Application Note

Failover through BGP route health injection

Document version: v1.2

Last update: 8th November 2013

Purpose

This application note aims to describe how to build a high available platform using **BGP** routing protocol to choose the best available ALOHA Load-Balancer.

Limitation

Currently, ALOHA Load-Balancer can only announce its own availability, whatever the status of the server farms. It means you could use this procedure to trigger a failover based on ALOHA availability, but not on server farm capacity or availability.
This kind of feature, Virtual IP route health injection based on server farm capacity or availability will come later.

That said, you can write your own script hosted on the ALOHA to update **BGP** configuration based on farms capacity.

Complexity

Versions concerned

– Aloha 4.2 and above

Changelog

Version	Description
1.2	Add Extreme Networks router configuration
1.1	Add Brocade router configuration
1.0	Initial release

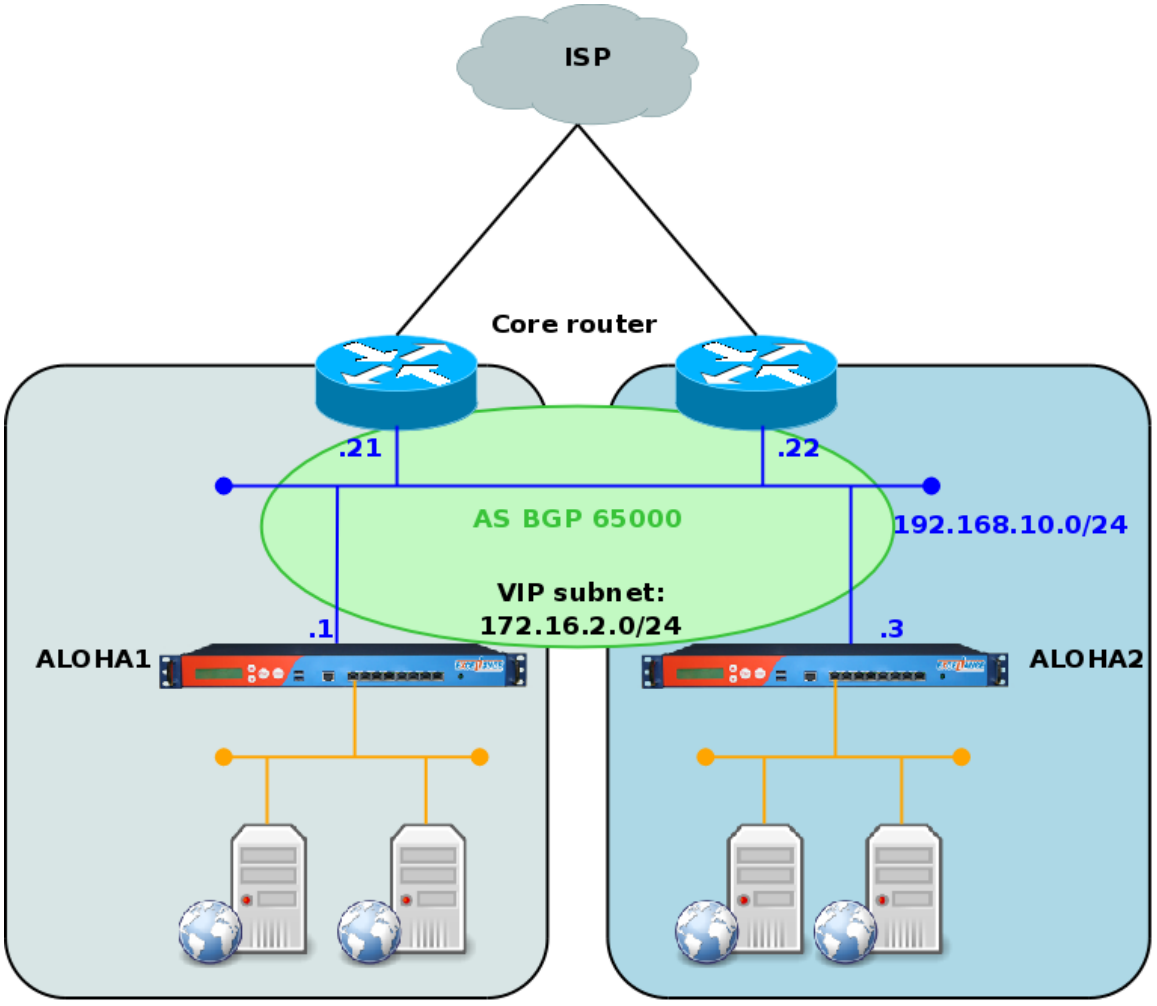
Synopsis

Usually, this type of architecture suits well when you have two datacenters or more, over a MAN or WAN. But it can be used in a single DC as well, over the LAN.
Principle is quite simple: building a **BGP Autonomous System** (aka AS) where the ALOHAs can

inject routes into your core routing network. The core routers will be configured to choose an ALOHA if it is available or failover to the second one: this is an **Active/Passive** infrastructure.

Diagram

The diagram below shows how things are working:



- The **core routers** will be configured to send traffic to **ALOHA1** and failover to **ALOHA2**.
- The BGP AS number is 65000, the routes injected by the **ALOHAs** are the subnet dedicated to Virtual IPs: 172.16.2.0/24.
- The **Core routers** and the **ALOHAs** can communicate through the subnet 192.168.10.0/24.

Your Virtual IP network could be public IPs as well

In the present appnote, we'll only provide configuration of core router #1 (192.168.10.21)

ALOHA BGP configuration

In the **ALOHA**, the dynamic routing service name is **bird**.

bird startup

On the ALOHA WUI, click on **Services** tab then scroll down and click on the link **advanced mode**.
Click on **OK** when prompted. *Advanced services are printed in red*.
Click on the **edit** icon on the **bird** service line:  .

Then comment or delete the line **no autostart**.
Now you can start **bird** by clicking the **start** icon:  .

