Dynamic IVRL on default VRF

AOS-CX 10.09

Vincent Giles
Technical Marketing Engineer
Agenda

1. Overview
2. Use Cases
3. Details and Caveats
4. Configuration
5. Best Practices
6. Troubleshooting
7. Demo
8. Additional Resources
Overview
Definitions

Acronyms

- **IVRL**  
  Inter VRF Route Leaking

- **DIVRL**  
  Dynamic Inter VRF Route Leaking

- **SIVRL**  
  Static Inter VRF Route Leaking

- **Routing table**  
  Valid routing entries selected from each active routing protocols based on the administrative distance

- **FIB**  
  Forwarding Information Base, active forwarding entries programmed into ASIC based on the routing table

- **RIB**  
  Routing Information Base, selected and non-selected candidate routes per routing protocol.
**Overview**

**Reminder**

- **IVRL** or Inter-VRF Route Leaking is a mechanism used to allow IP communication between VRFs and to “leak” IPv4/IPv6 routes from one VRF to one or multiple other VRFs.

- IVRL can be achieved either through:
  
  - **static IVRL** with usage of **static routes**:  
    A static route is configured in the destination VRF with both interface and IP Next-Hop from the source VRF.

    ```
    switch(config)# ip route 10.1.1.0/24 1/1/1 10.2.1.1 vrf VRF2
    vrf VRF1
      rd 192.168.1.1:1
      address-family ipv4 unicast
      route-target export 65001:1
      route-target import 65001:2
      exit-address-family
    vrf VRF2
      rd 192.168.1.1:2
      address-family ipv4 unicast
      route-target export 65001:2
      route-target import 65001:1
      exit-address-family
    ```

    leaks the prefix 10.1.0.0/24 into the “VRF2” VRF, which is reachable by the next-hop IP 10.2.1.1 on the interface 1/1/1 from “VRF1” VRF (not appearing in the static route command).

  - **Dynamic IVRL** using **MP-BGP**:  
    MP-BGP routes are exported from a source VRF with associated Route-Target, and imported on destination VRF with import Route-Target matching the value used for export.
<table>
<thead>
<tr>
<th>AOS-CX 10.07</th>
<th>AOS-CX 10.08</th>
<th>AOS-CX 10.09</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Static route-leaking between VRFs (including “default” VRF) is supported.</td>
<td>▪ Static route-leaking between VRFs (including “default” VRF) is supported.</td>
<td>▪ Static route-leaking between VRFs (including “default” VRF) is supported.</td>
</tr>
<tr>
<td>▪ Dynamic IVRL is supported between tenant VRFs.</td>
<td>▪ Dynamic IVRL is supported between tenant VRFs.</td>
<td>▪ Dynamic IVRL is supported between tenant and “default” VRFs.</td>
</tr>
<tr>
<td>▪ Not possible for “default” VRF.</td>
<td>▪ CLI configuration for route-target and route-distinguisher of “default” VRF is possible but <strong>not supported</strong> (not tested).</td>
<td>▪ CLI configuration for route-target and route-distinguisher of “default” VRF is possible and <strong>supported</strong>.</td>
</tr>
<tr>
<td>▪ No VRF route-map support.</td>
<td>▪ No VRF route-map support.</td>
<td>▪ No VRF route-map support.</td>
</tr>
</tbody>
</table>
Use cases

Inter-VRFs use-cases

- Between tenant VRF1 and tenant VRF2: not so frequent, some services like DHCP.
- Between services VRF and tenant VRFs: main use-case
  - Used for multiple services: DHCP, DNS, in-band network management and monitoring
  - Services VRF can be configured as a dedicated VRF called “SERVICES” VRF
  - Services VRF can not be the “mgmt” OOBM VRF name

```plaintext
switch(config)# vrf mgmt
VRF 'mgmt' already exists. Configurations under 'mgmt' vrf context are not allowed
```

- From AOS-CX 10.09, services VRF can now be the “default” VRF. Useful during migration/transition from one flat default VRF network to multiple VRFed/isolated networks. Most services for DNS and network management are already configured in default VRF.
AOS-CX 10.09 Enhancements

Dynamic IVRL on default VRF

- The previous configuration using non-default VRF have been extended to support default VRF.
- No schema change.
- Show running-config, show running-config vrf, show running-config vrf <name> have been enhanced to show vrf ‘default’.
- IPv4 and IPv6 Address-Families are supported under “default” VRF.
- EVPN route-targets are not supported under “default” VRF.

