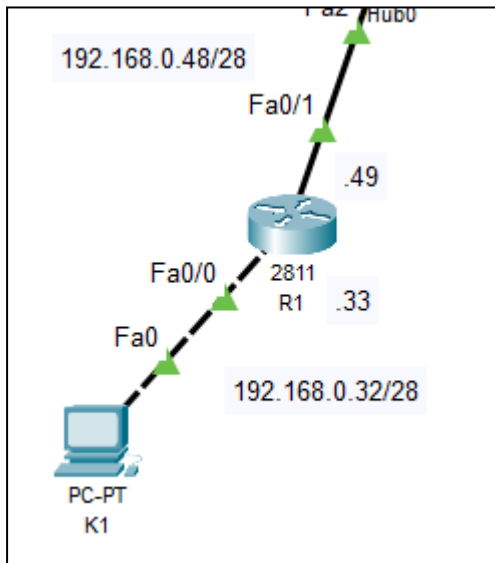


RIP v1:

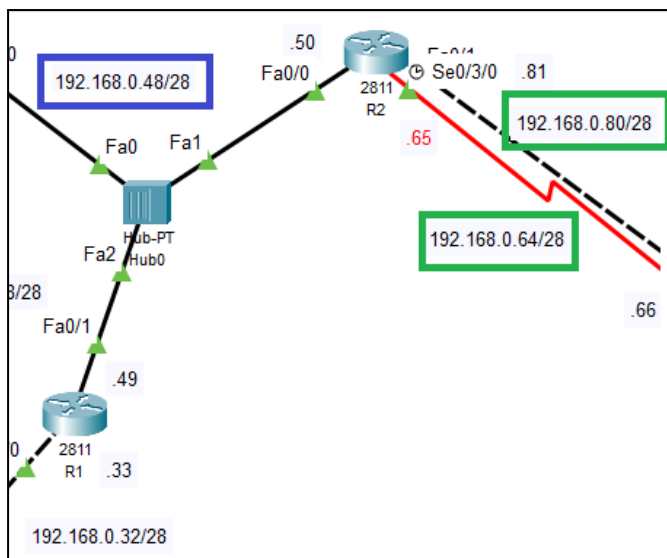
Rozważmy na razie jeden router R1 z protokołem routingu RIPv1:



Router R1 wysyła broadcastem informacje o znanych sobie sieciach (teraz tylko bezpośrednio połączonych), z metryką o jeden większą.

```
RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0 (192.168.0.33)
RIP: build update entries
      network 192.168.0.48 metric 1
RIP: sending v1 update to 255.255.255.255 via FastEthernet0/1 (192.168.0.49)
RIP: build update entries
      network 192.168.0.32 metric 1
```

Włączmy protokół RIPv1 na kolejnym routerze R2. Wysyła on informację broadcastem, które otrzymuje router R1.



Zauważmy, że router R2 nie wysłał informacji o sieci 192.168.0.48 “w lewo”, zgodnie z techniką split-horizon: **router nie wysyła informacji o trasach, które otrzymał od sąsiedniego routera, z powrotem do tego samego routera.**

Na dwóch poniższych zdjęciach widać informacje, które wysłał router R2.

```
RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0 (192.168.0.50)
RIP: build update entries
      network 192.168.0.64 metric 1
      network 192.168.0.80 metric 1
```