Repeat for both **ALOHAs**.

ALOHA bird configuration for route health injection

Based on the diagram above, below are the **BGP** configuration for both **ALOHAs**.
This configuration remains the same, whatever BGP router you are running on the core network.
In order to edit **bird** configuration, just click on the **edit** icon:  in the **Services** tab.

ALOHA1 configuration:

```
log syslog all;
router id 192.168.10.1;

protocol device {
 scan time 10;
}

protocol static VIPs {
 route 172.16.2.11/32 via 192.168.10.1;
 route 172.16.2.12/32 via 192.168.10.1;
 route 172.16.2.13/32 via 192.168.10.1;
}

protocol bgp {
 import none;
 export filter {
  if proto = "VIPs" then accept;
  reject;
 };
 local as 65000;
 source address 192.168.10.1;
 neighbor 192.168.10.21 as 65000;
}
```

ALOHA2 configuration:

```
log syslog all;
router id 192.168.10.3;

protocol device {
 scan time 10;
}

protocol static VIPs {
 route 172.16.2.11/32 via 192.168.10.3;
 route 172.16.2.12/32 via 192.168.10.3;
 route 172.16.2.13/32 via 192.168.10.3;
}

protocol bgp {
 import none;
 export filter {
  if proto = "VIPs" then accept;
  reject;
 };
 local as 65000;
 source address 192.168.10.3;
 neighbor 192.168.10.21 as 65000;
}
```

Once you have updated bird configuration, you have to reload them by clicking the **reload** icon: 
.

BGP routers configuration examples

This chapter introduces **BGP** configuration on different type of equipements.

These configurations are basic example and may require some tunning to fit in your environment.

bird router

bird is an opensource software and can be used on a BGP core network. Below is the **bird** configuration to accept BGP announces from **ALOHAs**:

```
# Configure logging
log syslog { info, remote, warning, error, auth, fatal, bug };

router id 192.168.10.21;

filter aloha_vip {
 if net ~ 172.16.2.0/24 then accept;
 else reject;
}

protocol kernel {
 scan time 10;
 import none;
 export all;
}

protocol device {
 scan time 10;
}

protocol bgp aloha1 {
 local as 65000;
 export none;
 import filter aloha_vip;
 source address 192.168.10.21;
 neighbor 192.168.10.1 as 65000;
 default bgp_local_pref 300;
}

protocol bgp aloha2 {
 local as 65000;
 export none;
 import filter aloha_vip;
 source address 192.168.10.21;
 neighbor 192.168.10.3 as 65000;
 default bgp_local_pref 200;
}
```

The weight (**bgp_local_pref**) is higher for **ALOHA1**, so it will be chosen first if it is available.

