© Michael Turek

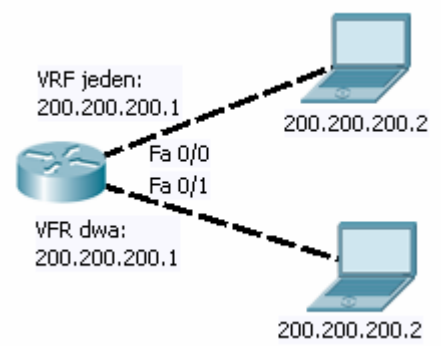
COMPUTER NETWORKS - LABORATORY 109

Subject:

Cisco IOS – VRF, Virtual Routing and Forwarding (with VRF Lite).

Task A: Configuring a VRF

1. *Virtual Routing and Forwarding* (VRF) virtualization allows to implement multiple routing systems in one physical IP router. The router, in addition to the classic IP routing system still operational, will be hosting virtual routers with their own sets its IP interfaces and routing tables. VRF introduces network isolation between individual virtual routers, so the same network IP addressing could be re-used in each virtual system. These routers are completely independent of each other.. The first task is to run in the VRF in Cisco router and check if it's possible to use the same IP network in different virtual routers.
2. Prepare a Cisco router and connect it with two computers, as shown below.



3. Create two instances of VRF (virtual routers):
   Router (config) #ip VRF one
   Router (config-VRF) #exit
   Router (config) #ip VRF two
   Router (config-VRF) #exit

   For each instance, define the so-called VRF route distinguisher, expressed as: <Autonomous System Identifier>: < exiting identifier (from the MPLS cloud)>. Route distinguisher is required to implement MPLS VRF technology with a use of locally defined VRF. In some versions of Cisco IOS route distinguishers are required when using the VRF alone. Autonomous System IDs values may now be fictional, but should be the same on both sides. Exiting identifiers should vary on both sides:
   Router (config) #ip VRF one
   Router (config-vrf) #rd 65500:1
   Router (config) #ip VRF two
   Router (config-vrf) #rd 65500:2

4. Assign two router physical interfaces to different VRFs:

Router(config)#int fa0 / 0
Router (config-if) #ip VRF forwarding one
Router (config) #int fa0 / 1
Router (config-if) #ip VRF forwarding two

Define an IP addresses of two interfaces to be the same (now it's possible, since interfaces are assigned to different and separated VRFs and there is no overlapping between IP networks):
Router (config) #int fa0 / 0
Router (config-if) #ip address 200.200.200.1
Router (config) #int fa0 / 1
Router (config-if) #ip address 200.200.200.1

5. Check the VRF routing tables configuration and the contents of VRFs:
   Router # show ip route VRF two
   Router # show ip route VRF one

   Router # show ip VRF one
   Router # show ip VRF two

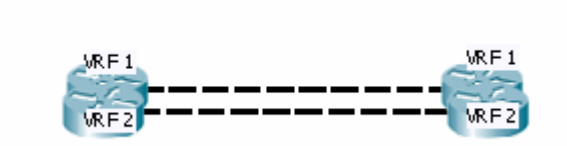
6. After configuring (the same way), test the communication doing "ping" checks. Note - due to the presence of VRF, ping tests should be run on the respective virtual routers:
   Router# ping VRF one 200.200.200.2
   Router# ping VRF two 200.200.200.2
   Check (Wireshark) to which computers ICMP datagrams were send (in both the cases above).

   Complete configuration of the router:

```
ip vrf one
exit
ip vrf two
exit

int fa0/0
ip vrf forwarding one
ip address 200.200.200.2 255.255.255.0
no sh
exit

int fa0/1
ip vrf forwarding two
ip address 200.200.200.2 255.255.255.0
no sh
exit
```

Task B: Dynamic routing protocols in the VRF instances.

