Python Programmability and Event Manager SE Remote Demo Script

Table of Contents

Introduction

This is a step-by-step guide on how to perform the Python Programmability and Event Manager SE Remote Demo, what to show and what to say while doing it.

Assumptions are made that the reader is already familiar with the demo set-up (please refer to "Python Programmability and Event Manager SE Remote Demo - Platform Information" document for further details).

CAUTION

It is imperative that under **no circumstances** should you modify any of the configurations applied on any element of the platform. Indeed, as this demo is open to any System Engineer that requests access, we must preserve the present configuration so that all users will be able to use the demonstration script as written.

Release Information

This first release of the Python Programmability and Event Manager SE Remote Demo is based on:

- AOS 7.3.4.450.R01 GA

Equipment Requirements

1. Your PC
2. RDP configuration file and credentials – emailed when demo is booked and approved.
3. A good, un-firewalled internet connection.

Demo Overview

The demo flow is summarised as follows:

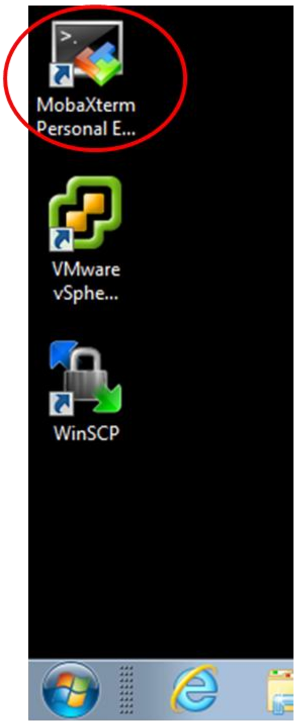
1. Connect to the E-Demo environment
2. Review configuration and statistics
3. Simulate link failure and observe automated response
4. Key Takeaways

Step 1: Connect to the E-Demo Environment

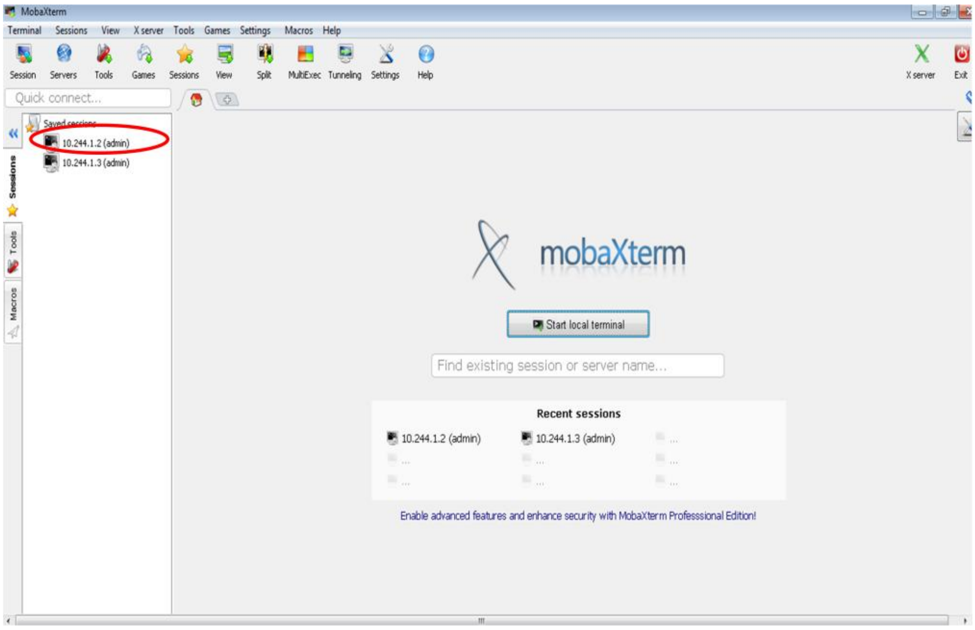
Refer to separate document: “Python Programmability and Event Manager SE Remote Demo – Platform Information”

