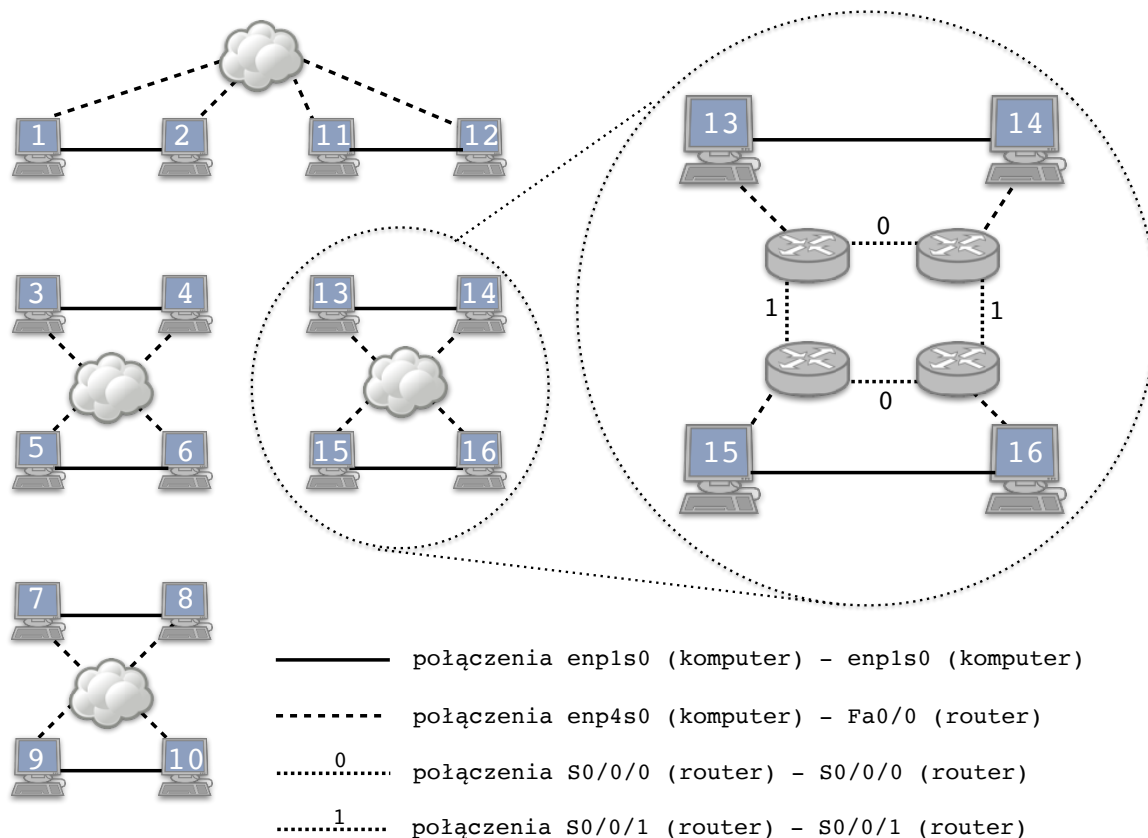


# Warsztaty z Sieci komputerowych

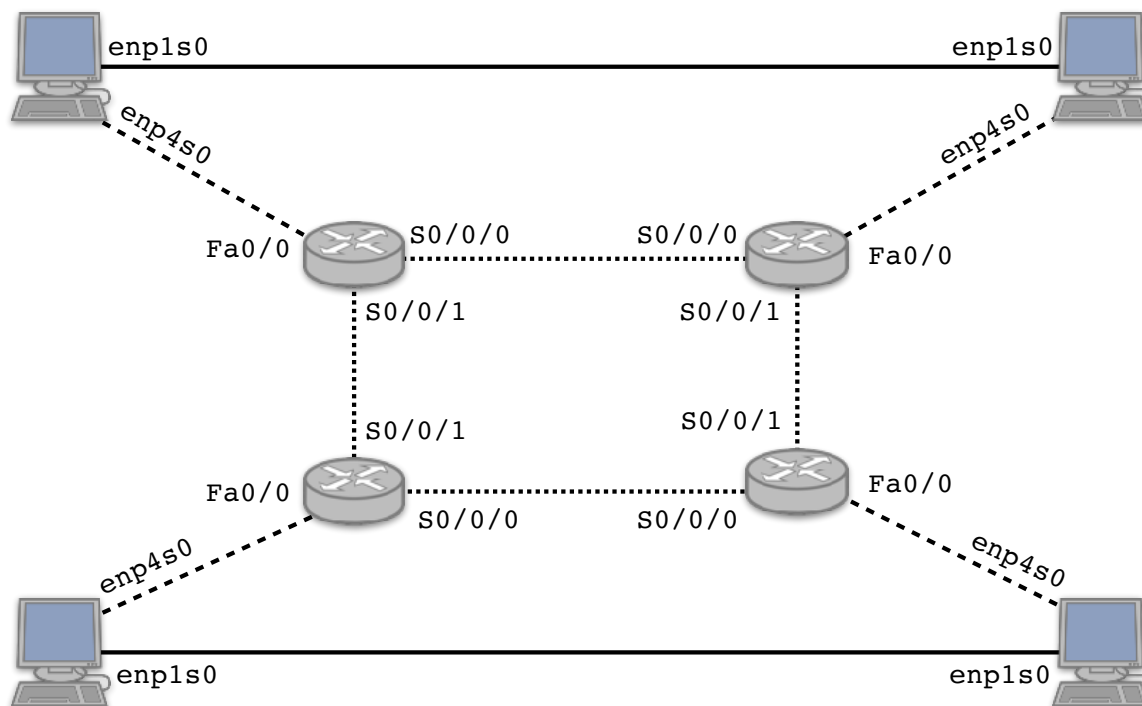
## Lista 3

Topologia sieci na te zajęcia została przedstawiona poniżej; każda czwórka komputerów jest osobną strukturą niepołączoną z niczym innym.



Celem dzisiejszych zajęć będzie skonfigurowanie (osobno w każdej czwórce) protokołu routingu dynamicznego RIP. Protokół ten będziemy konfigurować zarówno na routerach jak i na komputerach. Każdy komputer jest połączony z routerem nie tylko łączem ethernetowym, po którym będą przesyłane pakiety, ale także łączem szeregowym, służącym wyłącznie do konfiguracji.

Przed rozpoczęciem konfiguracji w 4-osobowych grupach ustalcie i zapiszcie na kartce adresy 10 sieci łączących Wasze komputery i routery. Sieci powinny być rozłączne adresowo. Wykorzystajcie pule adresów prywatnych 192.168.0.0/16 i 172.16.0.0/12. Nie wykorzystujcie masek sieci innych niż byłyby wybrane podczas routingu klasowego, tj. /24 dla sieci 192.168.x.0 i /16 dla sieci 172.x.0.0. Określcie też jakie adresy IP będą miały poszczególne interfejsy sieciowe.



**Zadanie 1.** Zdekonfiguruj interfejs `enp3s0` poleceniem `ifdown enp3s0`. Poleceniem

```
$> minicom
```

podłącz się do konsoli routera. Naciśnij klawisz **Enter**, aby przejść do znaku zachęty. Jeśli zostaniesz zapytany o wprowadzenie startowej konfiguracji (Would you like to enter the initial configuration dialog), wpisz `no` i naciśnij klawisz **Enter**. Jeśli zostaniesz zapytany o zakończenie autoinstalacji (Would you like to terminate autoinstall) wpisz `yes` i naciśnij klawisz **Enter**.

