing tools; iperf3 tests Next milestones: Problems: iperf3 does not work with all measurement routes: (sever-client) “Leo - PF1” and “DTN1 - GCP” still do not work correctly The data transfer experiments actually aren’t running on DTN1 currently, there is an issue with it that I have emailed Dr. Renfro about but cannot be addressed until next week Iperf3 still not working for leo-pf1 and dtn1-gcp routes Ping script on leo stopped recording results in file despite ping working when run outside of the script, trying to figure out what is wrong	

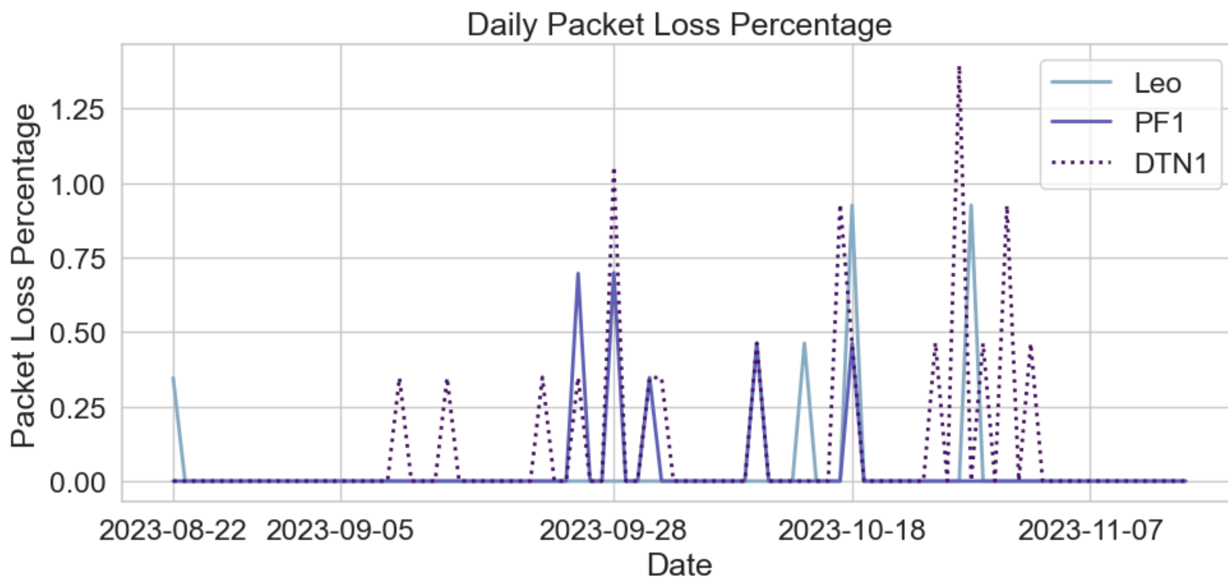
Plan & Timeline

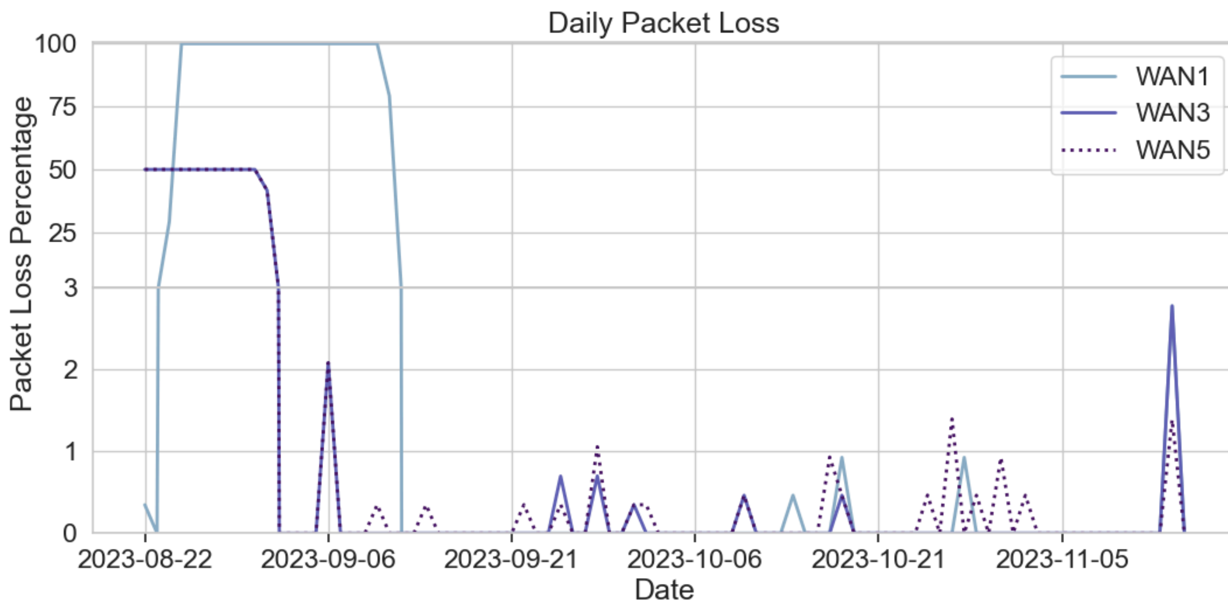
Oct 15 - 21	Reevaluate packet loss graphs, fix route diagrams, add graphs to paper
Oct 22 - 28	Make sure that data transfer experiments are successfully running, add graphs to paper
Oct 29 - Nov 4	

Check ripe atlas ping data to see if it is specified if probes are down when there is 100% packet loss - does not indicate status of probe when there is 100% packet loss. Should we just ignore instances of 100% packet loss? Just ignore 100% packet loss

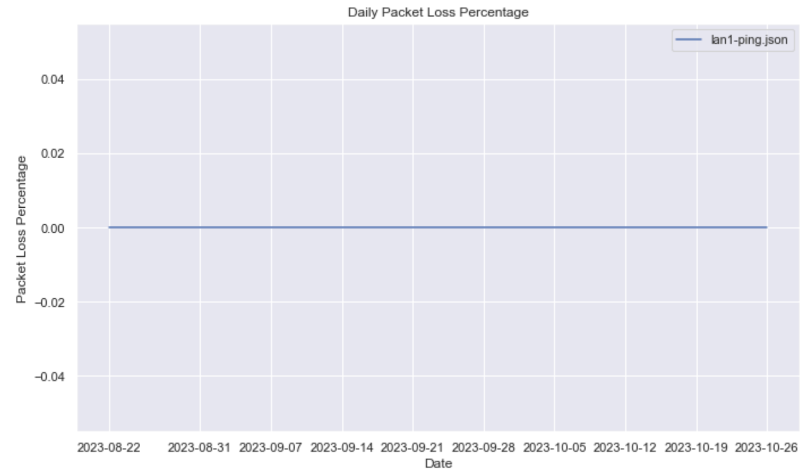


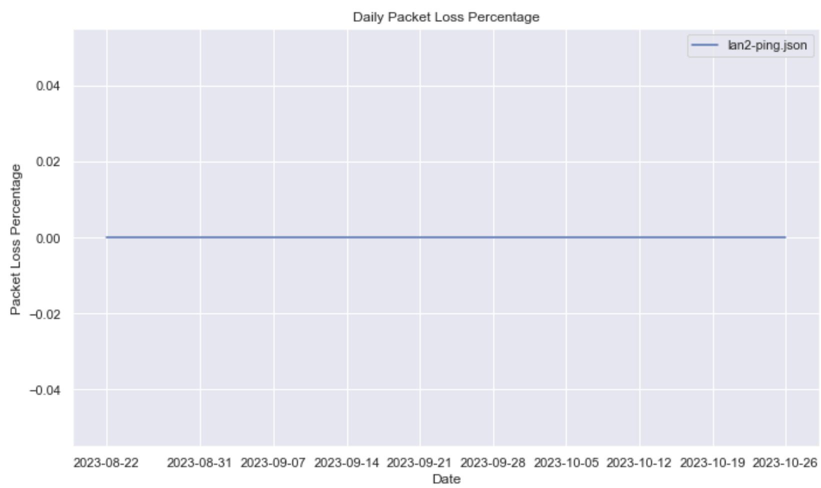
Leo (Red), Perfsonar1 (Blue), DTN1 (Green)

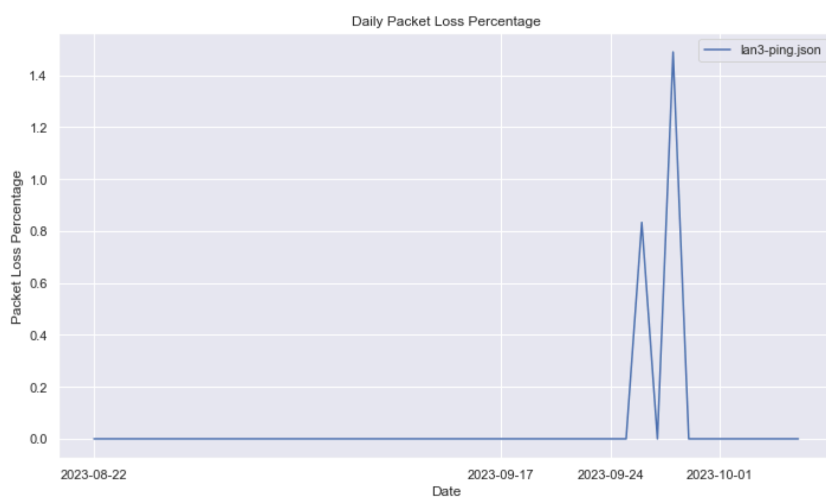


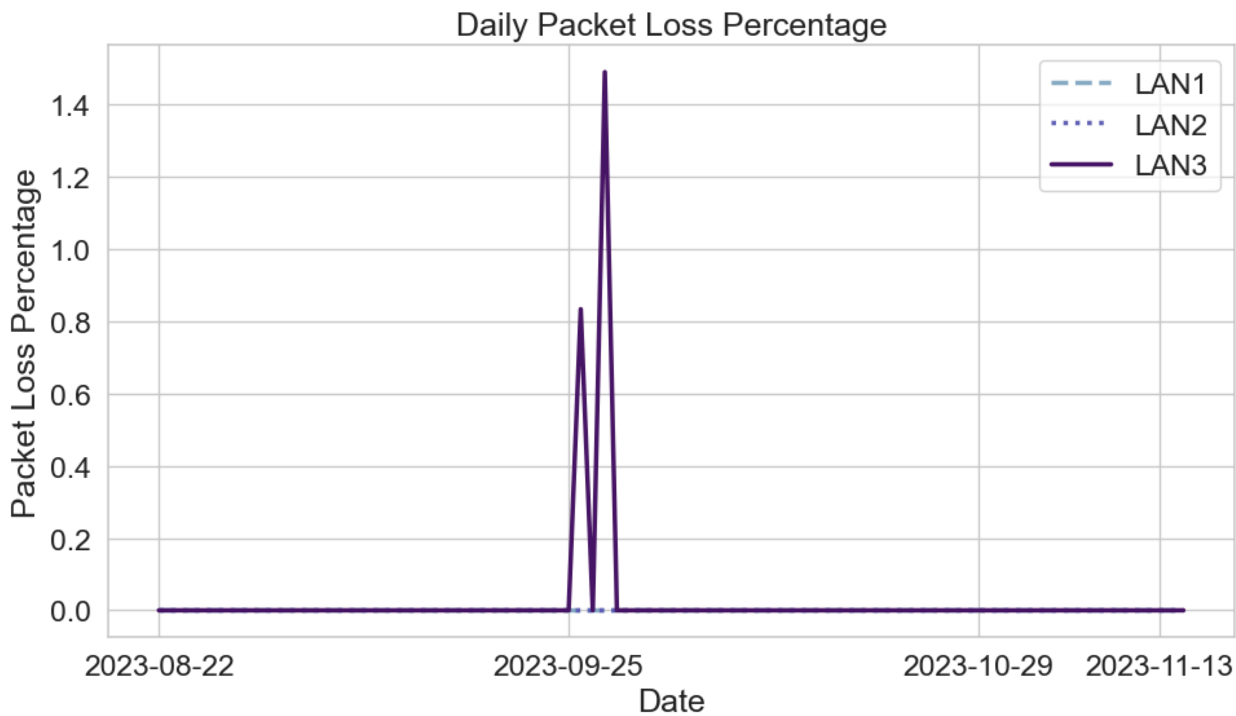


Analysis: When there is inbound traffic from RIPE Atlas, the DMZ experiences more spikes of packet loss than the campus network. While the campus network exhibits a period of 100% packet loss due to the campus node being down, both nodes of the DMZ experience 50% genuine packet loss. The Perfsonar1 node on the DMZ exhibits more instances of packet loss than the DTN1 node on the DMZ; it loses ~5% more packets than DTN1 over 3 months.

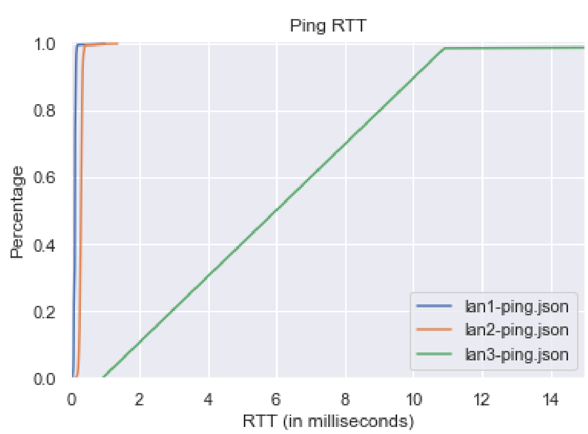

Daily Packet Loss Percentage
lan1-ping.json
Packet Loss Percentage
0.04
0.02
0.00
−0.02
−0.04
2023-08-22
2023-08-31
2023-09-07
2023-09-14
2023-09-21
2023-09-28
2023-10-05
2023-10-12
2023-10-19
2023-10-26
Date

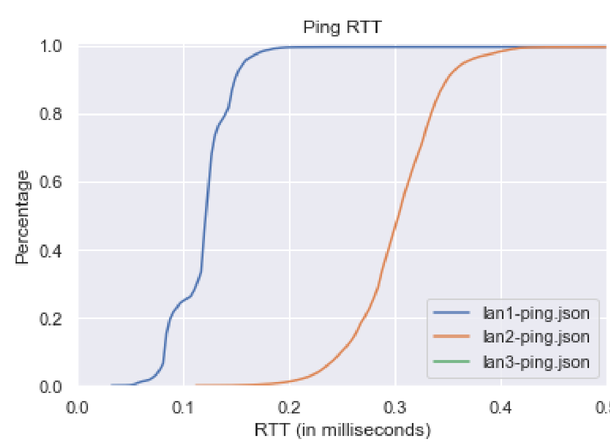

Daily Packet Loss Percentage
lan2-ping.json
Packet Loss Percentage
0.04
0.02
0.00
−0.02
−0.04
2023-08-22
2023-08-31
2023-09-07
2023-09-14
2023-09-21
2023-09-28
2023-10-05
2023-10-12
2023-10-19
2023-10-26
Date