Step 2: Review Configuration and Statistics

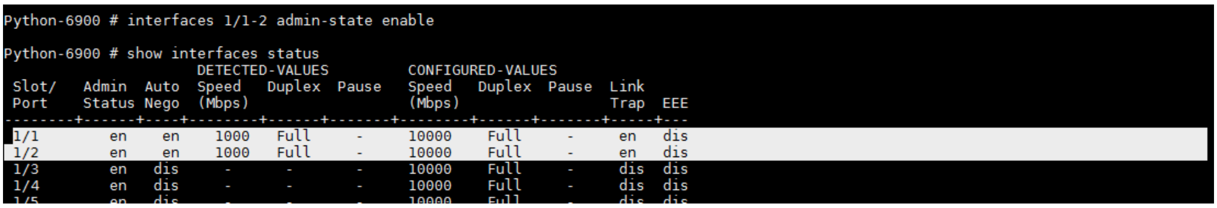
- ✓ Double click on the “MobaXTerm” shortcut on the desktop to launch the Moba XTerm application.



- ✓ On the left hand side panel, double click on the 10.244.1.2 (admin) terminal icon to launch an ssh session to the 6900 switch.


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Settings Help
X server Exit
Quick connect...
Saved sessions
10.244.1.2 (admin)
10.244.1.3 (admin)
Sessions Tools Macros
mobaXterm
Start local terminal
Find existing session or server name...
Recent sessions
10.244.1.2 (admin)
10.244.1.3 (admin)
Enable advanced features and enhance security with MobaXterm Professsional Edition!

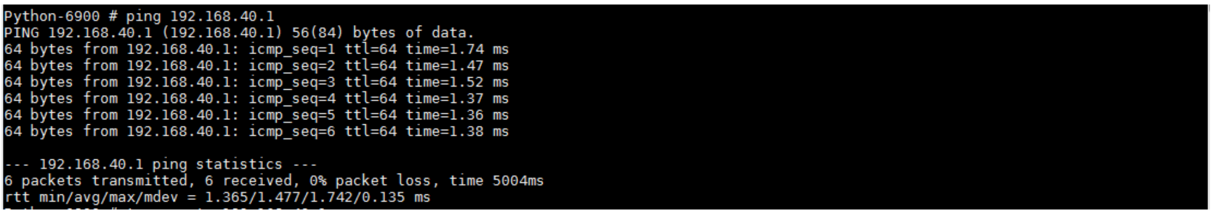
- ✓ Once connected to the 6900, make sure that both interfaces towards the 6400 switch are enabled.



```
Python-6900 # interfaces 1/1-2 admin-state enable

Python-6900 # show interfaces status
                      DETECTED-VALUES         CONFIGURED-VALUES
 Slot/   Admin  Auto  Speed   Duplex  Pause   Speed   Duplex  Pause  Link
 Port    Status Nego  (Mbps)                  (Mbps)                 Trap  EEE
--------+------+----+--------+------+-------+--------+------+-------+-----+---
 1/1        en    en    1000   Full     -     10000    Full     -     en   dis
 1/2        en    en    1000   Full     -     10000    Full     -     en   dis
 1/3        en   dis     -      -       -     10000    Full     -     dis  dis
 1/4        en   dis     -      -       -     10000    Full     -     dis  dis
 1/5        en   dis     -      -       -     10000    Full     -     dis  dis
```

- ✓ Ping the 6400's Loopback0 interface, which in this case represents the Internet.



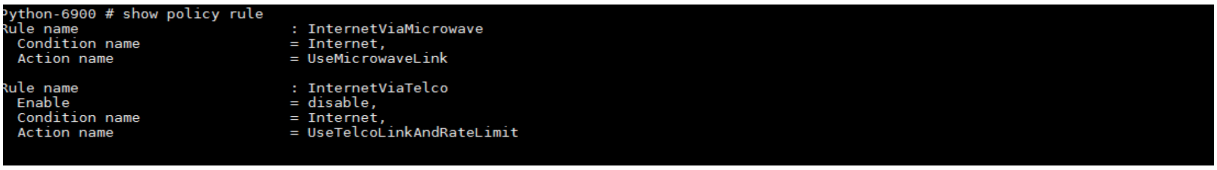
```
Python-6900 # ping 192.168.40.1
PING 192.168.40.1 (192.168.40.1) 56(84) bytes of data.
64 bytes from 192.168.40.1: icmp_seq=1 ttl=64 time=1.74 ms
64 bytes from 192.168.40.1: icmp_seq=2 ttl=64 time=1.47 ms
64 bytes from 192.168.40.1: icmp_seq=3 ttl=64 time=1.52 ms
64 bytes from 192.168.40.1: icmp_seq=4 ttl=64 time=1.37 ms
64 bytes from 192.168.40.1: icmp_seq=5 ttl=64 time=1.36 ms
64 bytes from 192.168.40.1: icmp_seq=6 ttl=64 time=1.38 ms

--- 192.168.40.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5004ms
rtt min/avg/max/mdev = 1.365/1.477/1.742/0.135 ms
```

- ✓ Show QoS policies.

