



pasja-informatyki.pl

Sieci komputerowe

**Funkcje warstwy sieciowej,
wprowadzenie do routingu.**

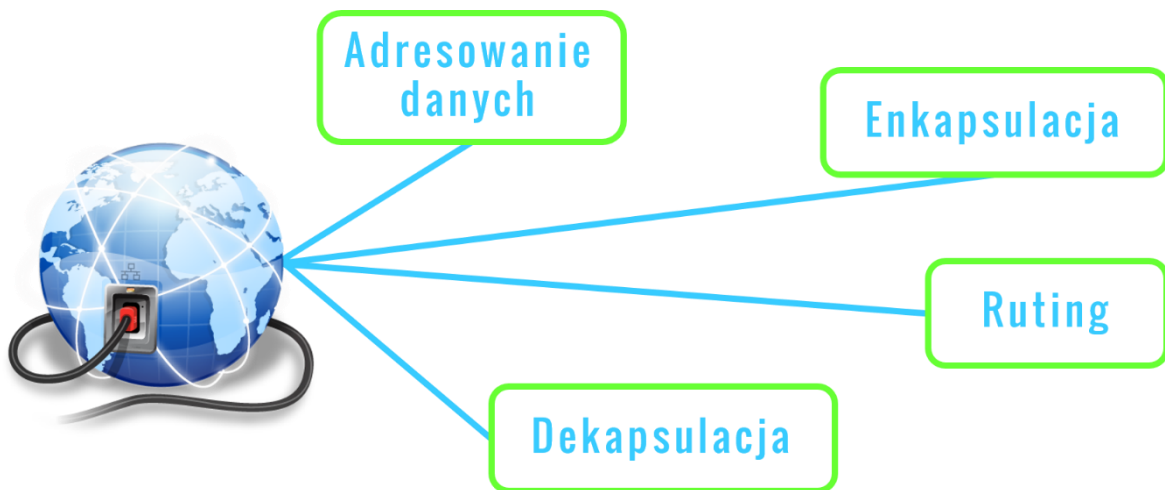
Damian Stelmach

Spis treści

Zadania warstwy sieciowej	3
Protokoły warstwy sieciowej	4
Adresowanie IPv4	8
Ruting	12
Testowanie warstwy sieciowej	18

Warstwa sieciowa (model ISO/OSI – 3 warstwa), zwana również **warstwą internetową** odbiera posegmentowane dane z warstwy transportowej, a następnie wykonuje działania, dzięki którym pakiet może zostać wysłany przez sieć. Do tych działań zaliczyć trzeba:

- **adresowanie danych** z wykorzystaniem adresów IP;
- **enkapsulację danych**, czyli przypisanie dodatkowych informacji wymaganych przez stosowany protokół warstwy sieci;
- **ruting**, czyli dobór najlepszej trasy dla pakietu;
- **dekapsulację**, czyli usunięcie tych dodatkowych informacji, kiedy pakiet osiągnie cel.



Wiemy, że komunikacją sieciową rządzą określone reguły, czyli protokoły komunikacyjne. Wiemy też, że każda z warstw wykorzystuje swoje, niezależne od innej warstwy protokoły. Nie inaczej jest z warstwą sieci, w której również one występują. Najpopularniejszym protokołem komunikacyjnym tej warstwy jest **protokół IPv4**. Najważniejszym powodem jego stosowania jest fakt, iż jest to protokół otwarty. To znaczy, nie jest on własnością żadnej firmy czy korporacji, dzięki czemu umożliwia komunikację pomiędzy urządzeniami różnych producentów. Po piętach depta mu już **protokół IPv6**, który również jest otwartym protokołem. Na chwilę obecną, wielu producentów urządzeń i oprogramowania stosuje równolegle te protokoły. Być może w przyszłości, IPv6 całkowicie wyprze IPv4, ale według mnie nie nastąpi to zbyt szybko. Oczywiście istnieją również protokoły stanowiące własność konkretnych firm, można wśród nich wymienić **protokół IPX**, stanowiący własność firmy **Novell** specjalizującej się tworzeniu sieciowych systemów operacyjnych, czy protokół **AppleTalk**, stworzony przez **Apple**. Z pełną stanowczością można jednak stwierdzić, że **protokół IPv4** to zdecydowanie najczęściej stosowany protokół warstwy sieci.

