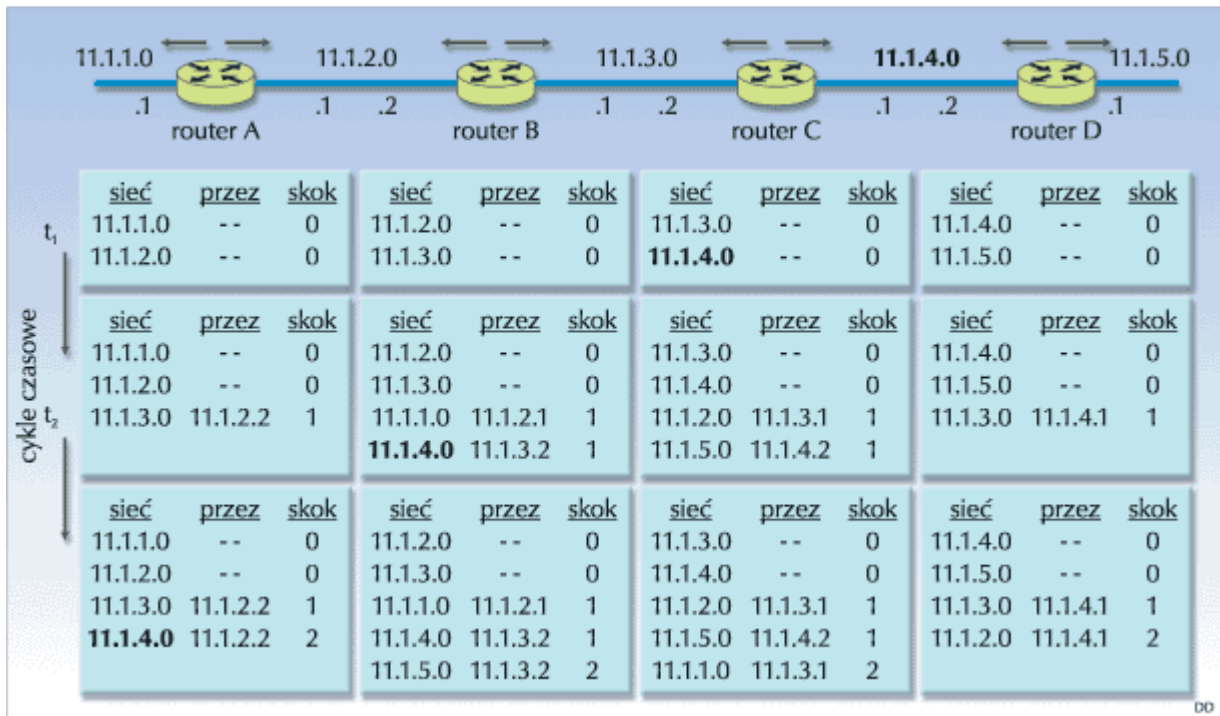
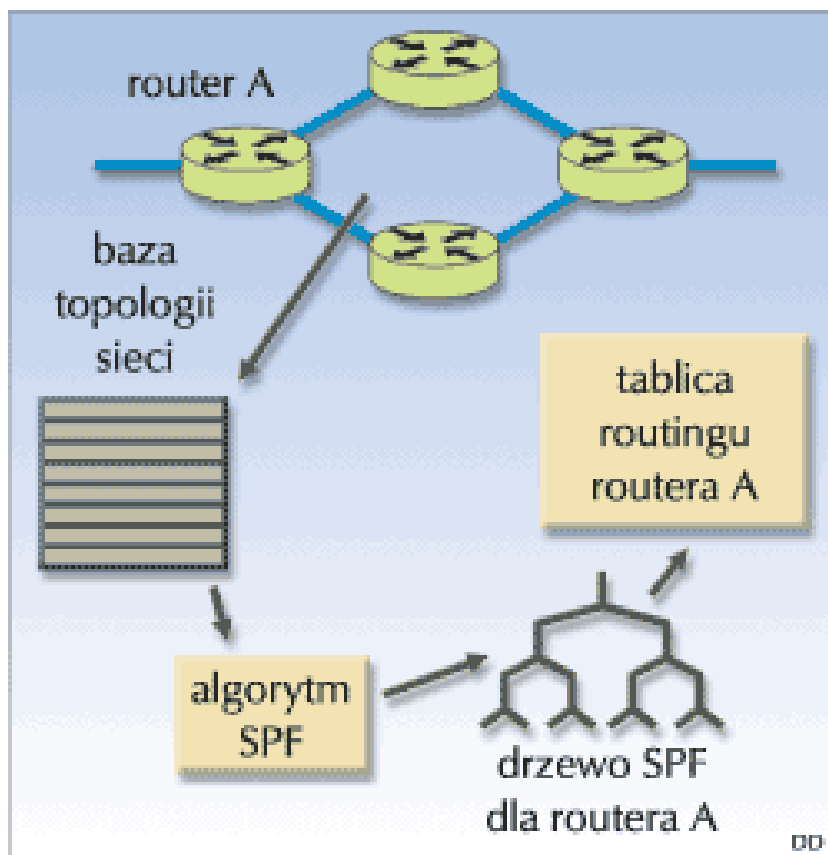
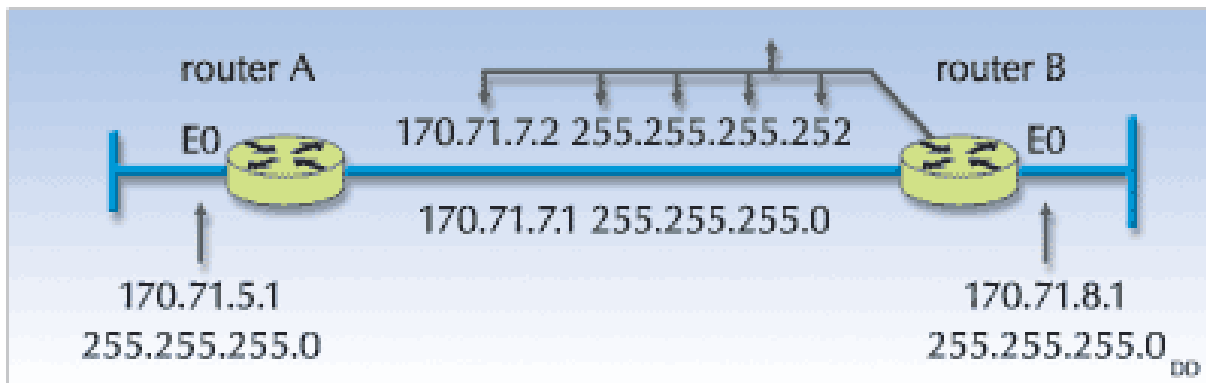