<u>What to say to the audience:</u>

"In the normal situation in which both links are operational and quality is acceptable, a QoS policy is applied such that all Internet traffic will use the Microwave link. The policy that allows usage of the Telco link for Internet traffic is disabled."

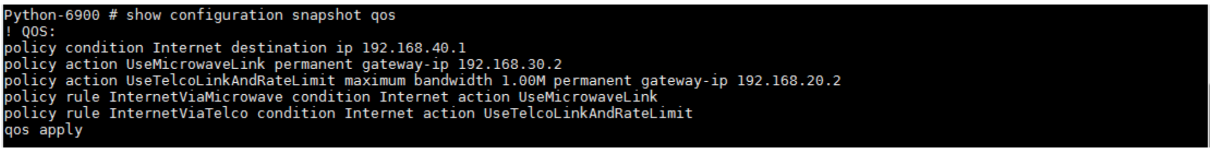


```
Python-6900 # show policy rule
Rule name                         : InternetViaMicrowave
  Condition name                  = Internet,
  Action name                     = UseMicrowaveLink

Rule name                         : InternetViaTelco
  Enable                          = disable,
  Condition name                  = Internet,
  Action name                     = UseTelcoLinkAndRateLimit
```

- ✓ Show QoS policies detail.

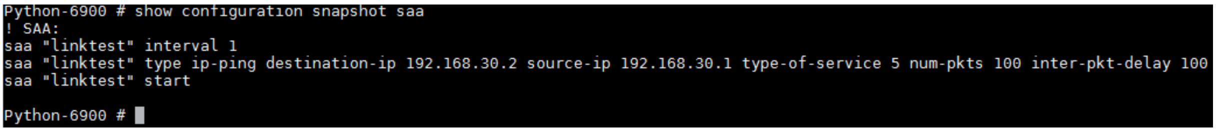
<u>What to say to the audience</u>:
"In the normal situation, all traffic towards the Internet, here represented by a single IP address, is forwarded through the Microwave link. When the Microwave link is failing or the quality is unacceptable, however, Internet traffic will be allowed on the Telco link but will be limited to 1Mbps."



```
Python-6900 # show configuration snapshot qos
! QOS:
policy condition Internet destination ip 192.168.40.1
policy action UseMicrowaveLink permanent gateway-ip 192.168.30.2
policy action UseTelcoLinkAndRateLimit maximum bandwidth 1.00M permanent gateway-ip 192.168.20.2
policy rule InternetViaMicrowave condition Internet action UseMicrowaveLink
policy rule InternetViaTelco condition Internet action UseTelcoLinkAndRateLimit
qos apply
```

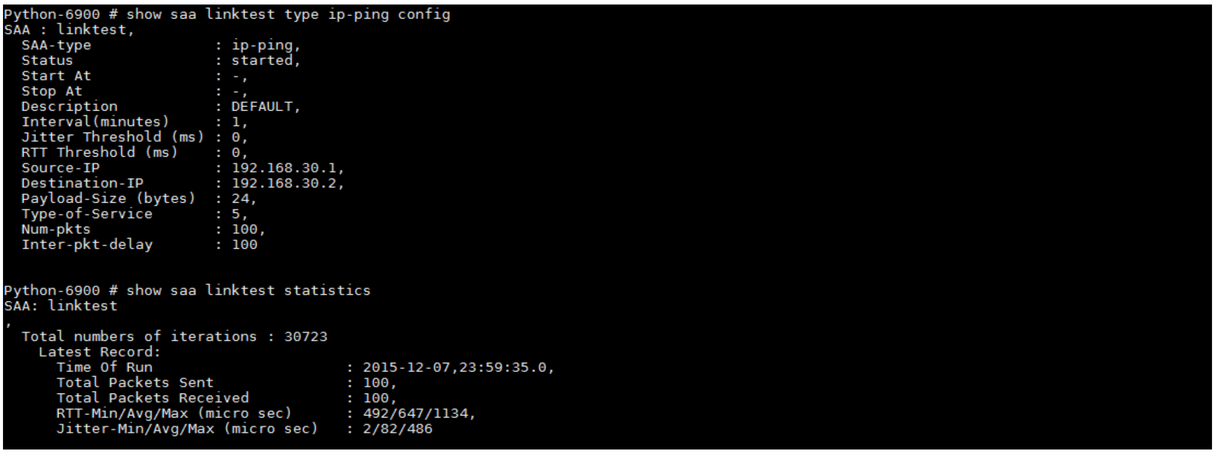
- ✓ Show SAA configuration.

