AlliedWare Plus™ OS

How To **Create Resilient Connections using the Host Attach solution**

Introduction

This How To Note describes how to use ping polling and triggers to achieve a redundant link failover mechanism, known as the Host Attach solution. This solution is useful on networks where you cannot use link aggregation or STP to protect the redundant link from loops.

What information will you find in this document?

This How To Note begins with the following information:

- "Which products and software versions does it apply to?" on page 1

Then it describes the configuration, in the following sections:

- "The typical network scenario" on page 2
- "The Host Attach solution" on page 2
- "Configuring the Host Attach solution on the VCStack" on page 5
- "Extending the configuration to handle link failures between stack members" on page 8
- "Full configuration script" on page 10

Which products and software versions does it apply to?

This configuration applies to AlliedWare Plus software version 5.2.1 and above, for the following Allied Telesis switches:

- SwitchBlade x908
- x900-12XT/S
- x900-24 Series

The typical network scenario

In the following example network, each member of the Virtual Chassis Stack (VCStack^TM) has a link to the Internet Gateway router. However, the router does not support LACP, so you cannot configure the links as an aggregated pair. The router also does not support STP, so you cannot configure STP to protect the redundant link from loops.
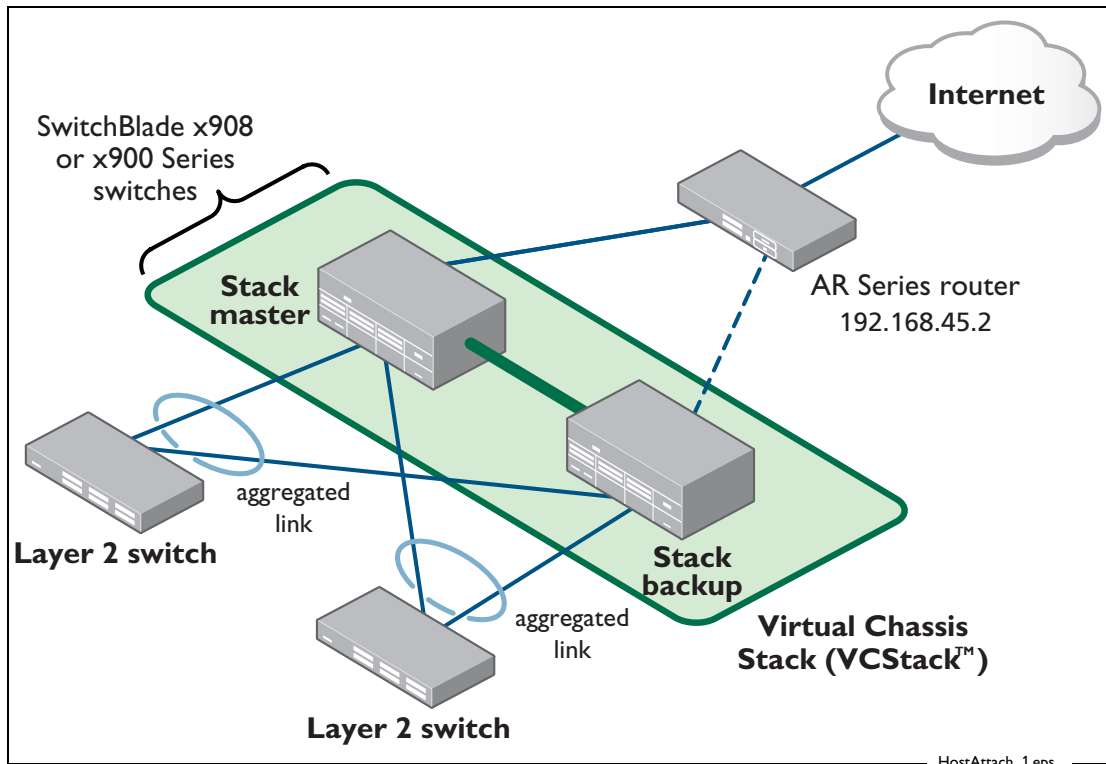

HostAttach_1.eps

The Host Attach solution

The solution is to use ping polling and triggers to achieve a redundant failover mechanism.

