



# CUSTOMIZING NETWORK LINK AGGREGATION USING PYTHON SCRIPTS IN OMNISWITCH

APPLICATION NOTE

# 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.

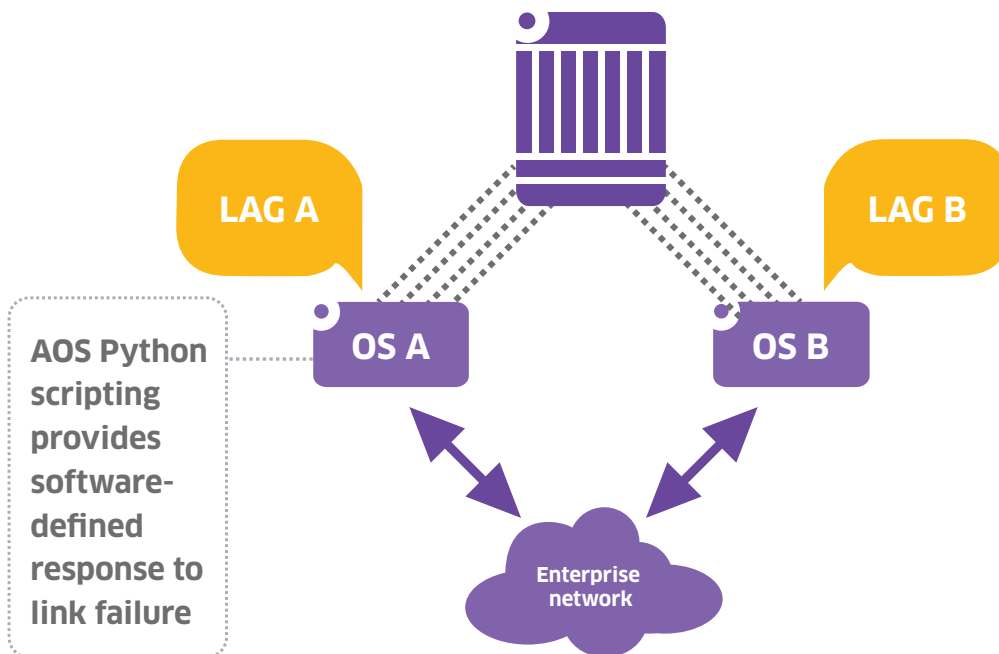
Alcatel-Lucent OmniSwitch® products can respond to network events quickly using the Event Manager of the Alcatel-Lucent Operating System (AOS) and its embedded Python® scripting feature.

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.

This application note demonstrates a combination of OmniSwitch features, including the AOS Event Manager and Python scripting, to provide a software-defined response failure of individual links in a link aggregation group. The software-defined result is customized to match the customer’s network policy needs, following the industry-standard link aggregation protocols. Another note describes how the AOS Event Manager and Python scripting can be used to extend policy-based routing.

## SOFTWARE-DEFINED RESPONSE TO A PARTIAL FAILURE OF A LINK AGGREGATION GROUP

Enterprise networks frequently use multiple link aggregation groups to provide higher bandwidth and increased network resiliency. The AOS Event Manager and Python scripting can provide software-defined responses when a link failure occurs.



The Enterprise network shown above uses two OmniSwitch units, OS A and OS B, and two link aggregation groups, LAG A and LAG B, to provide redundant access to a major router. There are four ports in each LAG identified as ports A1 through A4 and B1 through B4.

In this example, the customer has designed the network so that either LAG can provide sufficient bandwidth for all traffic and only one LAG is active at any time. The customer policy is to consider a LAG failed when half of the links in that LAG have failed.

The customer is concerned about partial failures where several links in a LAG fail, but at least one link remains operational. In that case, the traffic capacity is significantly reduced, but industry-standard link aggregation protocols do not automatically consider the LAG as failed.

The following solution implements a custom policy using the AOS Event Manager and Python scripting, along with AOS Web Services RESTful remote command execution.

The OmniSwitch Event Manager executes a custom Python script in OS A whenever a link fails. The Python script checks the number of remaining links in the LAG A group against a pre-defined limit and disables the entire LAG if the number of remaining links falls below the limit. The second LAG is then enabled remotely by OS A on OS B using the AOS RESTful API. All traffic then flows through the LAG B. This avoids operating the network with a partially-failed LAG and meets the goals set by the customer policy.

## EXAMPLE

This example demonstrates the AOS Event Manager and Python scripting.

The Event Manager is configured with event-action commands that execute the script `linkagg_script.py` when the `snmp linkDown` trap occurs.

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

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

## Pseudocode for linkagg\_script.py

```
If (SNMP trap is linkDown)
  If (the port is a member of LAG 1)
    Set value = the number of operational ports remaining in LAG 1
    If (number of remaining ports is below the customer's threshold)
      Execute local AOS command to Disable LAG 1 on OmniSwitch A
      Execute remote AOS command to Enable LAG 2 on OmniSwitch B
      Send email message to network manager's phone
    Else
      Log the port failed and the number of remaining ports
  Else
    linkDown event is not on a member of LAG 1. Ignore it.
```

## 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.