<u>What to say to the audience</u>:
"An SAA ICMP Echo test is run periodically on the Microwave link to assess the link quality. In normal circumstances, packet loss is low or nil and the Microwave link is used. Service Assurance Agent (SAA) enables customers to proactively identify network issues. SAA uses active monitoring to generate traffic in a continuous, reliable, and predictable manner, thus enabling the measurement of network performance and health."



```
Python-6900 # show configuration snapshot saa
! SAA:
saa "linktest" interval 1
saa "linktest" type ip-ping destination-ip 192.168.30.2 source-ip 192.168.30.1 type-of-service 5 num-pkts 100 inter-pkt-delay 100
saa "linktest" start

Python-6900 # █
```

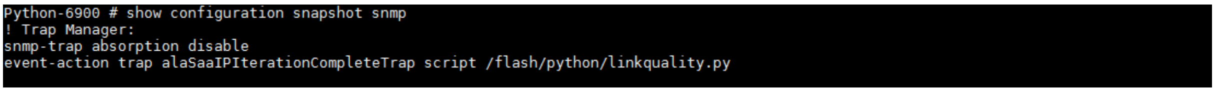


```
Python-6900 # show saa linktest type ip-ping config
SAA : linktest,
  SAA-type              : ip-ping,
  Status                : started,
  Start At              : -,
  Stop At               : -,
  Description           : DEFAULT,
  Interval(minutes)     : 1,
  Jitter Threshold (ms) : 0,
  RTT Threshold (ms)    : 0,
  Source-IP             : 192.168.30.1,
  Destination-IP        : 192.168.30.2,
  Payload-Size (bytes)  : 24,
  Type-of-Service       : 5,
  Num-pkts              : 100,
  Inter-pkt-delay       : 100


Python-6900 # show saa linktest statistics
SAA: linktest
,
  Total numbers of iterations : 30723
    Latest Record:
      Time Of Run                       : 2015-12-07,23:59:35.0,
      Total Packets Sent                : 100,
      Total Packets Received            : 100,
      RTT-Min/Avg/Max (micro sec)       : 492/647/1134,
      Jitter-Min/Avg/Max (micro sec)    : 2/82/486
```

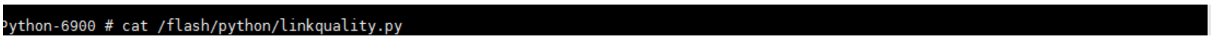
- ✓ Show SNMP configuration.
- ✓

<u>What to say to the audience</u>:
"A custom Python application/script is invoked when the SAA test iteration is complete. The Python script makes use of the SAA test results"



```
Python-6900 # show configuration snapshot snmp
! Trap Manager:
snmp-trap absorption disable
event-action trap alaSaaIPIterationCompleteTrap script /flash/python/linkquality.py
```

- ✓ Show the Python application on the switch by issuing the command below. Do not spend much time showing the script, it will be easier to explain the Flow Chart provided in the Slide Pack.

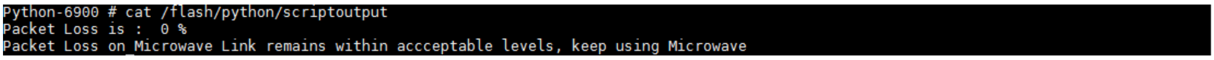


```
Python-6900 # cat /flash/python/linkquality.py
```

<u>What to say to the audience</u>:
"The Python application will look at the ICMP Echo test results. If Packet Loss is greater than 10%, and the Microwave Link is in use, it will disable the policy forcing Internet traffic through this link and will enable a policy that allows usage of the Telco link for Internet traffic but capped at 1Mbps. If Packet Loss is less than 5%, and the Microwave Link is NOT in use, it will undo the changes to force Internet traffic through the Microwave Link again. In any other case, no changes will be made. This is to avoid toggling between the two links".