Ping polling is a feature that allows you to create a polling instance to periodically ping a connected device. As long as your switch receives ping responses from the device, it considers the device reachable. If your switch does not receive a reply to a set number of ICMP Echo Requests, it considers that the host is unreachable. It continues to try to ping the device, at an increased rate. After it receives a set number of responses, it considers the device to be reachable again.

The trigger facility is a powerful mechanism that allows you to automate the execution of commands in response to certain events. The example network in this How To Note uses the trigger facility to respond to link-up and link-down detection by ping polling.

When the example network is functioning normally, only one port in the VCStack is an active link to the router. This link is in the data VLAN (VLAN 2) over which data flows to the Internet. Ping polling also occurs over the same link using the same VLAN.
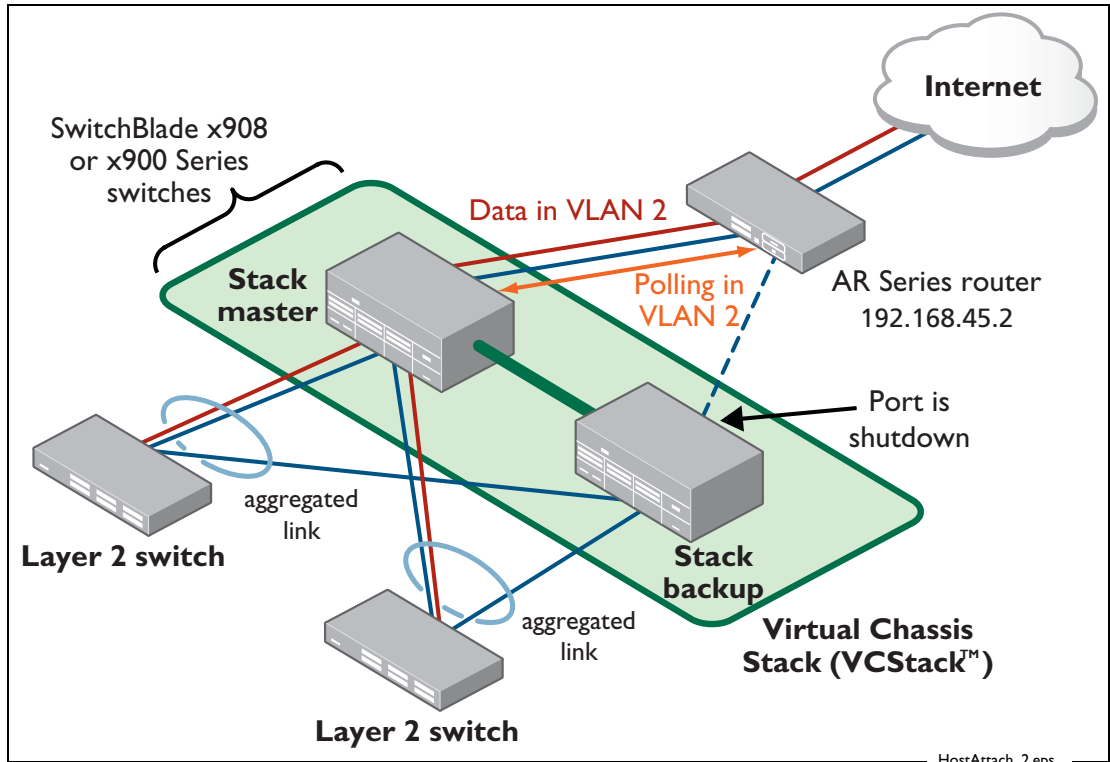
If the active link breaks then the polling instance detects this and activates a trigger. This trigger takes the port on the broken link out of the data VLAN, and puts the port on the end of the redundant link into that VLAN. Meanwhile, the polling instance continues to poll the router on the down link via an alternative VLAN created just for this purpose (VLAN 10).
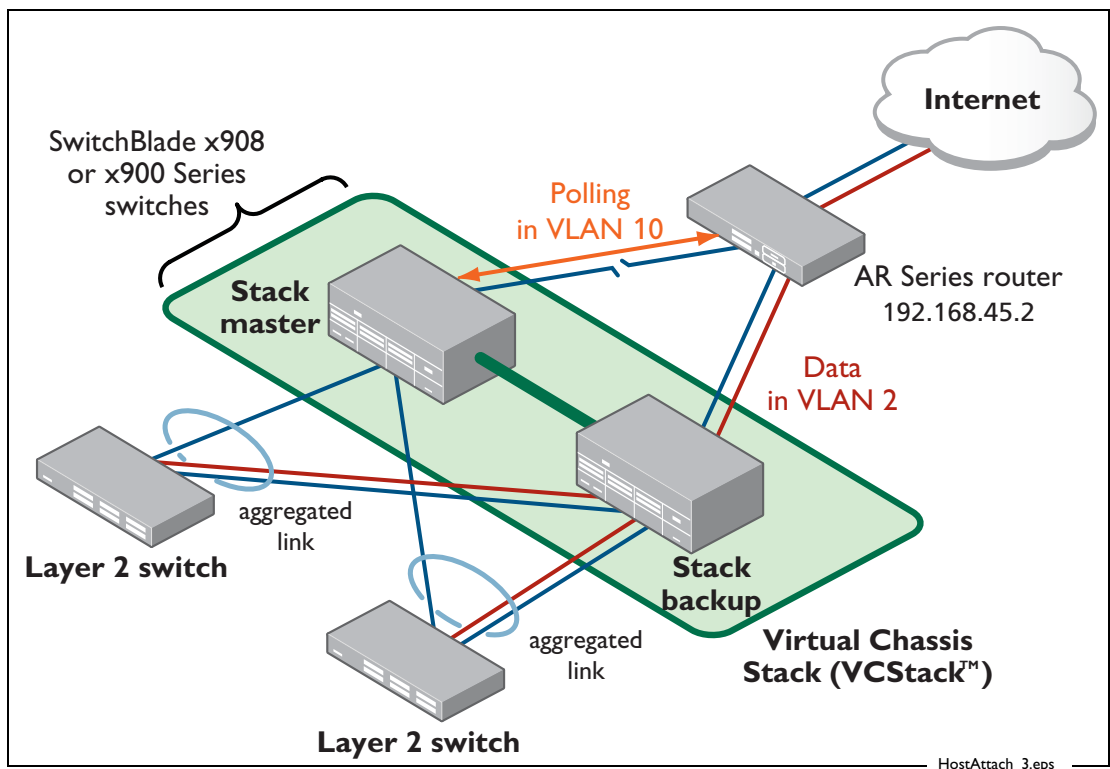
Ping polling attempts continue in VLAN 10 until the broken link is repaired. Once the polling instance can reach the router through the port again, it activates a trigger that moves the data VLAN back over to the original port.

The following figure shows the logic used by ping polling to determine whether the link is up.