Protokoły wektora odległości



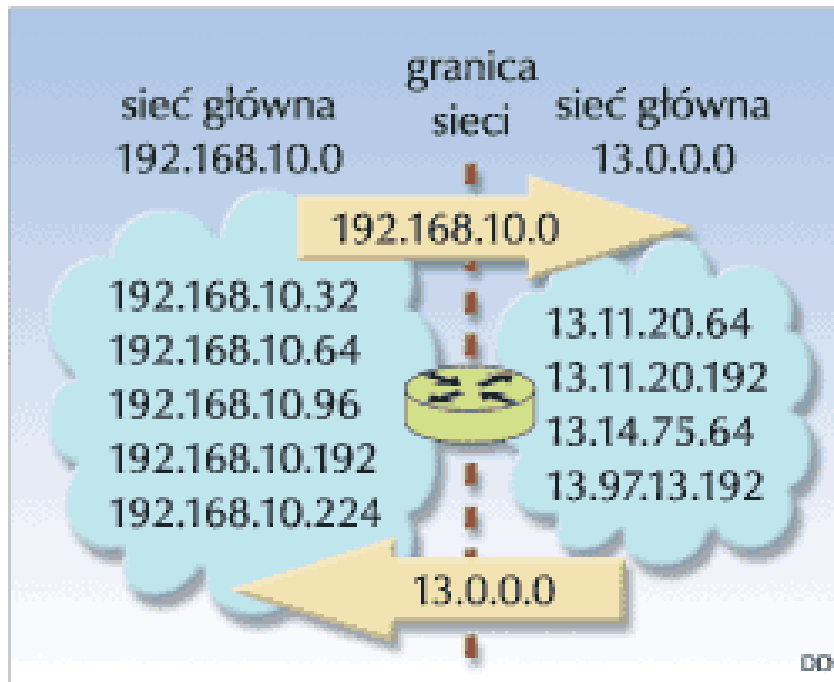
Protokoły stanu łącza



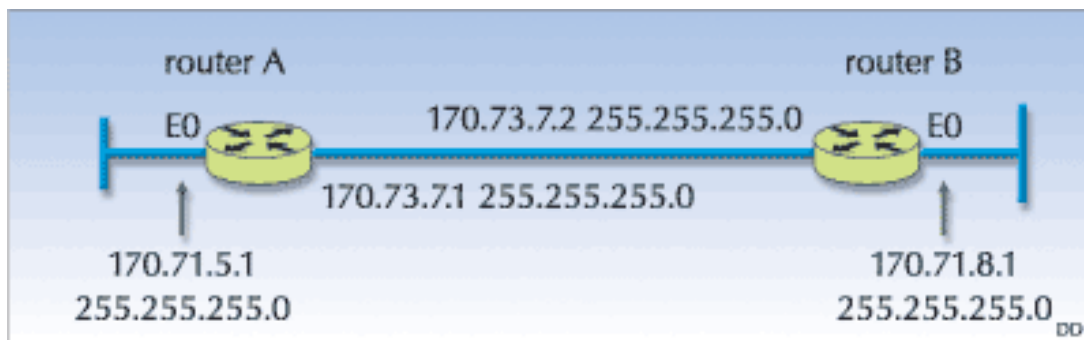


Dla protokołów klasowych stosowane są następujące zasady ogłaszania sieci lub podsieci:

- Jeżeli podsieć oraz interfejs, przez który jest ogłaszana, mają identyczną sieć główną oraz jednakowej długości maskę podsieci, to podsieć będzie ogłaszana poprawnie (poprawny adres sieci, ale bez maski). Na przykład router A ogłasza przez interfejs S0 (p. rysunek) podsieć 170.71.5.0 (sieć LAN), a router B interpretuje to ogłoszenie z maską podsieci własnego interfejsu S0 (w tym wypadku maska ma 30 bitów).
- Jeżeli podsieć ma taką samą sieć główną jak interfejs, przez który ma być ogłoszona, ale inną maskę podsieci, podsieć ta nie będzie w ogóle ogłaszana. Na przykład router B na rysunku nie będzie ogłaszał przez interfejs S0 podsieci 170.71.8.0, ponieważ jej maska (24 bity) jest niezgodna z maską interfejsu S0 (30 bitów).
- Jeżeli ogłaszana podsieć ma inną sieć główną niż interfejs, przez który jest ogłaszana, router wysyłający ogłoszenie dokonuje automatycznego przekształcenia na adres sieci głównej, czyli adres wynikający z klasy. Proces ten nazywany jest łączeniem tras na granicy sieci głównych.



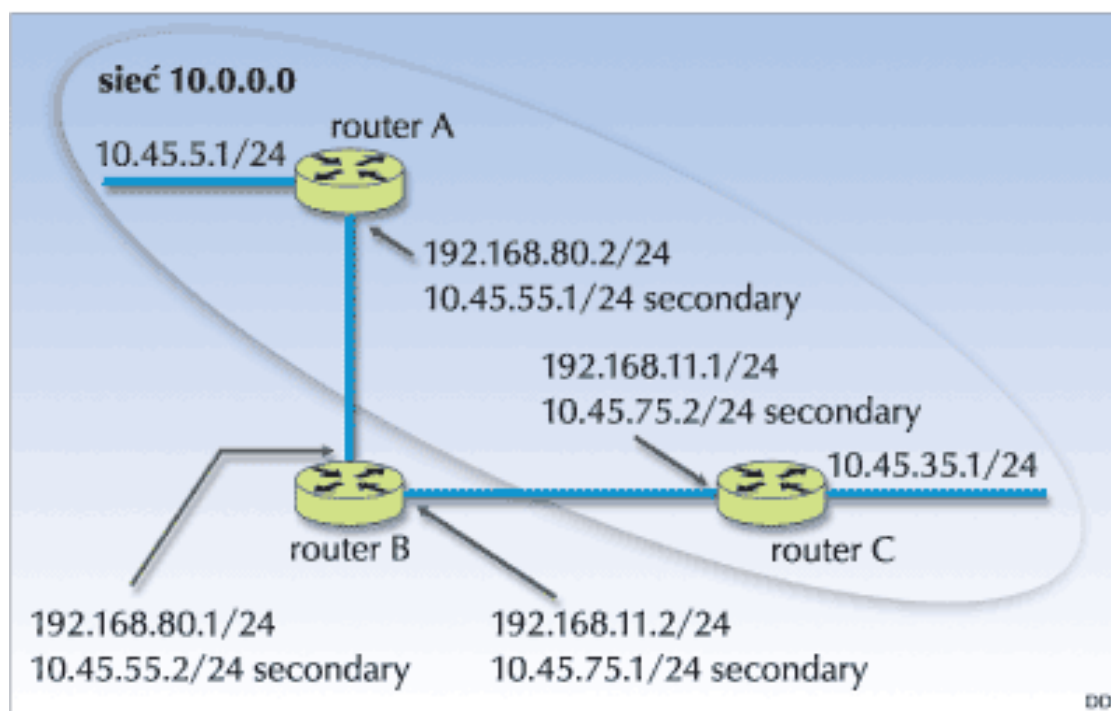
łączenie tras na granicy sieci głównych



brak obsługi sieci nieciągłych

Sieci nieciągłe to dwie podsieci tej samej sieci głównej rozdzielone inną siecią główną 170.71.5.0 i 170.71.8.0 rozdzielone przez 170.73.0.0

Jednym z problemów wynikających ze stosowania protokołów klasowych jest brak obsługi tzw. sieci nieciągłych. Sieci nieciągłe to dwie podsieci tej samej sieci głównej rozdzielone inną siecią główną - p. rysunek poniżej. Interfejsy Ethernet routerów A i B mają przypisane adresy IP z różnych podsieci sieci głównej 170.71.0.0. Na interfejsach szeregowych łączących routery wykorzystywana jest sieć główna 170.73.0.0. W tej sytuacji router A, ogłaszając sieć 170.71.5.0 do swojego sąsiada, będzie musiał dokonać przekształcenia na adres wynikający z klasy (granica sieci głównych). Ogłoszenie sumaryczne dociera do routera B, ale jest ignorowane, ponieważ router B ma dokładniejsze informacje o sieci 170.71.0.0, gdyż jest lokalnie podłączony do podsieci 170.71.8.0.



