

0. Konfiguracja routingu statycznego i dynamicznego

Wykonał: Patryk Kaniewski, 2 rok Informatyka ST PUSB Skierniewice.

1. Wprowadzenie

Routing jest to proces wybierania ścieżki w sieci lub pomiędzy różnymi sieciami. Przykładami routingu jest telefonia (dawniej operator fizycznie przełączający wtyczkę z naszym telefonem z innym gniazdem) jak i sieci komputerowe np. Internet. W sieciach komputerowych zazwyczaj jest to rozwiązane za pomocą tablic routingu, które przechowują informacje o sąsiednich sieciach by móc kierować ruch w odpowiednich kierunkach.

1.1 Cel ćwiczenia

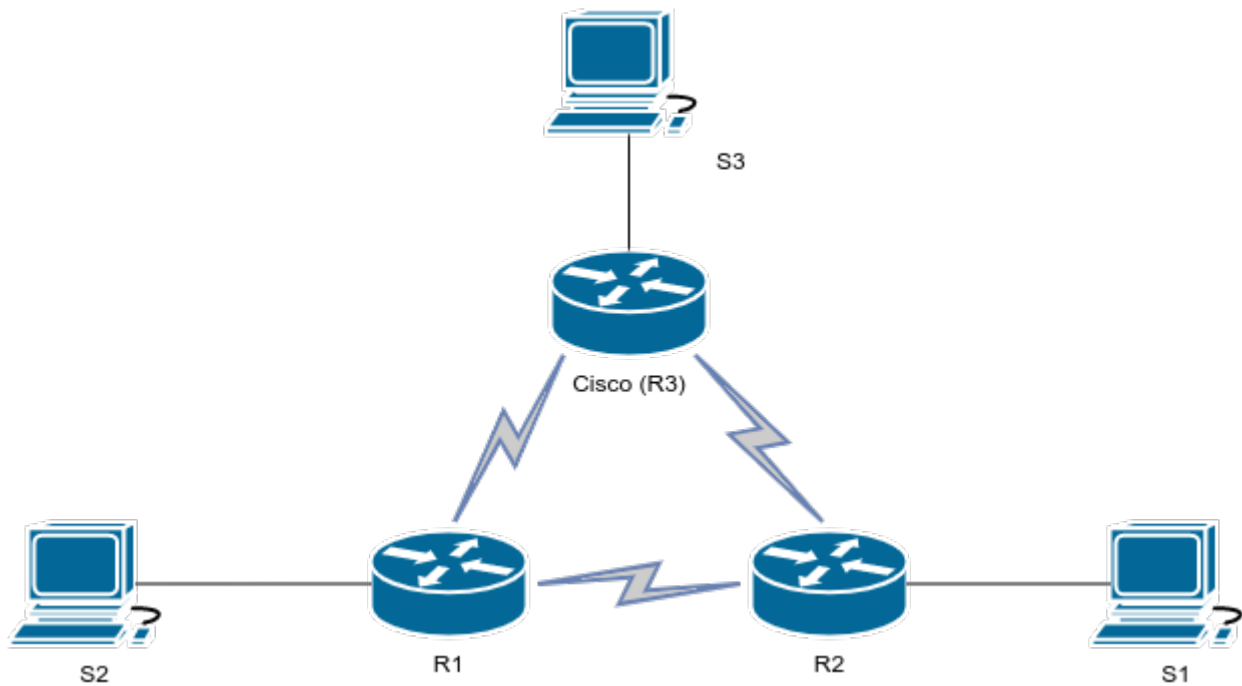
Konfiguracja i uruchomienie a następnie testowanie i monitorowanie zachowań protokołów routingu dynamicznego oraz zbadanie ich możliwości w różnych typowych i awaryjnych sytuacjach

1.2 Wymagania wstępne

- Podstawy CLI (Linux, Cisco IOS)
- Konfiguracja interfejsów sieciowych
- Konfiguracja usługi świadczącej routing dynamiczny (Linux: *quagga*)
- routing na podstawie stanu łącza - OSPFv2 (RFC 2328)
- routing na podstawie wektora odległości - RIPv2 (RFC 1723 i późniejsze RFC 2458)
- Narzędzia diagnostyczne sieci (*tcpdump, traceroute, ping*)