1. Unplug the router from the computer add second router, connect it with the two Ethernet cables, as shown:



2. Change the IP addressing in both IP Ethernet interfaces, in order to create two identical IP networks, embedded in two different VRFs, e.g.:

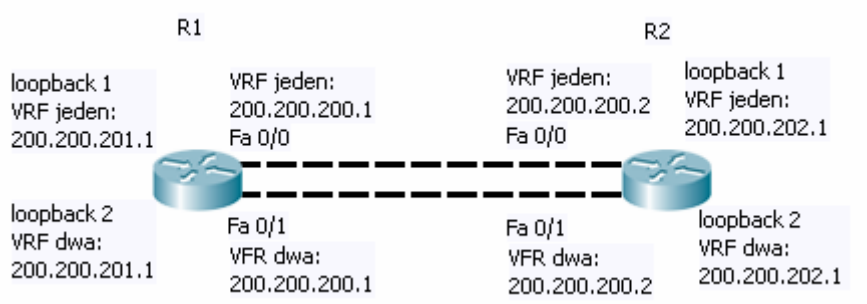
R1 (config) #int fa0 / 0
R1 (config-if) #ip VRF forwarding one
R1 (config-if) #ip address 200.200.200.1
R1 (config) #int fa0 / 1
R1 (config-if) #ip VRF forwarding two
R1 (config-if) #ip address 200.200.200.1

R2 (config) #int fa0 / 0
R2 (config-if) #ip VRF forwarding one
R2 (config-if) #ip address 200.200.200.2
R2 (config) #int fa0 / 1
R2 (config-if) #ip VRF forwarding two
R2 (config-if) #ip address 200.200.200.2

Test the configuration (ICMP is sending datagrams from each VRF instance to another, over a separated cable)
R1 # ping VRF one 200.200.200.2
R1 # ping VRF two 200.200.200.2

Add to both routers two loopback interfaces, adding these loopbacks the first and the second VRF respectively. These interfaces will be used to test the dynamic routing process (providing simulated networks to be used in dynamic routing processes). Assign IP addresses to networks (s usual, all networks in particular VRF must nave unique prefixes):
R1 (config) #int loopback 1
R1 (config-if) #ip VRF forwarding one
R1 (config-if) #ip address 200.200.201.1, 255.255.255.0
R1 (config) #int loopback 2
R1 (config-if) #ip VRF forwarding two
R1 (config-if) #ip address 200.200.201.1, 255.255.255.0

R2 (config) #int loopback 1
R2 (config-if) #ip VRF forwarding one
R2 (config-if) #ip address 200.200.202.1, 255.255.255.0
R2 (config) #int loopback 2
R2 (config-if) #ip VRF forwarding two
R2 (config-if) #ip address 200.200.202.1, 255.255.255.0



3. Create two OSPF processes in each router. These processes will be assigned to different VRFs:

Create an OSPF process in router R1, for the VRF one:
R1*(config) # router ospf 10 VRF one*
where 10 is just unique OSPF process ID.
Then assign networks to the OSPF process (only those consisting interfaces belonging to the VRF one):
R1*(config-router) #network 200.200.200.0 0.0.0.255 area 0*
R1*(config-router) #network 200.200.201.0 0.0.0.255 area 0*

Create second OSPF process in router R1 for VRF two, assigning networks having interfaces belonging to that VRF:
R1 (config) # router ospf 20 VRF two
where 20 (a process ID) should be different, comparing one previously defined one:
R1*(config-router) #network 200.200.200.0 0.0.0.255 area 0*
R1*(config-router) #network 200.200.201.0 0.0.0.255 area 0*

The router R2 define an analogous formula:

R2*(config)#router ospf 10 VRF one*
R2 *(config-router) #net 200.200.200.0 0.0.0.255 area 0*
R2*(config-router) #net 200.200.202.0 0.0.0.255 area 0*
R2*(config)#router ospf 20 VRF two*
R2*(config-router) #net 200.200.200.0 0.0.0.255 area 0*
R2*(config-router) #net 200.200.202.0 0.0.0.255 area 0*