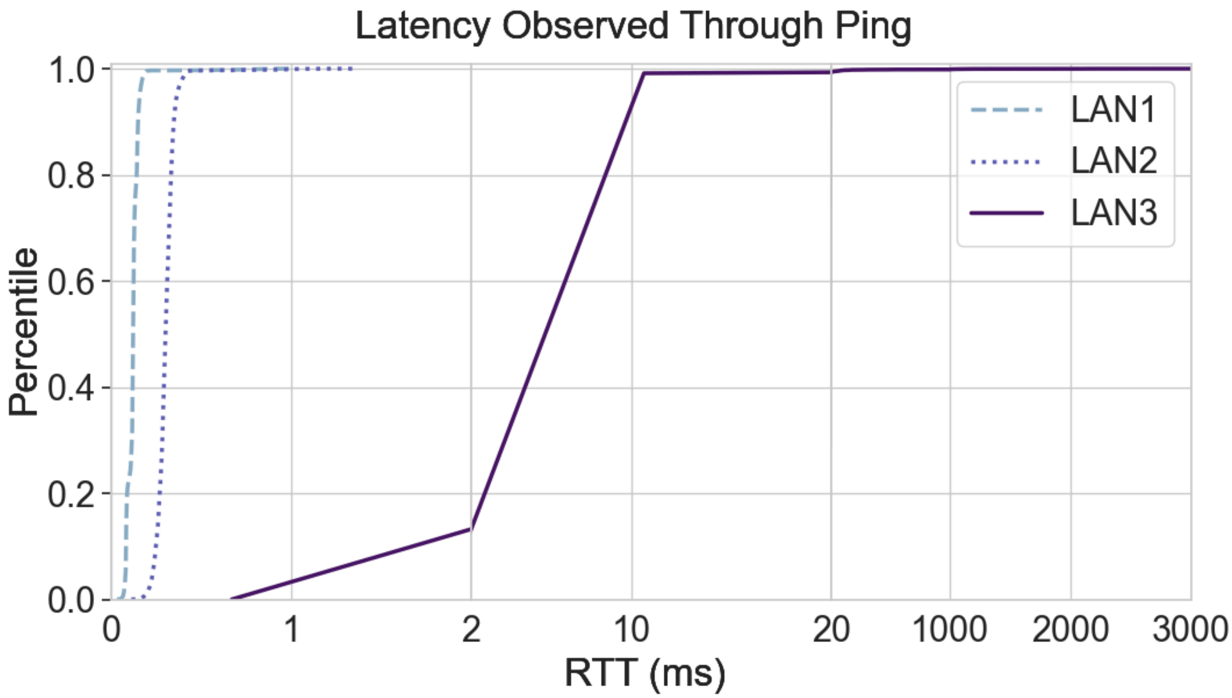

Daily Packet Loss Percentage
lan3-ping.json
Packet Loss Percentage
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2023-08-22
2023-09-17
2023-09-24
2023-10-01
Date



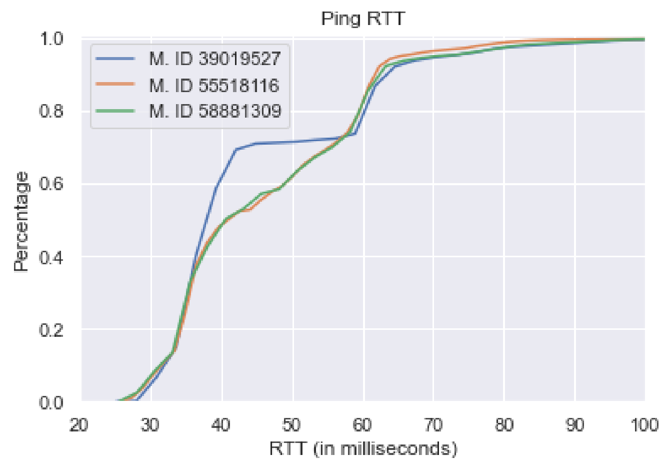
Analysis: During local traffic between local network nodes, the campus network node loses more packets than the DMZ nodes. The campus network loses ~2.3% more packets than the DMZ over the course of three months.

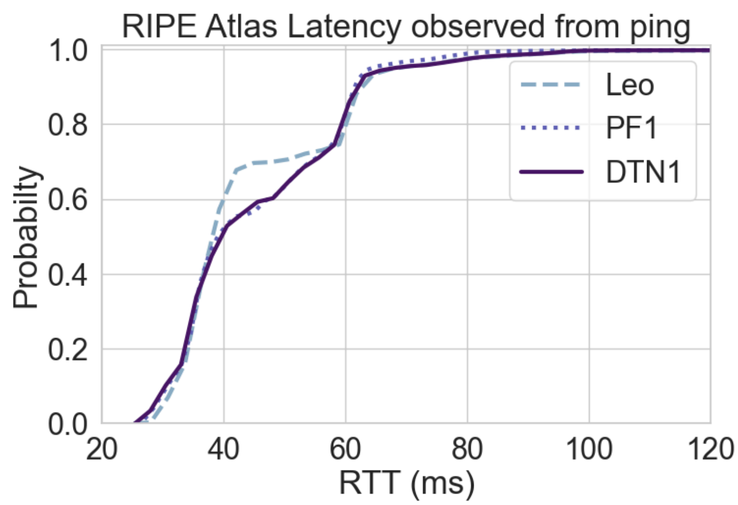


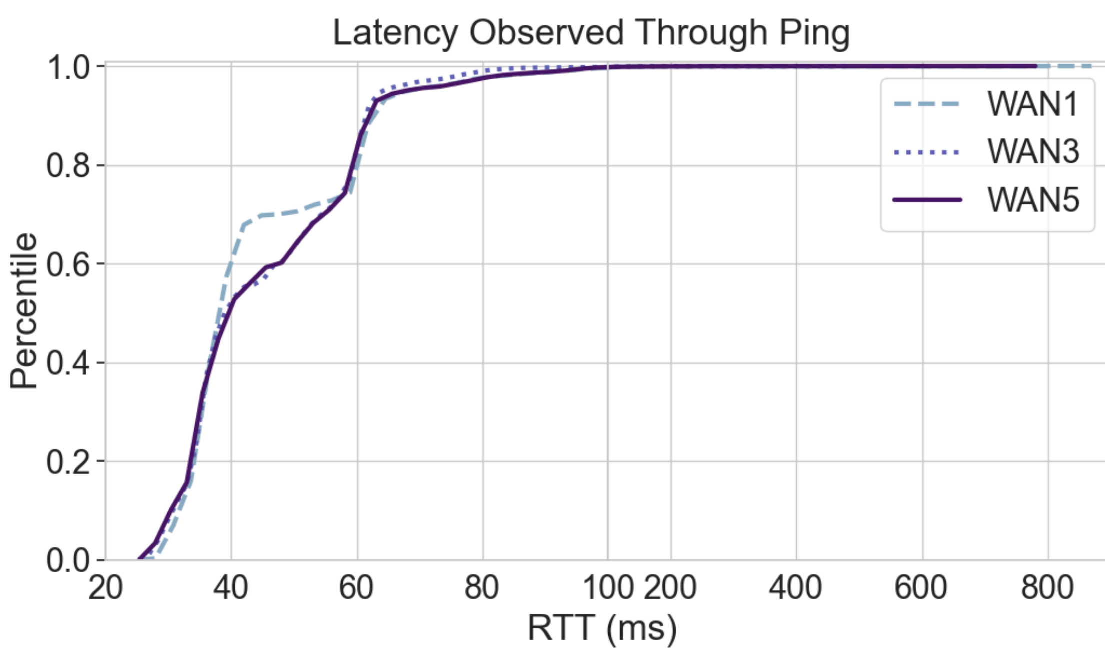




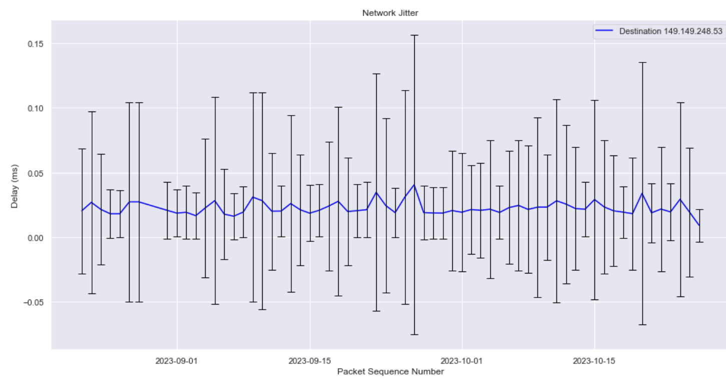
Analysis: During traffic between local nodes, the DMZ routes have significantly lower latency than the campus network. The DMZ's reported RTT is 80% - 99.5% (0.8 - 2998 ms) lower than that of the campus network's route. The route from DTN1 to DTN2 has 33% - 61.2% lower latency than the route from DTN1 to Perfsonar1.

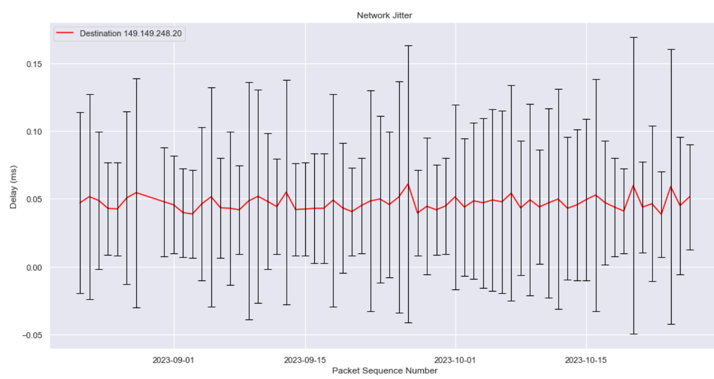

Ping RTT
M. ID 39019527
M. ID 55518116
M. ID 58881309
Percentage
RTT (in milliseconds)

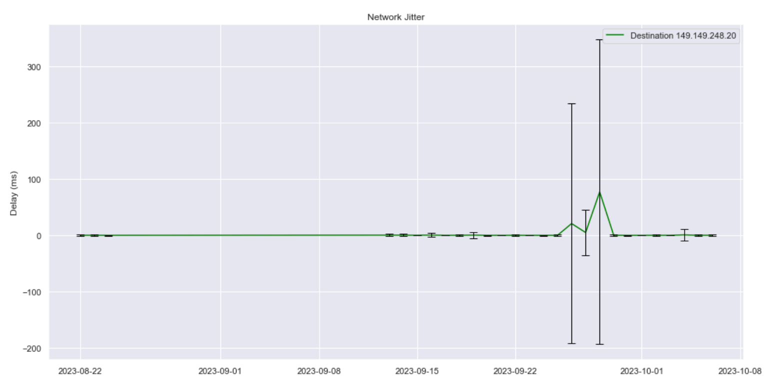

RIPE Atlas Latency observed from ping
Leo
PF1
DTN1
Probabilty
RTT (ms)

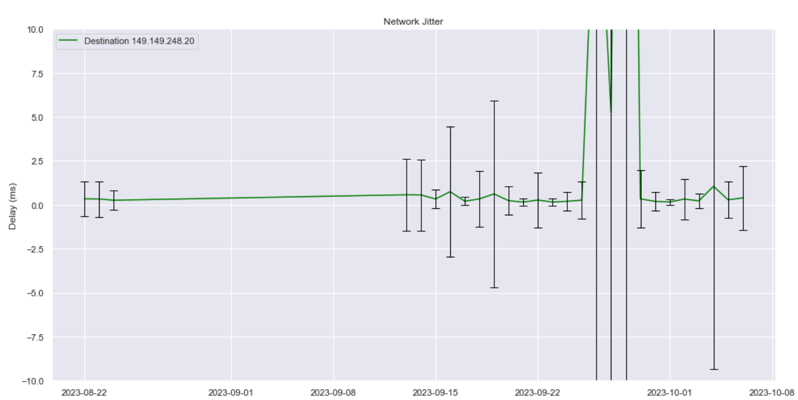

Latency Observed Through Ping
WAN1
WAN3
WAN5
Percentile
RTT (ms)

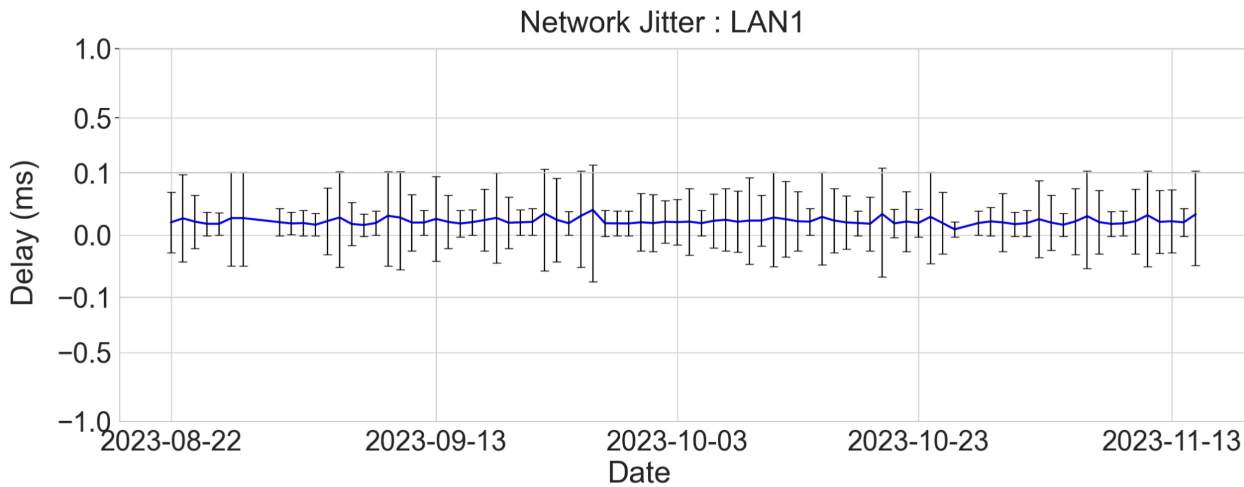
Analysis: When there is inbound WAN traffic from RIPE Atlas, both the campus network and the DMZ exhibit similar latency, with the campus network occasionally having a lower average latency than the DMZ by as much as ~20ms (5% - 30.5%) (unexpected result). This result is due to a period of time in which 2 of 4 sources of the ping measurements were not arriving at the DMZ despite arriving at the campus network. 1 of the 2 sources that completed its route to the DMZ was one of typical high latency, which affected the average latency poorly as the other 3 sources typically produce much lower latency. ***Takeaway: the campus network and the DMZ usually have similar latency with inbound traffic from RIPE.

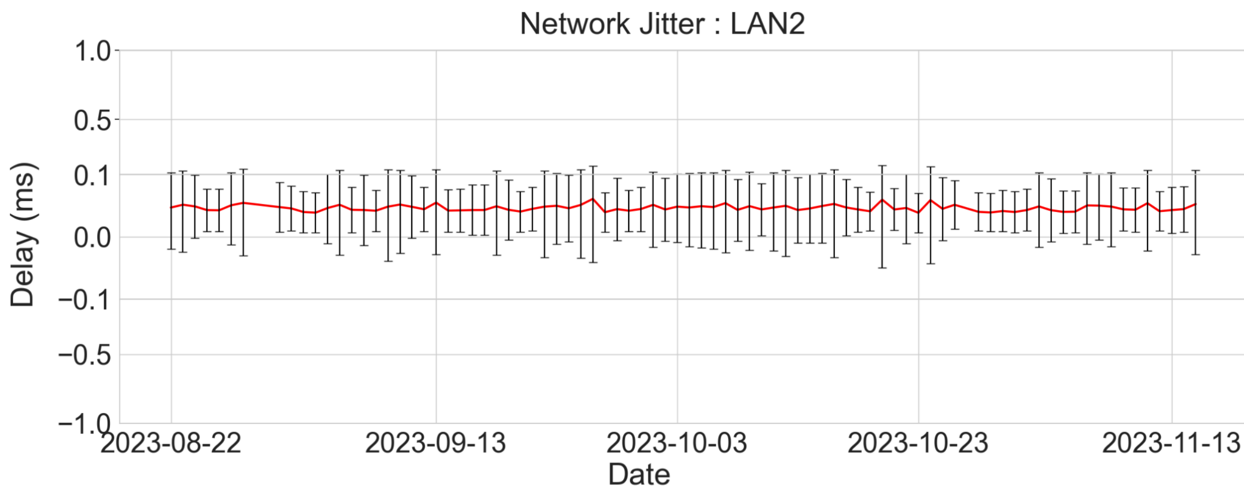

Network Jitter
Destination 149.149.248.53
Delay (ms)
0.15
0.10
0.05
0.00
-0.05
2023-09-01
2023-09-15
2023-10-01
2023-10-15
Packet Sequence Number