The **bird** route information table should look like this:

```
# birdc show route
BIRD 1.2.5 ready.
172.16.2.11/32   via 192.168.10.1 on eth0 [aloha1 16:35] * (100) [i]
                 via 192.168.10.3 on eth0 [aloha2 16:35] (100) [i]
172.16.2.13/32   via 192.168.10.1 on eth0 [aloha1 16:35] * (100) [i]
                 via 192.168.10.3 on eth0 [aloha2 16:35] (100) [i]
172.16.2.12/32   via 192.168.10.1 on eth0 [aloha1 16:35] * (100) [i]
                 via 192.168.10.3 on eth0 [aloha2 16:35] (100) [i]
```

Prefered route is the one with the star ***** and **bird** will use it first.
Let's confirm this by checking the router's routing table:

```
# ip route
172.16.2.13 via 192.168.10.1 dev eth0 proto bird
172.16.2.12 via 192.168.10.1 dev eth0 proto bird
172.16.2.11 via 192.168.10.1 dev eth0 proto bird
```

From a kernel point of view, only a single route is known.

- If **ALOHA1** fails, then **core router's bird** will update the router's routing table with **ALOHA2**'s IP for all of Virtual IPs.
- If **ALOHA1** stops annoucing one route, then **core router's bird** will update the router's routing table with **ALOHA2**'s IP as a destination for this particular Virtual IP.

Brocade

Brocade is one of the leader in the networking industry.
The configuration below shows how to configure **Brocade** BGP router to accept the **ALOHA** Route Health Injection:

```
ip prefix-list aloha_vip deny 0.0.0.0/0
ip prefix-list aloha_vip permit 172.16.2.0/24 le 32

router bgp
local-as 65000
neighbor aloha peer-group
neighbor aloha remote-as 65000
neighbor 192.168.10.1 peer-group aloha
neighbor 192.168.10.1 description aloha1
neighbor 192.168.10.3 peer-group aloha
neighbor 192.168.10.3 description aloha3
!
address - family ipv4
neighbor 192.168.10.1 activate
neighbor 192.168.10.1 route-map in local_pref_300
neighbor 192.168.10.1 prefix-list aloha_vip in
neighbor 192.168.10.3 activate
neighbor 192.168.10.3 route-map in local_pref_100
neighbor 192.168.10.3 prefix-list aloha_vip in
exit-address-family
exit

route-map local_pref_300 permit 10
set local-preference 300

route-map local_pref_100 permit 10
set local-preference 100
```

(sorry, no routing table output available)

Cisco

Cisco is one of the leader in the networking industry.
The configuration below shows how to configure **Cisco** BGP router to accept the **ALOHA** Route Health Injection:

```
!
configure terminal
!
ip prefix-list aloha_vip deny 0.0.0.0/0
ip prefix-list aloha_vip permit 172.16.2.0/24 le 32
!
router bgp 65000
 bgp router-id 192.168.10.21
 bgp log-neighbor-changes
 neighbor aloha peer-group
 neighbor aloha remote-as 65000
 neighbor 192.168.10.1 peer-group aloha
 neighbor 192.168.10.1 description aloha1
 neighbor 192.168.10.3 peer-group aloha
 neighbor 192.168.10.3 description aloha3
 !
 address-family ipv4
  neighbor 192.168.10.1 activate
  neighbor 192.168.10.1 localpref 300
  neighbor 192.168.10.1 prefix-list aloha_vip in
  neighbor 192.168.10.3 activate
  neighbor 192.168.10.3 localpref 200
  neighbor 192.168.10.3 prefix-list aloha_vip in
  no auto-summary
  no synchronization
 exit-address-family
!
exit
exit
!
```

Now, let's have a look at the router's routing table:

```
Router#sh ip bgp
BGP table version is 4, local router ID is 192.168.10.21
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i172.16.2.11/32 192.168.10.3               100    200 i
*>i               192.168.10.1               100    300 i
* i172.16.2.12/32 192.168.10.3               100    200 i
*>i               192.168.10.1               100    300 i
* i172.16.2.13/32 192.168.10.3               100    200 i
*>i               192.168.10.1               100    300 i
```

Cisco routing table is quite verbose: we can see the route weight and the currently selected route. We can clearly see as well that the routes were learnt through **iBGP**.