- 256 Route-Targets are supported in each VRF, including now in the default VRF context
- Redistribute local/connected/static/ospf can be used under “default” VRF context’s Address-Families to export routes from default VRF to tenant (non-default) VRF.

```plaintext
switch(config-vrf)# route-target both 1:1 evpn
EVPN configurations are not allowed under VRF default.
```
### 10.09 Platform Support

Dynamic IVRL on default VRF

<table>
<thead>
<tr>
<th>Platform</th>
<th>4100 6000 6100</th>
<th>6200</th>
<th>6300</th>
<th>6400</th>
<th>8320</th>
<th>8325</th>
<th>8360</th>
<th>8400</th>
<th>10000</th>
<th>Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic IVRL on default VRF</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Caveats

AOS-CX 10.09

- Cascaded IVRL on the same switch
  - On the same switch, only one level of Inter-VRF route leaking is supported.
    - Examples:
      - VRF1 to “default”
      - VRF1 to VRF2
  - On the same switch, more than one level of Inter-VRF route leaking, or cascaded IVRL, is not supported.
    - Example:
      - VRF1 to VRF2 with default being the intermediate VRF: VRF1 ➔ default ➔ VRF2
      - VRF2 won’t get routes from VRF1

- Multicast route leaking is not supported.

- As there is no route-map support yet for route-targets import/export rules, route filtering on dynamically leaked routes is not possible.

- Extended Community are lost through the route-leaking process.

- The total number of dynamic leaked routes among all VRFs is validated up to 16K per platform.
Configuration
Dynamic IVRL
VRF configuration

For DIVRL

- To leak routes from/to default VRF, the following must be configured:
  - In “default” VRF context:
    - route distinguisher
    - export route-target: one instance for value associated with “default” VRF or multiple instances (one per destination VRFs)
    - import route-target: multiple instances (one per source VRFs) or one instance for value associated with “default” VRF
  - In BGP “default” VRF context:
    - Redistribution of “default” VRF routes from other protocols: local, connected, static, ospf
    - or network command
    - or BGP routes received from BGP neighbor in “default” VRF
  - In “tenant” VRF context:
    - route distinguisher
    - Import/export route-targets
  - In BGP “tenant” VRF context:
    - Redistribution of “tenant” VRF routes from other protocols: local, connected, static, ospf
    - or network command
    - or BGP routes received from BGP neighbor in “tenant” VRF

Route-target **export** values must be different between IPv4 and IPv6 address-family.
VRF configuration

no vrf default

- The configurations under VRF cannot be deleted with “no vrf” command.
- Route-Distinguisher and Address-Family configurations have to be individually deleted under vrf context.

```
switch# show run vrf default
vrf default
  rd 192.168.2.1:0
  address-family ipv4 unicast
    route-target export 65001:0
    route-target import 65001:1
    route-target import 65001:2
  exit-address-family

switch(config)# no vrf default
All Layer 3 configurations associated with the VRF will be deleted.
Continue (y/n)? Y
Cannot delete default VRF.
switch(config)# show run vrf default
vrf default
  rd 192.168.2.1:0
  address-family ipv4 unicast
    route-target export 65001:0
    route-target import 65001:1
    route-target import 65001:2
  exit-address-family
```
Route leaking between default VRF and tenant VRFs

- no bgp peering in this example. (Can be configure if needed).
- Redistribution must be adjusted based on routing design per VRF, per AF.
Best Practices
Best Practices

Route-targets

- Route-Distinguisher value must be different on each switch and each VRF.
- One dedicated route-target import and export value per VRF, shared among every switch so that the same BGP extended community is used per VRF domain.
**Best Practices**

**Route-targets**

**“default” VRF route-target**

- Better control and troubleshooting due to dedicated route-target for “default” VRF.
- Single point of configuration: No requirement for awareness of tenant VRFs on BGP peers in the default VRF.