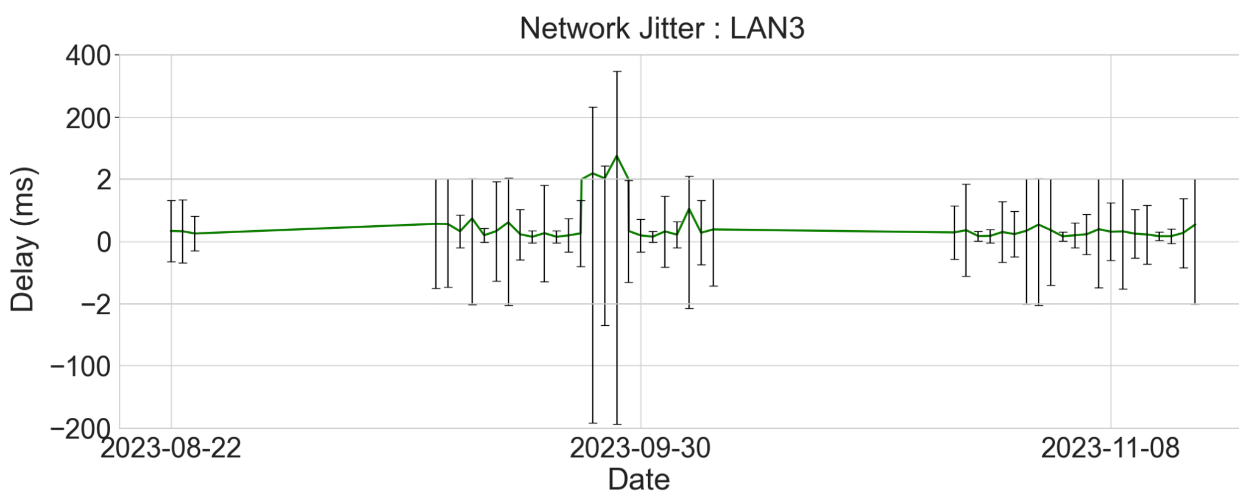

Network Jitter
Destination 149.149.248.20
Delay (ms)
0.15
0.10
0.05
0.00
-0.05
2023-09-01
2023-09-15
2023-10-01
2023-10-15
Packet Sequence Number


Network Jitter
Destination 149.149.248.20
Delay (ms)
300
200
100
0
-100
-200
2023-08-22
2023-09-01
2023-09-08
2023-09-15
2023-09-22
2023-10-01
2023-10-08
Packet Sequence Number

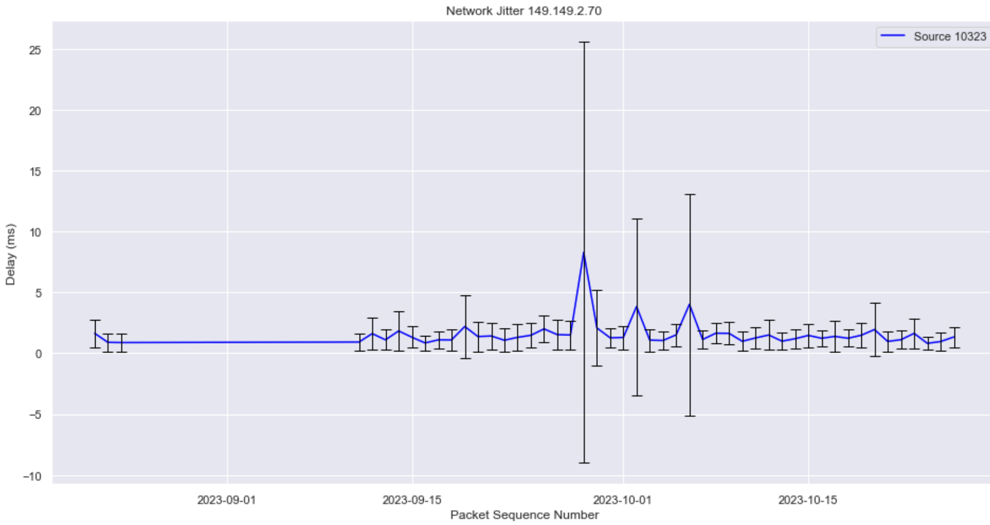

Network Jitter
Destination 149.149.248.20
Delay (ms)
10.0
7.5
5.0
2.5
0.0
-2.5
-5.0
-7.5
-10.0
2023-08-22
2023-09-01
2023-09-08
2023-09-15
2023-09-22
2023-10-01
2023-10-08
Packet Sequence Number

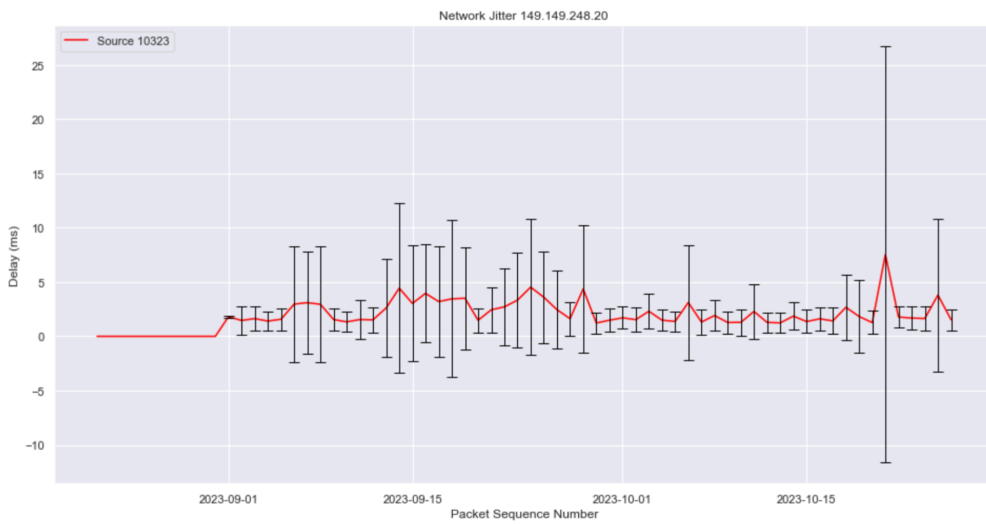


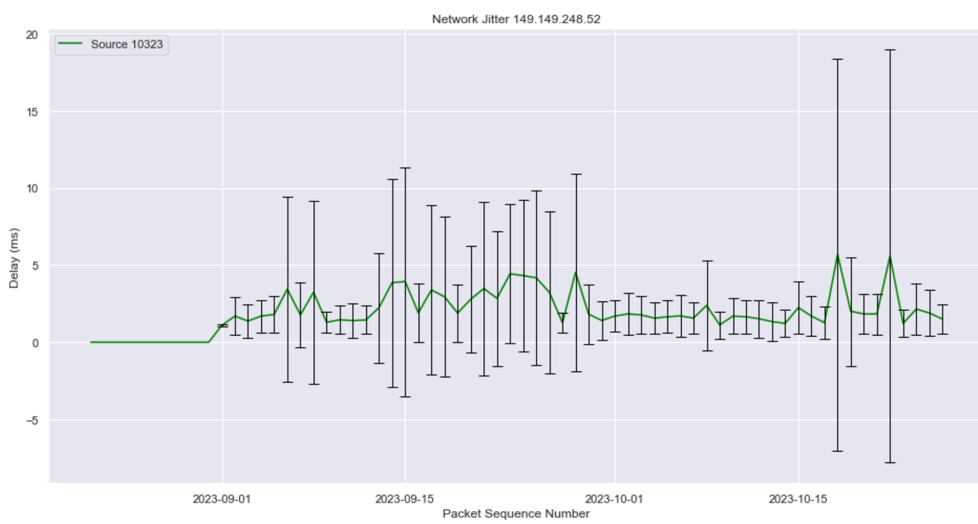


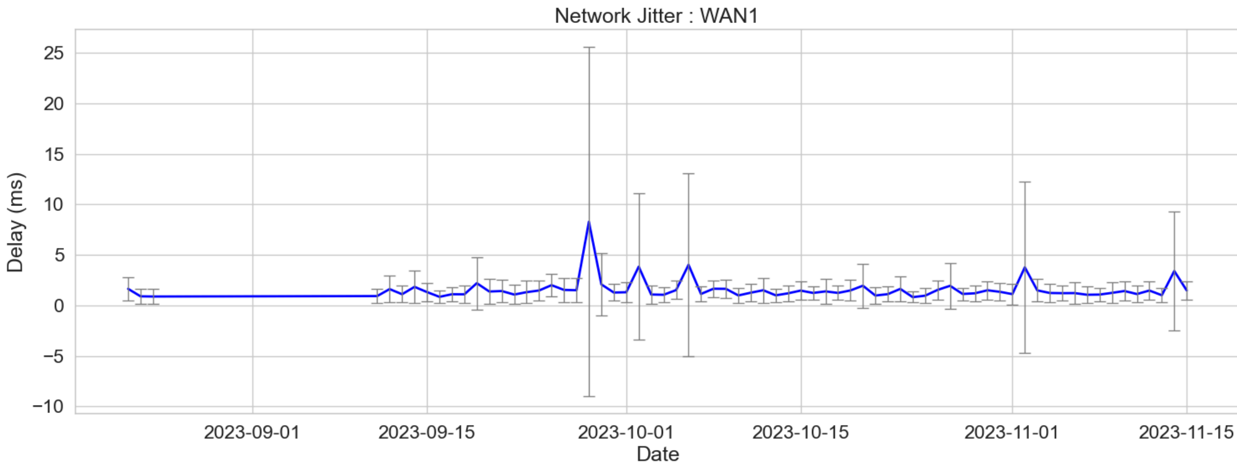


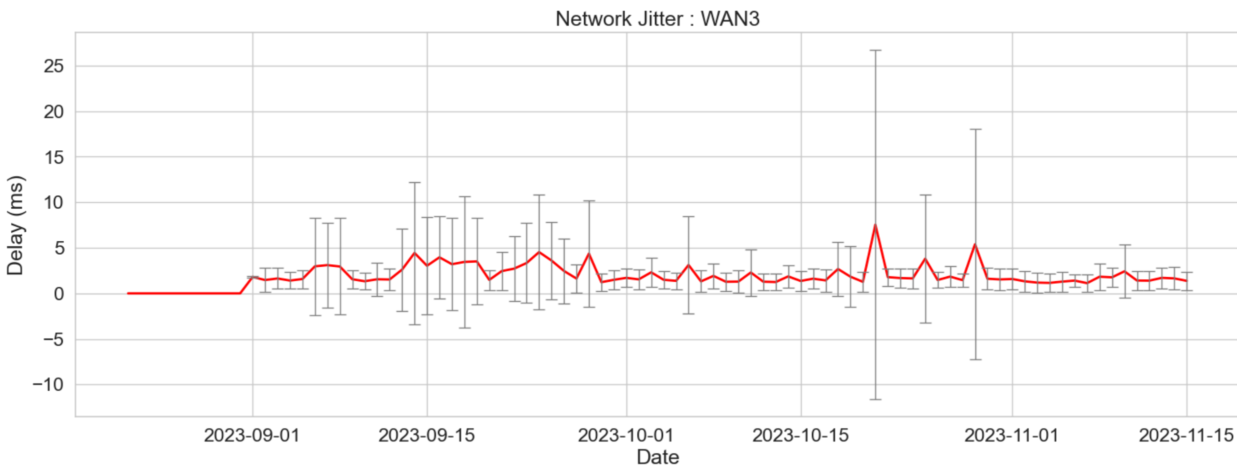
Analysis: With LAN traffic, the campus network presents significantly more jitter than the DMZ, with the campus network exhibiting moments of extreme deviation.

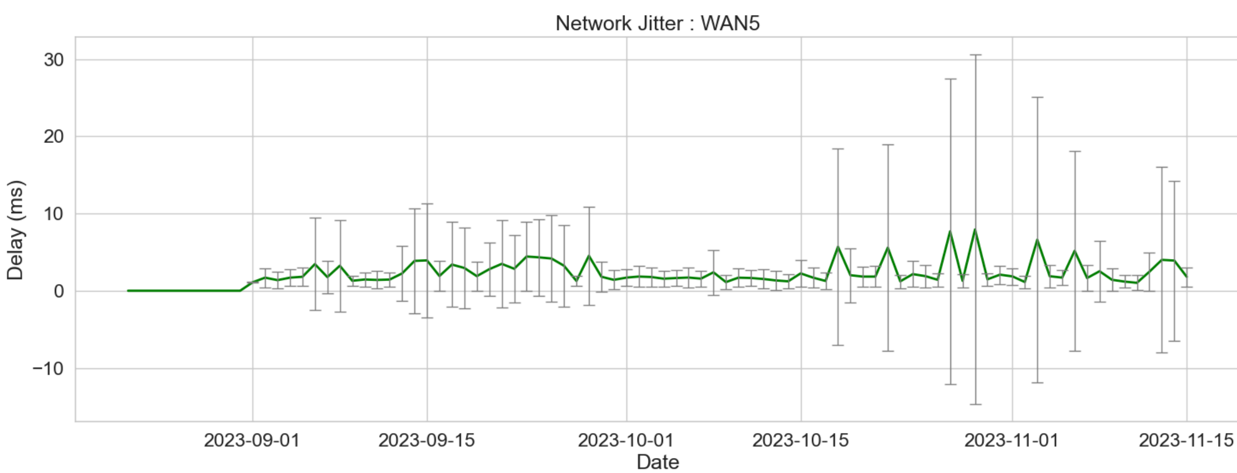

Network Jitter 149.149.2.70
Source 10323
Delay (ms)
Packet Sequence Number
2023-09-01
2023-09-15
2023-10-01
2023-10-15


Network Jitter 149.149.248.20
Source 10323
Delay (ms)
Packet Sequence Number
2023-09-01
2023-09-15
2023-10-01
2023-10-15

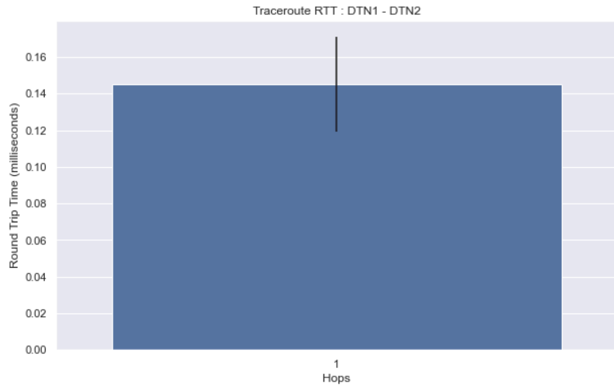

Network Jitter 149.149.248.52
Source 10323
Delay (ms)
Packet Sequence Number
2023-09-01
2023-09-15
2023-10-01
2023-10-15



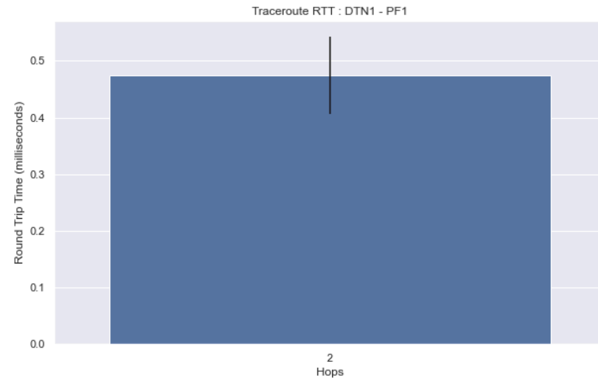




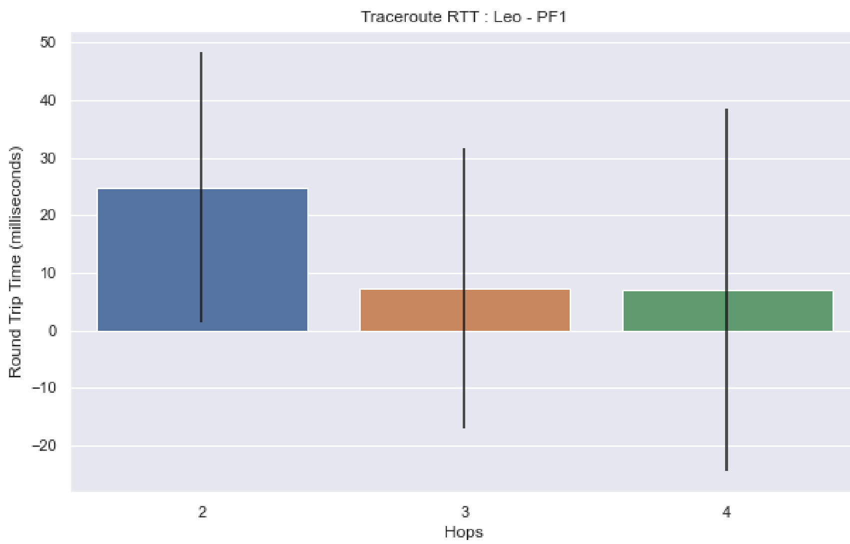
Analysis: With inbound WAN traffic from RIPE Atlas, the DMZ appears to have slightly higher jitter than the campus network and also exhibits more variation.