In some implementations of Cisco IOS, OSPF processes running in the VRFs will have problems with selecting a unique identifier (router-id) for OSPF node. In such cases declare these values manually, e.g.:
Router (config)# router ospf one VRF 10
Router (config-router) # router-id 5.6.7.8

4. Check OSPF interfaces status and OSPF neighbors in both routers:
Router # show ip ospf 10 interface
Router # show ip ospf 10 neighbor
Router # show ip ospf 20 interface
Router # show ip ospf 20 neighbor

Verify the OSPF database in both routes nab both VRFs:

Router # show ip ospf 10 database
Router # show ip ospf 20 database

In case of problems, run OSPF diagnostics:

Router # debug ip ospf event

Check the IP routing tables in both VRFs and finally - communication through the routes established by OSPF processes:

Router # show ip route VRF two
Router # show ip route VRF one
Router # ping VRF one source 200.200.202.1 200.200.201.1
Router # ping VRF two source 200.200.201.1 200.200.202.1

Complete configuration:

R1:

```
ip vrf one
ip vrf two

int loopback 1
 ip vrf forwarding one
 ip address 200.200.201.1 255.255.255.0
exit

int loopback 2
 ip vrf forwarding two
 ip address 200.200.201.1 255.255.255.0
exit

int fa0/0
 ip vrf forwarding one
 ip address 200.200.200.1 255.255.255.0
 no sh
exit

int fa0/1
 ip vrf forwarding two
 ip address 200.200.200.1 255.255.255.0
 no sh
exit

router ospf 10 vrf one
  network 200.200.200.0 0.0.0.255 area 0
  network 200.200.201.0 0.0.0.255 area 0

router ospf 20 vrf two
  network 200.200.200.0 0.0.0.255 area 0
  network 200.200.201.0 0.0.0.255 area 0
```

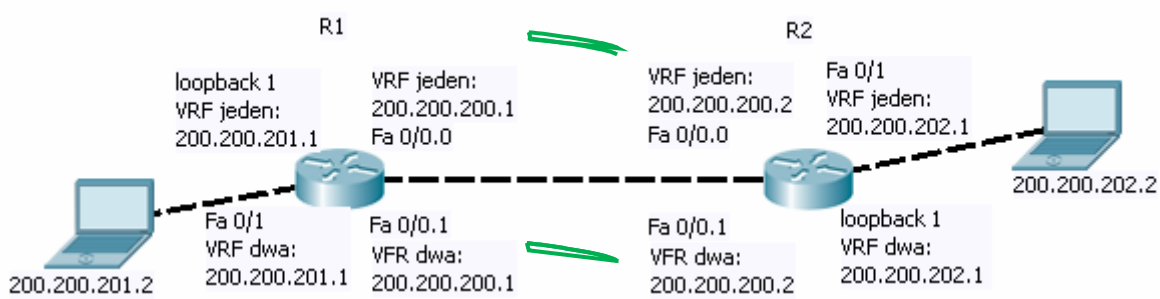
R2:

```
ip vrf one
ip vrf two

int loopback 1
 ip vrf forwarding one
 ip address 200.200.202.1 255.255.255.0
exit

int loopback 2
 ip vrf forwarding two
 ip address 200.200.202.1 255.255.255.0
exit

int fa0/0
 ip vrf forwarding one
 ip address 200.200.200.2 255.255.255.0
 no sh
exit

int fa0/1
 ip vrf forwarding two
 ip address 200.200.200.2 255.255.255.0
 no sh
exit

router ospf 10 vrf one
  network 200.200.200.0 0.0.0.255 area 0
  network 200.200.202.0 0.0.0.255 area 0

router ospf 20 vrf two
 network 200.200.200.0 0.0.0.255 area 0
 network 200.200.202.0 0.0.0.255 area 0
```

Task C: MultiVRF – multiple VRFs over one physical infrastructure

1. The current task is to create a physical Ethernet link supporting multiple VRFs between a pair of Cisco routers. Links between these routers will be divided into logical sub-links, assigned to particular VRFs and ending in router's sub-interfaces.