Note that in a real world scenario one would probably want to implement the same policies and changes at the other end of the links (i.e. at the Central Site) to avoid asymmetrical flows (e.g. outbound traffic using the Microwave Link whilst return traffic uses the Telco link). This would be done leveraging the RestFul API at the other end. In this demo, however, we are not doing that to keep it simple. Only mention this if the question comes up.

The script will not generate output to the console but rather to a file on the switch flash. You can check the last execution output by examining the contents of the output file as shown below.



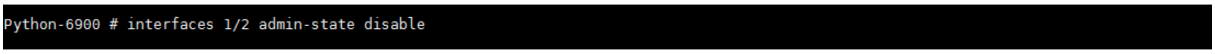
```
Python-6900 # cat /flash/python/scriptoutput
Packet Loss is :  0 %
Packet Loss on Microwave Link remains within accceptable levels, keep using Microwave
```

<u>What to say to the audience</u>:
"As we can see, when the packet loss is low, the Microwave link is used".

Step 3: Simulate Link Degradation and Observe Automatic Response

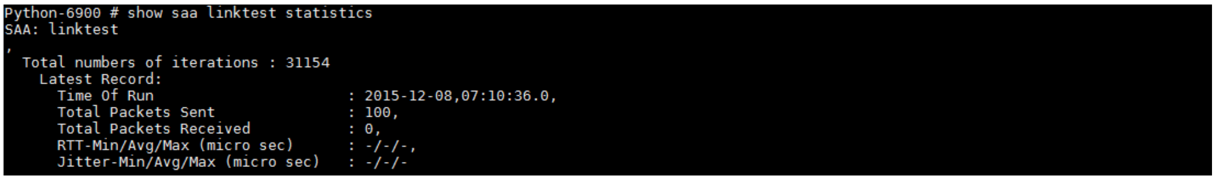
- ✓ From the 6900’s console, issue the command below to disable port 1/2 which represents the Microwave Link.

<u>What to say to the audience</u>:
“We will now disable the interface connected to the Microwave link to simulate degradation (or in this case, a failure) on that link.”



```
Python-6900 # interfaces 1/2 admin-state disable
```

The SAA tests run every minute. So it could take up to 1 minute for this to be detected and action to be taken. You can issue the command below to check packet loss measured in the last run.

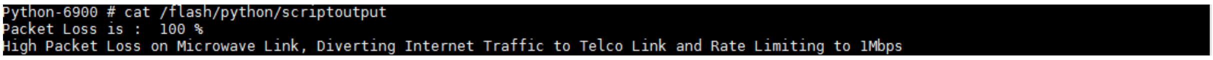


```
Python-6900 # show saa linktest statistics
SAA: linktest
,
  Total numbers of iterations : 31154
    Latest Record:
      Time Of Run                       : 2015-12-08,07:10:36.0,
      Total Packets Sent                : 100,
      Total Packets Received            : 0,
      RTT-Min/Avg/Max (micro sec)       : -/-/-,
      Jitter-Min/Avg/Max (micro sec)    : -/-/-
```

If there’s no packet loss, it means the test has not been executed since you disabled the interface. Wait a few more seconds and re-issue the command above until you see packet loss.

<u>What to say to the audience</u>:
“As we can see now the SAA agent has detected the packet loss on the Microwave link and the Python application will act upon it.”

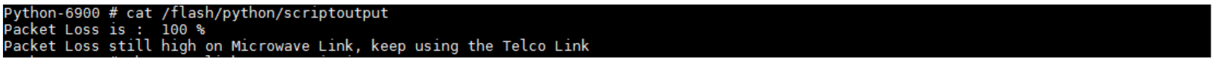
- ✓ Issue the command below to check the script output.



```
Python-6900 # cat /flash/python/scriptoutput
Packet Loss is :  100 %
High Packet Loss on Microwave Link, Diverting Internet Traffic to Telco Link and Rate Limiting to 1Mbps
```

<u>What to say to the audience</u>:
“Due to the high packet loss on the Microwave link, the Python application has automatically diverted Internet traffic to the Telco link whilst at the same time has set a 1Mbps cap on it.”

Note that if you wait more than 1 minute, the test will run again. But because action had already been taken in the previous run, no further changes are required and you will see the output below instead.