Protokół IPv4 został zaprojektowany w taki sposób, aby **nie wymagał dużej ilości danych sterujących** dodawanych w procesie enkapsulacji. Zapewnia tylko podstawowe funkcje, niezbędne do przesyłania pakietów od źródła do celu. Jest **bezpołączeniowy**, co oznacza, że nie ustanawia połączenia przed wysłaniem danych, działa w myśl zasady „**najlepiej, jak to możliwe**” (ang. **Best effort**), co oznacza, że nie wykorzystuje kontroli przepływu ani żadnych potwierdzeń dostarczania danych tak jak było to w protokole TCP, ale dokłada wszelkich starań, aby komunikacja przebiegała pomyślnie. Jest to również protokół **niezależny od nośnika**, to znaczy, że dane pomiędzy hostami mogą przesyłane być bez względu na zastosowane medium transmisyjne.



W jednej sieci możemy mieć przecież kabel typu skrętka, w drugiej światłowód, a w trzeciej fale radiowe. Protokół IP, będzie działał dokładnie tak samo w każdej z tych sieci. Problemem, jaki może się

pojawić podczas przesyłania danych przez różne media jest maksymalna wielkość pakietu, czyli wartość **MTU** (ang. **Maximum Transmission Unit**), jeśli pakiet jest zbyt duży, to połączony do sieci ruter podzieli go na mniejsze części. Proces ten nazywamy **fragmentacją** – kolejne pojęcie do naszego sieciowego słownika.

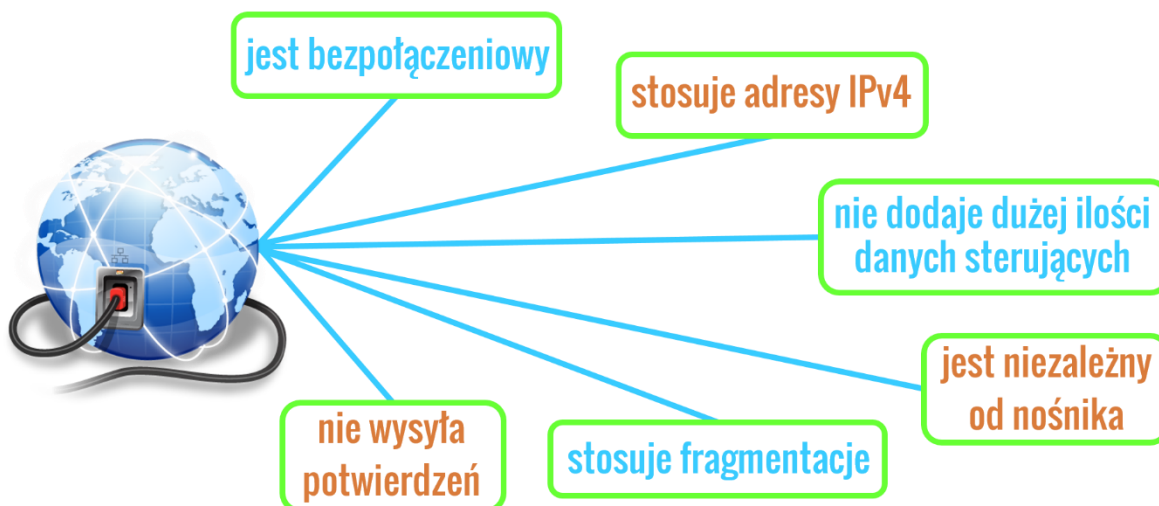
Aby łatwiej było zrozumieć działanie **protokołu IPv4** oraz to, w jaki sposób pakiety danych przesyłane są przez Internet zaprezentuje jego działanie na przykładzie paczki od cioci z Ameryki. Na paczkę składają się 3 kartony, połączone w jedną całość. Ciocia, zaadresowała prezent i dostarczyła go do firmy kurierskiej. Przy nadaniu paczki zrezygnowała ze wszystkich opcji dodatkowych, takich jak potwierdzenie odbioru czy śledzenie paczki. Pracownik firmy przykleił na kartony naklejkę z adresem docelowym oraz zwrotnym i przekazał paczkę dalej. Została ona, wraz z dziesiątką innych przesyłek, samochodem dowieziona do portu, gdzie zapakowano ją do kontenera i statkiem wyruszyła w podróż za ocean. W porcie docelowym kontener rozpakowano, posegregowano paczki, następnie samochodami rozwieziono do poszczególnych miast i tamtejszych punktów odbioru. Z punktu odbioru, samochodem, paczka ma zostać dowieziona pod dany adres, ale okazuje się, że trzy połączone kartony są zbyt duże, aby przewieźć je małym samochodem, dlatego kurier dzieli ją na pojedyncze kartony i w taki sposób Ci je dostarcza. Na koniec, kiedy paczka została odebrana, wykonywany jest telefon do cioci z podziękowaniem za prezent.