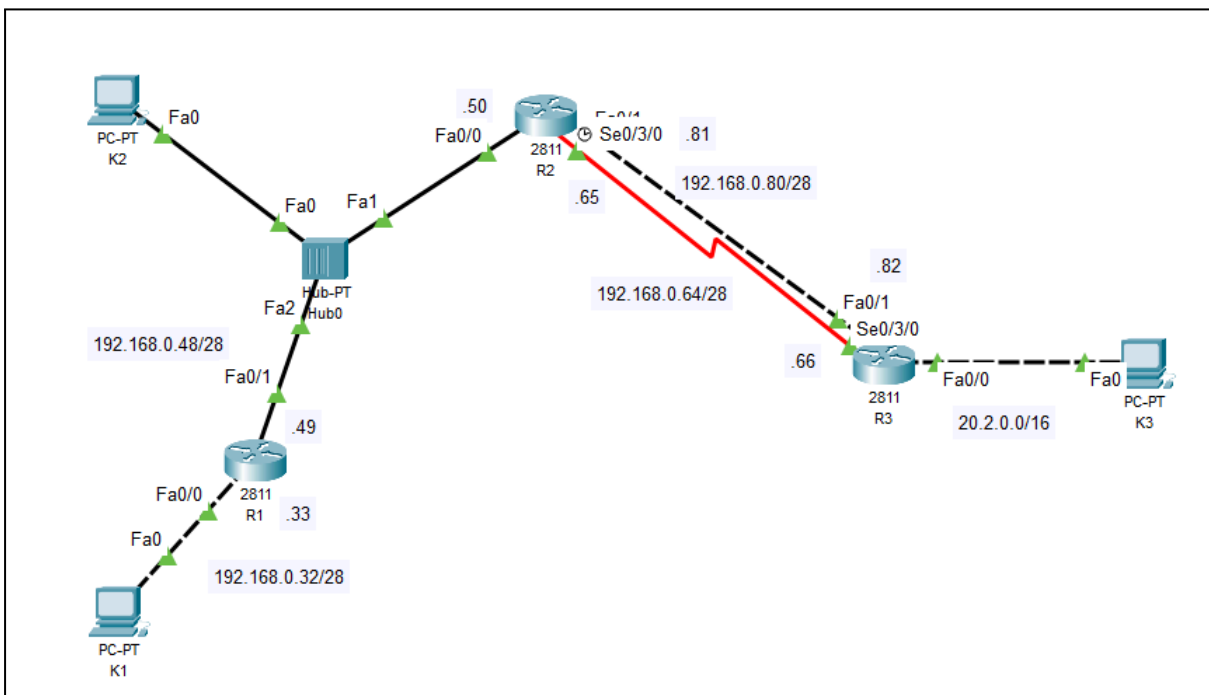
Na "prawą stronę" wysłał już sieć 192.168.0.48:

```
RIP: sending v1 update to 255.255.255.255 via FastEthernet0/1 (192.168.0.65)
RIP: build update entries
      network 192.168.0.32 metric 2
      network 192.168.0.48 metric 1
      network 192.168.0.80 metric 1
```

Na poniższym zdjęciu informacje, które otrzymał R1 od R2:

```
RIP: received v1 update from 192.168.0.50 on FastEthernet0/1
      192.168.0.64 in 1 hops
      192.168.0.80 in 1 hops
```

Rozważmy teraz całą sieć z protokołem RIPv1. Na poniższym zdjęciu tabela routingu RIP routera R1.



```
R1#show ip route rip
R    20.0.0.0/8 [120/2] via 192.168.0.50, 00:00:19, FastEthernet0/1
R    192.168.0.0/24 is variably subnetted, 6 subnets, 2 masks
R    192.168.0.64/28 [120/1] via 192.168.0.50, 00:00:19, FastEthernet0/1
R    192.168.0.80/28 [120/1] via 192.168.0.50, 00:00:19, FastEthernet0/1
```

Skąd routery wiedzą, jaką maskę przyporządkować danej sieci, skoro RIPv1 nie obsługuje VLSM?

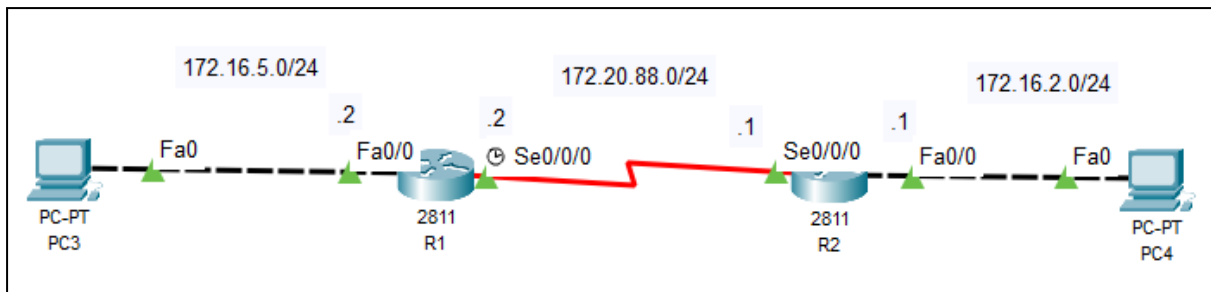
[Maski w protokole RIPv1](#)