Rozwiązaniem tego problemu jest na przykład zastosowanie [protokołu bezklasowego](#), który dzięki ogłaszaniu również maski podsieci, pozwala na wyłączenie automatycznego łączenia tras na granicy sieci głównych (ogłaszana jest poprawna długość podsieci), a router odbierający ogłoszenie może zapisać w tabeli routingu adres sieci IP o poprawnej długości. W przypadku sieci nieciągłych można posłużyć się także drugorzędnymi adresami IP należącymi do tej samej sieci głównej co nieciągłe podsieci. Adresy drugorzędne należy przypisać do wszystkich interfejsów na trasie między podsieciami nieciągłymi. Sieci 10.45.5.0 za routerem A oraz 10.45.35.0 za routerem C należą do tej samej sieci głównej 10.0.0.0 i są przykładem podsieci nieciągłych rozdzielonych inną siecią główną: 192.168.11.0 oraz 192.168.80.0. Dzięki przypisaniu adresów drugorzędnych należących do różnych podsieci tej samej sieci głównej 10.0.0.0, ogłaszanie informacji przez poszczególne interfejsy na trasie między routerem A i C nie wymaga łączenia tras na granicy sieci głównych. Ogłaszanie realizowane jest niezależnie dla poszczególnych adresów IP przypisanych do interfejsu. W omawianym przypadku przypisanie dwu adresów IP do interfejsu oznacza dwukrotny proces ogłaszania, niezależnie dla adresu głównego i drugorzędneho.

Protokół RIP

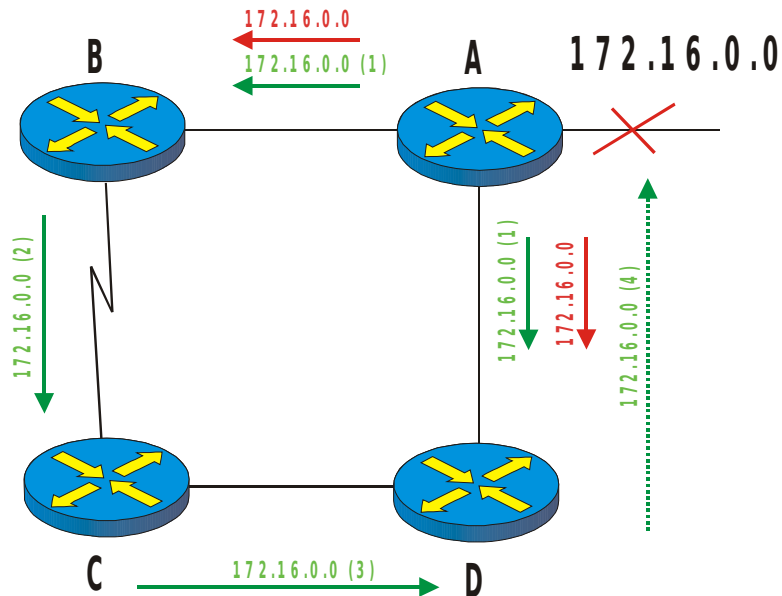
- 1970, XEROX, własne opracowanie
- 1982, Berkeley Software Distribution demon *routed* (później *gated*)
- 1988, RFC 1058
- protokół routingu typu “*distance vector*”
- algorytm Belmana-Forda (Forda-Fulkersona)
- protokół klasowy
- metryka oparta o ilość hopów (1-15)
- mechanizmy zapobiegania pętlom:
 - split horizon
 - poisoned reverse
 - triggered updates
 - zliczanie do nieskończoności

RIP v.1

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

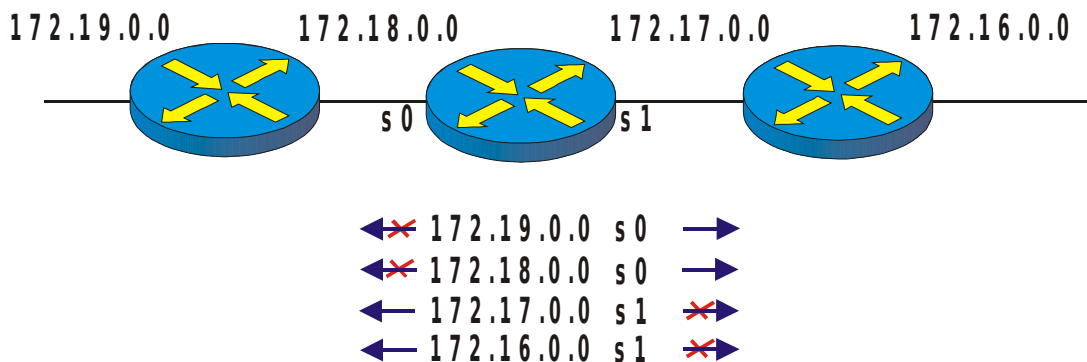
command	version	0
AFI=2 (IP)		0
IP address		
0		
0		
metric		
0		
AFI=2 (IP)		0
IP address		
0		
0		
metric		

Zliczanie do nieskończoności

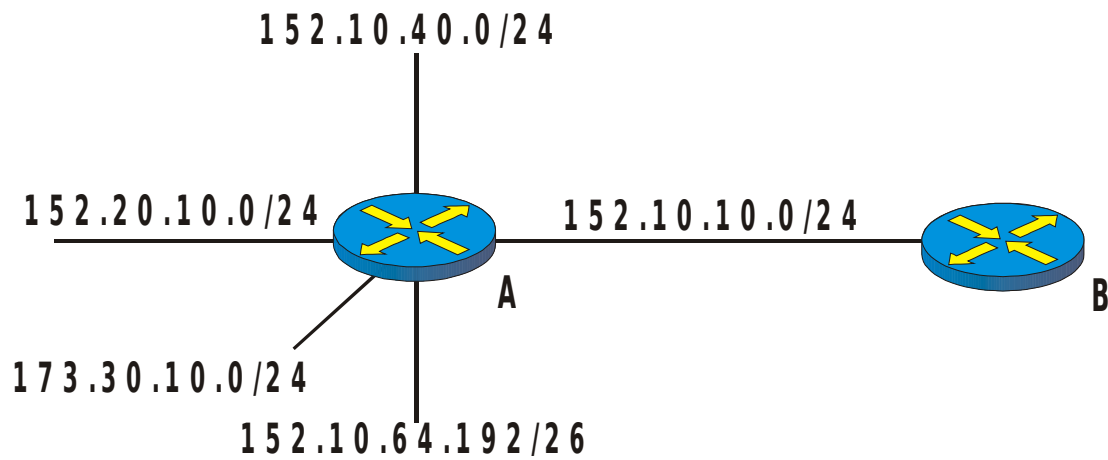


dla RIP nieskończoność to 16 skoków

SPLIT HORIZONT



- router nie rozgłasza sieci przyłączonych przez interfejsy, do których są one dowiązane
- router nie rozgłasza sieci otrzymanych od sąsiada w kierunku tego sąsiada