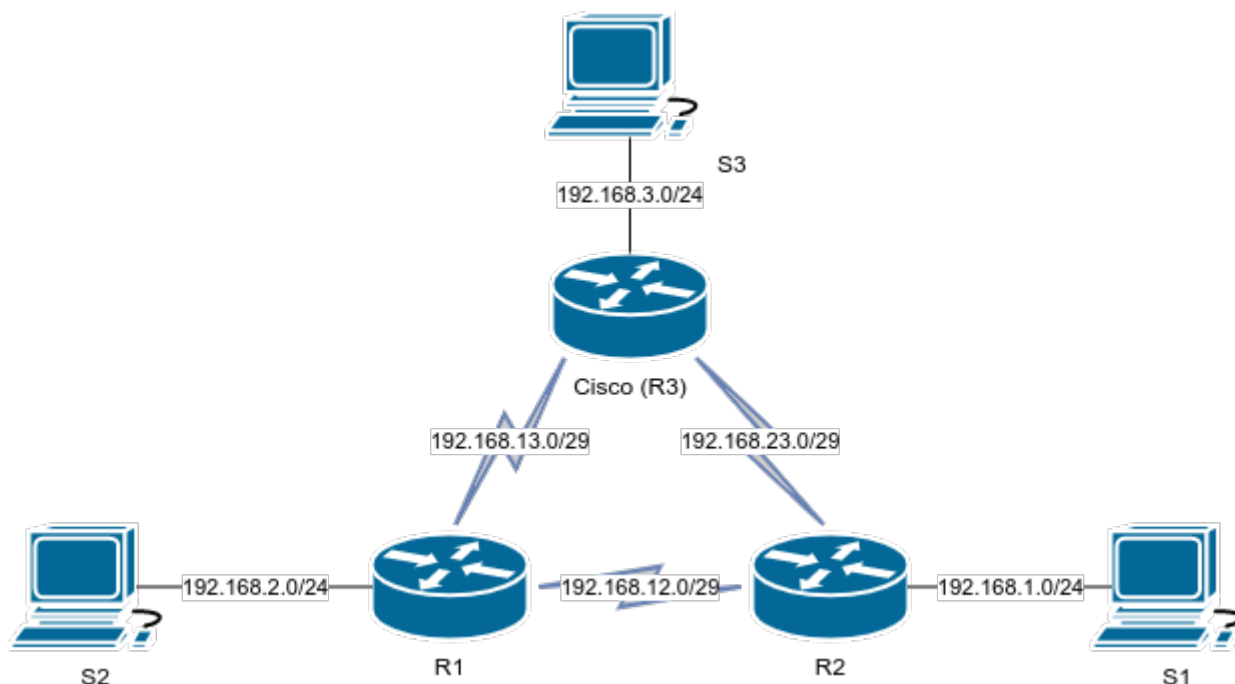
2. Przebieg ćwiczenia

Fizyczna konfiguracja sieci:



2.1 Przygotowanie stanowiska

Logiczna konfiguracja sieci:



Wyjaśnienie:

Routery łączymy w sieciach z dużą maską sieciową ponieważ nie potrzebujemy wiele adresów do sieci point-to-point. Maszyny robocze zostawiamy w domyślnych adresach lokalnych klasy C. Za pomocą tego możemy zweryfikować jak protokoły radzą sobie z maską sieci różnej długości (VLSM).

Konfiguracja interfejsów:

recznie

za pomocą skryptu: <https://github.com/p7tryk/administracja/blob/master/network.sh>

Instalacja programów diagnostycznych (jeśli potrzeba):

```
# apt install tcpdump traceroute ping
```

Włączenie routingu na R1,R2:

```
# Edycja sysctl.conf
```

```
+ net.ipv4.ip_forward = 1
```

2.2 Routing statyczny

Routing statyczny jest to najprostsza wersja routingu. Polega na podaniu następnego skoku jaki pakiet musi wykonać żeby dostać się do danej sieci.

2.2.1 Linux

Konfigurujemy routing statyczny ze stacji roboczej (S1,S2,S3) do przyłączonego do niej routera.

2.2.1.1 ip route add

```
# ip route [-6] add $adres [opcje] via $adres [opcje] dev $interfejs  
<zdjecie>
```

2.2.1.2 Default route

```
# ip route add default via $adres dev $interfejs
```

<zdjecie>

```
# edycja /etc/network/interfaces
```

<zdjecia>

2.2.2 Weryfikacja

Działanie połączenia:

<zdjecie>

Sprawdzanie tablic routingu:

<zdjecie>

2.3 Routing dynamiczny

W miarę wzrostu złożoności sieci komputerowych narosła potrzeba bardziej inteligentnego i *dynamicznego* systemu który mogłby reagować na zmiany w ruchu sieciowym czy awarie na połączeniach.

Dwoma głównymi metodami routingu dynamicznego w sieciach wewnętrznych jest routing oparty o *stan łącza* (*link state routing*) i *wektor odległości* (*distance-vector routing*).

