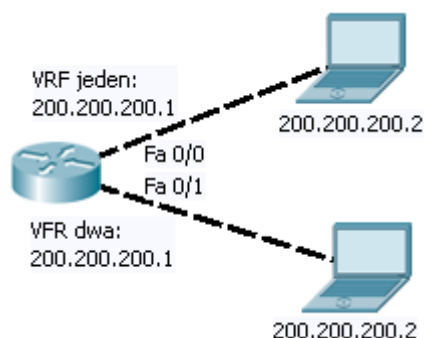


Tematyka:

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

Zadanie A: Uruchomienie VRF

1. *Virtual Routing and Forwarding (VRF)* umożliwia wprowadzenie wirtualizacji procesu rutowania IP. W routerze, poza klasycznym systemem rutowania IP zostaną zainstalowane routery wirtualne, posiadające własne i przypisane do siebie zbiory interfejsów IP i własne tablice rutowania. VRF wprowadza izolację sieci przetwarzanych przez poszczególne routery, więc adresacje IP sieci komunikowanych przez te routery są całkowicie niezależne od siebie. Mogą się także pokrywać (w poszczególnych procesach rutujących może pojawić się ta sama sieć IP). Pierwsze zadanie polega na uruchomieniu VRF w routerze Cisco i wykazaniu, że możliwe jest posługiwanie się tą samą siecią IP w różnych routerach wirtualnych.
2. Przygotuj do pracy jeden router Cisco i połącz go okablowaniem z dwoma stacjami PC, jak na rysunku:



3. Stwórz dwie instancje VRF (routery wirtualne):

```
Router(config)#ip vrf jeden
```

```
Router(config-vrf)#exit
```

```
Router(config)#ip vrf dwa
```

```
Router(config-vrf)#exit
```

Dla każdej instancji VRF zdefiniuj tzw. *route distinguisher* zapisany w formacie:

<Identyfikator Systemu Autonomicznego>:<Identyfikator wyjścia z chmury MPLS>. *Route distinguisher* wymagany jest przy integracji VRF z technologią MPLS. W niektórych wersjach systemu Cisco IOS jego zdefiniowanie jest jednak wymagane już przy użytkowaniu VRF Lite. Wartość Identyfikatora Systemu Autonomicznego może być fikcyjna lecz powinna być identyczna po obydwu stronach, wartość Identyfikatora wyjścia z chmury MPLS – różna po innych stronach:

```
Router(config)#ip vrf jeden
```

```
Router(config-vrf)#rd 65500:1
```

```
Router(config)#ip vrf dwa
```

```
Router(config-vrf)#rd 65500:2
```

4. Przypisz dwa interfejsy rutera kolejno do dwóch VRF, np.:

```
Router(config)#int fa0/0
Router(config-if)#ip vrf forwarding jeden
Router(config)#int fa0/1
Router(config-if)#ip vrf forwarding dwa
```

Zdefiniuj IDENTYCZNE adresy IP w obydwu interfejsach. Zauważ, że po przypisaniu interfejsów do różnych instancji VRF jest to możliwe (nie występuje overlapping sieci IP)

```
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. Sprawdź konfigurację VRF oraz zawartość tablic rutowania routerów wirtualnych:

```
Router#show ip vrf jeden
Router#show ip vrf dwa
Router#show ip route vrf dwa
Router#show ip route vrf jeden
```

6. Po skonfigurowaniu (także identycznie) stacji PC przetestuj komunikację pomiędzy nimi, a routerem. Uwaga - z uwagi na obecność VRF testy ping należy uruchamiać w odpowiednich routerach wirtualnych, np.:

```
Router#ping vrf jeden 200.200.200.2
Router#ping vrf dwa 200.200.200.2
```

Sprawdź w stacjach PC (*Wireshark*) do której z nich wysłane zostały datagramy ICMP w obydwu powyższych przypadkach.

Gotowa konfiguracja rutera:

```
ip vrf jeden
exit
ip vrf dwa
exit

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

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

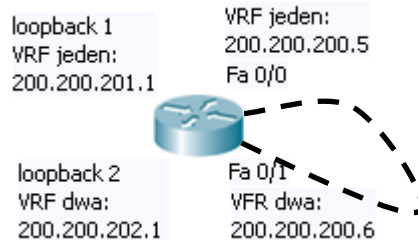
Zadanie B: Protokoły rutowania dynamicznego w instancjach VRF.