2. Prepare two Cisco routers and connect them with only one physical Ethernet link. Remove possible earlier configuration from these routers.



   The link between routers can be created using any technology that allows creation of sub-IP interfaces over its own addressing space (e.g. Ethernet, Serial encapsulation in frame-relay back-to-back, DSL, ATM, ATM-IMA, E-carrier / T - carrier, etc.). The present example will be using the Ethernet, as before. IP sub-interfaces will be used to connect each corresponding VRF, defined in routers to respective VRF in another router.

3. In both routers create two VRF instances:
   R1 (config) #ip VRF one
   R1 (config-VRF) #rd 65500:1
   R2 (config) #ip VRF two
   R2 (config-VRF) #rd 65500:1

4. In router's interfaces leading to the opposite router create sub-interfaces in a number same as VRF instances count. Configure the IEEE 802.1Q encapsulation for all sub-interfaces using various 802.1Q tag values (for each pair of sub-interfaces connected in both routers). Define (even) the same IP addresses for these sub-interfaces (each VRF instance is isolated, so no network overlapping occurs):

   R1 (config) # int fa0 / 0.1
   R1 (config-if) #ip VRF forwarding one
   R1 (config-if) #encapsulation dot1q 4
   R1 (config-if) #ip address 200.200.200.1, 255.255.255.0
   R1 (config) # int fa0 / 0.2
   R1 (config-if) #ip VRF forwarding two
   R1 (config-if) #encapsulation dot1q 2
   R1 (config-if) #ip address 200.200.200.1, 255.255.255.0

   R2 (config) # int fa0 / 0.1

R2 (config-if) #ip VRF forwarding one
R2 (config-if) #encapsulation dot1q 4
R2 (config-if) #ip address 200.200.200.2, 255.255.255.0
R2 (config) # int fa0 / 0.2
R2 (config-if) #ip VRF forwarding two
R2 (config-if) #encapsulation dot1q 2
R2 (config-if) #ip address 200.200.200.2, 255.255.255.0

5. Turn the interfaces on:
   R1 (config) #int fa0 / 0
   R1 (config-if) #no sh

6. Configure router's loopback interfaces to form a test IP networks linking in each VRF, for example:

   R1 (config) #int loopback 1
   R1 (config-if) #ip VRF forwarding one
   R1 (config-if) #ip address 200.200.201.1 255.255.255.0
   R2 (config) #int loopback 1
   R2 (config-if) #ip VRF forwarding two
   R2 (config-if) #ip address 200.200.202.1 255.255.255.0

7. Additionally, connect two computers and configure the IP networks there as well. Ethernet interfaces routers, connecting these computers, assign to the selected VRF, providing uniqueness of the IP network addressing in each VRF instance:

   R1 (config) #int fa 0/1
   R1 (config-if) #ip VRF forwarding one
   R1 (config-if) #ip address 200.200.201.1, 255.255.255.0
   R2 (config) #int fa 0/1
   R2 (config-if) #ip VRF forwarding two
   R2 (config-if) #ip address 200.200.202.1, 255.255.255.0

   For both VRF instances configure some IP routing system. Les's say – it will be static in VRF one, and dynamic (EIGRP) – in VRF two:

   R1 (config) #ip route VRF one 200.200.202.0, 255.255.255.0 200.200.200.2
   R2 (config) #ip route VRF one 200.200.201.0, 255.255.255.0 200.200.200.1

   R1 (config) #router EIGRP 10
   R1 (config-router) # address-family ipv4 VRF two
   R1 (config-router) # autonomous system-10
   R1 (config-router-af) #network 200.200.200.0 0.0.0.255
   R1 (config-router-af) #network 200.200.201.0 0.0.0.255
   R1 (config-router-af) #no auto-summary
   R1 (config-router-af) # exit-address-family
   R1 (config-router) #exit