Distance-vector routing polega na wyliczeniu liczby skoków (routerów) przez które pakiet przechodzi na danej drodze. Problemem w takiej sytuacji może być droga z małą przepustowością może być krótsza niż droga z dużą przepustowością (analogia świata rzeczywistego: jechanie przez centrum miasta vs. jechanie szybciej obwodnicą)

Link state routing natomiast polega na nadaniu danemu routerowi wagi zwykle oparte na szybkości połączenia i urządzenia. Rozwiązuje to problem obwodnicy ale zdecydowanie komplikuje działanie takiego systemu.

2.3.1 Quagga (Linux)

2.3.1.1 Instalacja quagga

```
# apt install quagga quagga-doc
```

```
# edycja /etc/quagga/daemons
```

```
+ zebra=yes
```

```
+ ospfd=yes
```

```
+ ripd=yes
```

stworzenie pustych plików konfiguracyjnych (inaczej usług się nie uruchomi *dependencies not met*)

```
# touch /etc/quagga/{zebra.conf,ospfd.conf,ripd.conf}
```

2.3.1.2 Konfiguracja podstawowa (OSPF)

uruchamiamy usługę:

```
# systemctl start ospfd.service
```

Łączymy się do servera ospfd za pomocą *telnet* lub *vttysh*:

numer portu możemy poznać za pomocą *nmap localhost | grep ospf*

```
$ enable //zeby przejść do trybu uprawnionego
```

```
# configure terminal //zeby przejść do trybu konfiguracji
```

```
# router ospf //zeby przejść do konfiguracji OSPF
```

```
# network $adres_sieci/$subnet area $area //podajemy sieć którą będziemy ogłaszać
```

Po dodaniu naszych sieci do rozgłoszenia wychodzimy z konfiguracji i komendą *write* zapisujemy obecną konfigurację do pliku.

2.3.1.3 Konfiguracja zaawansowana (OSPF)

do punktu 3.

```
# cost $koszt //manualne ustawienie kosztu przejścia przez interfejs (domyślnie kalkulowane z trybu prędkości interfejsu (np. 100 BASE-T))
```

```
# dead-interval $sekundy //po jakim czasie nie odpowiadający router sąsiedni będzie uznany za martwy
```

Lab1: Konfiguracja routingu statycznego i dynamicznego

```
# hello-interval $sekundy //specyfikuje co jak czas pakiet hello będzie wysyłany
# priority $numer //priorytet routera
# transmit-delay $sekundy //dodaje czas do pola czasu w pakiecie LSA
retransmit-interval $sekundy //ustawienie częstotliwości retransmisji dopóki pakiet LSA nie
zostanie potwierdzony
```

2.3.1.3 Konfiguracja podstawowa (RIP)

uruchamiamy usługę:

```
# systemctl start ripd.service
```

Łączymy się do servera ripd za pomocą *telnet* lub *vttysh*:

numer portu możemy poznać za pomocą *nmap localhost | grep ripd*

```
$ enable //zeby przejść do trybu uprawnionego
# configure terminal //zeby przejść do trybu konfiguracji
# router rip //zeby przejść do konfiguracji OSPF
# rip ip send version [1,2,1 2] //zeby wybrać wersję RIP której używamy
# [no] network $adres_sieci/$subnet //podajemy sieć którą będziemy ogłaszać
# [no] network $nazwainterfejsu //podajemy sieć którą będziemy ogłaszać (przez interfejs)
```

Po dodaniu naszych sieci do rozgłoszenia wychodzimy z konfiguracji i komendą *write* zapisujemy obecną konfigurację do pliku.

2.3.1.4 Konfiguracja zaawansowana (RIP)

do punktu 3.

```
# distance [1-255] [$adres/$maska] //ustawienie maksymalnego dystansu (liczby
skoków) dla sieci
# timers basic [$update $timeout $garbagecollect] //pokazuje/zmienia czasy
//$update – czas wysyłania aktualizacji do neighbour(sąsiadów)
//$timeout – czas do unieważnienia drogi
//$garbagecollect – czas do usunięcia nieważnej drogi
```

2.3.2 Cisco

TBD

2.3.3 Weryfikacja działania protokołu

<Ping, traceroute, ip route show, tcpdump (pakietow kontrolnych)? pomiedzy stacjami roboczymi, routing table routerów>

<zdjecia>

3. Testowanie „dynamiczności” routingu

3.1 Metodyka testowania

Włączamy tylko ten daemon którym się zajmujemy (RIP/OSPF)

1. Zdejmowanie interfejsów z r1,r2
2. Logowanie komunikacji pomiedzy routerami (tcpdump), tablic routingu
3. Analiza pakietów kontrolnych
4. Zmiana ustawień
5. GOTO 1

3.2 Wyniki

TBD

4. Wnioski

TBD

5. Bibliografia

TBD TBD

<https://www.nongnu.org/quagga/docs>

cisco docs