1. Odłącz od rutera stacje PC. Zmień adresację IP obydwu interfejsów Ethernet rutera tak, aby adresy były różne, ale interfejsy znajdowały się w tej samej sieci IP, np.:

```
Router(config)#int fa0/0
```

```
Router(config-if)#ip vrf forwarding jeden
Router(config-if)#ip address 200.200.200.6
Router(config)#int fa0/1
Router(config-if)#ip vrf forwarding dwa
Router(config-if)#ip address 200.200.200.5
```

Połącz obydwie interfejsy kablem z przeplotem jak na rysunku:



Przetestuj konfigurację (wysyłając datagramy ICMP z poszczególnych instancji VRF do innej instancji VRF):

```
Router#ping vrf jeden 200.200.200.5
Router#ping vrf dwa 200.200.200.6
```

2. Dodaj dwa interfejsy loopback przydzielając je odpowiednio do dwóch instancji VRF. Interfejsy te będą wykorzystywane do testowania procesu routowania dynamicznego (tworząc sieci przekazywane przez te procesy). Interfejsom nadaj adresy IP w unikatowych sieciach:

```
Router(config)#int loopback 1
Router(config-if)#ip vrf forwarding jeden
Router(config-if)#ip address 200.200.201.1 255.255.255.0
Router(config)#int loopback 2
Router(config-if)#ip vrf forwarding dwa
Router(config-if)#ip address 200.200.202.1 255.255.255.0
```

3. Doświadczenie będzie polegało na stworzeniu dwóch procesów OSPF w jednym routerze fizycznym. Procesy te będą przypisane do różnych VRF. Ponieważ adresy IP interfejsów Ethernet routera należą do tej samej sieci (choć interfejsy są przypisane do różnych VRF) po ich fizycznym połączeniu możliwe będzie skomunikowanie dwóch procesów OSPF pochodzących z różnych VRF.

Stwórz proces OSPF dla pierwszego VRF:

```
Router(config)#router ospf 10 vrf jeden
```

gdzie 10 to dowolnie wybrany unikatowy identyfikator procesu OSPF.

Przydziel do procesu OSPF tylko te sieci, które zawierają interfejsy należące do pierwszego VRF:

```
Router(config-router)#net 200.200.200.0 0.0.0.255 area 0
Router(config-router)#net 200.200.201.0 0.0.0.255 area 0
```

Stwórz w routerze drugi proces OSPF dla drugiego – przydzielając do niego sieci, które zawierają interfejsy należące do drugiego VRF:

```
Router(config)#router ospf 20 vrf dwa
```

gdzie 20 to inny niż w przypadku pierwszego procesu identyfikator procesu OSPF.

```
Router(config-router)#net 200.200.200.0 0.0.0.255 area 0
Router(config-router)#net 200.200.202.0 0.0.0.255 area 0
```

4. Sprawdź przy użyciu komend diagnostycznych OSPF, czy nastąpiło połączenie pomiędzy dwoma ruterami OSPF znajdującymi się w jednym ruterze fizycznym
- ```
Router# show ip ospf 10 interface brief
Router# show ip ospf 10 neighbor
Router# show ip ospf 20 interface brief
Router# show ip ospf 20 neighbor
```

Sprawdź, czy wiedza o trasach między dwoma ruterami wirtualnymi została wymieniona:

```
Router# show ip ospf 10 interface topology
Router# show ip ospf 20 interface topology
```

W razie problemów uruchom diagnostykę OSPF:

```
Router# debug ip ospf event
```

W przypadku niektórych implementacji Cisco IOS procesy OSPF uruchomione na tym samym urządzeniu będą miały problem z wybraniem rutera OSPF

*Designated router* dla sieci typu *broadcast* (którą jest np. Ethernet). *Designated router* nie występuje w sieciach typu *point-to-point*. W przypadku problemów można zadeklarować w routerze OSPF stosowanie zachowania jak dla sieci *point-to-point*:

```
Router(config)# int fa0/0
Router(config-if)# ip ospf network point-to-point
Router(config)# int fa0/1
Router(config-if)# ip ospf network point-to-point
```

Uwaga! Zachowanie to definiujemy indywidualnie dla interfejsów (one służą do przyłączania sieci). Musi być ono zunifikowane dla obydwu z nich

Sprawdź tablice rutowania IP dla obydwu VRF i możliwość komunikowania się przez po trasy ustalonych przez OSPF:

```
Router# show ip route vrf dwa
Router# show ip route vrf jeden
Router# ping vrf jeden 200.200.202.1 source 200.200.201.1
Router# ping vrf dwa 200.200.201.1 source 200.200.202.1
```

