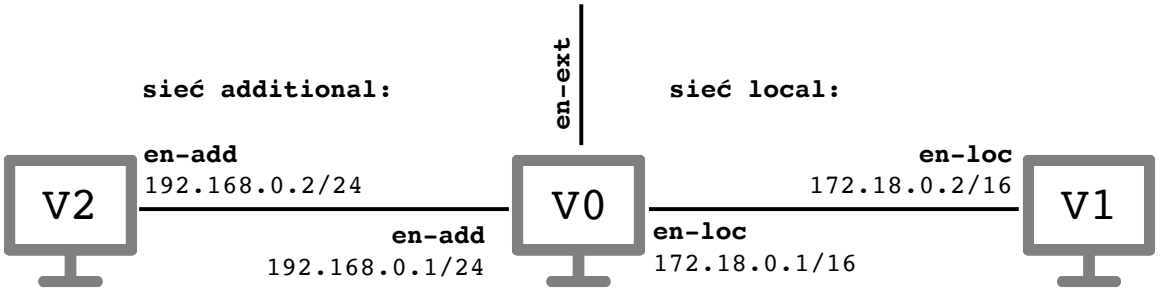
Warsztaty nr 4 z Sieci komputerowych

Konfiguracja początkowa 1

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



W tym celu wykonaj następujące kroki.

- Utwórz trzy maszyny:
  - *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`;
  - *Virbian2* z jedną kartą sieciową połączoną z siecią `additional`.
- Uruchom maszyny i nazwij ich interfejsy tak jak na rysunku powyżej. Aktywuj wszystkie interfejsy, ale nie przypisuj im jeszcze adresów IP.
- Uzyskaj konfigurację sieciową dla interfejsu `en-ext` maszyny *Virbian0* za pomocą DHCP (polecenie `dhcpcd` lub `dhclient`). Sprawdź jaki jest uzyskany przez maszynę adres IP, będziemy go poniżej oznaczać przez *`en-ext-IP`*.
- Włącz na wszystkich maszynach Wiresharka nasłuchującego na wszystkich interfejsach.

Tutorial 1

- Interfejsom `en-loc` przypisz adresy IP z sieci `172.18.0.0/16` jak na rysunku.
- Z maszyny *Virbian0* pingnij adres `1.1.1.1`; sprawdź, że otrzymywana jest odpowiedź.
- Dodaj maszynę *Virbian0* jako bramę domyślną dla maszyny *Virbian1*. Co się wydarzy, gdy z maszyny *Virbian1* pingniesz teraz adres `1.1.1.1`? Dlaczego?
- Skonfiguruj NAT na maszynie *Virbian0* dodając odpowiednie reguły za pomocą polecenia `nft`:

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

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 następująca:

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

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 *en-ext-IP*.

- Sprawdź, że teraz możesz pingnąć adres `1.1.1.1` z maszyny *Virbian1*. W Wiresharku na maszynie *Virbian0* zaobserwuj, że pakiety od maszyny *Virbian1* (a także odpowiedzi dla niej) są rejestrowane dwukrotnie, tj. przed podmianą źródłowego adresu IP i po niej.
  - Jakie jest pole TTL w takich pakietach przed i po podmianie?
  - Jakie adresy podmieniane są w pakietach z odpowiedziami?

  Sprawdź to również dla pakietów generowanych przez polecenie

  ```
  traceroute -N 1 -q 1 -n 1.1.1.1.
  ```

- Przypisz interfejsom `en-add` adresy IP z sieci `192.168.0.0/24` jak na rysunku powyżej. Z maszyny *Virbian1* pingnij maszynę *Virbian2*. Jaki jest adres źródłowy pakietów, które przychodzą do maszyny *Virbian2*?
- Problematyczna okazuje się niezbyt precyzyjna reguła NAT, która została zaaplikowana również dla pakietów, które nie wychodzą przez interfejs `en-ext`. Żeby to naprawić skasuj istniejące reguły poleceniem

  ```
  V0#> nft flush ruleset
  ```

  a następnie dodaj poprawne następującymi poleceniami:

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

- Sprawdź, że NAT dla pakietów wychodzących z maszyny *Virbian1* do internetu nadal działa, ale przy pinganiu maszyny *Virbian2* nie jest już zmieniany źródłowy adres IP.
- Dodaj trasę domyślną dla maszyny *Virbian2* wskazującą na maszynę *Virbian0*. Sprawdź, że maszyny *Virbian1* i *Virbian2* są teraz wzajemnie osiągalne za pomocą polecenia `ping`.
- Na maszynie *Virbian1* wpisz do pliku `/etc/resolv.conf` następujący wiersz:

  ```
  nameserver 1.1.1.1
  ```

  Spowoduje to, że zapytania DNS będą wysyłane do tego serwera.