Router korzysta z systemu operacyjnego Cisco IOS. Konsola routera umożliwia dopełnianie poleceń klawiszem **Tab** tak jak w powłocie. W dowolnym momencie można nacisnąć klawisz **?** i otrzymać kontekstową pomoc.

Na początku uzyskaj uprawnienia administratora poleceniem

```
Router> enable
```

Konsola IOS działa w dwóch trybach: *trybie konfiguracji* do którego wchodzimy poleceniem

```
Router# configure terminal
```

w którym znak zachęty to `Router(config)` i *trybie wydawania poleceń* (domyślny tryb po podłączeniu do konsoli), w którym znak zachęty to `Router`. W trybie konfiguracji niektóre polecenia powodują wejście głębiej w menu konfiguracji; aby wyjść o jeden poziom wyżej należy wpisać polecenie `exit`. Poziom menu w którym się aktualnie znajdujemy jest zazwyczaj odzwierciedlony w wyświetlanym znaku zachęty. Żeby zakończyć tryb konfiguracji (w dowolnym miejscu menu konfiguracyjnego) i przejść do trybu wydawania poleceń, należy wpisać polecenie

```
Router(config)# end
```

Można też skracać wpisywane polecenia i ich parametry: zostaną one wykonane pod warunkiem, że da się je jednoznacznie rozszerzyć do właściwych. Przykładowo zamiast polecenia `configure terminal` można wpisać polecenie `conf t`.