Przekładając to na komunikację z wykorzystaniem **protokołu IP** będzie tak:

- **pakiet został wysłany bez wcześniejszego poinformowania odbiorcy** – mamy tryb bezpołączeniowy;
- **w procesie enkapsulacji został nadany adres źródłowy i docelowy** – w naszym przykładzie to był adres zamieszkania odbiorcy jako adres docelowy, i adres zamieszkania cioci jako adres zwrotny;
- **pakiet nie został zaopatrzony w dużą ilość danych sterujących mogących spowolnić komunikację** - w tym celu ciocia zrezygnowała z opcji dodatkowych, czyli z potwierdzenia i śledzenia paczki;
- **pakiet dotarł do celu poprzez media światłowodowe, skrętkę i fale radiowe** - no bo paczka została dostarczona różnymi środkami transportu: statkiem, dużym samochodem, małym samochodem;
- **pakiet był za duży aby przesłać go w całości, przez jedną z sieci dlatego został pofragmentowany** - czyli paczka została w pewnym momencie podzielona, aby można było wykorzystać do transportu mały samochód;

- **protokół IP nie wysłał potwierdzenia, że pakiet został odebrany** – tak jak firma nie zapewniła cioci, że paczka dotarła na miejsce.

PROTOKÓŁ IPv4



Jak każdy protokół komunikacyjny, również IPv4 ma **ustandaryzowany nagłówek**, pozwalający na dodawanie informacji sterujących. Przykład typowego nagłówka **IPv4** widoczny jest poniżej.