- If **ALOHA1** fails, then **Cisco router** will update its routing table with **ALOHA2**'s IP for all Virtual IPs.

– If **ALOHA1** stops annoucing one route, then **Cisco router** will update its routing with **ALOHA2**'s IP for this particular Virtual IP.

Extreme Networks

The configuration below shows how to configure **Extreme Networks** BGP router to accept the **ALOHA** Route Health Injection:

```
# bgp configuration
configure bgp AS-number 65000
configure bgp routerid 192.168.10.21
configure bgp local-preference 300

create bgp peer-group aloha
configure bgp peer-group aloha remote-AS-number 65000
configure bgp peer-group aloha route-policy in aloha_vip
enable bgp peer-group aloha

create bgp neighbor 192.168.10.1 peer-group "aloha"
configure bgp neighbor 192.168.10.1 description "aloha1"
configure bgp neighbor 192.168.10.1 peer-group aloha acquire-all
enable bgp neighbor 192.168.10.1

create bgp neighbor 192.168.10.3 peer-group "aloha"
configure bgp neighbor 192.168.10.3 description "aloha2"
configure bgp neighbor 192.168.10.3 peer-group aloha acquire-all
enable bgp neighbor 192.168.10.3

enable bgp

# policy configuration
edit policy aloha1_vip
 entry filter1 {
  if match all {
   nlri 0.0.0.0/0 exact;
  } then {
   deny;
  }
 }
 entry filter2 {
  if match all {
   nlri 172.16.2.0/24;
  } then {
   permit;
   local-preference 300;
  }
 }

edit policy aloha2_vip
 entry filter1 {
  if match all {
   nlri 0.0.0.0/0 exact;
  } then {
   deny;
  }
 }
```

```
entry filter2 {
 if match all {
  nlri 172.16.2.0/24;
 } then {
  permit;
  local-preference 200;
 }
}
```

(sorry, no routing table output available)

OpenBGPd

OpenBGPd is the BGP daemon provided with **OpenBSD** operating system.
The configuration below shows how to configure **OpenBGPd** to accept the **ALOHA** RHI:

```
AS 65000
router-id 192.168.10.21

log updates

group aloha {
 remote-as 65000
 neighbor 192.168.10.1 {
  descr "aloha1"
  set localpref 300
  announce none
 }
 neighbor 192.168.10.3 {
  descr "aloha2"
  set localpref 200
  announce none
 }
}

deny from any
allow from group aloha inet prefixlen 24 - 32
```

Now, let's have a look at the router routing table:

```
# bgpctl show rib
flags: * = Valid, > = Selected, I = via IBGP, A = Announced
origin: i = IGP, e = EGP, ? = Incomplete

flags destination       gateway        lpref   med aspath origin
I*>   172.16.2.11/32     192.168.10.1     300     0 i
I*    172.16.2.11/32     192.168.10.3     200     0 i
I*>   172.16.2.12/32     192.168.10.1     300     0 i
I*    172.16.2.12/32     192.168.10.3     200     0 i
I*>   172.16.2.13/32     192.168.10.1     300     0 i
I*    172.16.2.13/32     192.168.10.3     200     0 i
```

OpenBGPd routing information base is quite verbose: we can see the route weight (**lpref**) and the currently selected route. We can clearly see as well that the route were learnt through **iBGP**.

- If **ALOHA1** fails, then **OpenBGPd** will update router's routing table with **ALOHA2**'s IP for all of Virtual IPs.
- If **ALOHA1** stops annoucing one route, then **OpenBGPd** will update router's routing with **ALOHA2**'s IP for this particular Virtual IP.