Gotowa konfiguracja rutera:

```
ip vrf jeden
ip vrf dwa

int fa0/0
ip vrf forwarding jeden
ip address 200.200.200.1 255.255.255.0
no sh
exit
int fa0/1
ip vrf forwarding dwa
ip address 200.200.200.2 255.255.255.0
no sh
exit
```

```
int lo0 0
ip vrf forwarding jeden
ip address 200.200.201.1 255.255.255.0
exit
```

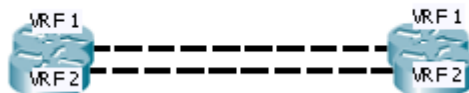
```
int lo0 1
ip vrf forwarding dwa
ip address 200.200.202.1 255.255.255.0
exit
```

```
router ospf 10 vrf jeden
net 200.200.200.0 0.0.0.255 area 0
net 200.200.201.0 0.0.0.255 area 0
exit
```

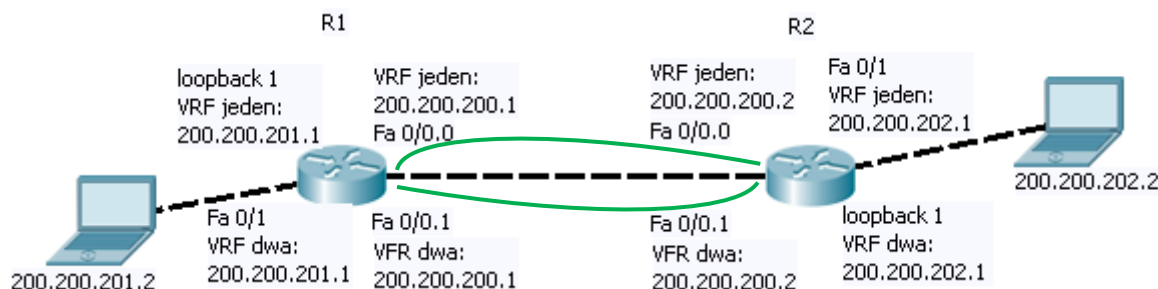
```
router ospf 20 vrf dwa
net 200.200.200.0 0.0.0.255 area 0
net 200.200.202.0 0.0.0.255 area 0
exit
```

## Zadanie C: MultiVRF – VRF w sieciach wielosegmentowych.

1. Bieżące zadanie ma na celu stworzenie sieci VRF sięgającej do wielu ruterów. Łąca między ruterami będą dzielone na logiczne pod-interfejsy – stanowiące połączenia poszczególnych VRF.



2. Przygotuj dwa routery Cisco łącząc je okablowaniem jak na rysunku i likwidując w nich ewentualną wcześniejszą konfigurację.



Łąca pomiędzy routerami może być utworzone z użyciem dowolnej technologii pozwalającej na tworzenie pod-interfejsów IP z własną adresacją (np. Ethernet, Serial z enkapsulacją Frame-Relay back-to-back, DSL, ATM, ATM-IMA, E-carrier/T-carrier itp.). Prezentowany przykład będzie wykorzystywał Ethernet. Pod-interfejsy IP zostaną wykorzystane jako łącza poszczególnych odpowiadających sobie routerów VRF znajdujących się w różnych routerach fizycznych..

3. W obydwu routerach stwórz po dwie instancje VRF:  
`R1(config)#ip vrf jeden`  
`R1(config-vrf)#rd 65500:1`  
`R2(config)#ip vrf dwa`  
`R2(config-vrf)#rd 65500:1`
4. W obydwu routerach w interfejsie prowadzącym do przeciwnego routera stwórz pod-interfejsy w liczbie odpowiadającej liczbie instancji VRF, jakie mają być komunikowane. Skonfiguruj enkapsulację IEEE 802.1Q pod-interfejsach stosując różne wartości tagu 802.1Q dla każdej pary pod-interfejsów. Nadaj pod-interfejsom adresy IP tak, aby w każdej instancji VRF pasowały do tej samej sieci IP:

```
R1(config)#int fa0/0.1
R1(config-if)#ip vrf forwarding jeden
R1(config-if)#encapsulation dot1q 1
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 dwa
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 jeden
R2(config-if)#encapsulation dot1q 1
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 dwa
R2(config-if)#encapsulation dot1q 2
R2(config-if)#ip address 200.200.200.2 255.255.255.0