A

152.10.0.0/16 is variably subnetted, 3 subnets, 2 masks

C **152.10.64.192/26** is directly connected, Loopback3

C 152.10.10.0/24 is directly connected, Ethernet0/0

C 152.10.40.0/24 is directly connected, Loopback5

152.20.0.0/24 is subnetted, 1 subnets

C 152.20.10.0/24 is directly connected, Loopback1

173.30.0.0/24 is subnetted, 1 subnets

C 173.30.1.0 is directly connected, Loopback2

B

152.10.0.0/24 is subnetted, 2 subnets

C 152.10.10.0 is directly connected, Ethernet0/0

R 152.10.40.0 [120/1] via 152.10.10.2, 00:00:16, Ethernet0/0

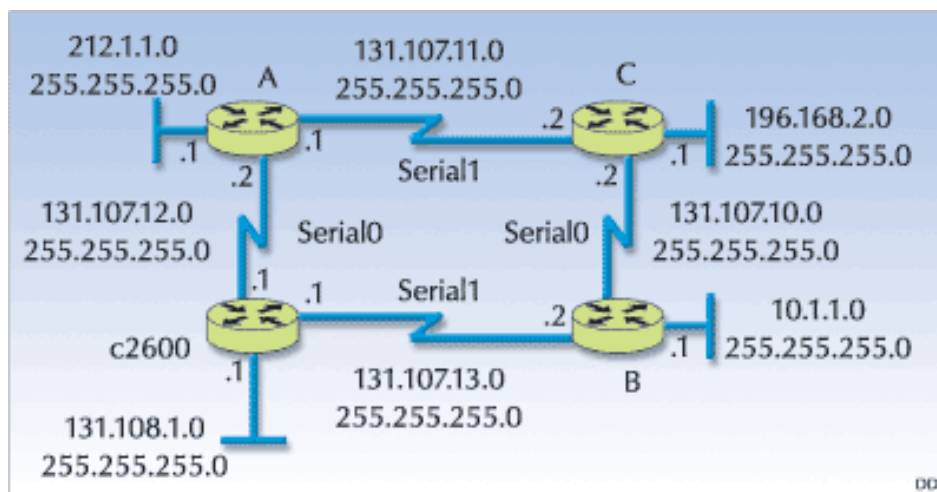
R 152.20.0.0/16 [120/1] via 152.10.10.2, 00:00:16, Ethernet0/0

R 173.30.0.0/16 [120/1] via 152.10.10.2, 00:00:16, Ethernet0/0

Włączenie routingu RIP

```
RouterA(config)#router rip
```

```
RouterA(config-router)#network 131.107.0.0
```



Konfiguracja routera c2600 jest następująca:

```
c2600(config)#router RIP
```

```
c2600(config-router)#network 131.107.0.0
```

```
c2600(config-router)#network 131.108.0.0
```

Konfiguracja pozostałych routerów będzie analogiczna (każdy z nich ogłaszać będzie inną sieć IP stosowaną w segmencie LAN).

```
Gateway of last resort is not set
```

```
R 212.1.1.0/24 [120/1] via 131.107.12.2, 00:00:03, Serial0/0
```

```
R 10.0.0.0/8 [120/1] via 131.107.13.2, 00:00:10, Serial0/1
```

```
R 196.168.2.0/24 [120/2] via 131.107.13.2, 00:00:10, Serial0/1  
[120/2] via 131.107.12.2, 00:00:04, Serial0/0
```

```
131.108.0.0/24 is subnetted, 1 subnets
```

```
C 131.108.1.0 is directly connected, Ethernet0/0
```

```
131.107.0.0/24 is subnetted, 4 subnets
```

```
R 131.107.10.0 [120/1] via 131.107.13.2, 00:00:10, Serial0/1
```

```
R 131.107.11.0 [120/1] via 131.107.12.2, 00:00:04, Serial0/0
```

```
C 131.107.12.0 is directly connected, Serial0/0
```

```
C 131.107.13.0 is directly connected, Serial0/1
```

polecenie **debug ip RIP**

```
00:53:23: RIP: received v1 update from 131.107.13.2
```


Protokół RIP - parametry czasowe

Update (czas aktualizacji) - czas wysyłania kolejnych aktualizacji. W protokole RIP domyślnie około 30 sekund.

Invalid (trasa nieważna) - zegar uruchamiany razem z ostatnią aktualizacją. Przy braku aktualizacji, po przekroczeniu tego czasu trasa oznaczana jest jako nieważna, ale nie jest automatycznie usuwana z tabeli routingu, tylko przechodzi w tryb przytrzymania trasy (hold down). W protokole RIP czas ten wynosi domyślnie 180 sekund (6 czasów aktualizacji).

Hold down (przytrzymywanie trasy) - czas przetrzymywania w tabeli routingu tras nieważnych (nieaktualizowanych). Trasa w tym trybie oznaczana jest w tabeli routingu jako "possibly down", ale jest cały czas wykorzystywana do wysyłania pakietów i router nie akceptuje ewentualnych ogłoszeń tej trasy od innych sąsiadów. Zastosowano tutaj zasadę, że lepiej przytrzymać w tabeli routingu trasę być może uszkodzoną, niż zbyt szybko przełączyć się na inną trasę do tej samej sieci, ale z wyższą metryką. W protokole RIP czas ten wynosi domyślnie 180 sekund (chyba że wcześniej skończy się czas flush).