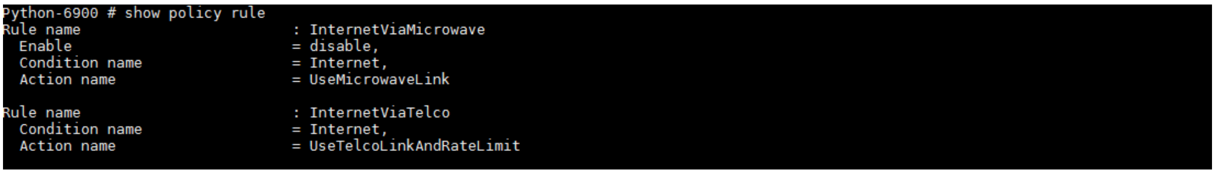


```
Python-6900 # cat /flash/python/scriptoutput
Packet Loss is :  100 %
Packet Loss still high on Microwave Link, keep using the Telco Link
```

In a real world scenario one would probably want to aggregate packet loss results over at least 5 minutes to avoid intermittent packet loss from triggering a policy change. But in this demo we won’t do that because we don’t want to wait 5 minutes.

We will now proceed to verify that the changes have been applied and that traffic continues to flow.

- ✓ Issue the command below to see changes applied to QoS policies



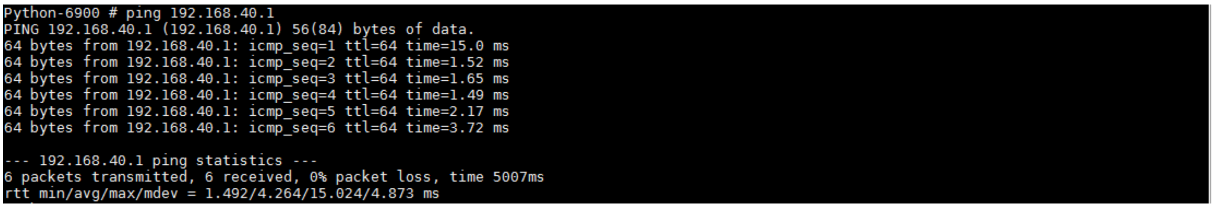
```
Python-6900 # show policy rule
Rule name                          : InternetViaMicrowave
  Enable                           = disable,
  Condition name                   = Internet,
  Action name                      = UseMicrowaveLink

Rule name                          : InternetViaTelco
  Condition name                   = Internet,
  Action name                      = UseTelcoLinkAndRateLimit
```

<u>What to say to the audience</u>:

“The Python app disabled use of the Microwave link due to its high packet loss and allowed use of the Telco link instead.”

- ✓ Verify changes by pinging the Internet IP address



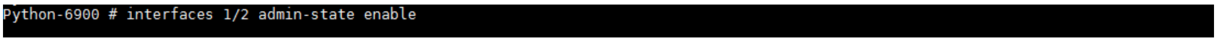
```
Python-6900 # ping 192.168.40.1
PING 192.168.40.1 (192.168.40.1) 56(84) bytes of data.
64 bytes from 192.168.40.1: icmp_seq=1 ttl=64 time=15.0 ms
64 bytes from 192.168.40.1: icmp_seq=2 ttl=64 time=1.52 ms
64 bytes from 192.168.40.1: icmp_seq=3 ttl=64 time=1.65 ms
64 bytes from 192.168.40.1: icmp_seq=4 ttl=64 time=1.49 ms
64 bytes from 192.168.40.1: icmp_seq=5 ttl=64 time=2.17 ms
64 bytes from 192.168.40.1: icmp_seq=6 ttl=64 time=3.72 ms

--- 192.168.40.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 1.492/4.264/15.024/4.873 ms
```

<u>What to say to the audience</u>:

“In this scenario we have shown how Python programmability can be used to automate a very precise response to a situation that is difficult to deal with using standard protocols and features: a link that is not completely down but rather has high packet loss. A further enhancement would be to detect failure of the Telco link and divert Corporate traffic to the Microwave link with higher priority than Internet traffic.”

- ✓ At this point you can re-enable the interface and wait for changes to be automatically reverted.



```
Python-6900 # interfaces 1/2 admin-state enable
```

Step 4: Wrap Up with Key Takeaways

Go back to the presentation and go over the key takeaways to summarize what was shown.

Refer to attached slides and speaker notes.

Contacts

eDemo support team:

edemo-support@al-enterprise.com