**Tenant VRF route-target**

- No configuration impact on tenant VRF.
- But, all BGP peers in the “default” VRF would need to get configured with all VRF Route-Targets.
Route-Leaking Troubleshooting

Troubleshooting Pre-requisite

1. Have a topology diagram with identified switch, interfaces ID and IP details (ROPs, Loopbacks, SVIs).
2. Generate a “show tech” when creating the TAC case.
3. Check physical cabling and transceivers/DACs (supported versus unsupported).
   “show system inventory transceiver”
4. Check interface state: “show interface brief” and “show interface error-statistics non-zero”
5. Check LLDP neighbor information “show LLDP neighbor”
6. Ping and traceroute between loopbacks or SVIs if relevant for debug.
Route-Leaking Troubleshooting

Troubleshooting Process

1. show running-config vrf
2. Check that RD is configured
3. Check that import route-target of VRFx corresponds to export route-target of VRFy targeted for import in VRFx
4. Check that export route-target of VRFx corresponds to import route-target of VRFz targeted for export from VRFx
5. Check in BGP that corresponding VRF instance has proper routes redistribution (like loopback, connected, static, ospf)
6. Check BGP neighbor if appropriate based on expected peering.
7. Check routes in BGP: show bgp ipv4|ipv6 unicast
8. Check RIB per VRF (inc. default): show ip|ipv6 rib vrf …
9. Check routes per VRF (inc. default): show ip|ipv6 route vrf …
10. Check events: show events -r -d hpe-routing
11. Check CPU/mem: top cpu / top memory
12. Enable vrf debugging: debug vrf all severity debug debug destination buffer show debug buffer module vrf
Demo
Topology

Simulator Lab: DIVRL with tenant VRF1 and VRF2 route-leaking to/from default VRF

- VRF1 Hosts should be able to communicate together in VRF1.
- No VRF1 Host should be able to communicate with VRF2 host.
- VRF1 and VRF2 Hosts should be able to communicate with Services Server located in “default” VRF.
hostname SW1
!
vrf VRF1
  rd 192.168.2.1:1
  address-family ipv4 unicast
  route-target export 65001:1
  route-target import 65001:0
  exit-address-family
vrf VRF2
  rd 192.168.2.1:2
  address-family ipv4 unicast
  route-target export 65001:2
  route-target import 65001:0
  exit-address-family
vrf default
  rd 192.168.2.1:0
  address-family ipv4 unicast
  route-target export 65001:0
  route-target import 65001:0
  exit-address-family
!
vlan 1,1115,1125
!
interface 1/1/1
  no shutdown
  vrf attach VRF1
description to SW2
  ip address 192.168.115.0/31
interface 1/1/2
  no shutdown
description to SW3
  no routing
  vlan trunk native 1
  vlan trunk allowed 1115,1125
interface 1/1/9
  no shutdown
description to SRV
  ip address 10.5.50.1/24

interface vlan 1115
  vrf attach VRF1
  ip address 192.168.115.2/31
interface vlan 1125
  vrf attach VRF2
  ip address 192.168.125.0/31
!
  ip route 10.11.96.0/20 192.168.115.1 vrf VRF1
  ip route 10.11.119.0/24 192.168.115.3 vrf VRF1
  ip route 10.12.0.0/16 192.168.125.1 vrf VRF2
!
router bgp 65001
  bgp router-id 192.168.2.1
  address-family ipv4 unicast
  redistribute connected
  redistribute static
  exit-address-family
!
vrf VRF1
  address-family ipv4 unicast
  redistribute connected
  redistribute static
  exit-address-family
!
vrf VRF2
  address-family ipv4 unicast
  redistribute connected
  redistribute static
  exit-address-family
Configuration