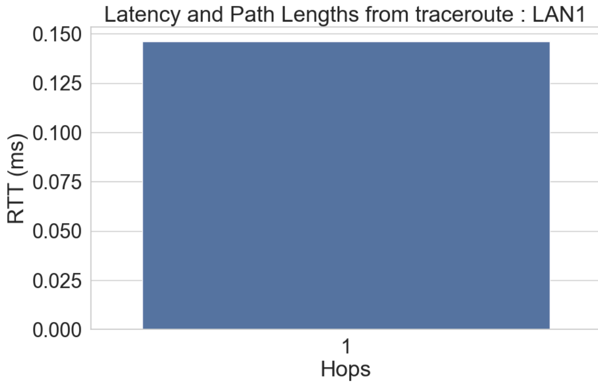
Traceroute RTT : DTN1 - DTN2
Round Trip Time (milliseconds)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
1
Hops

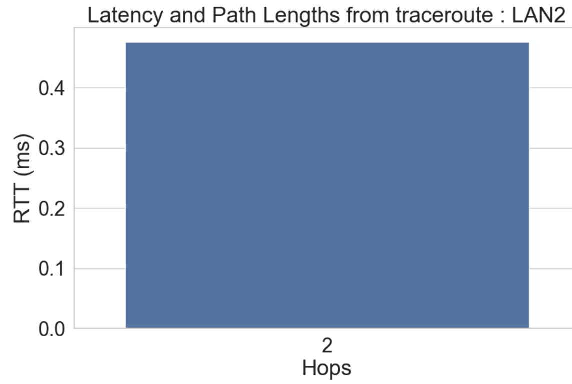


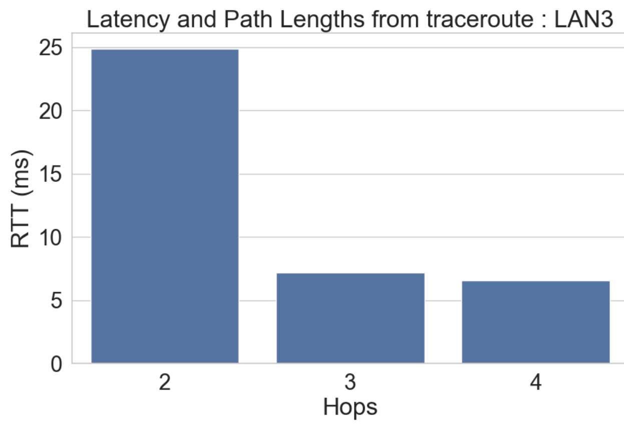
Traceroute RTT : DTN1 - PF1
Round Trip Time (milliseconds)
0.0
0.1
0.2
0.3
0.4
0.5
2
Hops



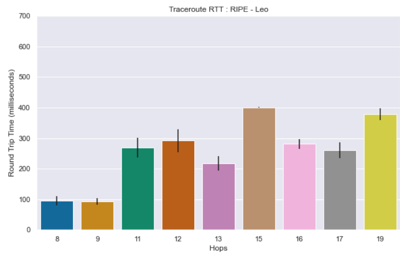
Traceroute RTT : Leo - PF1
Round Trip Time (milliseconds)
50
40
30
20
10
0
−10
−20
2
3
4
Hops

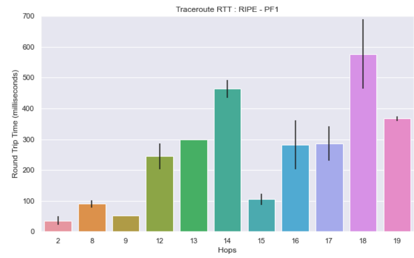


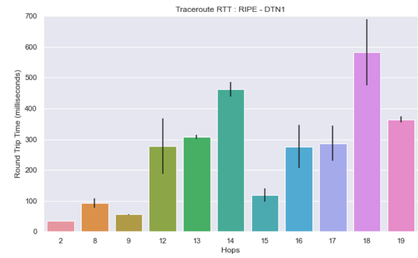


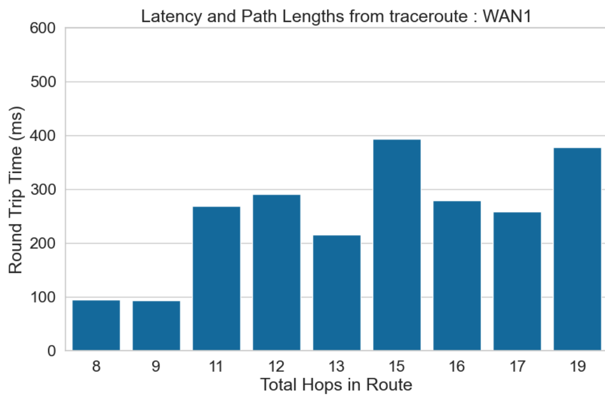


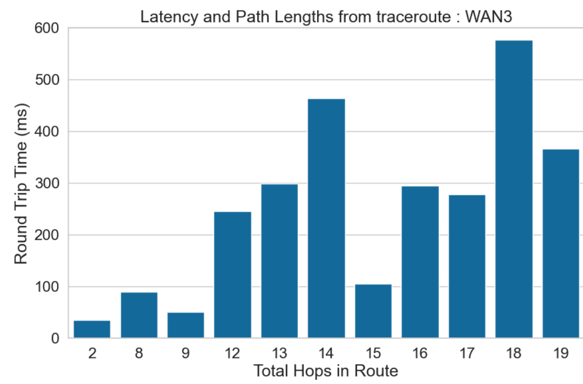
Analysis: When traffic is internal, the DMZ significantly outperforms the campus network and follows more consistent paths. The DMZ performs 98% faster than the campus network, considering common route lengths.

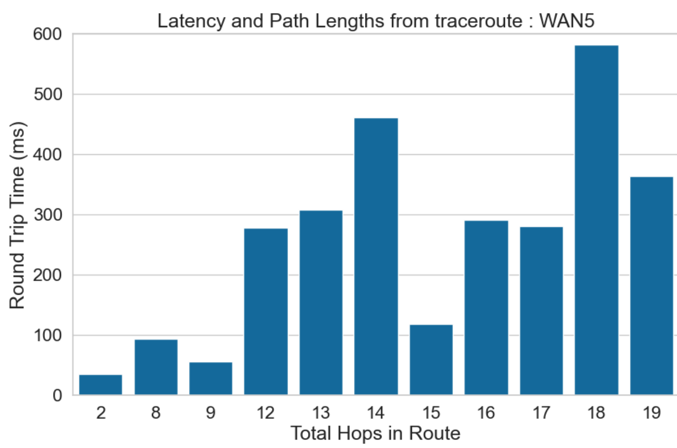












Analysis: During inbound WAN traffic, both the DMZ and the campus network have a majority of the same route lengths, with the DMZ having one route length that is significantly lower (6 hops lower) than the campus network. Both networks have moments of outperforming and underperforming the other.

For: November 6 - 13 LATEST UPDATES		*Last updated: Nov 27, 2023*
Status	Description	Comments
Last completed *Due date:*	61. Plotted GCP data 62. Plotted Extern data	
Working on currently/next	Currently working on : experimenting with cloud host; graph comparison models; iperf3 tests; set up measurements to run over break Next to work on: finish testing tools Next milestones: Problems: iperf3 does not work with all measurement routes: (sever-client) “Leo - PF1” and “DTN1 - GCP” still do not work correctly Iperf now works for leo-pf1 and pf1-gcp, but it not longer works for the other routes	

Plan & Timeline

Nov 6 - 13	Start iperf3 tests, graph GCP data
Nov 14 - 19	
Nov 20 - 26	

Check iperf3 permissions again

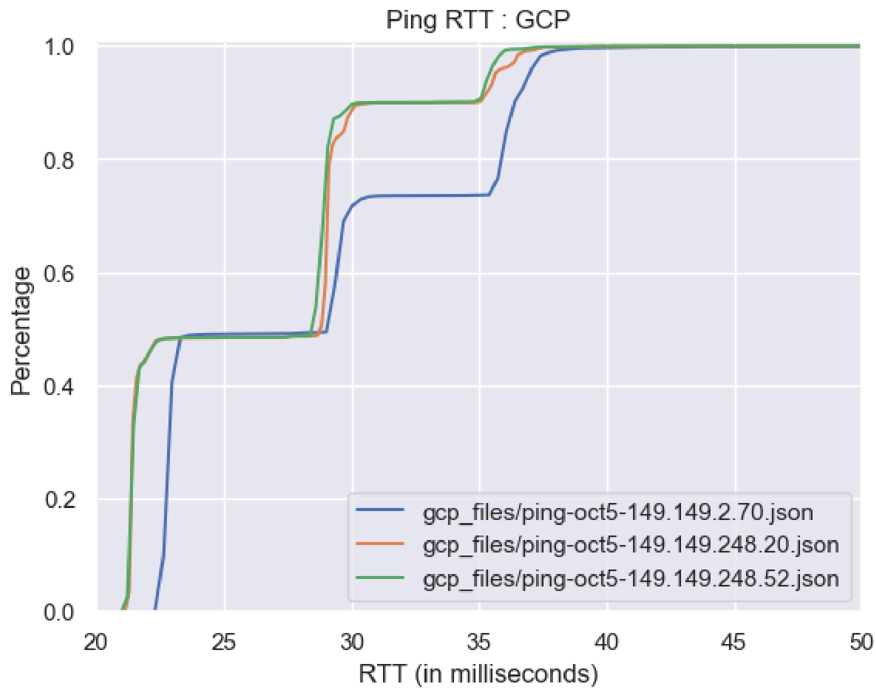
Make sure that gcp ip address is the same after rebooting it

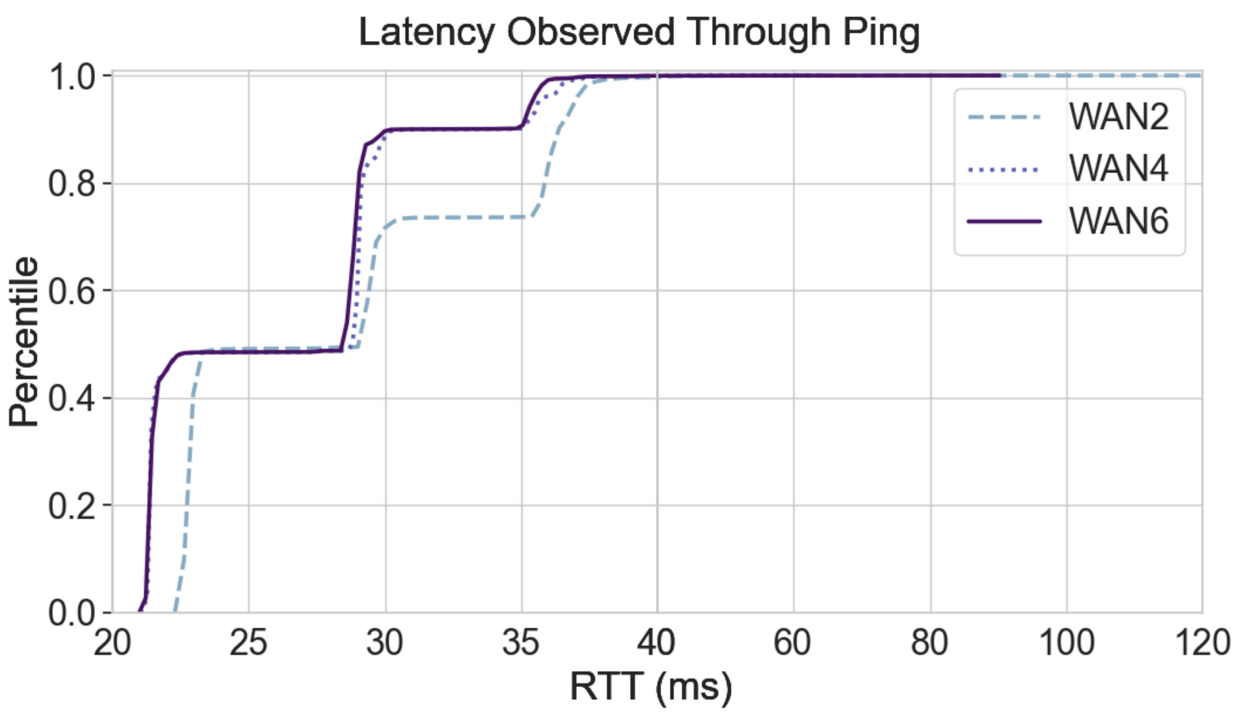
Paper points: Just because you have a DMZ, that does not guarantee better performance.

Try to start iperf3 measurements. Can do leo - pf1 pf1 - gcp if dtn1 - gcp does not work

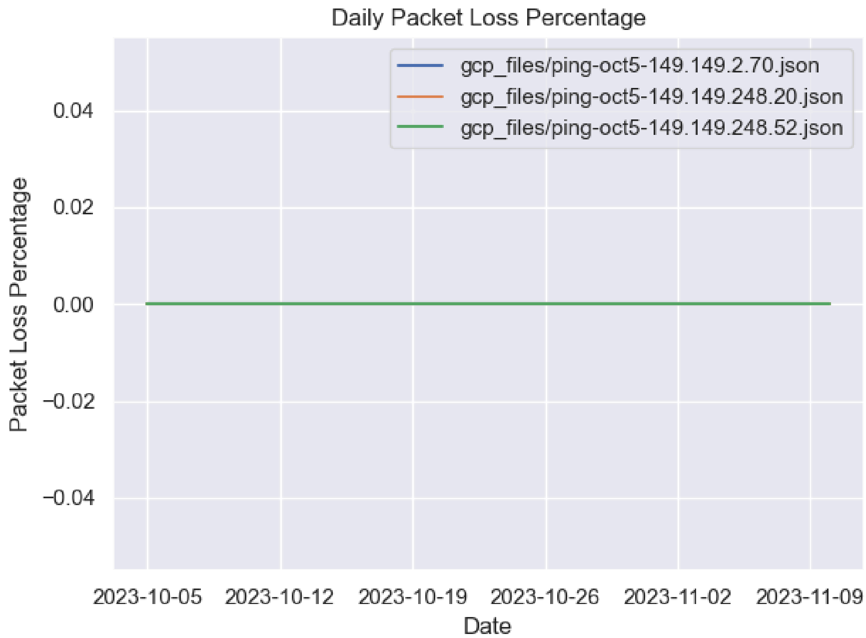
Make comparison graphs with GCP & extern data

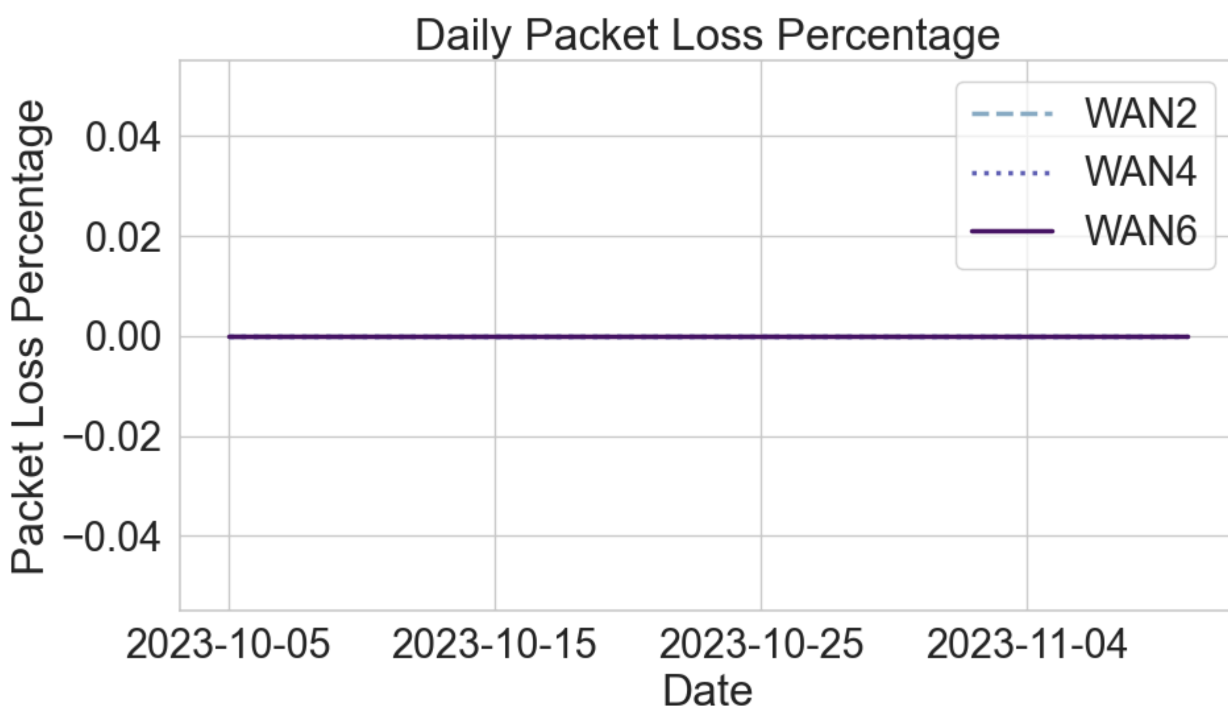
Make notes on observations of each experiment. Set up measurements to run over break.
Examine Data Transfer data.