Flush (usuwanie trasy z tabeli) - czas, po którym trasa nieaktualizowana usuwana jest z tabeli routingu. Zegar ten uruchamiany jest razem z ostatnią aktualizacją trasy. Ponieważ w protokole RIP czas ten wynosi domyślnie 240 sekund, więc w praktyce trasa nieaktualizowana usunięta zostanie z tabeli routingu już po 60 sekundach trybu hold down (plus 180 sekund czasu invalid).

```
on Serial0/1
```

```
00:53:23: 10.0.0.0 in 1 hops
00:53:23: 131.107.10.0 in 1 hops
00:53:23: 131.107.11.0 in 2 hops
00:53:23: 196.168.2.0 in 2 hops
00:53:29: RIP: sending v1 update to 255.255.255.255
    via Serial0/0 (131.107.12.1)
00:53:29: subnet 131.107.10.0, metric 2
00:53:29: subnet 131.107.13.0, metric 1
00:53:29: network 10.0.0.0, metric 2
00:53:29: network 131.108.0.0, metric 1
```

Modyfikowacja zegarów RIP

```
timers basic update invalid holddown flush.
```

Aby zilustrować stosowanie poszczególnych czasów, posłużymy się przedstawionym wcześniej układem czterech routerów i w tabeli routingu routera c2600 będziemy obserwować sieć 212.1.1.0 (LAN za routerem A), która nie będzie poprawnie aktualizowana. W tym celu w konfiguracji protokołu RIP na routerze A należy wyłączyć wysyłanie ogłoszeń przez interfejs Serial 0 poleceniem:

```
RouterA (config-router) #passive-interface Serial0
```

Przez pierwsze 180 sekund trasa pokazywana jest poprawnie i wykorzystywana jest podczas przesyłania pakietów do sieci 212.1.1.0 (tylko czas ostatniej aktualizacji zwiększa się powyżej 30 sekund):

```
R 212.1.1.0/24 [120/1] via 131.107.12.2, 00:01:13, Serial0/0
```

Po upływie około 3 minut kończy się czas invalid i trasa przechodzi w tryb hold down, w którym nadal będzie wykorzystywana do obsługi ruchu dla sieci 212.1.1.0:

```
R 212.1.1.0/24 is possibly down, routing via 131.107.12.2,  
Serial0/0
```

Teoretycznie trasa może pozostawać w trybie hold down przez 180 sekund, ale już po 60 sekundach kończy się czas flush (zegar ten uruchamiany jest razem z ostatnią aktualizacją) i trasa usuwana jest z tabeli routingu, a na jej miejsce pojawia się nowy wpis, prowadzący do sieci 212.1.1.0 przez router B z wyższą metryką równą 3:

```
R 212.1.1.0/24 [120/3] via 131.107.13.2, 00:00:10, Serial0/1
```

Po wyłączeniu komendy **passive-interface**, do tabeli routingu natychmiast powraca pierwotna trasa do sieci 212.1.1.0, ponieważ jest ogłaszana z lepszą metryką (1 skok).

Protokół IGRP

Standardowo jeden pakiet protokołu RIP może przesłać maksymalnie 25 pozycji z tabeli routingu. Podczas stosowania metody uwierzytelniania z przesyłaniem hasła jawnym tekstem pierwsza pozycja w pakiecie RIP zawiera hasło. W metodzie opartej na algorytmie MD5 dwie pozycje w pakiecie RIP (pierwsza i ostatnia) zarezerwowane są na 128-bitowy podpis cyfrowy.

Protokół IGRP opracowany został przez firmę Cisco w celu wyeliminowania niektórych ograniczeń protokołu RIP. Jedną z najważniejszych zmian jest znacznie większy dopuszczalny rozmiar sieci. W protokole RIP najdłuższa ścieżka mogła mieć tylko 15 skoków, w protokole IGRP zwiększono tę wartość do 255 (domyślnie limit ustawiony jest na 100 skoków). Jako protokół wektora odległości i protokół klasowy IGRP podlega takim samym zasadom pracy, jak protokół RIP i w wielu punktach jest do niego podobny. Poszczególne sieci ogłaszane są do sąsiadów przez wszystkie włączone interfejsy z wykorzystaniem komunikacji rozgłoszeniowej. Zmieniono jednak wartości liczników czasowych w porównaniu z protokołem RIP (ich znaczenie jest identyczne), co pokazuje poniższa tabela:

W porównaniu z RIP znacznie zoptymalizowano format pakietu IGRP, a protokół przenoszony jest bezpośrednio przez warstwę IP jako protokół nr 9 (RIP wykorzystuje UDP). Kolejną ciekawą zmianą jest możliwość rozłożenia ruchu pakietów na kilka tras o niejednakowej metryce, prowadzących do tej samej sieci. Jedną z najważniejszych cech protokołu IGRP jest jednak zupełnie inny sposób obliczania metryki trasy. W protokole RIP koszt trasy opierał się tylko na liczbie skoków do sieci docelowej. IGRP przesyła i monitoruje liczbę skoków, ale tylko w celu

sprawdzania, czy trasa nie jest zbyt długa (255 skoków maksymalnie). Liczba skoków nie jest w ogóle brana pod uwagę przy wyliczaniu metryki. W zamian uwzględnia się 5 następujących parametrów:

- Przepustowość - wartość stała definiowana w konfiguracji interfejsu, ustawiana poleceniem **bandwidth**. Im większa wartość, tym bardziej preferowana trasa. Domyślnie **uwzględniana** w metryce.
- Opóźnienie - wartość stała definiowana w konfiguracji interfejsu, ustawiana poleceniem **delay**. Im mniejsza wartość, tym bardziej preferowana trasa. Domyślnie **uwzględniana** w metryce.
- Pewność - wartość dynamicznie mierzona na poziomie interfejsu i wyrażana jako liczba z przedziału od 1 do 255. Im większa wartość (mniej błędów na interfejsie), tym bardziej preferowana trasa. Domyślnie **nieuwzględniana** w metryce.
- Obciążenie - wartość dynamicznie mierzona na poziomie interfejsu i wyrażana jako liczba z przedziału od 1 do 255. Im mniejsza liczba (mniej obciążony interfejs), tym bardziej preferowana trasa. Domyślnie **nieuwzględniana** w metryce.
- MTU - maksymalny rozmiar pola danych ramki przesyłanej w danym segmencie. Protokół IGRP śledzi najmniejszą wartość MTU na trasie, ale obecnie w ogóle **nie uwzględnia** tego parametru w metryce.

Aktualne wartości powyższych parametrów wyświetlić można poleceniem **show interfaces**. Ponieważ Pewność i Obciążenie są parametrami rzeczywistymi, zmieniającymi się w trakcie pracy interfejsu, oznaczałoby to również "ciągłą" zmianę metryk. Aby temu zapobiec, wprowadzono rozwiązanie, w którym parametry te wyznaczane są z dokładnością do 5 minut na podstawie średniej ważonej z odczytów wykonywanych co 5 sekund. Kompletny wzór, na podstawie którego wyznacza się metrykę ma postać:

Czas	Wartość
Update	od 72 do 90 sekund
Invalid	270 sekund
Hold down	280 sekund
Flush	630 sekund

$$\text{metryka} = [k1 \cdot BW_{IGRP(\min)} + (k2 \cdot BW_{IGRP(\min)}) / (256 - \text{LOAD}) + k3 \cdot DLY_{IGRP(\text{sum})}] \cdot [k5 / \text{RELIABILITY} + k4]$$

Zwróćmy uwagę na stosowane we wzorze stałe od k1 do k5. Jeżeli stała k5 przyjmuje wartość 0, ostatni składnik wzoru (nawias kwadratowy) nie jest w ogóle uwzględniany (wynik mnożenia zawsze byłby zerowy). Domyślnie stała k1=k3=1, a pozostałe stałe mają wartość 0, co oznacza, że powyższy wzór przekształca się do następującego:

$$\text{metryka} = BW_{IGRP(\min)} + DLY_{IGRP(\text{sum})}$$

Wartości stałych od k1 do k5 można zmienić poleceniem `metric weights tos k1 k2 k3 k4 k5`