Przykładowo: Router R1 widzi, że informacja o sieci 192.168.0.64 przyszła od routera R2 na interfejs R1.49 (192.168.0.49). Router R1 sprawdza, czy te dwa adresy należą do tej samej sieci. Istotnie, są to adresy klasy C, sieci 192.168.0.0. Zatem, router nie dokona sumaryzacji i zastosuje taką samą maskę, jak na interfejsie, czyli /28.

Z kolei, otrzymując informacje o sieci 20.0.0.0, zobaczy, że jest to adres sieci klasy A, czyli inny od sieci na interfejsie - dokona zatem sumaryzacji i zastosuje maskę /8.

Zauważmy też, że to router R3 wpięrw dokonał sumaryzacji 20.2.0.0/16, do 20.0.0.0/8, wysyłając informacje do routera R2 (bo sieć interfejsu była inna).

Protokół RIPv1 nie działa także dla **sieci niespójnych** (mimo zastosowania tych samych masek).



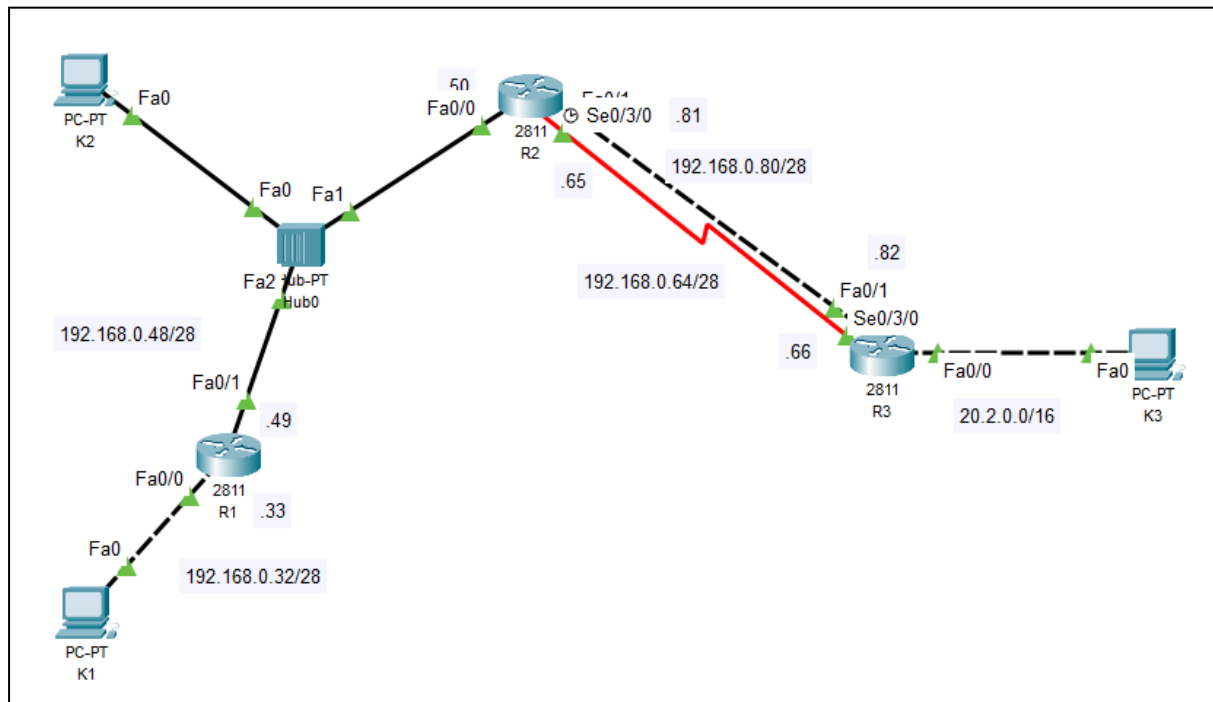
Router R1 chce przesłać informacje o sieci 172.16.5.0 na interfejsie R1.2 (na prawo). Sprawdza, czy rozgłaszana sieć i sieć na interfejsie jest ta sama. Są to sieci klasy B, są różne, więc router dokona sumaryzacji na sieć: 172.16.0.0 (/16). Następnie, **router R2** musi zdecydować, czy akceptuje otrzymaną trasę, czy odrzuca. Jeśli:

- otrzymana sieć (172.16.0.0) należy do tej samej sieci głównej (A, B, C, D), co sieć interfejsu (172.20.88.0), to odrzuć trasę. W tym przypadku są to różne sieci, więc przechodzimy do drugiego punktu.
- sprawdź w swojej tabeli routingu, czy na pozostałych interfejsach, nie ma dokładniejszego, szczegółowego wpisu dot. tej sieci (wpis jest dokładniejszy, gdy ma dłuższą maskę, podsieć). Gdy router posiada dokładniejszy wpis, odrzuci otrzymaną trasę, wpp. akceptuje trasę.

Nasz router R2 ma sieć bezpośrednio połączoną, 172.16.2.0/24, jest to podsieć sieci 172.16.0.0/16, dokładniejszy wpis. **Router R2 odrzuci zatem otrzymaną trasę.** Skoro odrzucił, to nie jest w stanie komunikować się z siecią 172.16.5.0/24 !

W protokołach takich jak **RIP v1**, które nie przekazują masek podsieci, routery muszą polegać na regułach klasowego routingu. Jeśli router ma bardziej szczegółowe informacje o podsieciach danej sieci, zignorowanie aktualizacji pomaga uniknąć konfliktów routingu i zapewnia, że trasy w tabeli routingu są jak najbardziej precyzyjne.

RIP v2:



Router R1 otrzymuje:

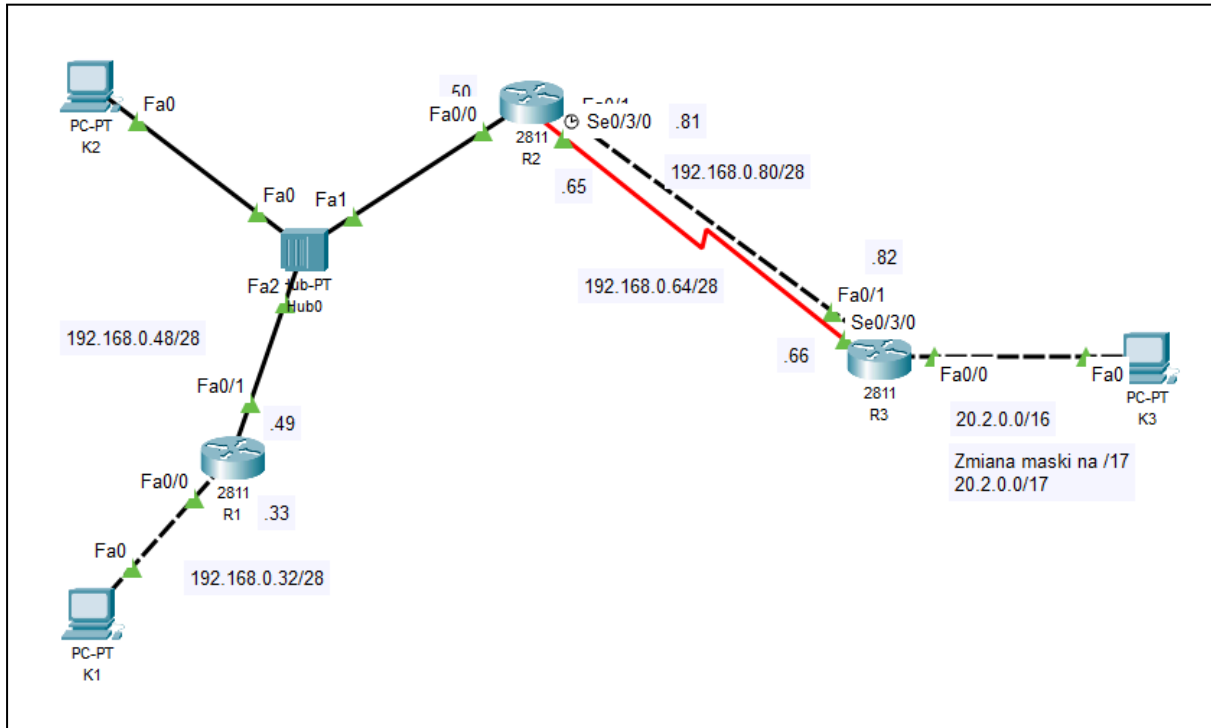
```
RIP: received v2 update from 192.168.0.50 on FastEthernet0/1
  20.2.0.0/16 via 0.0.0.0 in 2 hops
  192.168.0.64/28 via 0.0.0.0 in 1 hops
  192.168.0.80/28 via 0.0.0.0 in 1 hops
```