Zacznij od skonfigurowania konsoli, wydając (w trybie konfiguracji) polecenia

```
Router(config)# line console 0
Router(config-line)# logging synchronous
Router(config-line)# transport preferred none
```

a następnie wyjdź z trybu konfiguracji (poleceniem `end`).

**Zadanie 2.** W trybie wydawania poleceń wyświetl dostępne interfejsy sieciowe:

```
Router# show ip interface brief
```

Dostępne powinny być cztery interfejsy:

- FastEthernet0/0 (w skrócie Fa0/0),
- FastEthernet0/1 (nie będziemy go używać),
- Serial0/0/0 (S0/0/0) oraz
- Serial0/0/1 (S0/0/1).

Uwaga: na niektórych routerach zamiast dwóch ostatnich interfejsów dostępne są `Serial0/1/0` i `Serial0/1/1`; należy wtedy zmienić odpowiednie poniższe polecenia. Zacznijemy od konfiguracji interfejsów sieciowych routera. Wejdź w tryb konfiguracji wpisując `configure terminal` i następnie wejdź w tryb konfiguracji interfejsu Fa0/0 poleceniem

```
Router(config)# interface Fa0/0
```

Zauważ, że znak zachęty zmienił się na `Router(config-if)`. Na początku trzeba aktywować interfejs poleceniem

```
Router(config-if)# no shutdown
```

Potem przypisz interfejsowi Fa0/0 adres IP i maskę sieci (w formacie czterech liczb dziesiętnych oddzielonych kropkami) poleceniem

```
Router(config-if)# ip address adres maska
```

Jeśli podasz błędny wpis, możesz go usunąć poleceniem `no ip address adres maska`. Wyjdź do trybu wydawania poleceń (poleceniem `end`) i ponownie wyświetl dostępne interfejsy sieciowe poleceniem

```
Router# show ip interface brief
```

Wypróbuj także polecenia

```
Router# show ip interface
Router# show interface
```

W analogiczny sposób skonfiguruj dwa pozostałe interfejsy: Serial0/0/0 i Serial0/0/1. Bieżącą konfigurację możesz wyświetlić poleceniem

```
Router# show running-config
```

Z każdego routera pingnij pozostałe. Które adresy IP są dostępne?

Na komputerze skonfiguruj interfejs `enp4s0` (ten połączony z interfejsem Fa0/0 routera)

```
#> ip link set enp4s0 up
#> ip addr add adres/maska dev enp4s0
```

a następnie dodaj trasę domyślną prowadzącą przez interfejs Fa0/0 routera:

```
#> ip route add default via adres_interfejsu_Fa0/0
```

Wyświetl tablice routingu na komputerze i routerze:

```
#> ip route
Router# show ip route
```

Poleceniem `ping` sprawdź, które adresy IP przypisane komputerom lub routerom w sieci są dostępne z komputera. Czy osiągalne są wszystkie interfejsy najbliższego routera? A interfejsy innych routerów?

**Zadanie 3.** Włączymy teraz na routerze protokół RIP w wersji 2. W tym celu wejdź do konfiguracji protokołu RIP wydając w trybie konfiguracji polecenie

```
Router(config)# router rip
```

Będąc w tym trybie włącz wersję drugą protokołu poleceniem

```
Router(config-router)# version 2
Router(config-router)# no auto-summary
```

Następnie włącz rozgłaszanie informacji o wszystkich trzech sąsiadujących z routerem sieciach poleceniami

```
Router(config-router)# network adres_sieci_1
Router(config-router)# network adres_sieci_2
Router(config-router)# network adres_sieci_3
```

Adres sieci należy wpisać bez maski sieci. W trybie wydawania poleceń możesz włączyć wyświetlanie wszystkich aktualizacji protokołu RIP poleceniem

```
Router# debug ip rip
```

Powyższe informacje (i inne włączone informacje diagnostyczne) można wyłączyć poleceniem

```
Router# undebug all
```

O działaniu RIP można przekonać się również oglądając co jakiś czas wynik działania poleceń:

```
Router# show ip rip database
Router# show ip protocol
Router# show ip route
```

W uruchomionym na komputerze Wiresharku obejrzyj wysyłane przez router komunikaty RIP. O jakich sieciach dostajesz powiadomienie?