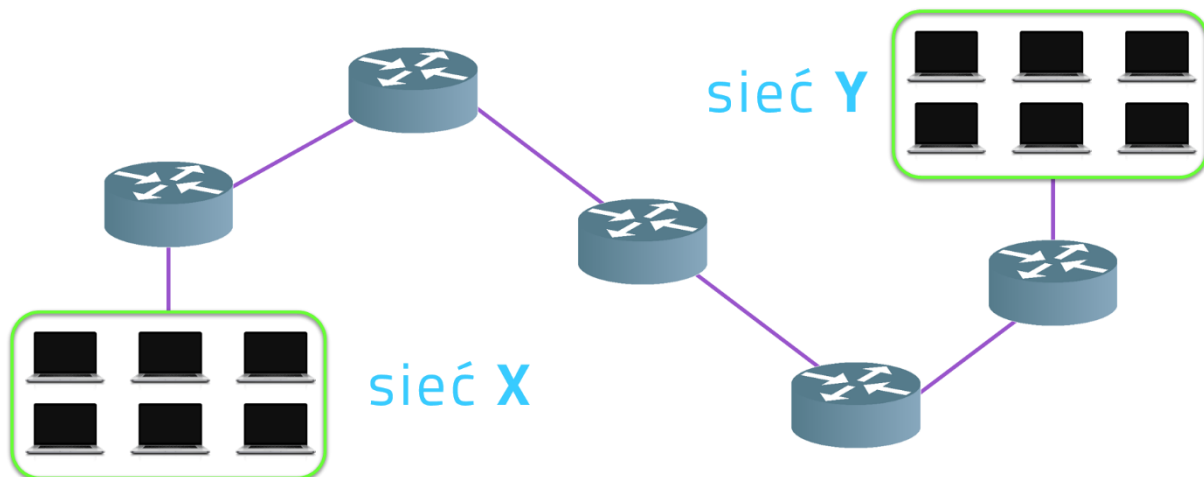
Wersja	IHL	Typ usługi	Długość pakietu	
Identyfikacja			Flaga	Przesunięcie fragmentu
TTL	Protokół		Suma kontrolna nagłówka	
Adres źródłowy				
Adres docelowy				
Opcje			Wypełnienie	

- **Docelowy adres IP** – adres IP odbiorcy danych;
- **Źródłowy adres IP** - adres IP nadawcy danych;
- **Czas życia (TTL)** – 8 bitowe pole, które określa pozostały czas życia pakietu. Wartość TTL jest zmniejszana o co najmniej 1 za każdym razem, gdy pakiet przechodzi przez ruter (tj. za każdym przeskokiem). Kiedy wartość osiąga 0, ruter porzuca pakiet i jest on usunięty ze strumienia

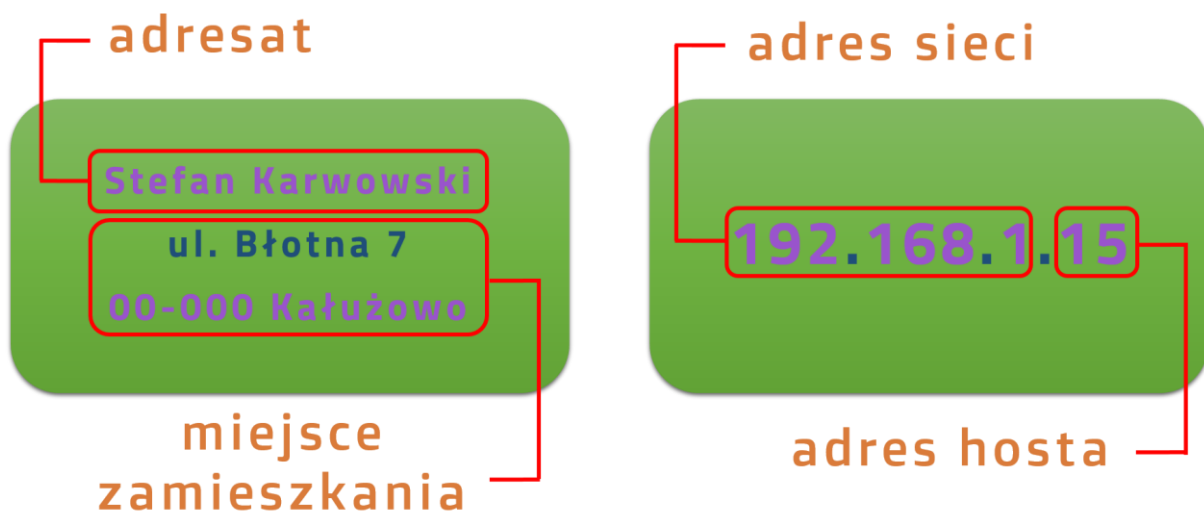
danych w sieci. Mechanizm ten chroni pakiety, które nie mogą osiągnąć celu przed przekazywaniem w nieskończoność pomiędzy ruterami w tzw. pętli rutingu. Jeżeli pętle rutingu byłyby dozwolone, sieć byłaby przeciążona pakietami danych, które nigdy nie osiągną swojego celu. Zmniejszanie wartości TTL przy każdym przeskoku zapewnia, że w końcu osiągnie wartość 0 i pakiet, którego pole TTL równe jest 0 będzie odrzucone.