A wysła przykładowo:

```
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.0.33)
RIP: build update entries
  20.2.0.0/16 via 0.0.0.0, metric 3, tag 0
  192.168.0.48/28 via 0.0.0.0, metric 1, tag 0
  192.168.0.64/28 via 0.0.0.0, metric 2, tag 0
  192.168.0.80/28 via 0.0.0.0, metric 2, tag 0
```

RIPv2 obsługuje VLSM, przesyła także maski, a pakiety są już wysyłane na adres multicastowy **224.0.0.9**

Aby zasymulować zmianę topologii, zmienię maskę sieci 20.2.0.0/16 na 20.2.0.0/17.



Gdy timer dobieje do 180 sekund, router zorientuje się, że trasa jest niedostępna. Rozpocznie się procedura.

```
R2#show ip route rip
 20.0.0.0/16 is subnetted, 1 subnets
R    20.2.0.0 [120/1] via 192.168.0.66, 00:00:05, FastEthernet0/1
      [120/1] via 192.168.0.82, 00:00:05, Serial0/3/0
 192.168.0.0/24 is variably subnetted, 7 subnets, 2 masks
R    192.168.0.32/28 [120/1] via 192.168.0.49, 00:00:15, FastEthernet0/0
```

```
R2#show ip route rip
 20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R    20.2.0.0/16 [120/1] via 192.168.0.66, 00:01:08, FastEthernet0/1
      [120/1] via 192.168.0.82, 00:01:08, Serial0/3/0
R    20.2.0.0/17 [120/1] via 192.168.0.66, 00:00:15, FastEthernet0/1
      [120/1] via 192.168.0.82, 00:00:15, Serial0/3/0
 192.168.0.0/24 is variably subnetted, 7 subnets, 2 masks
R    192.168.0.32/28 [120/1] via 192.168.0.49, 00:00:24, FastEthernet0/0
```

Po 180 sekundach bez otrzymania aktualizacji od sieci 20.2.0.0/16, router R2 rozpocznie procedurę. Oznacza tę trasę, jako niedostępną, przypisuje jej metrykę 16 (nieskończoność) i rozgłasza pozostałym routerom tę nowinę (tj. rozgłasza tę trasę z metryką 16):

```
R2#show ip route rip
 20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R    20.2.0.0/16 is possibly down, routing via 192.168.0.66, 00:03:01 FastEthernet0/1
      is possibly down, routing via 192.168.0.82, 00:03:01 Serial0/3/0
R    20.2.0.0/17 [120/1] via 192.168.0.66, 00:00:13, FastEthernet0/1
      [120/1] via 192.168.0.82, 00:00:13, Serial0/3/0
 192.168.0.0/24 is variably subnetted, 7 subnets, 2 masks
R    192.168.0.32/28 [120/1] via 192.168.0.49, 00:00:26, FastEthernet0/0
```

```
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.0.50)
RIP: build update entries
    20.2.0.0/16 via 0.0.0.0, metric 16, tag 0
    20.2.0.0/17 via 0.0.0.0, metric 2, tag 0
    192.168.0.64/28 via 0.0.0.0, metric 1, tag 0
    192.168.0.80/28 via 0.0.0.0, metric 1, tag 0
```