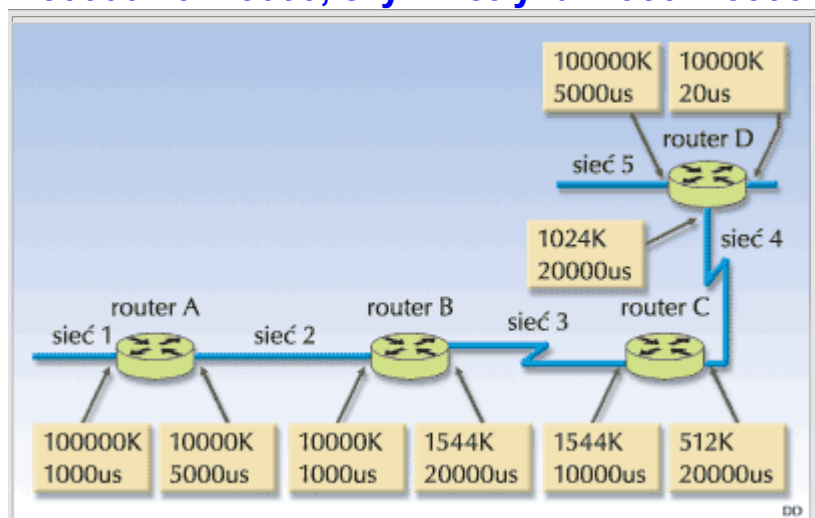
Składnik $BW_{IGRP(\min)}$ oblicza się, dzieląc 107 przez najmniejszą Przepustowość na całej trasie, natomiast $DLY_{IGRP(\text{sum})}$ jest sumą opóźnień wprowadzanych przez wszystkie interfejsy wyjściowe na całej trasie wyrażoną w dziesiątkach ms.

Wyznamy teraz metrykę przykładowej trasy prowadzącej z routera A do sieci 5 za routerem D (patrz rys. poniżej).

Najniższa przepustowość na trasie z routera A do sieci 5 wynosi 512 kb/s, stąd:

$$BW_{IGRP(\min)} = 10^7 / 512 = 19531 \text{ oraz}$$

$$DLY_{IGRP(\text{sum})} = 50000 / 10 = 5000, \text{ czyli metryka} = 19531 + 5000 = 24531$$



Protokół IGRP konfigurujemy podobnie jak protokół RIP, za pomocą głównego polecenia konfiguracyjnego **router** oraz podkomend **network**, których znaczenie i działanie jest takie samo, jak w protokole RIP. Zasadniczą różnicą jest stosowanie opcji **area** w poleceniu **router**, wskazującej identyfikator obszaru autonomicznego, w tym wypadku zwanego również domeną routingu. W przeciwieństwie do protokołu RIP, routery pracujące z protokołem IGRP mogą zostać logicznie przydzielone do różnych obszarów, w ramach których wymieniają się informacje. Standardowo routery pracujące w dwu różnych obszarach nie wymieniają informacji o sieciach. W naszym przykładowym układzie czterech routerów połączonych w pętli (patrz str. 69), konfiguracja routera c2600 mogłaby być taka:

```
c2600 (config) #router IGRP 10
c2600 (config-router) #network 131.107.0.0
c2600 (config-router) #network 131.108.0.0
```

Konfiguracja pozostałych routerów będzie analogiczna. Założono, że wszystkie routery pracują w obszarze 10. Zawartość tabeli routingu wyświetlić można poleceniem **show ip route**:

```
Gateway of last resort is not set
I 212.1.1.0/24 [100/80135] via 131.107.12.2, 00:00:21,
Serial0/0
I 10.0.0.0/8 [100/80225] via 131.107.13.2, 00:00:22,
Serial0/1
I 196.168.2.0/24 [100/82135] via 131.107.12.2, 00:00:22,
Serial0/0
    [100/82135] via 131.107.13.2, 00:00:22, Serial0/1
    131.108.0.0/24 is subnetted, 1 subnets
C 131.108.1.0 is directly connected, Ethernet0/0
    131.107.0.0/24 is subnetted, 4 subnets
I 131.107.10.0 [100/82125] via 131.107.13.2, 00:00:22,
Serial0/1
I 131.107.11.0 [100/82125] via 131.107.12.2, 00:00:22,
Serial0/0
C0 131.107.12.0 is directly connected, Serial0/0
C 131.107.13.0 is directly connected, Serial0/1
c2600#
```

Ponownie wyświetlane są dwie trasy do sieci 196.168.2.0 z jednakową metryką 82135. Wiarygodność protokołu IGRP wynosi 100. Wyobraźmy sobie teraz, że na routerze c2600, w konfiguracji interfejsu Serial 0/0 zwiększono parametr **Opóźnienie**, wykonując komendę **delay 160000** (wartość w dziesiątkach ms). Spowoduje to zwiększenie metryki prowadzącej do sieci 196.168.2.0 przez ten interfejs i po pewnym czasie trasa ta, jako gorsza, zostanie z tabeli routingu usunięta. Aktualny stan parametrów wpływających na metrykę wyświetlić można dla interfejsu S0/0 poleceniem **show interfaces S0/0**.

Bardzo ciekawą cechą protokołu IGRP jest możliwość rozłożenia ruchu pakietów na kilka tras o różnych metrykach. W omawianym powyżej przykładzie trasa prowadząca do sieci 196.168.2.0 przez interfejs S0/0 została usunięta z tabeli routingu z powodu gorszej metryki. Jeśli jednak w konfiguracji protokołu IGRP na routerze c2600 zastosujemy polecenie **variance zakres**, w tabeli routingu oprócz trasy najlepszej pojawią się również te, których metryka nie będzie niższa niż iloczyn zakresu i najlepszej metryki. **W naszym przykładzie najlepsza metryka trasy przez interfejs S0/1 wynosi 82135. Trasa przez interfejs S0/0 jest mniej niż 3 razy gorsza (zmieniono opóźnienie na 160000).** Aby w tabeli routingu pojawiły się obie trasy, należy wykonać następujące polecenia:

```
c2600(config)#router IGRP 10
c2600(config-router)#variance 3
```

Skutek obserwujemy w poleceniu **sh ip route**:

```
I 196.168.2.0/24 [100/240135] via 131.107.12.2,
00:00:16, Serial0/0
[100/82135] via 131.107.13.2, 00:00:16, Serial0/1
```

Jeżeli teraz na routerze c2600 włączone zostanie przełączanie pakiet po pakiecie (**no ip route-cache**), pakiety do sieci 196.168.2.0 rozdzielane będą między poszczególne interfejsy w stosunku 1 do 3 (wynika to ze stosunku metryk, a nie z komendy variance 3), co sprawdzić można po wydaniu polecenia **debug ip packet**. Polecenie **variance 3** oznacza, że w tabeli routingu pojawią się też trasy z metryką dwa razy gorszą od najlepszej, ale nie cztery razy gorszą.