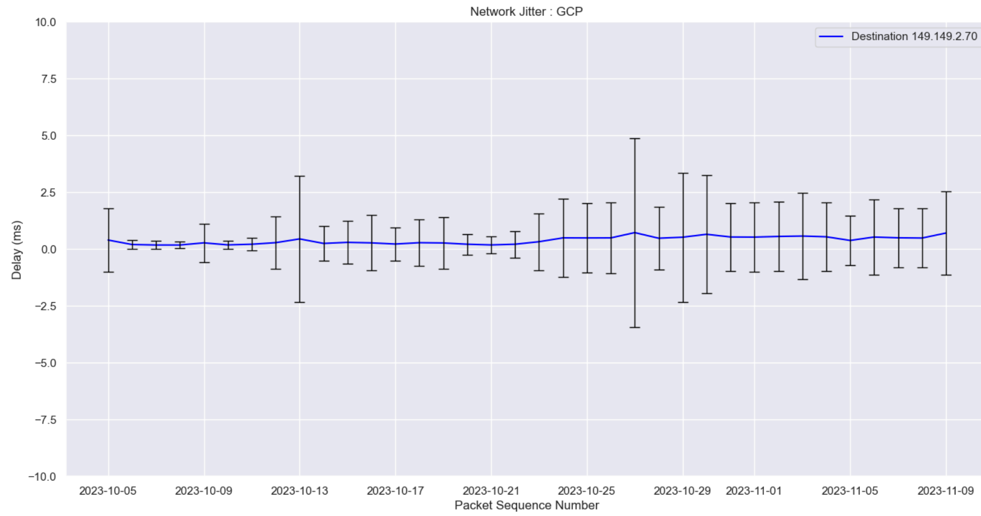


Analysis: When traffic is inbound from Google Cloud, both nodes on the DMZ tend to have similar latency, with an occasional ~2% difference. The campus network tends to have higher latency but occasionally has similar latency to the DMZ. The campus network has a ~2% - 24% higher latency than the DMZ.

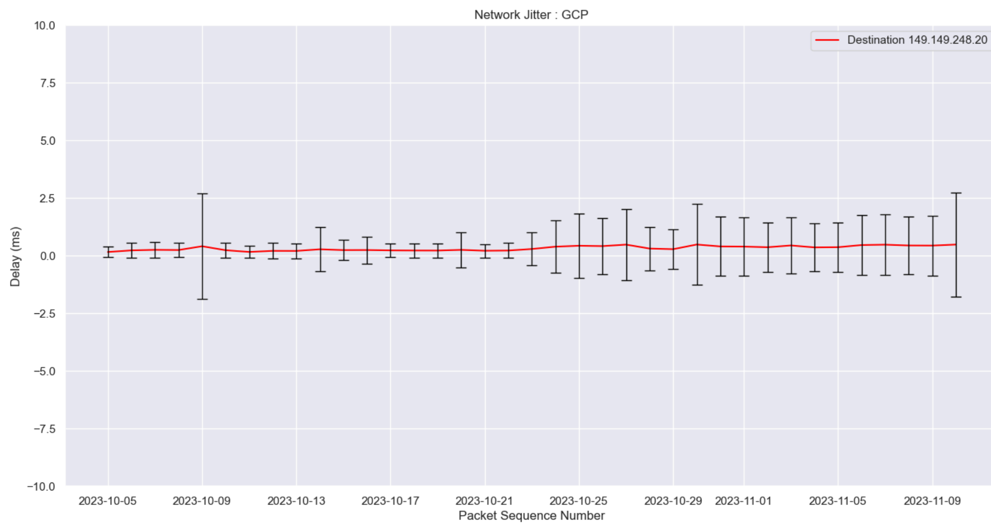




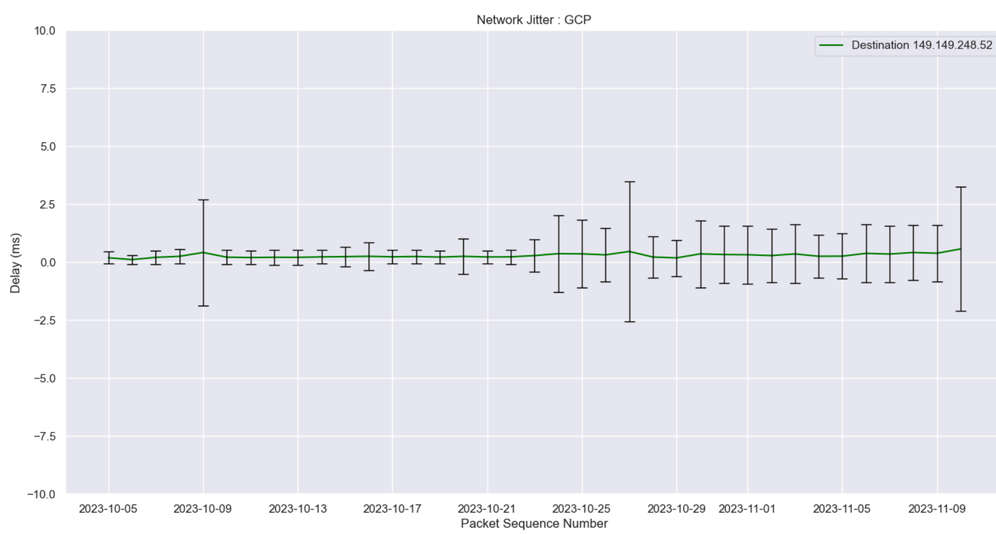
Analysis: When traffic is incoming from the cloud, there is no pattern of packet loss across DMZ or campus network.



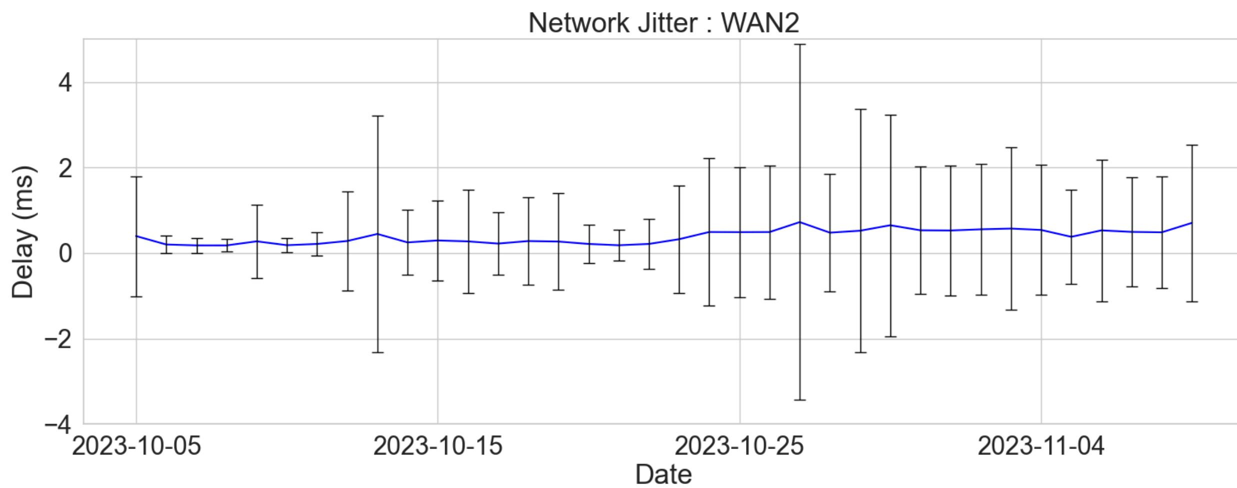
Network Jitter : GCP
Destination 149.149.2.70
Delay (ms)
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
2023-10-05
2023-10-09
2023-10-13
2023-10-17
2023-10-21
2023-10-25
2023-10-29
2023-11-01
2023-11-05
2023-11-09
Packet Sequence Number

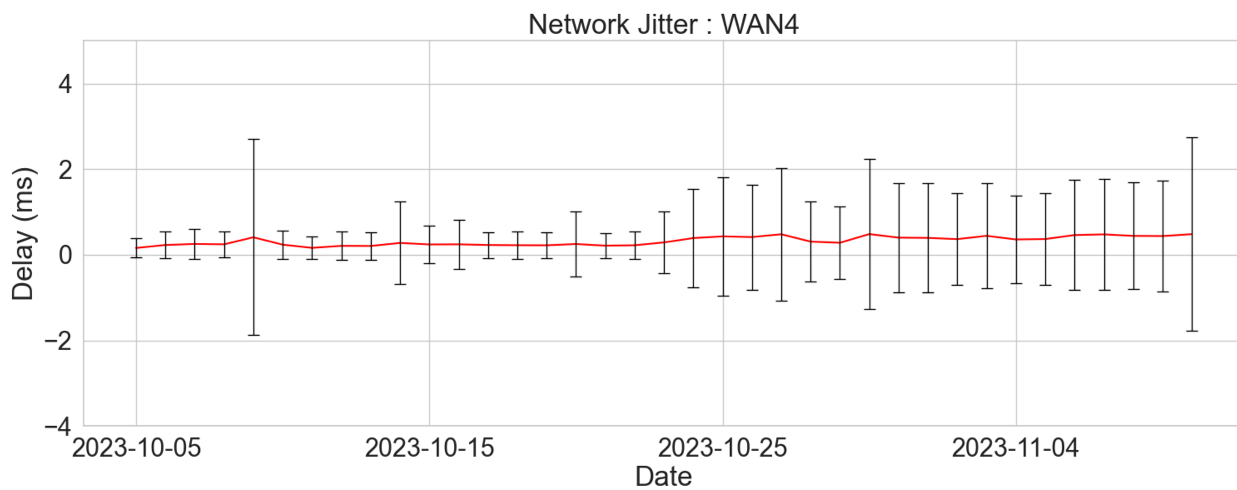


Network Jitter : GCP
Destination 149.149.248.20
Delay (ms)
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
2023-10-05
2023-10-09
2023-10-13
2023-10-17
2023-10-21
2023-10-25
2023-10-29
2023-11-01
2023-11-05
2023-11-09
Packet Sequence Number

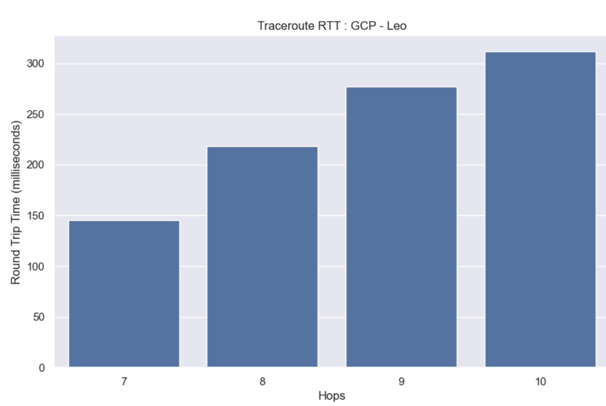


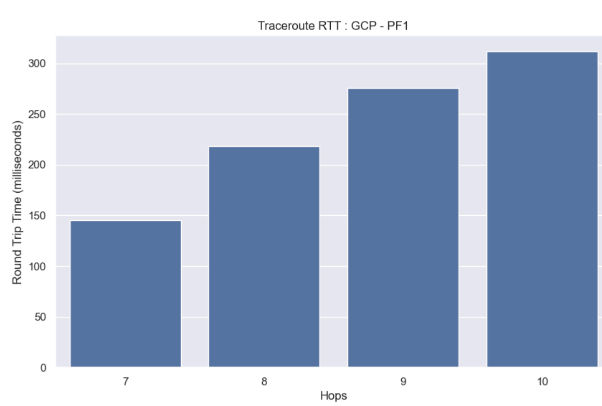
Network Jitter : GCP
Destination 149.149.248.52
Delay (ms)
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
2023-10-05
2023-10-09
2023-10-13
2023-10-17
2023-10-21
2023-10-25
2023-10-29
2023-11-01
2023-11-05
2023-11-09
Packet Sequence Number

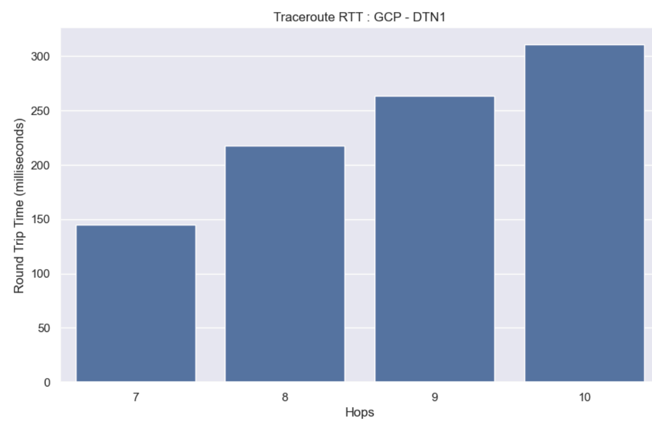


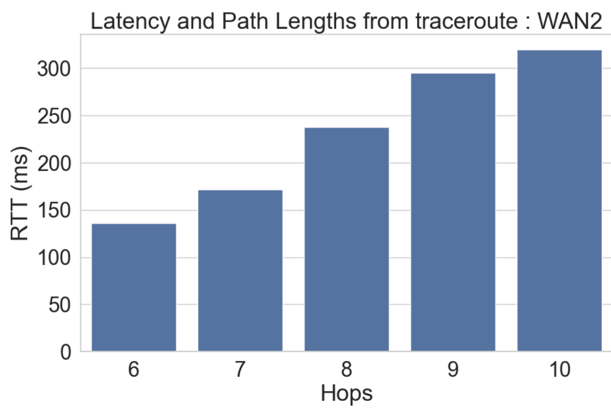


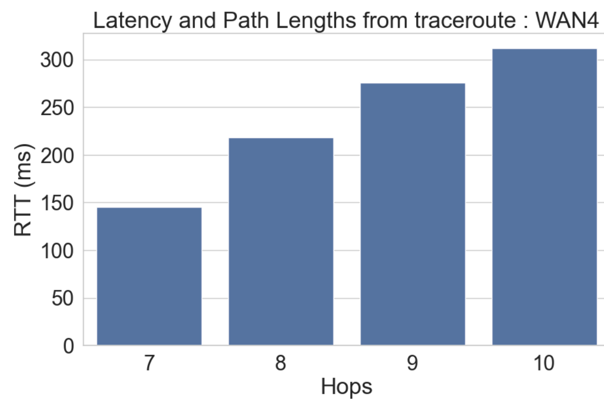
Analysis: With inbound WAN traffic from the cloud, both the campus network and the DMZ exhibit similar jitter patterns.

