CUSTOMIZING NETWORK LINK OUTAGE RESPONSES USING PYTHON SCRIPTS IN OMNISWITCH

APPLICATION NOTE

Alcatel·Lucent Enterprise

INTRODUCTION

Today's enterprise networks provide mission-critical resources for employees, customers and business partners. Networks must be flexible, and the network operations staff must be able to trust that network issues are resolved quickly to minimize business impact.

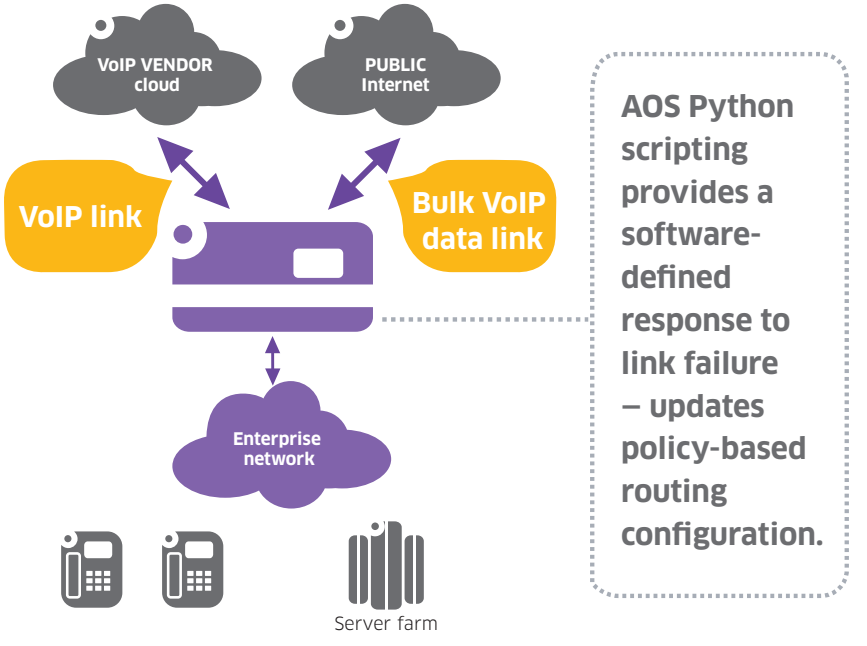
Alcatel-Lucent OmniSwitch® products can be customized to respond adequately to specific network events using the Event Manager of the Alcatel-Lucent Operating System (AOS) and its embedded Python® scripting feature.

Specific network events can trigger Python scripts to provide fast, real-time, software-defined responses tailored to the requirements of the enterprise. For example, the customers can set up different responses for link failures depending on the location and type of a link.

A SOFTWARE-DEFINED RESPONSE TO A LINK FAILURE UPDATES THE POLICY-BASED ROUTING CONFIGURATION

Enterprise networks frequently include multiple primary and backup links with different costs, latencies and bandwidth characteristics. Policy-based routing (PBR) and other configuration mechanisms are used to direct traffic over the appropriate links based on security policies, bandwidth and latency or other technology requirements as defined based on business needs. This approach optimizes the use of network resources and provides the best network performance when all links are available. However, a policy may require changes to the behavior when one or more links fail.

Enterprise networks can use the Event Manager and Python scripting to provide an embedded software-defined response to link failures and other network events. The link failure response can then update the network to minimize disruption with a nearly real-time reaction. The advantage is that the update is not dependent on the availability of a third-party management system to execute the necessary actions.



The enterprise network shown above has two links to the Internet. One link provides high-quality VoIP and video, and the other link delivers high bandwidth for bulk data transfers and network redundancy. The VoIP link is provisioned by the service provider for minimum latency with moderate bandwidth, whilst the main bulk data Internet link is a lower-cost link connection that delivers high bandwidth but without quality-of-service (QoS) guarantees.

When both links are operational, the network configuration directs all VoIP traffic over the VoIP link and all other traffic over the bulk data Internet link.

The AOS Event Manager feature executes a Python script when a link state changes. The Python "link down" script modifies the network configuration appropriately whenever a key link fails.

- When the VoIP link fails, the Python link down script updates the OmniSwitch policy-based routing configuration to direct the VoIP traffic to the bulk link. The link down script also updates the QoS configuration to limit non-VoIP bandwidth, ensuring good VoIP performance.
- When the bulk data Internet link fails, the Python link down script updates the configuration to allow a moderate amount of bulk data to use the VoIP link but without impacting VoIP traffic.

The Python "link up" script restores the normal configuration when the links return to operational status. It also reports the total down time to the network management team by e-mail.

EXAMPLE

This example demonstrates the AOS Event Manager and its Python scripting.

The Event Manager is configured with event-action commands that execute the **pbr_saa_script.py** script when the s**nmp linkDown** or the **snmp linkUp** trap occurs. In this example, the same script handles both cases for simplicity, but two separate scripts could be used.

```
! Trap Manager:
snmp-trap absorption disable
event-action trap linkDown script /flash/python/pbr_saa_script.py
event-action trap linkUp script /flash/python/pbr_saa_script.py
```

AOS passes the trap type and other parameters to the script when a **linkDown** or **linkUp** trap **occurs**. The script then checks the parameters trap and **data[ifIndex]** to determine what happened.

```
if trap == 'linkDown':
  if data2['ifIndex'] == 1012:
print('linkdown on port 1/1/12')
<Python code executed when port 1/12 goes down>
elif trap == 'linkUp':
  if data2['ifIndex'] == 1012:
     <Python code executed when port 1/12 goes up>
print('linkup on port 1/1/12')
```

In this example, a **linkDown** trap occurs when link 1/12 fails. The Python script then enables port 2/1/11 and updates the QoS configuration that implements policy-based routing.

A **linkUp** trap occurs when link 1/12 returns to operational state. The Python script then disables port 2/1/11 and updates the QoS/PBR configuration. The code is summarized here:

```
if trap == 'linkDown':
  if data2['ifIndex'] == 1012:
     print('linkdown on port 1/1/12')
     os.system("interfaces 2/1/11 admin-state enable")
     os.system("qos flush")
     os.system("qos apply")
     os.system("policy condition pbr01 source ip 1.1.1.2")
     os.system("policy action pbr01 permanent gateway-ip 172.16.0.192")
     os.system("policy rule pbr01 condition pbr01 action pbr01")
     os.system("qos apply")
elif trap == 'linkUp':
  if data2['ifIndex'] == 1012:
     print('linkup on port 1/1/12')
     os.system("interfaces 2/1/11 admin-state disable")
     os.system("qos flush")
     os.system("qos apply")
     os.system("policy condition pbr01 source ip 1.1.1.2")
     os.system("policy action pbr01 permanent gateway-ip 172.16.2.11")
     os.system("policy rule pbr01 condition pbr01 action pbr01")
     os.system("qos apply")
```

CONCLUSION

The AOS Event Manager and Python scripting provide a fast, flexible, software-defined response to network events. The response supports and enforces network policies defined by the networking staff. Responses can include enabling or disabling links or changing network priorities. It can also include gathering information to document the failure and notifying network staff by e-mail.

REFERENCES

See “OmniSwitch AOS XXX Switch Management Guide”, chapter “Web Services, CLI Scripting, and OpenFlow”. The document is located at http://enterprise.alcatel-lucent.com/UserGuides. This application note applies to AOS release 7.3.4 or later.

Alcatel·Lucent Enterprise