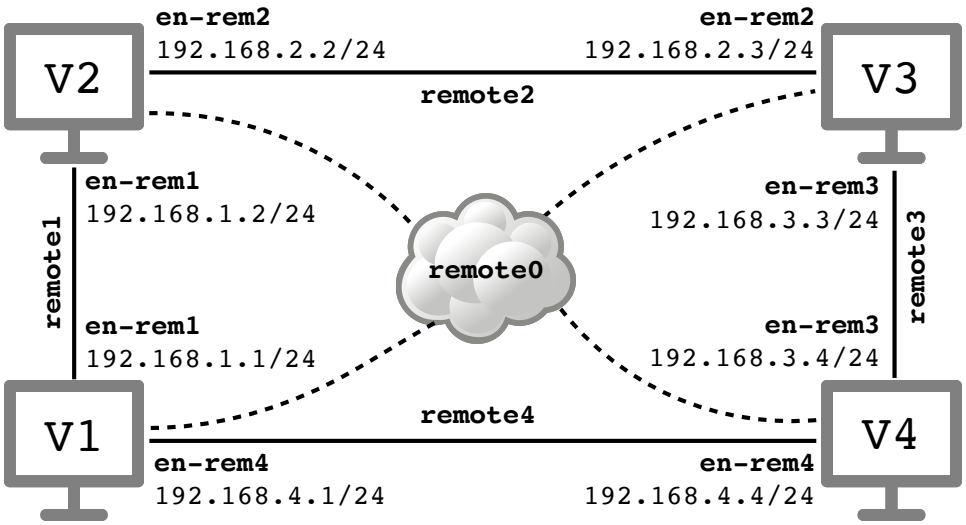
- Sprawdź, że na maszynie *Virbian1* można również korzystać z przeglądarki uruchamiając Firefoksa i otwierając w nim stronę `example.com` lub inną. Obejrzyj w Wiresharku przechodzące przez maszynę *Virbian0* zapytania HTTP (przed i po podmianie adresu źródłowego).

- Upewnij się, że źródłowe adresy IP są podmieniane w odpowiedni sposób.
- Z jakiego protokołu warstwy transportowej korzystają pakiety HTTP? Czy NAT zmienia w nich porty źródłowe bądź docelowe? Jeśli nie, to czy wyobrażasz sobie sytuację, w której NAT musiałby zmienić porty?
- Czy NAT zmienia w tych pakietach jakieś inne pola? Czy zmienia pola w nagłówku HTTP?

- Zdekonfiguruj interfejsy sieciowe i wyłącz maszyny.

Konfiguracja początkowa 2

Teraz osiągniemy konfigurację sieci z rysunku poniżej.



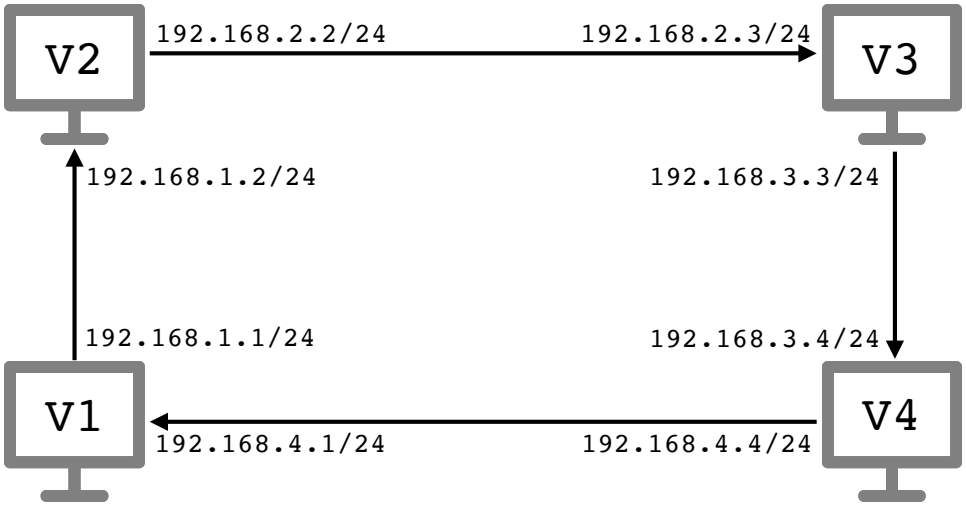
W tym celu wykonaj następujące kroki.

- Utwórz cztery maszyny *Virbian1*–*Virbian4*. Każda z nich powinna mieć dwa interfejsy połączone z odpowiednimi wirtualnymi sieciami `remote1`–`remote4`. Interfejs połączony z siecią `remote`i należy na maszynie wirtualnej nazwać `en-rem`i tak jak na rysunku powyżej. Dodatkowo na każdej maszynie powinien być interfejs nazwany `en-all` (niezaznaczony na rysunku) połączony (linie przerywane) z wirtualną siecią `remote0`.
- Na każdej maszynie aktywuj dwa interfejsy sieciowe `en-rem`i; interfejsy `en-all` pozostaw nieaktywne. Aktywnym interfejsom przypisz adresy IP jak na rysunku powyżej. Zauważ, że interfejsy podpięte do sieci `remote`i 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 2

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 brama musi leżeć w bezpośrednio podłączonej sieci: przykładowo brama domyślna 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.



- Na maszynie *Virbian1* poleceniem `ping` sprawdź, że 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?
- Na maszynie *Virbian1* wykonaj polecenia

  ```
  traceroute -N 1 -q 1 -n adres_IP
  ```

  żeby wyświetlić ścieżkę do każdego z ośmiu interfejsów.

