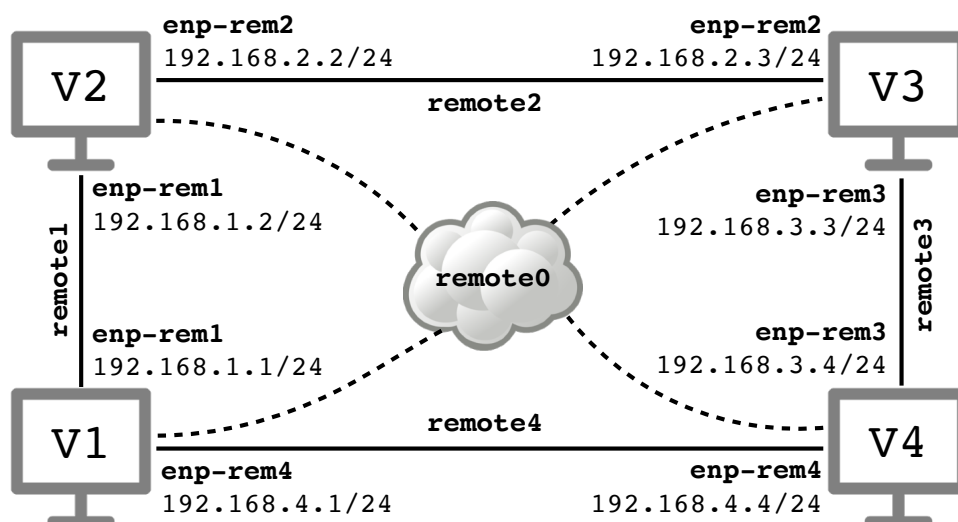


Warsztaty z Sieci komputerowych

Lista 4

Konfiguracja początkowa

Celem tej części jest osiągnięcie topologii sieci jak na rysunku poniżej.

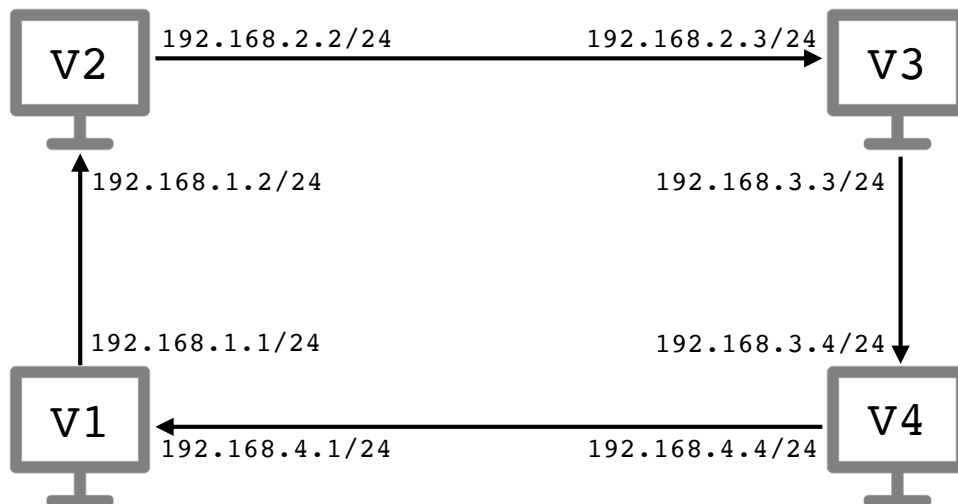


- ▶ Na każdej z czterech maszyn wirtualnych *Virbian1–Virbian4* powinny być dwa interfejsy połączone z odpowiednimi wirtualnymi sieciami *remote1–remote4*. Interfejs połączony z siecią *remotei* należy w maszynie wirtualnej nazwać *enp-remi* tak jak na rysunku powyżej. Dodatkowo na każdej maszynie powinien być interfejs nazwany *enp-all* (niezaznaczony rysunku) połączony (linie przerywane) z wirtualną siecią *remote0*.
- ▶ Na każdej maszynie aktywuj dwa interfejsy sieciowe *enp-remi*; interfejsy *enp-all* pozostaw nieaktywne. Aktywnym interfejsom przypisz adresy IP jak na rysunku powyżej. Zauważ, że karty podpięte do sieci *remotei* mają adresy IP z klasy 192.168.*i*.0/24.
- ▶ Poleceniem `ip route` sprawdź, że tablica routingu każdej maszyny zawiera dokładnie dwa wpisy dotyczące bezpośrednio połączonych z nią sieci. Sprawdź dostępność bezpośrednio połączonych maszyn poleceniem `ping`.

Tutorial #1

Uruchom Wiresharka na wszystkich maszynach nasłuchującego na wszystkich interfejsach.

- Będziemy teraz przekazywać wszystkie pakiety do celu zgodnie ze wskazówkami zegara. Jako bramę domyślną dla każdej maszyny ustaw maszynę, która jest następna w cyklu (tj. tak jak pokazują strzałki na rysunku poniżej). Pamiętaj, że bramą powinna być osiągalna bezpośrednio karta sieciowa: przykładowo bramą domyślną dla komputera *Virbian2* powinna być równa 192.168.2.3 a nie 192.168.3.3. Upewnij się, że tablica routingu każdej maszyny zawiera dokładnie trzy wpisy.