Po przekroczeniu dozwolonej liczby skoków, dla określonej sieci ustawia się parametr DLYIGRP na wartość 0xFFFFF, co oznacza, iż sieć jest nieosiągalna.

Poszczególne parametry pracy protokołu IGRP (także RIP, jeśli jest włączony) wyświetlić można poleceniem **show ip protocol**. Zwróćmy uwagę na parametry czasowe, wartości stałych od k1 do k5, dozwoloną rozpiętość sieci (100), ogłaszane do sąsiadów sieci (podane klasowo), adresy sąsiadów, od których odbierane są

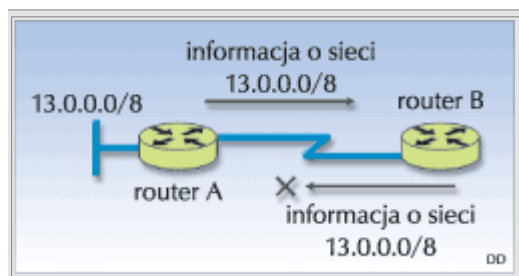
informacje:

```
c2600#sh ip protocol
Routing Protocol is "igrp 10"
  Sending updates every 90 seconds, next due in 9 seconds
  Invalid after 270 seconds, hold down 280, flushed after 630
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  IGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  IGRP maximum hopcount 100
  IGRP maximum metric variance 3
  Redistributing: igrp 10
  Routing for Networks:
    131.107.0.0
    131.108.0.0
  Routing Information Sources:
```

```
Gateway Distance Last Update
 131.107.12.2 100 00:01:03
 131.107.13.2 100 00:01:20
Distance: (default is 100)
```

Zapobieganie pętlom

Zarówno w sieciach wykorzystujących protokół RIP, jak i tych pracujących z protokołem IGRP może zdarzyć się, że routery zaczną odsyłać do siebie nawzajem pakiety, sądząc, że ten drugi wie, jak dostarczyć pakiet do sieci docelowej. Jest to przypadek pętli, która oprócz układu dwu sąsiednich routerów może obejmować swoim zasięgiem także większe grupy routerów (są to pętle rozległe). Wprowadzono kilka mechanizmów zabezpieczających sieć przed powstawaniem pętli. Pierwszy z nich, nazywany dzieleniem horyzontu (split horizon) oznacza, iż router nie może ogłaszać przez konkretny interfejs sieci, których nauczył się przez ten interfejs. Tę regułę nazywamy także blokadą ogłaszania wstecznego albo zakazem "uczenia swojego nauczyciela". Na zamieszczonym powyżej rysunku router B otrzymuje od sąsiada ogłoszenie o sieci 13.0.0.0 przez interfejs S0. Router B nie może wysłać informacji o sieci 13.0.0.0 z powrotem przez ten sam interfejs.



Reguła dzielenia horyzontu jest domyślnie włączona w konfiguracji interfejsów. Po jej wyłączeniu można doprowadzić do powstania pętli między routerami, co zaobserwujemy w przykładowym układzie czterech routerów fizycznie połączonych w pętlę z włączonym protokołem RIP. Aby wyłączyć regułę dzielenia horyzontu, w konfiguracji interfejsu należy wykonać komendę: **no ip split-horizon**. Skutek sprawdzamy w poleceniu **sh ip int s0/0**:

```
Serial0/0 is up, line protocol is up
Internet address is 131.107.12.1/24
Split horizon is disabled
```

Dla naszego przykładowego scenariusza dzielenie horyzontu należy wyłączyć w obydwu interfejsach szeregowych routera c2600 oraz w interfejsie s0 routera A i s1 routera B (sąsiedzi routera c2600). Dodatkowo na routerze c2600 usuwamy adres IP z interfejsu Ethernet 0/0 (131.108.1.1), co spowoduje usunięcie informacji o sieci 131.108.1.0 z tablicy routingu routera c2600. W przypadku systemu operacyjnego 11.3 router c2600 nie informuje o tym fakcie swoich sąsiadów, którzy w efekcie przysyłają do niego informacje o sieci 131.108.0.0 ze zwiększoną metryką (równą 2). Router c2600 musi te informacje zaakceptować, gdyż nie ma innych, lepszych i sam ogłasza sieć 131.108.0.0 do sąsiadów, ponownie zwiększając liczbę skoków (3). Routery A i B również muszą to ogłoszenie zaakceptować (mimo że zwiększa się metryka), gdyż router c2600 jest oryginalnym nauczycielem, od którego dowiedziały się o sieci 131.108.0.0. Tak powstaje pętla, na skutek której pakiety do sieci 131.108.0.0 router c2600 kierować będzie do routera A, a router A odsyłać będzie do c2600. W kolejnych cyklach czasowych, sieć 131.108.0.0 jest ogłaszana z coraz większą metryką, co bardzo dobrze można zaobserwować po uruchomieniu procesu śledzenia **deb ip RIP** na routerze c2600:

```
08:29:45: RIP: sending v1 update to 255.255.255.255 via
Serial0/1 (131.107.13.1)
08:29:45: network 131.108.0.0, metric 9
08:29:47: RIP: received v1 update from 131.107.13.2 on Serial0/1
08:29:47: 131.108.0.0 in 10 hops
```

Dla powyższych ogłoszeń zawartość tabeli routingu routera c2600 będzie następująca:

```
R 131.108.0.0/16 [120/10] via 131.107.13.2, 00:00:01, Serial0/1
```

W przypadku protokołu RIP pętla dla sieci 131.108.0.0 automatycznie się skończy po przekroczeniu dozwolonej liczby skoków (15), gdyż sieć ta zostanie oznaczona jako nieosiągalna (z metryką równą 16).

Pętle rozległe powstające w większej grupie routerów eliminuje się poprzez regułę dzielenia horyzontu z zatruciem wstecznym (split horizon with poisoned reverse). W przeciwieństwie do zwykłej reguły dzielenia horyzontu, tym razem zezwala się na ogłaszanie wsteczne, ale tylko w bardzo specyficznych sytuacjach. Wyobraźmy sobie parę routerów, w której router A regularnie ogłasza do routera B pewną sieć, za każdym razem zwiększając metrykę tej trasy (co może być skutkiem powstania pętli w innej części sieci). Router B "zgadza się" na pierwsze podniesienie metryki przez router A, ale przy kolejnym ogłoszeniu z ponownie większą metryką router B "domyśla się", iż w sieci powstała pętla i "zatrzuwa" trasę poprzez wysłanie ogłoszenia z powrotem do routera A, ale z metryką oznaczającą sieć nieosiągalną (16 dla protokołu RIP).