ping_flowchart.eps
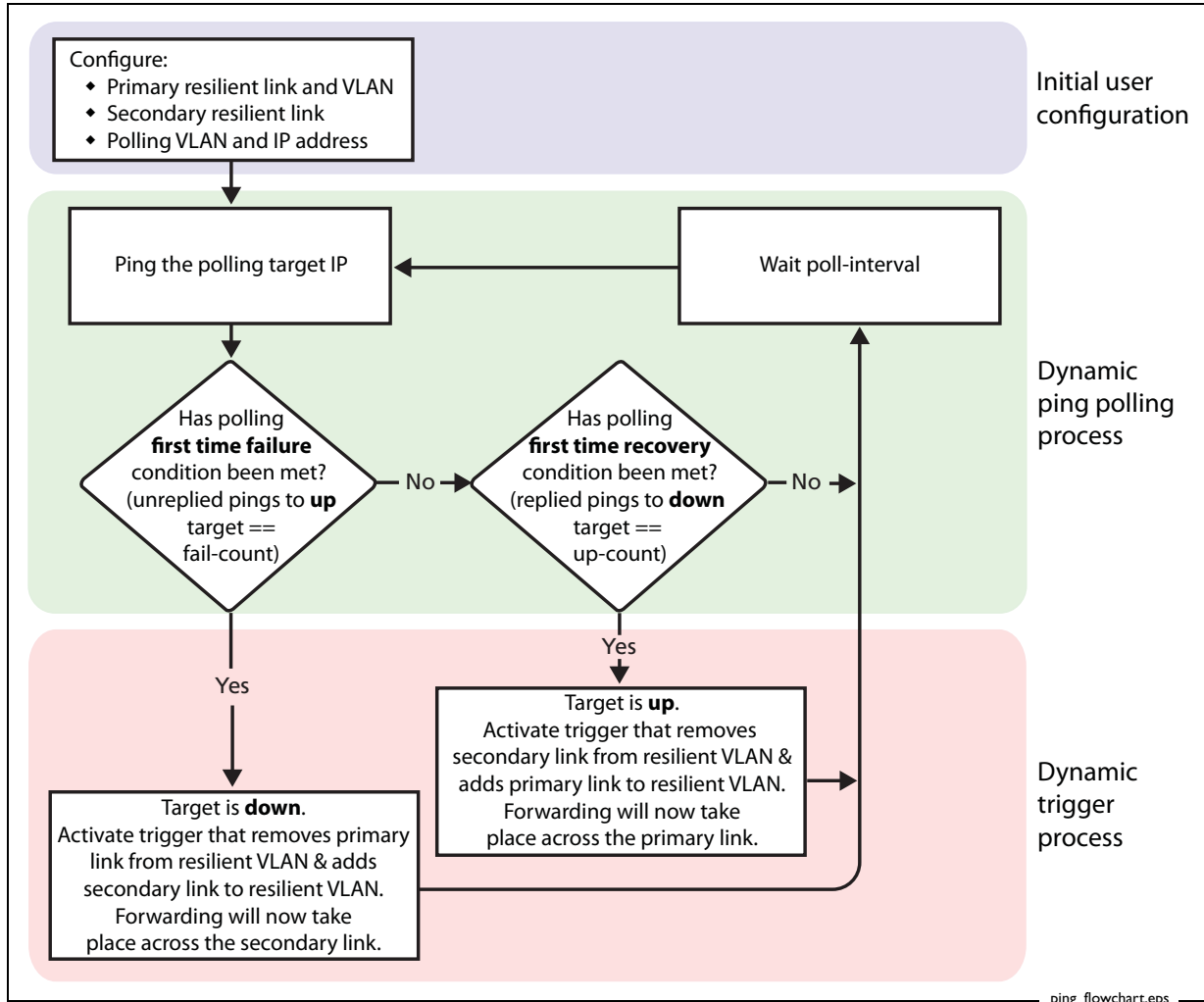
Configuring the Host Attach solution on the VCStack

In this example, the active link is port1.0.10 on the stack master switch. The redundant link is port2.0.10 on the stack backup switch.

Setting up the normal state of the ports

Use the following steps to create the initial configuration between the switches and the router:

1. Create VLANs 2, 10, and 11

Create the VLANs used for the solution. You will need three VLANs:

- a data VLAN–this is used for the data, and is associated with whichever port is active.
- a ping polling VLAN–this is used by the polling instance to poll the broken link.
- a parked VLAN–this is used on the redundant port to ensure it will not cause loops even if inadvertently enabled.