```

5. Włącz interfejsy IP w ruterach:

```

R1(config)#int fa0/0
R1(config-if)#no sh

```

6. Skonfiguruj interfejsy loopback w ruterach, tworząc testowe sieci IP przypisując je do poszczególnych VRF, na przykład:

```

R1(config)#int loopback 1
R1(config-if)#ip vrf forwarding jeden
R1(config-if)#ip address 200.200.201.1 255.255.255.0
R2(config)#int loopback 1
R2(config-if)#ip vrf forwarding dwa
R2(config-if)#ip address 200.200.202.1 255.255.255.0

```

7. Dodatkowo podłącz do ruterów dwie stacje PC i skonfiguruj tam sieci IP. Interfejsy Ethernet ruterów także przypisz do wybranych VRF pilnując unikatowości adresacji sieci IP w każdej instancji VRF:

```

R1(config)#int fa 0/1
R1(config-if)#ip vrf forwarding dwa
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 jeden
R2(config-if)#ip address 200.200.202.1 255.255.255.0

```

Dla obydwu instancji VRF skonfiguruj system rutowania IP. W przykładzie dla VRF jeden zostały zdefiniowane statyczne reguły rutowania IP, dla VRF dwa – EIGRP:

```

R1(config)#ip route 200.200.202.0 255.255.255.0 200.200.200.2 vrf jeden
R2(config)#ip route 200.200.201.0 255.255.255.0 200.200.200.1 vrf jeden

```

```

R1(config)#router eigrp 10
R1(config-router)#address-family ipv4 vrf dwa
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 dwa
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. Na koniec włącz wszystkie skonfigurowane wcześniej interfejsy Ethernet (*no shut*)

Gotowa konfiguracja ruterów:

R1:

```

ip vrf jeden
rd 65500:1
exit
ip vrf dwa
rd 65500:2
exit

int fa 0/0
no sh
exit

int fa0/0.1
ip vrf forwarding jeden
encapsulation dot1q 1
ip address 200.200.200.1 255.255.255.0
exit
int fa0/0.2
ip vrf forwarding dwa
encapsulation dot1q 2
ip address 200.200.200.1 255.255.255.0
exit

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

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

router eigrp 10
address-family ipv4 vrf dwa
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 jeden 200.200.202.0 255.255.255.0
200.200.200.2

```

R2:

```

ip vrf jeden
rd 65500:3
exit
ip vrf dwa
rd 65500:4
exit

int fa 0/0
no sh
exit

int fa0/0.1
ip vrf forwarding jeden
encapsulation dot1q 1
ip address 200.200.200.2 255.255.255.0
exit
int fa0/0.2
ip vrf forwarding dwa
encapsulation dot1q 2
ip address 200.200.200.2 255.255.255.0
exit

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

loopback 1
ip vrf forwarding dwa
ip address 200.200.202.1 255.255.255.0
no sh
exit

router eigrp 10
address-family ipv4 vrf dwa
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 jeden 200.200.201.0 255.255.255.0
200.200.200.1

```

9. Sprawdź funkcjonowanie obydwu sieci VRF oraz zawartość tablic rutowania IP dla poszczególnych VRF:

```
R1#show ip route vrf jeden
R1#show ip route vrf dwa
R2#show ip route vrf jeden
R2#show ip route vrf dwa
R1#ping vrf jeden 200.200.202.1 source 200.200.201.1
R1#ping vrf dwa 200.200.202.2 source 200.200.201.1
R2#ping vrf jeden 200.200.201.1 source 200.200.202.1
R2#ping vrf dwa 200.200.201.2 source 200.200.202.1
```

Diagnostyka EIGRP w VRF:

```
R1#show ip eigrp vrf dwa interfaces
R1#show ip eigrp vrf dwa neighbors
R1#show ip eigrp vrf dwa topology
R1#debug ip eigrp vrf dwa summary
R1#debug ip eigrp vrf dwa notifications
```

Reset sąsiadów EIGRP i tablicy rutowania IP w VRF:

```
R1#clear ip eigrp vrf dwa neighbors
R1#clear ip route vrf dwa *
```

10. Sprawdź komunikację pomiędzy stacjami PC (Wireshark). Czy jest możliwa? Przenieś jeden interfejs w ruterze, prowadzący do stacji PC, do innego VRF:

```
R2(config)#int fa 0/1
R2(config-if)#ip vrf forwarding jeden
```

Uwaga! Konieczne jest ponowne zdefiniowanie adresu IP – dodatkowo ponieważ teraz przynależy do z innej instancji VRF konieczne jest zdefiniowanie tego adresu tak, aby nie kolidował z loopback 1 czy PC:

```
R2(config0if)#ip address 200.200.202.3 255.255.255.0
```