Powróćmy teraz do scenariusza omawianego powyżej, w którym wyłączono dzielenie horyzontu między routerem c2600 i jego sąsiadami (między tymi routerami automatycznie wyłączone jest też zatrucie wsteczne). W tym scenariuszu doprowadziliśmy do powstania pętli między routerem c2600 i jego sąsiadami, a skutek tego przenosi się również na router C. Między routerem A i C włączone jest dzielenie horyzontu oraz zatrucie wsteczne. Router A zaczyna ogłaszać sieć 131.108.0.0 do routera C z coraz większą metryką. Przy drugim podniesieniu metryki router C zatrzuwa trasę do sieci 131.108.0.0, wysyłając do routera A ogłoszenie z metryką 16. Router A otrzymuje więc informację, że do sieci 131.108.0.0 na pewno nie można pójść przez router C:

```
08:34:49: RIP: sending v1 update to 255.255.255.255 via Serial1
(131.107.11.2)
08:34:49: RIP: build update entries
08:34:49: network 131.108.0.0 metric 16
```

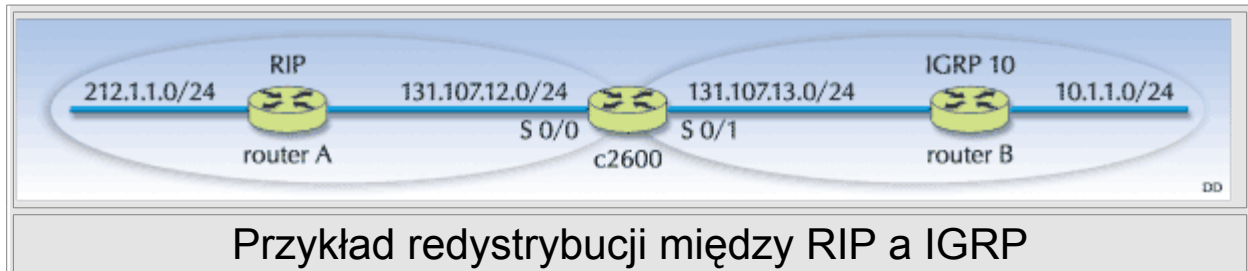
Kolejny mechanizm, którego celem jest przyspieszenie aktualizacji tabel routingu, to tzw. aktualizacja wymuszona (triggered update). Niezależnie od regularnych czasów aktualizacji (około 30 sekund dla protokołu RIP) router wysyła natychmiastowo do swoich sąsiadów informacje o sieci, dla której zmieniła się metryka (na lepszą bądź gorszą). Aktualizacja wymuszona przesyła tylko tę sieć, dla której zmieniła się metryka, a nie całą tabelę routingu, nie zakłóca też terminarza aktualizacji regularnych.

Przykładem aktualizacji wymuszonej jest włączenie polecenia **shutdown** w konfiguracji interfejsu. Sieć IP tego interfejsu jest natychmiastowo usuwana z tabeli routingu, a do sąsiadów wysyłane jest ogłoszenie, iż sieć ta jest nieosiągalna. W zależności od systemu operacyjnego ogłaszana sieć zostanie od razu usunięta z tabeli routingu innych routerów bądź ustawiona w tryb przytrzymywania (hold down). Pamiętajmy, że usunięcie adresu IP z interfejsu routera z systemem operacyjnym 11.3 nie powoduje aktualizacji wymuszonej (w wersji 12.0 już tak).

Redystrybucja danych routingu

Proces redystrybucji danych o routingu między różnymi źródłami informacji najczęściej konfiguruje się w środowisku, w którym uruchomione są różne protokoły routingu dynamicznego. Jego celem jest przekazanie informacji o sieciach zebranych w ramach jednego protokołu do innego protokołu routingu dynamicznego. Można sobie wyobrazić rozbudowaną sieć, w której część urządzeń pracuje z wykorzystaniem protokołu RIP, a pozostała część korzysta z protokołu IGRP. Dzięki redystrybucji wszystkie routery będą posiadały informacje o wszystkich segmentach sieci.

Specyficzną odmianą procesu redystrybucji jest rozgłaszanie tras statycznych przy wykorzystaniu protokołu routingu dynamicznego. Poprzez redystrybucję realizuje się także wymianę danych między różnymi obszarami w ramach protokołu IGRP. **Podstawowym problemem przy przekazywaniu informacji z jednego źródła do innego jest zmiana metryki. Nie można bowiem przenieść tras z protokołu RIP, gdzie metryka to liczba skoków mniejsza niż 16, do protokołu IGRP, w którym koszt trasy wyznaczany jest według zupełnie innych kryteriów.** W trakcie konfigurowania procesu redystrybucji w ramach określonego protokołu routingu dynamicznego należy podać wartość metryki, jaką będą otrzymywały wszystkie pobierane z innego źródła trasy. Prześledzimy teraz scenariusz, w którym router c2600 będzie realizował redystrybucję między protokołem RIP oraz IGRP (patrz rysunek).



Sieć z lewej strony routera c2600 (interfejs S0/0 oraz router A) korzysta z protokołu RIP, sieć z prawej strony (interfejs S0/1 oraz router B) stosuje protokół IGRP. **Redystrybucję włącza się na routerze brzegowym dla kilku źródeł informacji; w tym wypadku jest to router c2600, na którym uruchomiono zarówno protokół RIP, jak i IGRP. Podstawowym poleceniem konfiguracyjnym jest komenda **redistribute parametr**** wykonywana wewnątrz procesu, do którego dane pobierane są z zewnętrznego źródła wskazanego jako parametr. Przykładowa konfiguracja routera c2600 mogłaby być następująca:

```
c2600 (config)#router rip
c2600 (config-router)#redistribute igrp 10 metric 5
c2600 (config-router)#passive-interface Serial 0/1
c2600 (config-router)#network 131.107.0.0
c2600 (config)#router igrp 10
c2600 (config-router)#redistribute rip
c2600 (config-router)#default-metric 1000 100 255 1 1500
c2600 (config-router)#passive-interface Serial 0/0
c2600 (config-router)#network 131.107.0.0
```

Zwróćmy uwagę na dwa sposoby określania metryki dla pobieranych tras. **Metrykę można zdefiniować, stosując opcję **metric** w komendzie **redistribute**.** W powyższym przykładzie trasy pobierane do protokołu RIP z obszaru 10 protokołu IGRP będą miały metrykę 5. Można także zdefiniować domyślną metrykę dla wszystkich poleceń **redistribute**, stosując komendę **default-metric** z odpowiednimi parametrami. W powyższym przykładzie polecenie **default-metric** wymagało aż pięciu parametrów, z których liczona jest metryka w protokole IGRP (Przepustowość, Opóźnienie, Pewność, Obciążenie, MTU). Dla protokołu RIP wymagany jest tylko jeden parametr - liczba skoków. Polecenie **passive-interface** blokuje ogłaszanie danego protokołu przez wskazany interfejs. Gdybyśmy dodatkowo chcieli w ramach protokołu RIP ogłaszać trasy statycznie wpisane do tabeli routingu routera c2600, należałoby wykonać następujące polecenie:

```
c2600 (config)#router rip
c2600 (config-router)#redistribute static metric 5
```

Redystrybucję między dwoma obszarami protokołu IGRP.