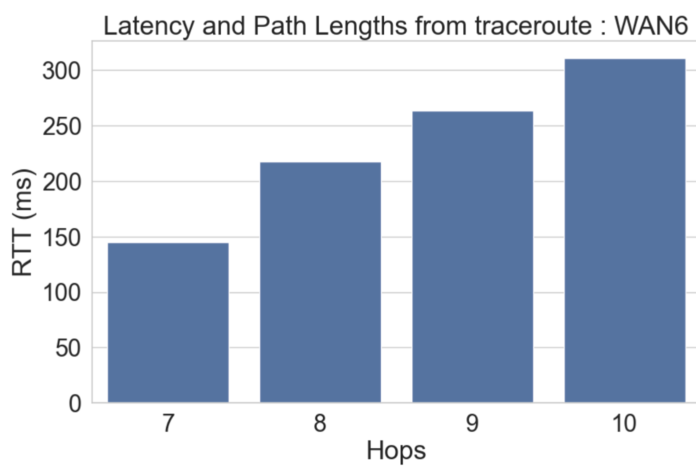






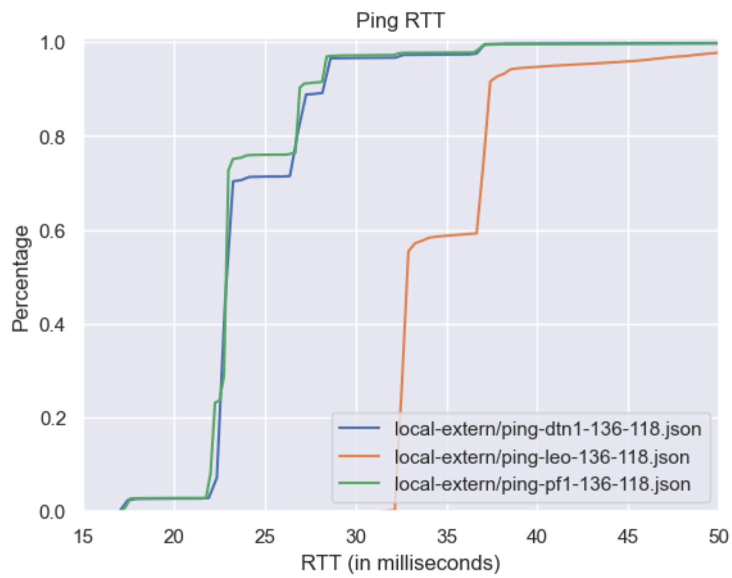


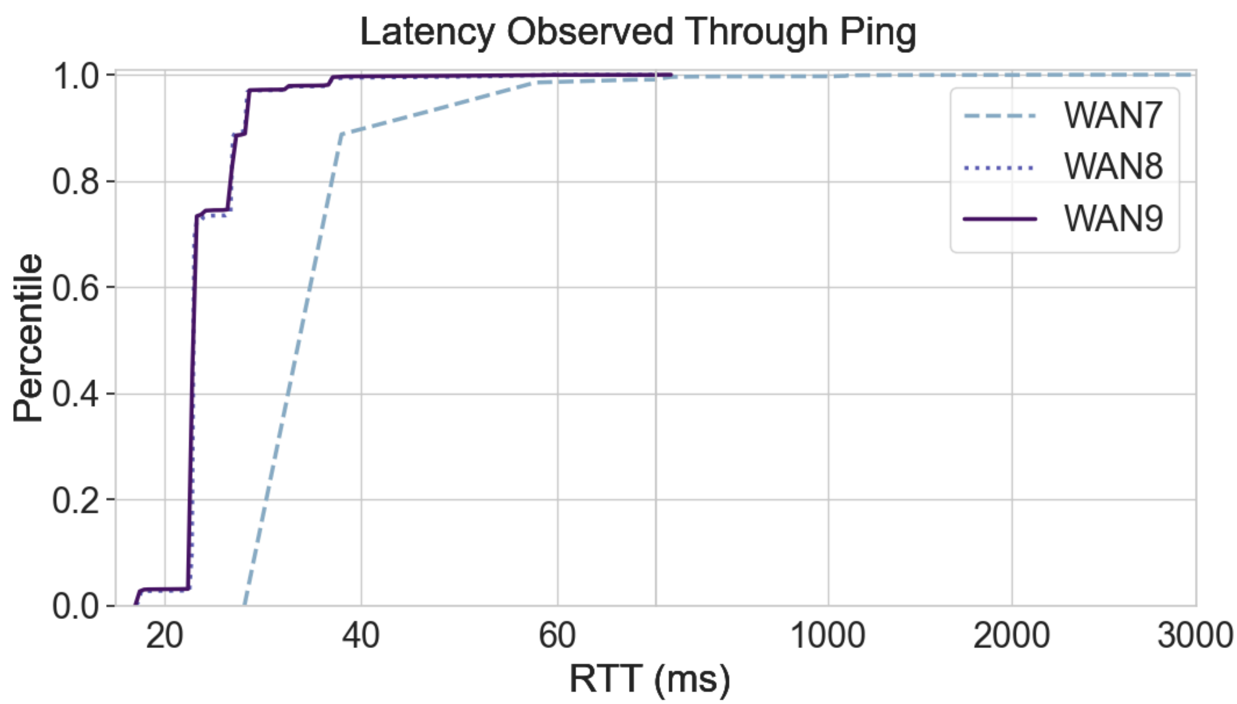




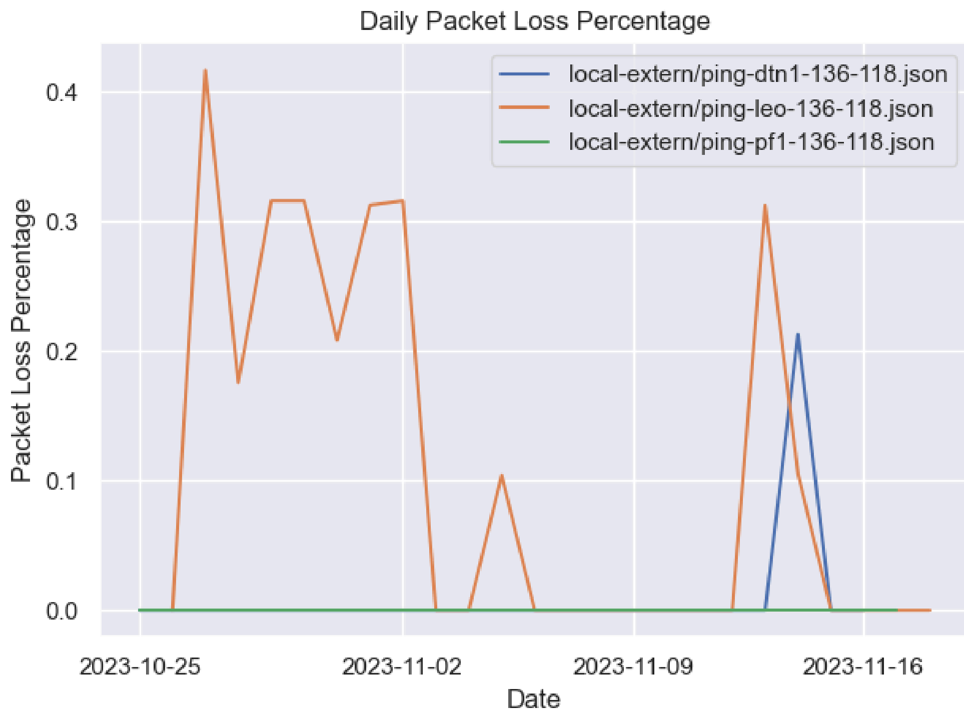
REVIEW Analysis: When traffic is inbound from Google Cloud, the campus network experiences a route path that is one hop shorter than the route paths that the DMZ nodes take. But the DMZ exhibits lower latency at all route lengths in common with the campus network by ~3% - 6.78%.

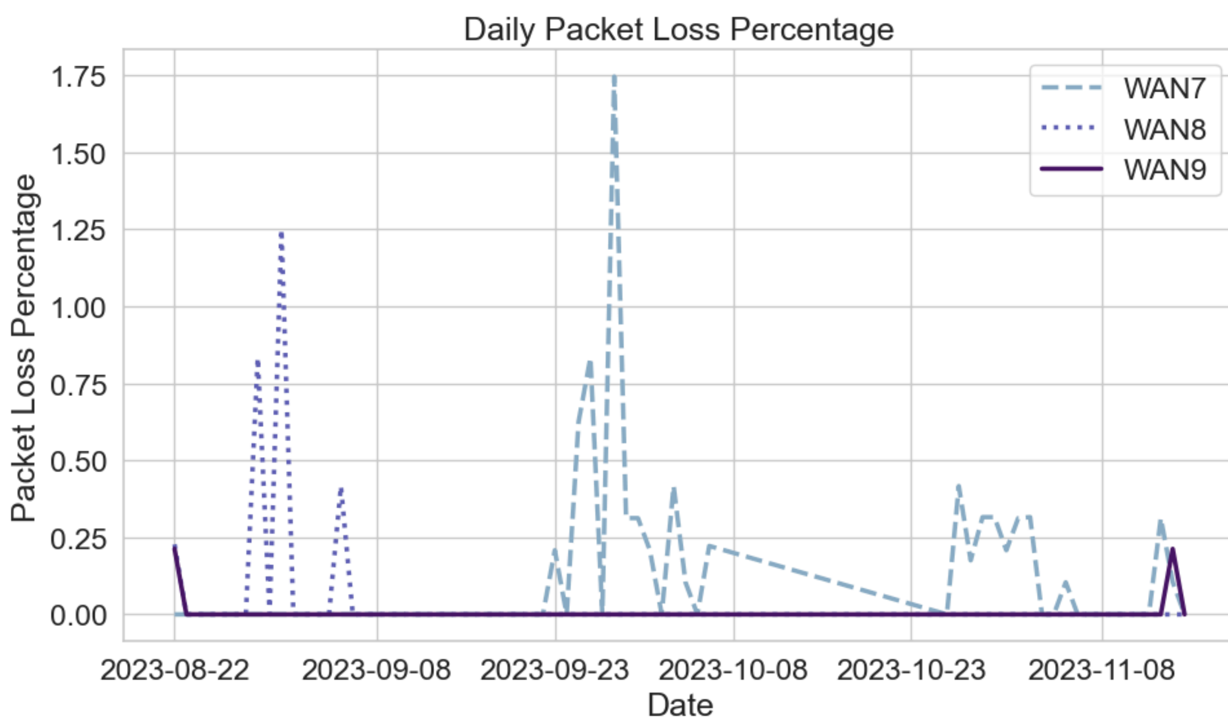
Local Network - External





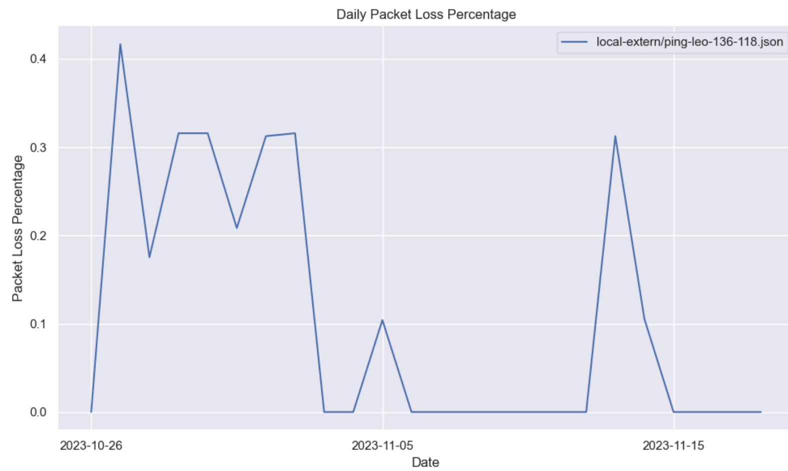
Analysis: When traffic is outbound to perfSONAR nodes, both nodes on the DMZ exhibit similar latency while the campus network experiences latency that is 30.43% - 83% slower.

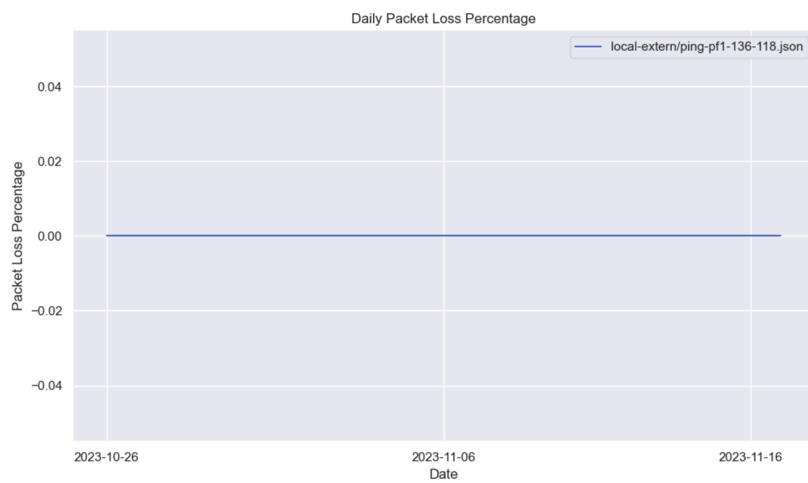


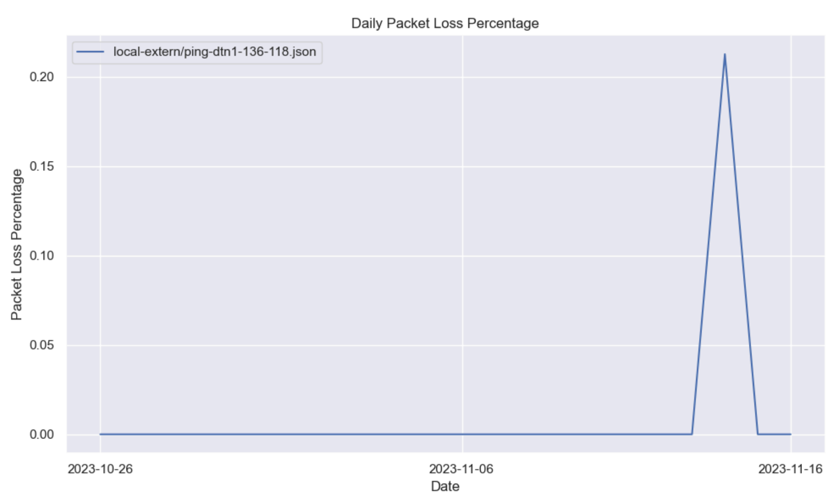


Analysis: When traffic is outbound to external perfSONAR nodes, the campus network experiences more packet loss than the DMZ as a whole, but the Perfsonar1 node experiences more packet loss than the DTN1 node. Perfsonar1 exhibits ~2% more packet loss than DTN1.

The campus network exhibits ~4.2% more packet loss than Perfsonar1 and ~5.7% more packet loss than DTN1.