   R2 (config) #router EIGRP 10
   R2 (config-router) # address-family ipv4 VRF two

R2 (config-router) # autonomous system-10
R2 (config-router-af) #network 200.200.200.0 0.0.0.255
R2 (config-router-af) #network 200.200.202.0 0.0.0.255
R2 (config-router-af) #no auto-summary
R2 (config-router-af) # exit-address-family
R2 (config-router) #exit

8. Finally, turn all remaining Ethernet interfaces on in routers (no shut)

Complete router configuration:

R1:

```
ip vrf one
rd 65500:1
exit
ip vrf two
rd 65500:2
exit

int fa 0/0
 no sh
exit

int fa0/0.1
 ip vrf forwarding one
 encapsulation dot1q 4
 ip address 200.200.200.1 255.255.255.0
exit
int fa0/0.2
 ip vrf forwarding two
 encapsulation dot1q 2
 ip address 200.200.200.1 255.255.255.0
exit

int loopback 1
 ip vrf forwarding one
 ip address 200.200.201.1 255.255.255.0
 no sh
exit

int fa0/1
 ip vrf forwarding two
 ip address 200.200.201.1 255.255.255.0
 no sh
exit

router eigrp 10
 address-family ipv4 vrf two
  autonomous-system 10
  network 200.200.200.0 0.0.0.255
  network 200.200.201.0 0.0.0.255
  no auto-summary
  exit-address-family
 exit

ip route vrf one 200.200.202.0 255.255.255.0
200.200.200.2
```

R2:

```
ip vrf one
rd 65500:3
exit
ip vrf two
rd 65500:4
exit

int fa 0/0
 no sh
exit

int fa0/0.1
 ip vrf forwarding one
 encapsulation dot1q 4
 ip address 200.200.200.2 255.255.255.0
exit
int fa0/0.2
 ip vrf forwarding two
 encapsulation dot1q 2
 ip address 200.200.200.2 255.255.255.0
exit

int fa0/1
 ip vrf forwarding one
 ip address 200.200.202.1 255.255.255.0
 no sh
exit

int loopback 1
 ip vrf forwarding two
 ip address 200.200.202.1 255.255.255.0
 no sh
exit

router eigrp 10
 address-family ipv4 vrf two
 autonomous-system 10
  network 200.200.200.0 0.0.0.255
  network 200.200.202.0 0.0.0.255
  no auto-summary
  exit-address-family
 exit

ip route vrf one 200.200.201.0 255.255.255.0
200.200.200.1
```

9. Check both the routing tables for each VRF:
   R1 # show ip route VRF one
   R1 # show ip route VRF two
   R2 # show ip route VRF one
   R2 # show ip route # two

   and check the communication:

   R1 # ping VRF one 200.200.202.1 source 200,200. 201.1
   R1 # ping VRF two source 200.200.202.2 200.200.201.1
   R2 # ping VRF one source 200.200.201.1 200.200.202.1
   R2 # ping VRF two source 200.200.201.2 200.200.202.1

   Additional diagnostics (EIGRP in VRFs):

   R1 # show ip eigrp VRF two interfaces
   R1 # show ip eigrp neighbors VRF two
   R1 # show ip eigrp topology VRF two
   R1 # debug ip eigrp summary VRF two
   R1 # debug ip eigrp notifications VRF two

   Neighbors and reset the EIGRP routing table IP VRF:

   R1 # clear ip eigrp neighbors two VRF
   *R1 # clear ip route * VRF two*

10. Check the communication between both computers and loopbacks in all VRFs.
    Try to move one router's interface to another VRF
    R2 (config) #int loop 1
    R2 (config-if) #ip VRF forwarding one
    Caution: It is necessary to redefine the IP address after change - it now belongs to another VRF instance, and was cleared automatically after VRF change to avoid possible overlapping:
    R2 (config0if) #ip address 200.200.202.3, 255.255.255.0

    Check connectivity (regarding that interface) again.