- **Protokół** - ta 8-bitowa wartość, określająca wykorzystany protokół warstwy wyższej, np. **UDP** lub **TCP**.
- **Typ usługi (ToS)** - zawiera 8-bitową wartość, która używana jest do określenia priorytetu każdego pakietu.
- **Przesunięcie fragmentu** – pole stosowane podczas rekonstrukcji podzielonego przez ruter pakietu. Wskazuje porządek, w jakim ma być ustawiony każdy z pakietów podczas rekonstrukcji.
- **Flaga MF** (ang. More Fragments) – pojedynczy bit używanym z polem przesunięcia fragmentu do podziału i rekonstrukcji pakietów. Gdy bit **flagi MF** jest ustawiony, oznacza to, że dany fragment nie jest ostatnim fragmentem pakietu. Kiedy host odbierający zauważy przybywający pakiet z ustawioną wartością **MF=1**, sprawdza pole przesunięcia fragmentu, gdzie należy umieścić ten fragment podczas rekonstrukcji pakietu. Kiedy host odbierający zauważy przybywający pakiet z ustawioną wartością **MF=0** i niezerową wartość w polu przesunięcia fragmentu, umieszcza ten fragment jako ostatni kawałek rekonstruowanego pakietu.
- **Flaga DF** (ang. Don't Fragment) – pojedynczy bit, który jeśli jest ustawiony wskazuje, że fragmentowanie pakietu jest niedozwolone. Jeżeli **flaga DF** jest ustawiona, wtedy fragmentacja tego pakietu nie jest dozwolona.
- **Wersja** - zawiera numer wersji protokołu IP (w tym wypadku to będzie **IPv4**).
- **Długość nagłówka (IHL)** – określa rozmiar nagłówka pakietu.
- **Długość pakietu** – to pole podaje w bajtach całkowitą wielkość pakietu, zawierającą nagłówki oraz dane.
- **Identyfikacja** – to pole jest używane do jednoznacznego identyfikowania fragmentów podzielonego pakietu IP.
- **Suma kontrolna nagłówka** – pole używane jest do sprawdzenia błędów nagłówka pakietu.
- **Opcje** – jest to miejsce na dodatkowe pola w nagłówku IPv4 do obsługi innych usług. Jest ono jednak rzadko używane.

Jednym z kluczowych zadań warstwy sieci jest adresowanie. Adresowanie w sieciach IP jest bardzo podobne do adresowania stosowanego przez nas, ludzi. Oczywiście tylko na poziomie logicznym, mechanizmy adresowania są różne. Hosty w sieciach zostały pogrupowane w celu łatwiejszego nimi zarządzania i adresowania.



Podobnie jak u ludzi, też mieszkamy w miastach, na konkretnych ulicach. Dzięki temu wspomniana wcześniej paczka od cioci z Ameryki łatwo mogła dotrzeć do odbiorcy. Najpierw została wysłana promem do Polski, potem ciężarówką do Twojego miasta, następnie mniejszym autem pod wskazaną ulicę i numer domu. Bardzo podobnie jest z adresowaniem hostów. Pakiet przesyłany pomiędzy sieciami, najpierw trafia do sieci, do której przynależy host, a potem przesyłany jest już do konkretnego hosta. Ten typ adresowania nazywany jest **adresowaniem hierarchicznym**, bo najpierw odczytywane są informacje ogólne, czyli w przypadku przesyłania danych adres sieci, potem dopiero te szczegółowe, czyli adres IP konkretnego hosta.