SW2 and SW3

```
hostname SW2
!
vlan 1,110-111
!
interface 1/1/1
  no shutdown
description to HostA
no routing
vlan access 110
interface 1/1/2
  no shutdown
description to HostB
no routing
vlan access 111
interface 1/1/9
  no shutdown
description to SW1
  ip address 192.168.115.1/31
!
interface vlan 110
  ip address 10.11.110.1/24
interface vlan 111
  ip address 10.11.111.1/24
!
ip route 0.0.0.0/0 192.168.115.0
```

```
hostname SW3
vrf VRF1
vrf VRF2
!
vlan 1,119-120,1115,1125
!
interface 1/1/1
  no shutdown
description to HostA
no routing
vlan access 119
interface 1/1/2
  no shutdown
description to HostB
no routing
vlan access 120
interface 1/1/9
  no shutdown
description to SW1
  no routing
  vlan trunk native 1
  vlan trunk allowed 1115,1125
!
interface vlan 119
  vrf attach VRF1
  ip address 10.11.119.1/24
interface vlan 120
  vrf attach VRF2
  ip address 10.12.120.1/24
interface vlan 1115
  vrf attach VRF1
  ip address 192.168.115.3/31
interface vlan 1125
  vrf attach VRF2
  ip address 192.168.125.1/31
!
ip route 0.0.0.0/0 192.168.115.2 vrf VRF1
ip route 0.0.0.0/0 192.168.125.0 vrf VRF2
```
SW1 BGP RIB

Per VRF

SW1# show bgp vrf VRF1 ipv4 unicast
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF: VRF1
Local Router-ID 192.168.115.2

<table>
<thead>
<tr>
<th>Network</th>
<th>Nexthop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 10.5.50.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 10.11.96.0/20</td>
<td>192.168.115.1</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 10.11.199.0/24</td>
<td>192.168.115.3</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.115.0/31</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.115.2/31</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
</tbody>
</table>

Total number of entries 5

SW1# show bgp vrf VRF2 ipv4 unicast
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF: VRF2
Local Router-ID 192.168.125.0

<table>
<thead>
<tr>
<th>Network</th>
<th>Nexthop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 10.5.50.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 10.11.96.0/20</td>
<td>192.168.115.1</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 10.11.119.0/24</td>
<td>192.168.115.3</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.115.0/31</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.115.2/31</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.125.0/31</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
</tbody>
</table>

Total number of entries 3

SW1# show bgp ipv4 unicast
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF: default
Local Router-ID 192.168.2.1

<table>
<thead>
<tr>
<th>Network</th>
<th>Nexthop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 10.5.50.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 10.11.96.0/20</td>
<td>192.168.115.1</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 10.11.119.0/24</td>
<td>192.168.115.3</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 10.12.0.0/16</td>
<td>192.168.125.1</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.115.0/31</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.115.2/31</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.125.0/31</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
</tbody>
</table>

Total number of entries 7
### SW1 FIB

#### Per VRF

**SW1# sh ip route vrf VRF1**

Displaying ipv4 routes selected for forwarding

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Nexthop</th>
<th>Interface</th>
<th>VRF(egress)</th>
<th>Origin/ Type</th>
<th>Distance/ Metric</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.50.0.24</td>
<td>-</td>
<td>1/1/9</td>
<td>default</td>
<td>B/I</td>
<td>[200/0]</td>
<td>00h:42m:01s</td>
</tr>
<tr>
<td>10.11.96.0/20</td>
<td>192.168.115.1</td>
<td>1/1/1</td>
<td>-</td>
<td>S</td>
<td>[1/0]</td>
<td>03h:28m:24s</td>
</tr>
<tr>
<td>192.168.115.3</td>
<td>-</td>
<td>vlan1115</td>
<td>-</td>
<td>C</td>
<td>[0/0]</td>
<td>-</td>
</tr>
<tr>
<td>192.168.115.31</td>
<td>1/1/1</td>
<td>-</td>
<td>L</td>
<td>[0/0]</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>192.168.115.32</td>
<td>1/1/1</td>
<td>vlan1115</td>
<td>-</td>
<td>L</td>
<td>[0/0]</td>
<td>-</td>
</tr>
</tbody>
</table>