Use the commands:

```
vlan database
  vlan 2 name data
  vlan 10 name pingpoll
  vlan 11 name parked
  vlan 2,10,11 state enable
```

2. Give the data VLAN an IP address

The ping poll will poll the IP address of the router. Give VLAN 2 an IP address in the same range so that the polling instance uses this VLAN to reach the router. Use the commands:

```
interface vlan2
  ip address 192.168.45.1/24
```

3. Configure the port from the stack master to the router

This configures the initial state of port1.0.10, with this port as the active link to the router. Use the commands:

```
interface port1.0.10
  switchport access vlan 2
  spanning-tree portfast
```

4. Configure the redundant port from the stack backup to the router

Add this port to the parked VLAN, and configure it to be manually blocked. Use the commands:

```
interface port2.0.10
   switchport access vlan 11
   shutdown
   spanning-tree portfast
```

Configuring the scripts, polling, and triggers

Use the following steps to create the scripts, poll instance, and triggers for the Host Attach solution:

1. Create the link-down script (linkdown.scp)

The link-down script is triggered when the polling instance determines that the link between the stack master and the router is down. The script:

- adds the down port to VLAN 10–this act also removes VLAN 2 from this port

- removes the IP address from VLAN 2, and adds it to VLAN 10. The ping poll instance now polls across VLAN 10 on port 1.0.10.

- adds the redundant link to VLAN 2 and enables this port

The script must remove the ping poll from the data VLAN before the data VLAN is moved to the active link, otherwise the ping poll could receive an answer from the router, and then wrongly re-enable the down link.

The commands in the script are:

```
enable
configure terminal
interface port1.0.10
   switchport access vlan 10
interface vlan2
   no ip address
interface vlan10
   ip address 192.168.45.1/24
interface port2.0.10
   switchport access vlan 2
   no shutdown
```

2. Create the link-up script (linkup.scp)

Next create the link-up script (linkup.scp). This is triggered when ping polling determines that the link between the stack master and the router has re-established. The script:

- adds the redundant link to VLAN 11–this act also removes VLAN 2 from this port. The port is manually blocked again.

- removes the IP address from VLAN 10, and assigns it to VLAN 2.

- Adds port1.0.10 to VLAN 2. The polling instance can now poll the router through this link.

The commands in the script are:

```
enable
configure terminal
interface port2.0.10
  switchport access vlan 11
  shutdown
interface vlan10
  no ip address
interface vlan2
  ip address 192.168.45.1/24
interface port1.0.10
  switchport access vlan 2
```

3. Configure the polling instance to poll the router

Configure the switch to poll the router. Use the commands:

```
ping poll 1
  ip 192.168.45.2
  active
```

The switch will begin polling the router via VLAN 2 on the active port.

4. Create the triggers activated by the polling instance

Create trigger 1, which activates when the polling instance determines that the link is down. Use the commands:

```
trigger 1
  type ping-poll 1 down
  script 1 flash:/linkdown.scp
```

Create trigger 2, which activates once the polling instance determines that the link is up. Use the commands:

```
trigger 2
  type ping-poll 1 up
  script 1 flash:/linkup.scp
```

Extending the configuration to handle link failures between stack members

So far this How To Note has only discussed using the Host Attach solution for link failures. These are by far the most common failure events within networks. However, there is also the possibility of the stack failing. You can extend the Host Attach configuration described in this How To Note to deal with a stack failure as well.

The VCStack feature-set includes trigger events related to stack failure and recovery. To extend this Host Attach solution, we will use the following VCStack event triggers:

- **masterfail**–this event trigger indicates that a master failover event has occurred on the stack.

- **member join**–this event trigger indicates that a stack member has rejoined the stack.

In the example network, the primary active link is on the stack master, so, the solution must act only if the stack master fails. If the backup member fails, then no action is necessary.