Rozbudowany tutorial dotyczący adresowania IP, wraz z omówieniem jak wykonywać obliczenia na adresach IPv4, znajdziesz na kanale

Pasja Informatyki, dostępny jest pod tym linkiem:

<https://youtu.be/t3lceGltjig>

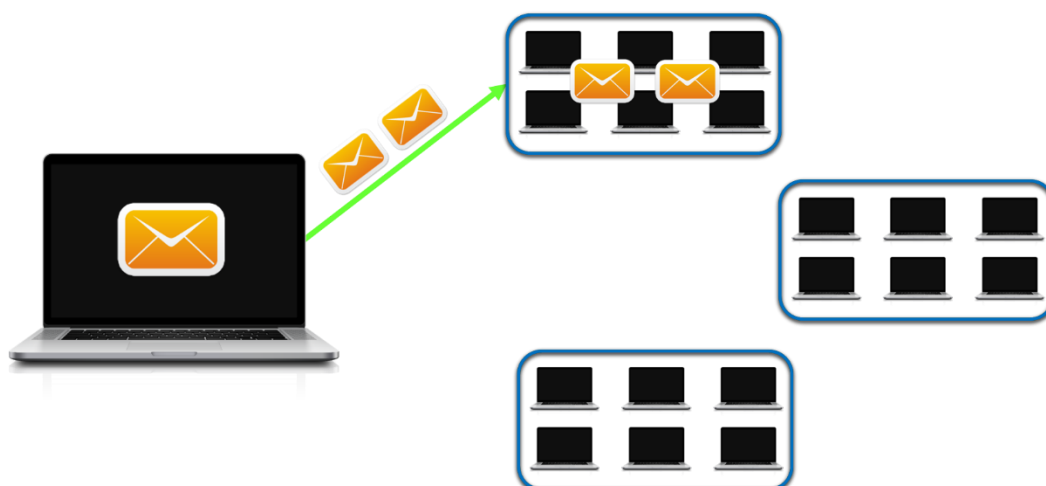
W sieciach komputerowych hosty mogą się ze sobą komunikować na trzy sposoby:

- *z wykorzystaniem transmisji pojedynczej*, (ang. Unicast);
- *poprzez rozsyłanie grupowe*, (ang. Multicast);
- *poprzez rozgłaszanie*, (ang. Broadcast).

Transmisja typu **unicast** stosowana jest najczęściej, wykorzystywana jest w typowych połączeniach pomiędzy dwoma hostami. Przykładowo, kiedy klient wysyła żądanie do serwera, wykorzystuje do tego właśnie transmisję **unicast**.



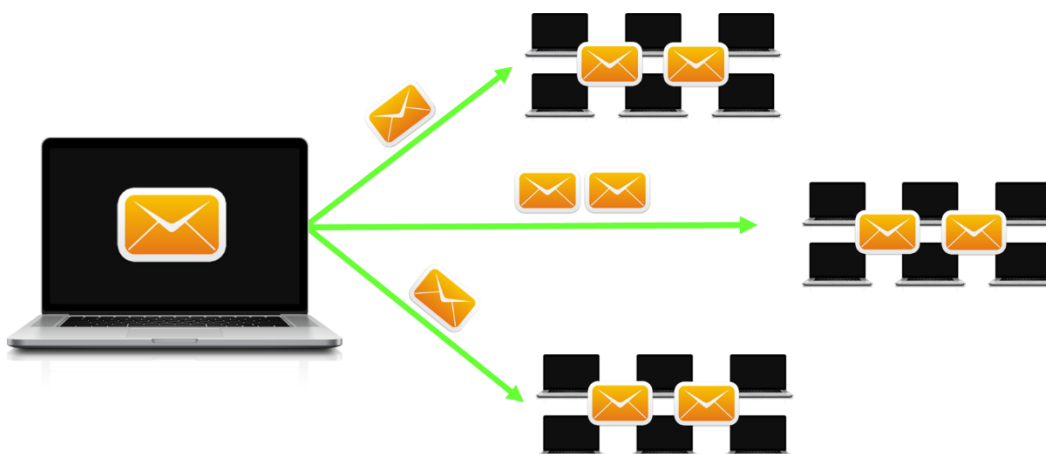
Zastosowanie transmisji grupowej, czyli **multicast** w znacznym stopniu pozwala zredukować zużycie pasma w sieci ponieważ nie wysyła się pojedynczych pakietów do wielu hostów, tak jak byłoby to realizowane z użyciem transmisji pojedynczej, lecz wysyła się jeden pakiet, który trafić może do wielu odbiorców jednocześnie.