Total Route Count: 7

**SW1# sh ip route vrf VRF2**

Displaying ipv4 routes selected for forwarding

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Nexthop</th>
<th>Interface</th>
<th>VRF(egress)</th>
<th>Origin/ Type</th>
<th>Distance/ Metric</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.50.0.24</td>
<td>-</td>
<td>1/1/9</td>
<td>default</td>
<td>B/I</td>
<td>[200/0]</td>
<td>00h:41m:19s</td>
</tr>
<tr>
<td>10.12.0.0/16</td>
<td>192.168.125.1</td>
<td>vlan1125</td>
<td>-</td>
<td>L</td>
<td>[0/0]</td>
<td>-</td>
</tr>
<tr>
<td>192.168.115.31</td>
<td>1/1/1</td>
<td>vlan1115</td>
<td>-</td>
<td>L</td>
<td>[0/0]</td>
<td>-</td>
</tr>
</tbody>
</table>

Total Route Count: 4

**SW1# sh ip route**

Displaying ipv4 routes selected for forwarding

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Nexthop</th>
<th>Interface</th>
<th>VRF(egress)</th>
<th>Origin/ Type</th>
<th>Distance/ Metric</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.50.0.24</td>
<td>-</td>
<td>1/1/9</td>
<td>default</td>
<td>B/I</td>
<td>[200/0]</td>
<td>00h:42m:31s</td>
</tr>
<tr>
<td>192.168.115.1</td>
<td>vlan1125</td>
<td>-</td>
<td>S</td>
<td>[1/0]</td>
<td>03h:28m:54s</td>
<td></td>
</tr>
<tr>
<td>192.168.125.1</td>
<td>vlan1125</td>
<td>-</td>
<td>C</td>
<td>[0/0]</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>192.168.125.31</td>
<td>vlan1125</td>
<td>-</td>
<td>L</td>
<td>[0/0]</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Total Route Count: 8
Route-Targets in “default” VRF

SWI# sh bgp ipv4 unicast 10.5.50.0/24

<table>
<thead>
<tr>
<th>Network</th>
<th>Peer</th>
<th>Metric</th>
<th>Weight</th>
<th>Best</th>
<th>Type</th>
<th>Originator ID</th>
<th>Aggregator ID</th>
<th>Atomic Aggregate</th>
<th>RFD Flaps</th>
<th>AS-Path</th>
<th>Cluster List</th>
<th>Communities</th>
<th>Ext-Communities</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.5.50.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>Yes</td>
<td>external</td>
<td>0.0.0.0</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>RT: 65001:0</td>
<td></td>
</tr>
</tbody>
</table>

Source VRF: default

Route-target is lost during import

SWI# sh bgp vrf VRF1 ipv4 unicast 10.11.119.0/24

<table>
<thead>
<tr>
<th>Network</th>
<th>Peer</th>
<th>Metric</th>
<th>Weight</th>
<th>Best</th>
<th>Type</th>
<th>Originator ID</th>
<th>Aggregator ID</th>
<th>Atomic Aggregate</th>
<th>RFD Flaps</th>
<th>AS-Path</th>
<th>Cluster List</th>
<th>Communities</th>
<th>Ext-Communities</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.11.119.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>Yes</td>
<td>external</td>
<td>0.0.0.0</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>RT: 65001:1</td>
<td></td>
</tr>
</tbody>
</table>

Destination VRF: VRF1

Route-target is lost during import
Traffic objectives

ICMP tests

- VRF1 Hosts should be able to communicate together in VRF1.
- No VRF1 Host should be able to communicate with VRF2 host.
- VRF1 and VRF2 Hosts should be able to communicate with Services Server located in “default” VRF.
Thank you

vincent.giles@hpe.com