There are a few scenarios that can trigger a master failover event. An important aspect of this solution is to ensure that the script activates the redundant link only when the backup switch no longer has contact with the original stack master (and becomes a stack master). The solution determines whether the original stack master is still in contact with the stack member by examining the output of the **show stack** command (on the new stack master) to see if it reports one active switch or two. If it reports 2 active switches, then no action is taken. If it reports just one active switch, then the solution changes the configuration to activate the redundant link.

When the stack recovers from the fail event, then the **member join** trigger is activated. In this case, the trigger's action returns the configuration to its original state by calling the link-up script.

The additional configuration steps follow:

1. Create the trigger that activates when the stack fails

This trigger activates the script. Use the commands:

```
trigger 3
  type stack master-fail
  script 1 flash:/masterdown.scp
```

2. Create the trigger that activates when the stack recovers

This trigger activates the linkup.scp script, configured earlier in this How To Note. Use the commands:

```
trigger 4
   type stack member join
   script 1 flash:/linkup.scp
```

3. Create the script activated by trigger 3 (masterdown.scp)

This script is required to activate a more powerful ASH script. The commands in the script are:

```
enable
activate masterdown.sh
```

4. Create the ASH script (masterdown.sh)

This script is written as an ASH shell script. The first 2 lines of the script looks at the number of switches listed in the **show stack** output. If more than one stack member is reported, then the script stops at this point. If there is only one stack member reported, then the script runs the CLI configuration commands required to change the active links. These are the same commands that are in linkdown.scp.

The commands in the script are:

```
n=`echo -e "show stack" | imish | grep 00 | wc -l`
if [ $n -lt 2 ]
then
  echo -e "enable\n
            configure terminal\n
            interface port1.0.10\n
              switchport access vlan 10\n
            interface vlan2\n
              no ip address\n
            interface port2.0.10\n
              switchport access vlan 2\n
              no shutdown\n
            interface vlan10\n
              ip address 192.168.45.1/24\n" | imish
  fi
```

Full configuration script

The full configuration script for the VCStack:

```
ping-poll 1
  ip 192.168.45.2
  active

vlan database
  vlan 2 name data
  vlan 10 name pingpoll
  vlan 11 name parked
  vlan 2,10-11 state enable

interface port1.0.10
  switchport access vlan 2
  spanning-tree portfast

interface port2.0.10
  switchport access vlan 11
  shutdown
  spanning-tree portfast

interface vlan2
  ip address 192.168.45.1/24

trigger 1
  type ping-poll 1 down
  script 1 flash:/linkdown.scp

trigger 2
  type ping-poll 1 up
  script 1 flash:/linkup.scp

trigger 3
  type stack master-fail
  script 1 flash:/masterdown.scp

trigger 4
  type stack member join
  script 1 flash:/linkup.scp
```

The configuration script in linkdown.scp:

```
enable
configure terminal
interface port1.0.10
  switchport access vlan 10
interface vlan2
  no ip address
interface vlan10
  ip address 192.168.45.1/24
interface port2.0.10
  switchport access vlan 2
  no shutdown
```

The configuration script in linkup.scp:

```
enable
configure terminal
interface port2.0.10
  switchport access vlan 11
  shutdown
interface vlan10
  no ip address
interface vlan2
  ip address 192.168.45.1/24
interface port1.0.10
  switchport access vlan 2
```

The configuration script in masterdown.scp:

```
enable
activate masterdown.sh
```

The ASH script in masterdown.sh:

```
n=`echo -e "show stack" | imish | grep 00 | wc -l`
if [ $n -lt 2 ]
then
  echo -e "enable\n
          configure terminal\n
          interface port1.0.10\n
            switchport access vlan 10\n
          interface vlan2\n
            no ip address\n
          interface port2.0.10\n
            switchport access vlan 2\n
            no shutdown\n
          interface vlan10\n
            ip address 192.168.45.1/24\n" | imish
fi
```

C613-16140-00 REV A

Connecting The (IP) World

Allied Telesis