Multicast może być stosowany przez routery do wymiany informacji związanych z routowaniem, jak również do dystrybucji oprogramowania. W transmisji multicast stosuje się specjalną pulę adresów, zwaną adresami grupowymi, w protokole IPv4 to jest zakres, który widoczny poniżej:

od **224.0.0.0** do **239.255.255.255**

Broadcast, czyli rozgłaszanie, z kolei polega na wysyłaniu pakietów do wszystkich hostów w danej sieci. Wykorzystywany jest do tego specjalny adres, adres rozgłoszeniowy, więc nie jest tak, że w pakiecie IP zapisane są adresy wszystkich hostów w sieci. Byłoby to technicznie nie możliwe, to raz, a dwa transmisję rozgłoszeniową stosuje się na przykład wówczas, kiedy nieznany jest adres konkretnego urządzenia. Ten rodzaj transmisji wykorzystywany jest najczęściej w sieciach lokalnych, rzadko zdarza się, że rozgłaszanie stosuje się do komunikacji z hostami spoza danej sieci lokalnej.

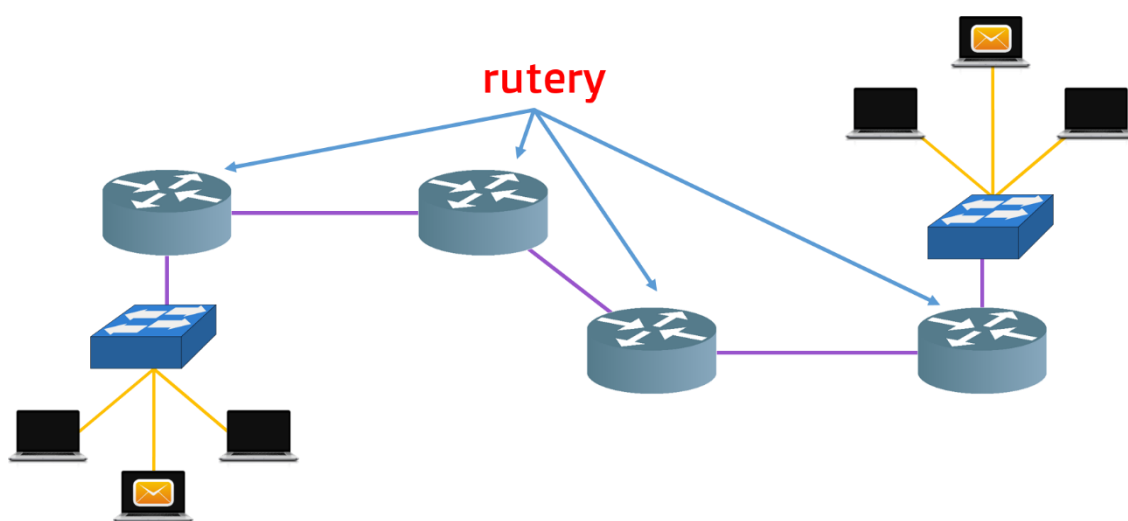


W całej **puli adresów IPv4** wyodrębniono grupy adresów, tzw. **specjalnego zastosowania**. Są to adresy, które niestosowane są do komunikacji w sieciach rozległych. Wśród tych adresów specjalnych znajdują się adresy, tak zwanych **pętli zwrotnych** (ang. **loopback**). Adres pętli zwrotnej to nic innego jak adres samego siebie, każdy