Gdy pozostałe routery widzą trasę z metryką 16, także oznaczają ją jako niedostępną i rozgłaszają tę wiadomość dalej. Następnie, włączają czasomierz (domyślnie 240 sekund¹) - jeśli nie otrzymają aktualizacji do tej trasy, usuwają ją z tablicy routingu.

Cały proces zbieżności trwa zatem: 180s + 240s = 7 minut.

Pamiętajmy, że był włączony dzielony horyzont - router R2 nie otrzymywał informacji o trasie od żadnego routera. Jeśli wyłączymy dzielony horyzont, router R2 będzie otrzymywał informacje o tej trasie od routera R1, ale z większą metryką (na razie będzie ją ignorować). Po 180 sekundach router R1 będzie chciał rozpocząć procedurę rozgłaszania i zatruwania tej trasy poprzez wysłanie metryki 16 - jeśli w odpowiednim momencie otrzyma informację od routera R1 o tej trasie, to zaktualizuje metrykę i nie rozpocznie tej procedury. Niestety, zacznie się wtedy zliczanie do nieskończoności:

```
R2(config)#do show ip route rip
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R    20.2.0.0/16 [120/11] via 192.168.0.49, 00:00:25, FastEthernet0/0
R    20.2.0.0/17 [120/1] via 192.168.0.66, 00:00:24, FastEthernet0/1
    [120/1] via 192.168.0.82, 00:00:24, Serial0/3/0
192.168.0.0/24 is variably subnetted, 7 subnets, 2 masks
R    192.168.0.32/28 [120/1] via 192.168.0.49, 00:00:25, FastEthernet0/0

R1(config-router)#do show ip route rip
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R    20.2.0.0/16 [120/12] via 192.168.0.50, 00:00:10, FastEthernet0/1
R    20.2.0.0/17 [120/2] via 192.168.0.50, 00:00:10, FastEthernet0/1
192.168.0.0/24 is variably subnetted, 6 subnets, 2 masks
R    192.168.0.64/28 [120/1] via 192.168.0.50, 00:00:10, FastEthernet0/1
R    192.168.0.80/28 [120/1] via 192.168.0.50, 00:00:10, FastEthernet0/1
```

Router R2 ma trasę do 20.2.0.0/16, która prowadzi przez router R1. Router R1 ma trasę do 20.2.0.0/16, która prowadzi przez router R2. Metryka będzie zwiększać się, aż do 16.

Czas zbieżności będzie wtedy znacznie dłuższy (nawet 15 minut²).

¹ Czasomierz mogą włączyć wcześniej, wtedy czas oczekiwania będzie mniejszy niż 240s.

² Nie jestem pewien poprawności. Wywnioskowałem to z: 180 s + 16 * 30s (zliczanie) + 240s (flush) = 900 s = 15 min.

Ograniczenia stosowalności protokołu RIPv1:

- najdłuższa ścieżka między dowolnymi, dwoma routerami musi zawierać łącznie mniej niż 16 routerów
- nie obsługuje VLSM, sieci o różnych długości maskach, sieci niespójnych
- generuje dużo ruchu w sieci przez wysyłanie pakietów na broadcast
- słaba metryka - uwzględnia jedynie liczbę hopów
- wolna zbieżność

Ograniczenia stosowalności protokołu RIPv2:

- najdłuższa ścieżka między dowolnymi, dwoma routerami musi zawierać łącznie mniej niż 16 routerów
- słaba metryka - uwzględnia jedynie liczbę hopów
- wolna zbieżność