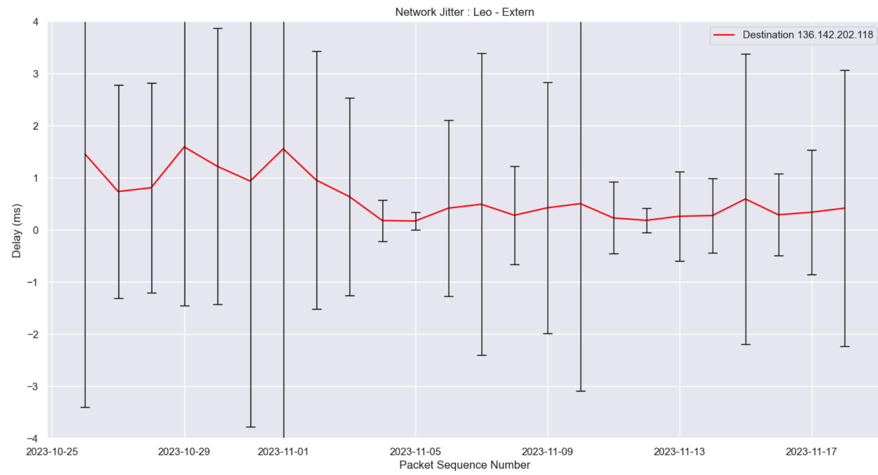


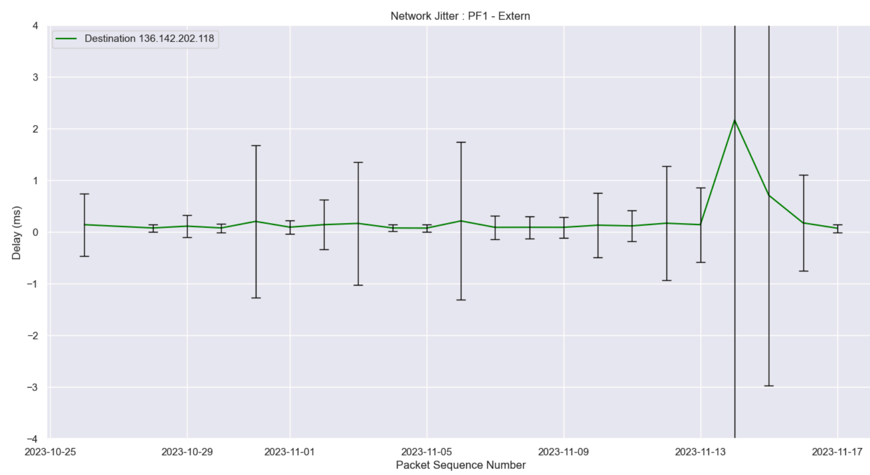


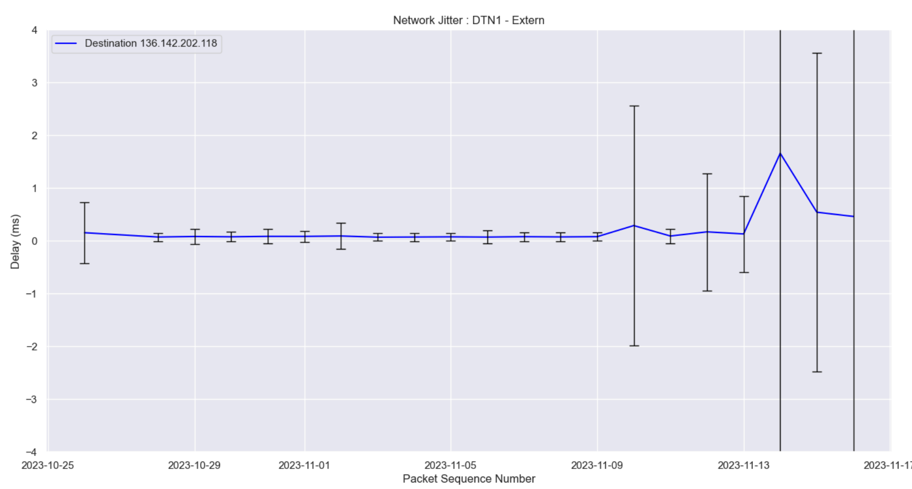


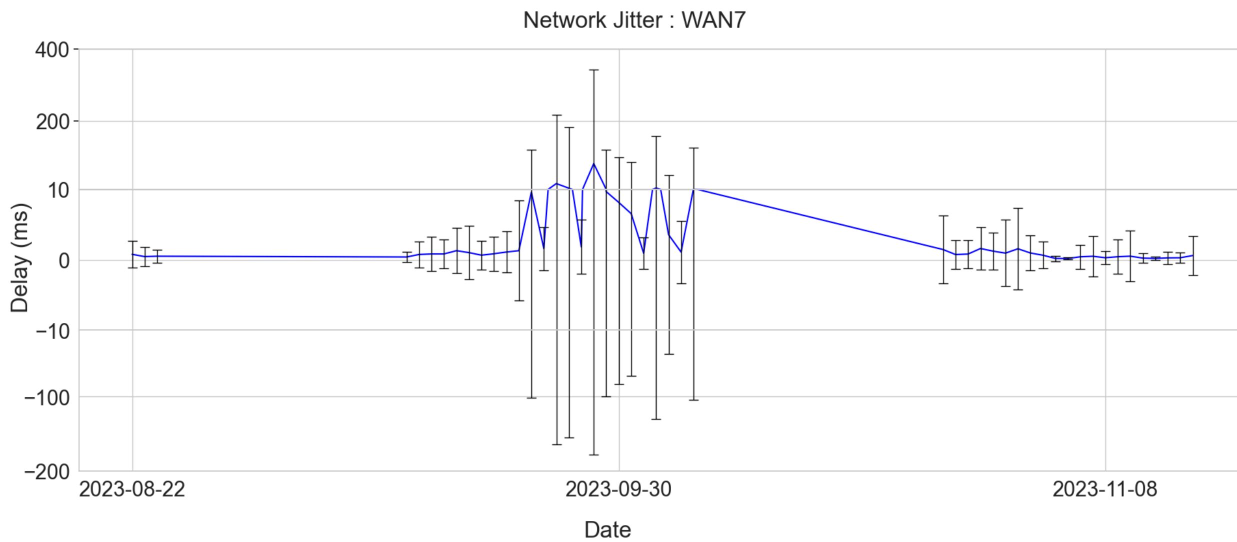
Analysis: During outbound WAN traffic, the campus network has more occurrences of packet loss than the DMZ.

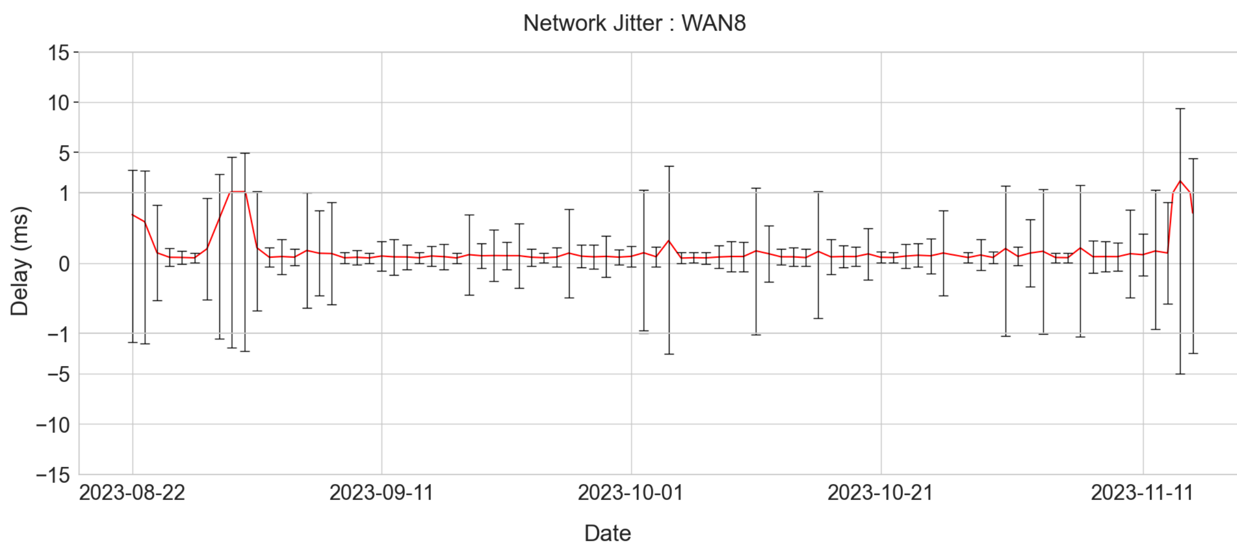
The only record of packet loss on the DMZ route is ~0.20% less than the highest amount of packet loss on the campus network route.

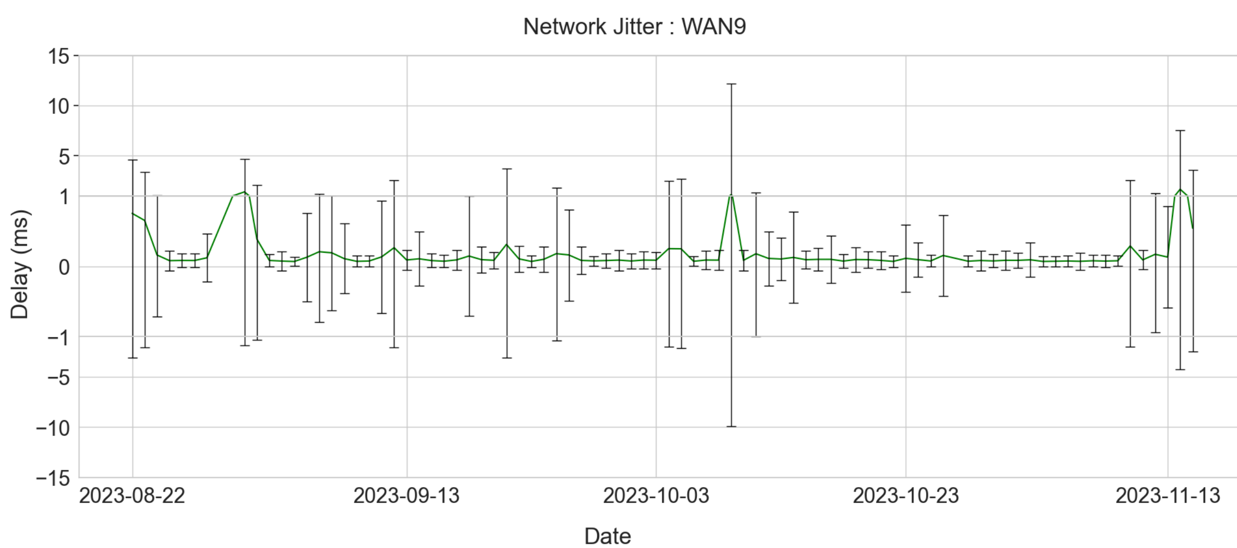




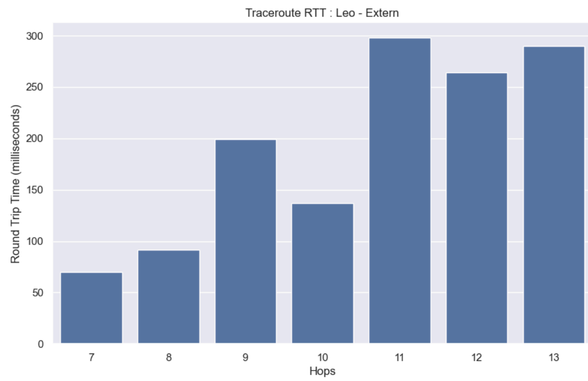


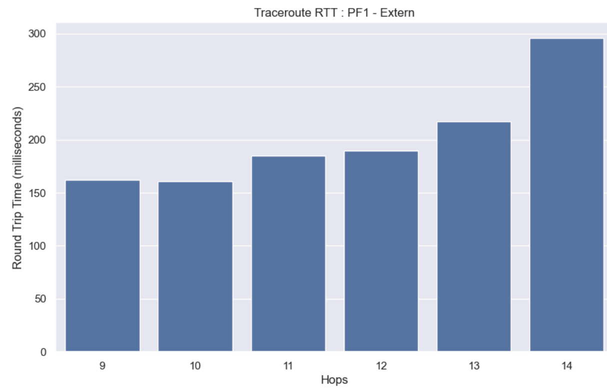

Network Jitter : WAN7
400
200
10
0
−10
−100
−200
Delay (ms)
2023-08-22
2023-09-30
2023-11-08
Date

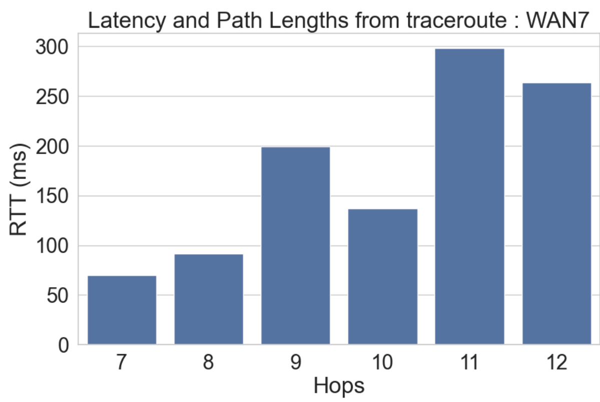

Network Jitter : WAN8
15
10
5
1
0
-1
-5
-10
−15
Delay (ms)
2023-08-22
2023-09-11
2023-10-01
2023-10-21
2023-11-11
Date

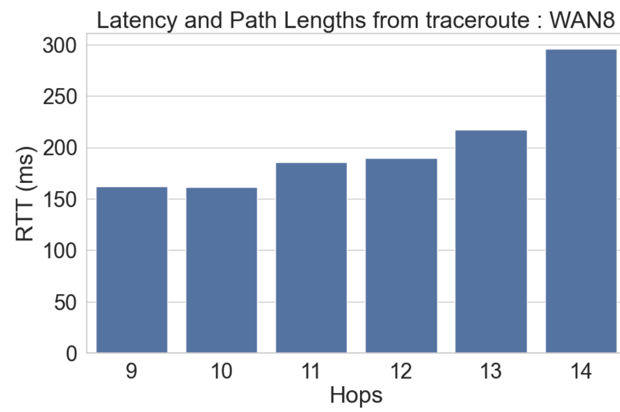

Network Jitter : WAN9
15
10
5
1
0
-1
-5
−10
−15
Delay (ms)
2023-08-22
2023-09-13
2023-10-03
2023-10-23
2023-11-13
Date

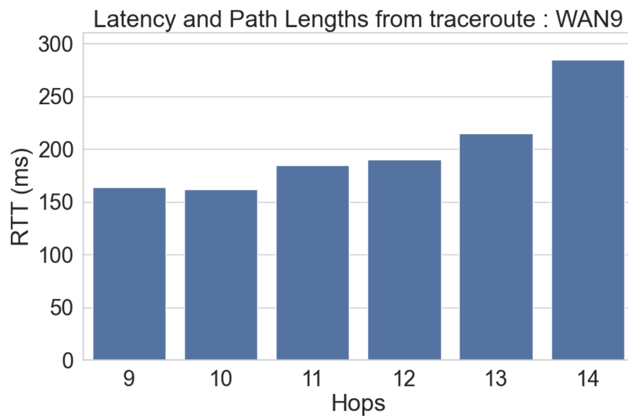
Analysis: With outbound WAN traffic, the campus network typically exhibits higher jitter with more deviation than the DMZ











Analysis: When traffic is outbound to external perfSONAR nodes, the campus network experiences routes with ranges 1-2 hops shorter than DMZ routes. But, the DMZ tends to have a latency 20% - 36.7% lower than the campus network (gathered from common path lengths observed through traceroute). However, there is a point when the campus network performs faster than the DMZ by 12.5% at 10 hops. The DTN1 node on the DMZ has a latency, at the longest path length, that is ~6.75% lower than that of the Perfsonar1 node on the DMZ.

For: January 15 - 21 LATEST UPDATES		*Last updated: Jan 18, 2024*
Status	Description	Comments
Last completed *Due date:*	63. Analyzed existing graphs 64.	
Working on currently/next	Currently working on : experimenting with cloud host; graph comparison models; iperf3 tests; regraph RIPE data Next to work on: finish testing tools Next milestones: Problems: iperf dtn1 (server) - dtn2 (client), dtn1 (server) - pf1 (client) still not working	

Plan & Timeline

Jan 15 - 21	Test iperf3 connections; revisit measurements scheduled for break; analyze existing graphs
Jan 22 - 28	Analyze existing graphs, add to paper
Jan 29 - Feb 4	Revisit data transfer experiments; combine tcpdump files

Possible conferences to consider:

- IMC 2024 - Madrid - Paper Deadline May 8
- TCM 2024 - Dresden - Paper Deadline March 8
- ICCCN 2024 - Hawaii - Paper Deadline March 1

Experiments still remaining:

- Gateway measurements (complete)
- BGP measurements (complete)
- IPerf measurements (in progress)
- Application Performance

Take note of takeaways for the paper

- The presence of a DMZ does not guarantee better performance.
- WAN : While the DMZ may have better end-to-end latency (ping results), the campus network might demonstrate more efficient routing paths (traceroute results). The DMZ may be optimized for shorter distances while the campus network's routing efficiency is more pronounced over longer distances.

For: February 5 - 11 LATEST UPDATES		*Last updated: February 8, 2024*
Status	Description	Comments
Last completed *Due date:*	65. IPerf tests set up 66. Gateway tests set up 67. Regraphed RIPE Atlas Packet Loss	
Working on currently/next	Currently working on : graph comparison models Next to work on: finish testing tools Next milestones: Problems:	

Plan & Timeline

Feb 5 - 11	Set up iperf3 tests, set up gateway tests, reevaluate/quantify results, register paper
Feb 12 - 18	Reevaluate/quantify results, write test setup and results section
Feb 19 - 25	Graph iperf3 throughput, graph gateway data, graph data transfer data

IPerf tests: use –json or convert alternatively?

Why does campus network have lower latency than DMZ when using RIPE Atlas?
Observations:
- Between Aug 22, 2023 and Sept 1, 2023, PF1 and DTN1 are both losing 50% packets with traceroute. Only 2/4 probes send signals that reach the final destination, why?
- Leo measurements use the exact same probes as PF1 and DTN1 but have 0% packet loss during that timeframe.
- The measurements that get through to PF1 and DTN1 have higher latency, the ones that don’t get through have lower latency on Leo, so there are no lower latency measurements to bring down the average latency on PF1 and DTN1.
- Leo was down from Aug 25, 2023 to Sept 12, 2023, so it has less comparison points?

Points for 02/13 Meeting:
- Data Transfer Experiments with updated URLs do provide more insight using Wireshark, but each measurement file is over 2GB in size, is this an issue?
- 1:50-55 AM, 7:50-55AM, 1:50-55PM, 7:50-55PM Aug 22, 2023-Sept 1, 2023, find hangups for RIPE Atlas probes.
- Traceroute from RIPE to PF1 on October 1, 2023 at 1:53 AM records as a two hop route, but the two hops are * * * and PF1 address. Happens to DTN1 on September 27, 2023

at 7:53 PM. This does not seem right. Probe continuously switches between 2 hop route and 13 hop route, then produces no routes, then later returns with very high latency routes.