  - Jaki protokół warstwy transportowej i port wykorzystują wysyłane zapytania? Sprawdź w pliku `/etc/services`, czy jest to port jakiejś znanej usługi.
  - Czy na wyświetlanej przez `traceroute` ścieżce są adresy IP bliższych czy dalszych interfejsów?
  - Dla jakich adresów docelowych obserwujesz pakiety z odpowiedziami wracające zgodnie ze wskazówkami zegara? Czy wracają one w ten sposób od wszystkich wyświetlanych przez `traceroute` interfejsów?
- Usuń trasy domyślne z tablic routingu. Sprawdź, że zmiany odniosły skutek wyświetlając bieżącą tablicę poleceniem `ip route`.
- Zdekonfiguruj interfejsy sieciowe i wyłącz maszyny.

Wyzwanie

Twoim celem jest skonfigurowanie tablic 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
  ```

  i sprawdź jej aktywność w pliku dziennika poleceniem `journalctl -u frr`.

- Skonfiguruj protokół OSPF na każdej maszynie.

  - W konsoli `vtysh` wyświetl bieżącą tablicę routingu poleceniem

    ```
    virbian# show ip route
    ```

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

  - Włącz OSPF dla podłączonych sieci poleceniami

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

    gdzie `192.168.x.0/24` to zakres CIDR odpowiadający jednej z bezpośrednio podłączonych sieci.

    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
     network 192.168.y.0/24 area 0
    ```

- W Wiresharku na wszystkich maszynach zaobserwuj przesyłane pakiety protokołu OSPF.

  - Jakie są adresy IP, do których są wysyłane pakiety protokołu OSPF? Czy są to adresy zwykłe, multicastowe czy rozgłoszeniowe?
  - Czy OSPF używa protokołu transportowego? Jeśli tak, to jakiego?
  - Jakie typy pakietów OSPF są przesyłane? Jak często są przesyłane?

- W konsoli `vtysh` wyświetl tablicę routingu protokołu OSPF poleceniem `show ip ospf route`. W nawiasach kwadratowych powinny być wyświetlane poprawnie policzone odległości do wszystkich czterech sieci, gdzie koszt pojedynczej krawędzi jest równy 100. Przykładowo na maszynie *Virbian2* powinna być wyświetlana następująca tablica routingu OSPF:

```
N   192.168.1.0/24  [100] area: 0.0.0.0
                    directly attached to en-rem1
N   192.168.2.0/24  [100] area: 0.0.0.0
                    directly attached to en-rem2
N   192.168.3.0/24  [200] area: 0.0.0.0
                    via 192.168.2.3, en-rem2
N   192.168.4.0/24  [200] area: 0.0.0.0
                    via 192.168.1.1, en-rem1
```

- Poleceniem `ip link` wyłącz jeden z aktywnych interfejsów maszyn i sprawdź, jak zmienia się tablica routingu. Z której maszyny widzisz sieć w odległości 400? Włącz ten interfejs ponownie i sprawdź, że tablica routingu wraca do poprzedniego stanu.
- Wyświetl tablicę routingu w zwykłej powłoce poleceniem `ip route`. Jakie jest pole `proto` dla różnych tras?
- Poleceniami `ping` i `traceroute` sprawdź osiągalność interfejsów wszystkich maszyn.
- Na każdej maszynie włącz teraz protokół OSPF również dla sieci *remote0*.
  - Na wszystkich maszynach poleceniem `ip` aktywuj interfejs `en-all` i przypisz mu adres `172.16.20.`x`/16`, gdzie $x \in \{1, 2, 3, 4\}$ jest numerem maszyny.
  - W konsoli `vtysh` włącz protokół OSPF dla nowej sieci `172.16.0.0/16` wydając polecenia

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

    Zaobserwuj przesyłane pakiety OSPF i zmiany w tablicy routingu.
  - Sprawdź, że wyświetlana poleceniem `ip route` tablica routingu zawiera po trzy trasy do każdej z sieci, które nie są bezpośrednio sąsiadujące (protokół OSPF wylicza do nich taką samą odległość). Przykładowo wynik tego polecenia dla maszyny *Virbian4* powinien być następujący:

    ```
    172.16.0.0/16 dev en-all proto kernel scope link src 172.16.0.3
    192.168.1.0/24 nhid id1 proto ospf metric 20
        nexthop via 192.168.2.2 dev en-rem2 weight 1
        nexthop via 172.16.20.2 dev en-all weight 1
        nexthop via 172.16.20.1 dev en-all weight 1
    192.168.2.0/24 dev en-rem2 proto kernel scope link src 192.168.2.3
    192.168.3.0/24 dev en-rem3 proto kernel scope link src 192.168.3.3
    192.168.4.0/24 nhid id2 proto ospf metric 20
        nexthop via 192.168.3.4 dev en-rem3 weight 1
        nexthop via 172.16.20.1 dev en-all weight 1
        nexthop via 172.16.20.4 dev en-all weight 1
    ```

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

Marcin Bieńkowski