- Poleceniem `ping` sprawdź, że z każdej maszyny osiągalne są *wszystkie* interfejsy innych maszyn. Prześledź w Wiresharku ścieżki komunikatów *ICMP echo request* i *ICMP echo reply*. Czy zawsze suma tych ścieżek daje pełny cykl? Dlaczego?
- Z maszyny wirtualnej *Virbian1* wykonaj polecenie `traceroute` do adresów IP przypisanych interfejsom innych maszyn. Wykorzystaj opcję `-n`, aby przyspieszyć działanie programu (wyłącza ono odpytywanie DNS).
- Usuń trasy domyślne z tablic routingu. Sprawdź, że zmiany odniosły skutek wyświetlając bieżącą tablicę poleceniem `ip route`.

Tutorial #2

Skonfigurujemy teraz tablice routingu za pomocą protokołu routingu dynamicznego OSPF.

- Na każdej maszynie w pliku `/etc/frr/daemons` zmień wiersz zawierający `ospfd=no` na `ospfd=yes`. Następnie uruchom usługę `frr` poleceniem

```
Vi#> systemctl start frr
```

Aktywność usługi routingu dynamicznego RIP możesz sprawdzić poleceniem `systemctl status frr`: w wyświetlanych komunikatach powinien znajdować się napis `ospfd state -> up`.

- Na każdej maszynie uruchom konsolę `vttysh`. W razie potrzeby przypomnij sobie, jak z niej korzystać na podstawie listy zadań z poprzednich warsztatów. W konsoli `vttysh` wyświetl bieżącą tablicę routingu poleceniem

```
virbian# show ip route
```

Wyświetlane powinny być trasy do dwóch bezpośrednio podłączonych sieci `enp-remi`.

- Wejdź w tryb konfiguracji routingu OSPF poleceniami

```
virbian# configure terminal
virbian(config)# router ospf
```

Następnie włącz protokół OSPF dla sieci przyłączonych do interfejsów `enp-remi`. W tym celu wykonaj polecenia

```
virbian(config-router)# network 192.168.x.0/24 area 0
virbian(config-router)# network 192.168.y.0/24 area 0
```

Jeśli pomylisz się przy wpisywaniu, sieć można usunąć poleceniem

```
virbian(config-router)# no network adres_sieci area 0
```

- Wyjdź z trybu konfiguracji i wyświetl aktualną konfigurację poleceniami

```
virbian(config-router)# end
virbian# show running-config
```

Upewnij się, że są w niej informacje takie jak

```
router ospf
 network 192.168.x.0/24 area 0.0.0.0
 network 192.168.y.0/24 area 0.0.0.0
```

a następnie zapisz bieżącą konfigurację poleceniem

```
virbian# copy running-config startup-config
```

- Obejrzyj w Wiresharku przesyłane pakiety protokołu OSPF. Czy protokół OSPF korzysta z warstwy transportowej czy też jest osadzony bezpośrednio w pakietach IP?
- Okresowo wyświetlaj bieżącą tablicę routingu poleceniem `show ip route` w powłoce `vttysh` i poleceniem `ip route` w zwykłym terminalu.
- Polecenie `show ip route` w powłoce `vttysh` niestety nie wyświetla wyliczonej odległości do znanych sieci. Aby ją wyświetlić, wykonaj polecenia `show ip route 192.168.x.0/24` dla każdej z sieci `remotex` i wynikach znajdź wpisy `metric`.
- Po zakończeniu budowania tablic poleceniami `ping` i `traceroute` sprawdź osiągalność interfejsów wszystkich maszyn.
- Na wszystkich maszynach poleceniem `ip` aktywuj interfejs `enp-all` i przypisz mu adres `172.16.16.x/16`, gdzie $x \in \{1, 2, 3, 4\}$ jest numerem maszyny.

- W każdej maszynie włącz protokół OSPF również dla nowej sieci 172.16.0.0/16. W tym celu w trybie konfiguracji narzędzia `vttysh` wykonaj polecenia

```
virbian(config)# router ospf
virbian(config-router)# network 172.16.0.0/16 area 0
```

Zaobserwuj przesyłane pakiety OSPF i zmiany w tablicy routingu. Zauważ, że w przypadku równej odległości dostępnych jest kilka tras do celu w tablicy routingu (w programie `frr`) natomiast nie ma ich w tablicy przekazywania (w programie `ip`).