TCPDUMP 5 sets of captures and then analysis. Interarrival Interpacket Delay, RTT, Loss and Jitter, Total Download Time, Number of Retransmissions (flax), Window Size

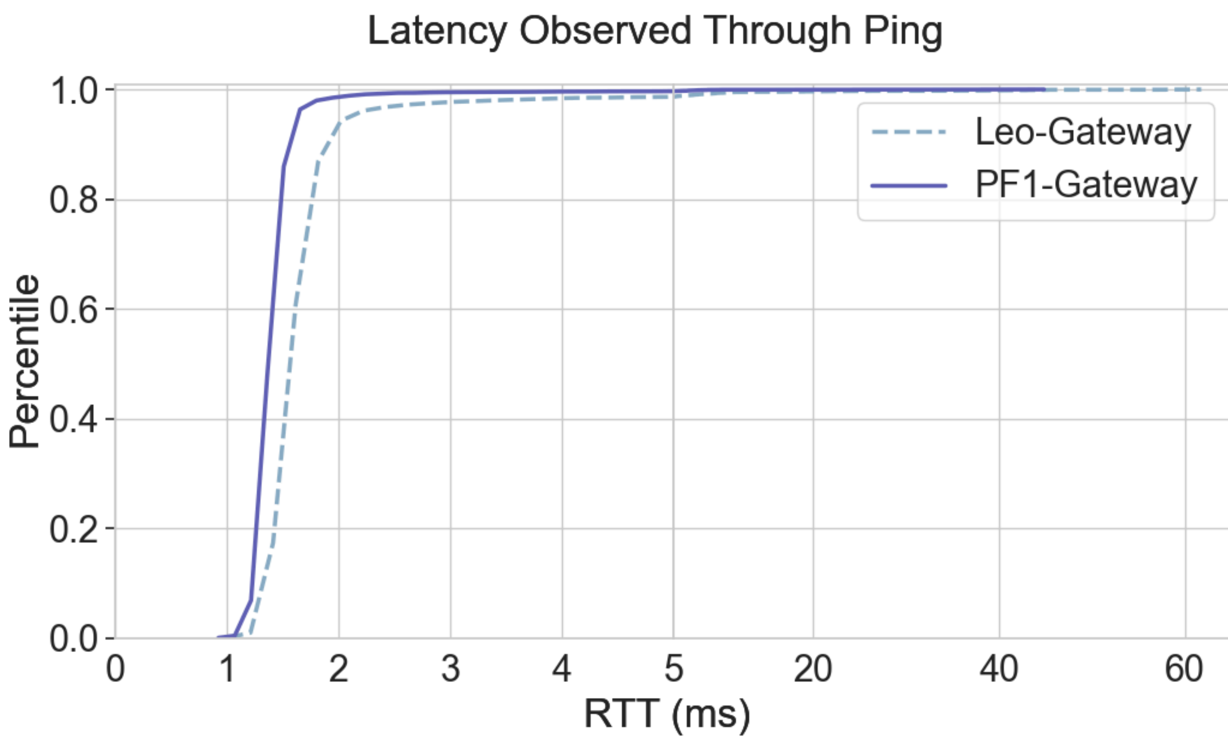
Data Transfer Observations:
Leo:

- 41 retransmissions / 5,611,917 packets
- 10172 packets lost / 5,611,917 packets

DTN1:

- 13 retransmissions / 242,261 packets
- 1 packet lost / 242,261 packets

Gateway Latency:



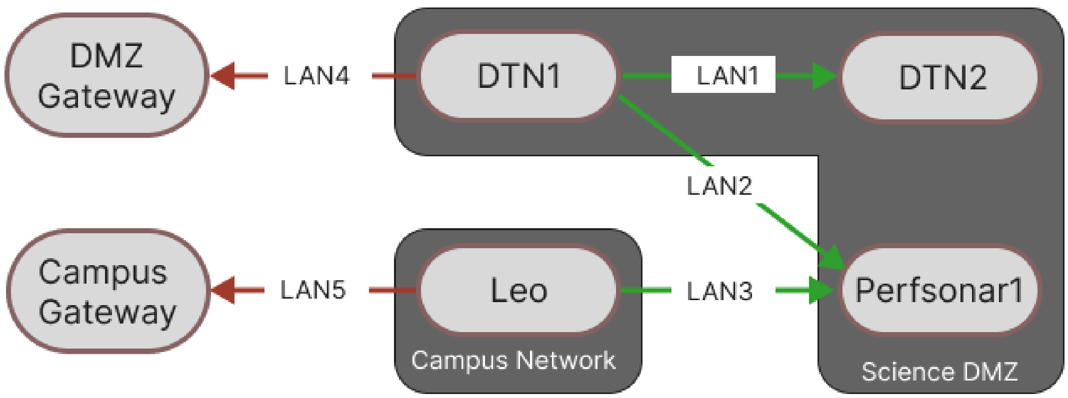
For: March 18 - 24 LATEST UPDATES		*Last updated: March 20, 2024*
Status	Description	Comments
Last completed *Due date:*	68.	
Working on currently/next	Currently working on : Next to work on: Next milestones: Problems:	

Plan & Timeline

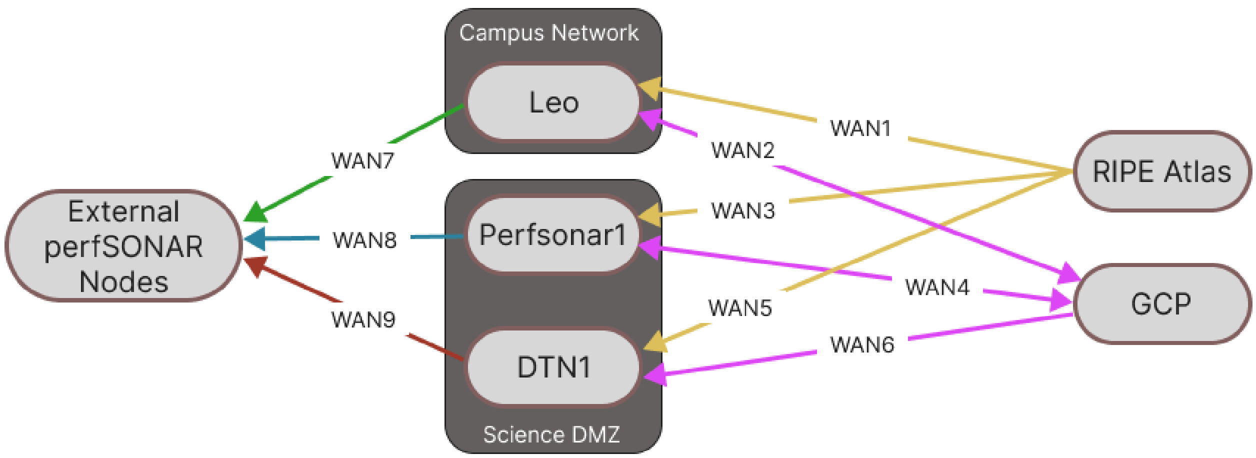
Mar 18 - 24	Find backup venue; clean up lab notebook; set up video experiments; upload remaining scripts to github
Mar 25 - 31	
Apr 1 - 7	

Backup venues:

- AMC IMC 2024 - May 15th submission date
- IEEE HPCC 2024 - August 15th submission date
- IEEE eScience 2024 ? - May 6th submission date



Experiments Run on the Local Area Network:
- Ping Tests are run on LAN1-LAN5
- Traceroute Tests are run on LAN1-LAN3 (but results were not used in first draft of paper)



Experiments Run on the Wide Area Network:
- Ping Tests are run on WAN1-WAN9
- Traceroute Tests are run on WAN1-WAN9
- Iperf3 tests are run on WAN2 and WAN4
- Data Transfer Tests are run on Leo and DTN1

Points to Note from Review of Paper Submission:

Review #3A
===
===

Paper summary

The paper presents results of a longitudinal study comparing the performance of a scientific DMZ (i.e., a network dedicated to research) and a regular campus network. Measurements were performed at a low pace during two years from various vantage points located in the US. Classic metrics are discussed.

Strengths

- longitudinal study
- careful analysis

Weaknesses

- motivations behind the work are lacking
- measurements vantage points all located in the US (lack of diversity)
- results show high level trends

Comments for authors

Although authors performed a longitudinal study (which is always welcomed and appreciated), I don't see the interest and actual motivations behind measuring basic metrics (delay, jitter, throughput) for "science DMZ". Why is it important to measure that?

External measurement nodes (PerfSonar and RIPE Atlas) are all located in the US. Why not consider more widespread machines? Scientific data exchanges are not limited to the US.

Performing measurements at a low pace is done for obvious security and traffic reasons (I get that point) but, in the end, averaging measurements results on a per day basis provides only a high level overview of the network performance. Bursts and particular situations are not necessarily highlighted. BTW, the mean is, by default, sensitive to noise data. It would thus have been better to accompany the mean with noise/error measurements, such as the confidence interval.

"DMZ exhibits lower latency at all route lengths in common with the campus network by ~3% - 6.78%" => this corresponds to a latency reduction between 9 and 18ms. I'm not convinced this demonstrates strong incentives in deploying a dedicated science DMZ (confirmed by the takeaways of Sec. IV.A).

Sec. IV.B: "Internally, we find the latency between the campus and DMZ nodes to be very low. However, given that the path length is minimal, the effect of the firewall is really pronounced here." => It would have been interesting to indeed confirm the responsibility of the firewall (through middlebox measurements, such as tracebox). The same applies for the packet loss measurements.

It is reasonable to think that traffic has not been stopped on both scientific DMZ and campus network during the measurements. What is the impact of regular traffic on the measurement results?

Minor comments:
- Introduction would benefit from more references to base statements such as producing petabytes of data within a short amount of time or data transfer to various scientific facilities.
- Line plots (e.g., Fig. 5 and Fig. 6) are too small and quite hard to read.

* *

Review #3B
===
===

Paper summary

The authors compared various network metrics related to their campus network and their science DMZ network. They ran many experiments and show their delay, throughput, path lengths, packet loss, jitter, and other metrics.

Strengths
----
* A study case on the performance of their Science DMZ vs Campus Internet, which is not widely exploited in the literature

Weaknesses
----
* Experiment design seem rather arbitrary and not fully justified or following performance evaluation conventions
 * Results presented in a way that is hard to assess the network performance

Comments for authors

Thanks very much for submitting this paper, it really fills a void in the literature, even if it is a use case of a single Science DMZ and its associated campus.

I was intrigued by the introduction and its motivations, but I think this paper can be further improved to make it more robust and a greater contribution to the community.

How to improve the paper

1 . W.r.t. Experiments:

* First and foremost, see this paper:
https://conferences.sigcomm.org/imc/2004/papers/p263-paxson.pdf

* Please design your experiments following RFC2544. It's classic for performance evaluation -- they take into account multiple parameters that were not evaluated here -- like packet size, and so on

* Please increase coverage of your vantage points: why are you using only 5 Atlas probes? Why not one from a different AS?

* Why do you only use Google Cloud ? Why not any other network or university?

* Why do you evaluate only a download of a linux distro? Where is it hosted? How does that impact the results?

There are so many parameters that can influence your results that the way is presented right now it makes it harder to generalize

* Which OSes do you run on Leo, Perfsonar1 and DTN1? How does the OS pre-config settings for TCP impact your results? Which distro do you use?

2. W.r.t analysis

* For latency, instead of reporting only averages, also plot quantiles (1st and 3rd), and median
* Also 99%ile is sometimes nice
* Why do you plot the number of hops and latency? Why does that even matter for a DMZ? I mean, the goal is to get somewhere faster

* Please use `hyperref`in your paper to make your links clickable, it is easier for the reader.

* `On the contrary, scientific data is often at`: -> "In contrast, scientific data is often at`

* Fix your atlas ref [20]:

```
` ``
@misc{ripeatlas,
  title={{RIPE Atlas}},
  author={{RIPE Network Coordination Centre}},
  year={2024},
  HOWPUBLISHED = {\url{https://atlas.ripe.net}}
}

@Article{ripeatlas2015ipj,
  author  = "{RIPE NCC Staff}",
  title   = "{RIPE Atlas: A Global Internet Measurement Network}",
  journal = "{Internet Protocol Journal (IPJ)}",
  year    = "2015",
  volume  = "18",
  number  = "3",
  pages   = "2--26",
  month   = "Sep",
}
```

``
* can you make fig1 and 2 vectorial?

* *

Review #3C
===
===

Paper summary

The paper studies Science DMZ networks, and compares their behavior to general-purpose campus networks through a two-year quantitative network measurement study. The study found that Science DMZs exhibit lower latency, higher throughput, and reduced jitter compared to general-purpose networks. Despite some non-intuitive results (e.g., longer routes to external destinations), Science DMZs offer benefits for researchers for carefully-tuned use cases. The paper emphasizes the importance of tailored network infrastructure for scientific data transfers and highlights the value of Science DMZs in academic and research communities.

Strengths

+ The paper presents a two-year quantitative network measurement study comparing Science DMZs to general-purpose networks

+ By focusing on the practical implications of Science DMZs for scientific applications, the paper addresses a critical need in the research community for optimized data transfer environments

+ The study uncovers nuanced results, such as lower latency and higher throughput in Science DMZs, while also highlighting potential limitations and the need for tailored network tuning

+ The paper is well written with clear takeaways.

Weaknesses

- While the paper compares Science DMZs to general-purpose networks, the study could take the findings one step further by associating the results to security policies and enforcement mechanisms used in science DMZs. Note that the cost of security policies and enforcement mechanisms is unclear today.

- Could you comment on validation here? For example, did you reach out to the network operators and run some of your findings by them?

Comments for authors

Thank you for submitting this paper to TMA. I enjoyed reading this paper. In addition to addressing the weaknesses above, the paper could be further strengthened by suggesting clear avenues for future research, e.g., exploring the scalability of Science DMZs or conducting comparative studies across different types of research networks.

* *

Review #3D
==
===

Paper summary

The paper compares and contrasts the performance of networked applications in a science DMZ to that in a general-purpose network.

Strengths

- Analyzes the performance in Science DMZs, which seem quite commonly used for research.

Weaknesses

- Experiments and analyses have many technical issues.

Comments for authors

Thank you for submitting your work to TMA!

Given the paper's clear definition of what a science DMZ constitutes—"an [network] environment mostly free from competing traffic flows and complex security middleware ... that often impede data transfer performance"—the key findings that a DMZ exhibits lower latency, higher throughput, and lower jitter (compared to a general-purpose network) are hardly surprising.

The claim that there are more than 200 DMZs in the US alone cites a search query on the NSF portal as evidence, but not all of the 233 results returned are proposing the construction of a new DMZ. Even if that was the case, it is not proof of the claim of 200 DMZs being deployed or operated in the US. Many awards in the search results probably propose different types of analyses or running different experiments on DMZs.

The argument in the background about short flows finding “moderate bandwidth, latency, and loss rates” sufficient seems misleading, if not technically incorrect. Short flows benefit, for instance, substantially from small latencies than from large bandwidths. Besides, what does “moderate” refer to here? What additional assumptions are you making when you claim that “moderate losses” are acceptable?

The analyses presented in section IV (Results) are, unfortunately, quite shallow and simplistic.
- Figures 3a and 3d do not include measurements to DTN1. Even if we set that issue aside, what is the value of looking at the number of hops? Since you ran traceroute (and not *paris-traceroute*), the number of hops revealed could be a misleading measure! The bizarre bi-modal RTT distribution as a function of hops probably is suggestive of measurement issues.
- At what level are you measuring the hops—AS-level or IP-level?
- The networks being compared are different—by design or construction. The latencies, hence, will differ. In this context, the results presented in Fig. 3 appear to contribute little to no

information.
- Comparing paths from a vantage point to RIPE Atlas and GCP nodes is quite misleading. What RIPE Atlas nodes did you select, and (since their locations are typically known with reasonable accuracy) how far were they compared to the vantage points? Were the GCP nodes selected deployed at comparable distances? None of these are explained in the paper.
- Large hop counts do not necessarily mean high latencies. Your own measurements highlight that.
- DMZs, I would assume, are configured to optimize performance of applications running within the DMZ network. With that reasoning most of the latency measurements look unsurprising.

Link to Documentation

https://docs.google.com/document/d/1zztw-dIrwzQPed-zD7ClNxXr2kL4wC65d6EWpfDr2ZY/edit?usp=sharing