Poczekaj aż stan tablicy routingu osiągnie stan stabilny. Następnie poleceniem `ping` sprawdź, które adresy IP przypisane w sieci są dostępne z komputera. Adresy IP z 8 sieci (ze wszystkich dotychczas zdefiniowanych) powinny być osiągalne.

**Zadanie 4.** Za pomocą polecenia `traceroute` znajdźcie trasę pomiędzy dwoma komputerami przechodzącą przez trzy routery. Wyłączcie jeden z interfejsów wykorzystywanych na tej trasie (`Serial0/0/0` lub `Serial0/0/1`). W tym celu na odpowiednim routerze wykonajcie polecenia

```
Router(config)# interface nazwa_interfejsu
Router(config-if)# shutdown
```

Jak zmieniła się trasa pokazywana przez poprzednie polecenie `traceroute`? Włączcie ponownie interfejs poleceniem

```
Router(config-if)# no shutdown
```

i znowu sprawdźcie trasy wyznaczane przez `traceroute`.

Wykonajcie powyższe zadanie dla innej trasy obserwując dodatkowo wymieniane komunikaty włączając ich wyświetlanie poleceniem

```
Router# debug ip rip
```

a także okresowo monitorując stan tablicy poleceniami

```
Router# show ip rip database
Router# show ip protocol
Router# show ip route
```

**Zadanie 5.** W tym zadaniu skonfigurujemy protokół RIP również na komputerach. Rozpocznij od zdekonfigurowania bieżących ustawień interfejsu `enp4s0` na komputerze poleceniami

```
#> ip route del default via adres_bramy_domyślnej
#> ip addr flush enp4s0
```

Linuksowym programem implementującym protokoły routingu dynamicznego jest **quagga**, której interaktywna konfiguracja jest bardzo podobna do konfiguracji routerów Cisco. Polecenia konfigurujące Quaggę należy wydawać w specjalnym terminalu konfiguracyjnym; uruchom go teraz przez

```
$> vtysh
```

Wejdź w tryb konfiguracji (znak zachęty powinien zmienić się na **host(config)**) i skonfiguruj interfejsy sieciowe **enp1s0** i **enp4s0**. Uwaga: tym razem adresy IP należy podawać w postaci CIDR.

```
host(config)# interface enp1s0
host(config-if)# no shutdown
host(config-if)# ip address adres/maska
host(config-if)# exit
host(config)# interface enp4s0
host(config-if)# no shutdown
host(config-if)# ip address adres/maska
host(config-if)# exit
```

Następnie sprawdź, czy zmiany zostały przyjęte przez jądro, uruchamiając w innym terminalu linuksowe polecenia **ip addr** i **ip route**. (W tablicy przekazywania powinny być informacje o dwóch bezpośrednio podłączonych sieciach).

Uruchom program **wireshark**, żeby zobaczyć jakie pakiety będą wymieniane. W Quagdze włącz protokół RIP dla obu sąsiadujących sieci poleceniami

```
host(config)# router rip
host(config-router)# version 2
host(config-router)# network adres_sieci_1
host(config-router)# network adres_sieci_2
```

Uwaga: tym razem adres sieci należy podać z maską, czyli np. **192.168.7.0/24**.

W Wiresharku obejrzyj pakiety przesyłane podczas tworzenia tablicy routingu. Okresowo oglądaj tablicę routingu poleceniem **ip route** a także w terminalu konfiguracyjnym Quaggi poleceniem **show ip route**. Po zakończeniu budowania tablic poleceniami **ping** i **traceroute** sprawdź, że wszystkie sieci są dostępne.

Na routerach wyłącz interfejsy **Serial0/0/0**. Połączenia stworzą teraz 8-elementowy cykl (element to komputer lub router). Ile zajęło protokołowi RIP zauważenie tej sytuacji i rekonfiguracja tras?

Lista i materiały znajdują się pod adresem <http://www.ii.uni.wroc.pl/~mbi/dyd/>.

*Marcin Bieńkowski*