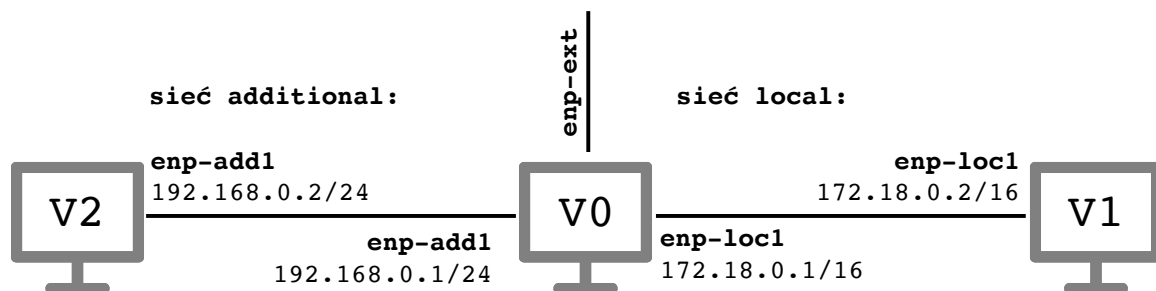
- Poleceniem `ip link set down dev interfejs` wyłącz niektóre z interfejsów, tak aby uzyskać w tablicy routingu ścieżkę o długości co najmniej 3 (wartość w polu `metric` powinna wynosić co najmniej 300). Czy umiesz uzyskać ścieżkę o długości 4?
- Zatrzymaj usługę OSPF poleceniem

```
Vi#> systemctl stop frr
```

Następnie zdekonguruj wszystkie interfejsy i wyłącz maszyny wirtualne.

Wyzwanie #1

Twoim pierwszym zadaniem jest osiągnięcie konfiguracji z rysunku poniżej.



W tym celu wykonaj następujące kroki.

- Utwórz maszyny wirtualne:
 - ▢ *Virbian0*, która będzie miała trzy karty sieciowe: pierwszą z domyślną konfiguracją sieciową (NAT), zaś drugą i trzecią połączoną z wirtualnymi sieciami `local` i `additional`;
 - ▢ *Virbian1* z jedną kartą sieciową połączoną z siecią `local` oraz
 - ▢ *Virbian2* z jedną kartą sieciową połączoną z siecią `additional`.
- Uruchom maszyny i nazwij ich interfejsy tak jak na rysunku powyżej. Uzyskaj konfigurację sieciową dla interfejsu `enp-ext` maszyny *Virbian0* za pomocą DHCP (polecenie `dhclient`). Sprawdź jaki jest uzyskany przez maszynę adres IP, będziemy go poniżej oznaczać przez `enp-ext`.
- Przypisz obu interfejsom `enp-loc1` i obu interfejsom `enp-add1` adresy IP z sieci 172.18.0.0/16 i 192.168.0.0/24 jak na rysunku. Sprawdź osiągalność interfejsów bezpośrednio połączonych maszyn poleceniem `ping`.

- Z maszyny *Virbian0* pingnij adres 8.8.8.8; zaobserwuj, że otrzymywana jest odpowiedź.
- Dodaj maszynę *Virbian0* jako bramę domyślną dla maszyny *Virbian1*. Co się wydarzy, gdy z maszyny *Virbian1* pingniesz teraz adres 8.8.8.8? A co jeśli pingniesz maszynę *Virbian2*?
- Skonfiguruj funkcję NAT na maszynie *Virbian0* dodając odpowiednie reguły za pomocą polecenia `nft`:

```
V0#> nft add table ip my_table
V0#> nft add chain ip my_table my_rules \
    {type nat hook postrouting priority srcnat\; }
V0#> nft add rule my_table my_rules ip saddr 172.18.0.0/16 snat to enp-ext
```

Jeśli pomylisz się przy wpisywaniu, wszystkie reguły można usunąć poleceniem `nft flush ruleset`. Bieżące reguły `nft` można wyświetlić poleceniem `nft list ruleset`. Ich obecna zawartość powinna być równa

```
table ip my_table {
    chain my_rules {
        type nat hook postrouting priority srcnat; policy accept;
        ip saddr 172.18.0.0/16 snat to enp-ext
    }
}
```

Dzięki tym regułom *Virbian0* przetworzy pakiety o adresach źródłowych z zakresu 172.18.0.0/16 przechodzące przez tę maszynę (i nie kończące na niej trasy). Adres źródłowy takich pakietów zostanie zmieniony na *enp-ext*.

- Sprawdź, że dzięki temu możesz pingnąć adres 8.8.8.8 z maszyny *Virbian1*.
Zaobserwuj na Wiresharku na maszynie *Virbian0* że pakiety od maszyny *Virbian1* (a także odpowiedzi dla niej) są rejestrowane dwukrotnie, tj. przed podmianą źródłowego adresu IP i po niej. Jakie adresy podmieniane są w pakietach z odpowiedziami?
- Sprawdź, że na maszynie *Virbian1* można również korzystać z innych usług internetowych (np. uruchamiając Firefoksa na stronie `example.com`).
- Spróbuj teraz pingnąć z maszyny *Virbian1* maszynę *Virbian2*. Co i dlaczego obserwujesz na Wiresharku na maszynie *Virbian2*?

Materiały do kursu znajdują się w systemie SKOS: <https://skos.ii.uni.wroc.pl/>.

Marcin Bieńkowski