komputer w sieci, oprócz właściwego adresu IP wykorzystywanego do komunikacji, ma przypisany również adres samego siebie, najczęściej jest to adres **127.0.0.1**. Ponadto, każdy adres z puli służy do testowania poprawności konfiguracji protokołu IPv4 na hoście. Innym rodzajem adresów specjalnych, są **adresy łącza lokalnego**, (ang. **Link-local**). Tego typu adresy stosuje się wówczas, kiedy nie jest dostępna inna konfiguracja adresów IP, czyli np. w przypadku kiedy **serwer DHCP nie jest dostępny**. Transmisja danych z wykorzystaniem łącza lokalnego może odbywać się tylko w obrębie sieci lokalnej, w której pracuje dany host. No i jeszcze jedna, ostatnia już grupa adresów specjalnych, czyli adresy typu **TEST-NET**. Podobnie jak adresy lokalnego łącza służą one do komunikacji

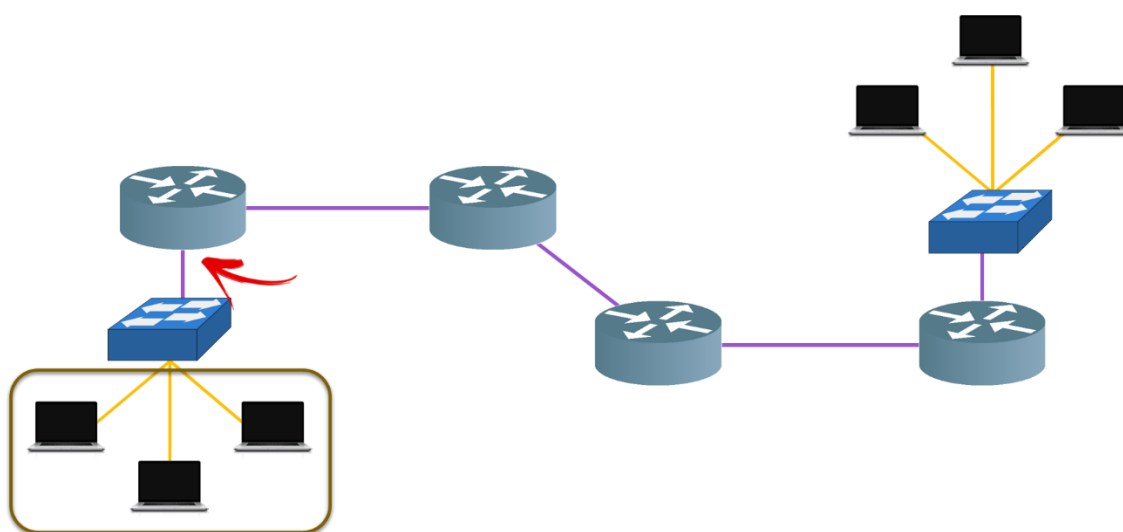
tylko w obszarze sieci lokalnej, stosowane są w celach edukacyjnych. Mogą być wykorzystywane w dokumentacji czy też w przykładach, np. podczas zajęć z sieci. Nie powinny natomiast być używane na stałe. Zakresy adresów specjalnych widoczne są w tabeli poniżej.

Zakres adresów	Nazwa
127.0.0.1 ÷ 127.255.255.254	Pętla zwrotna (ang. Loopback)
169.254.0.1 ÷ 169.254.255.254	I-Łcze lokalne (ang. Local-Link)
192.0.2.0 ÷ 192.0.2.254	Edukacyjne (ang. Test-Net)

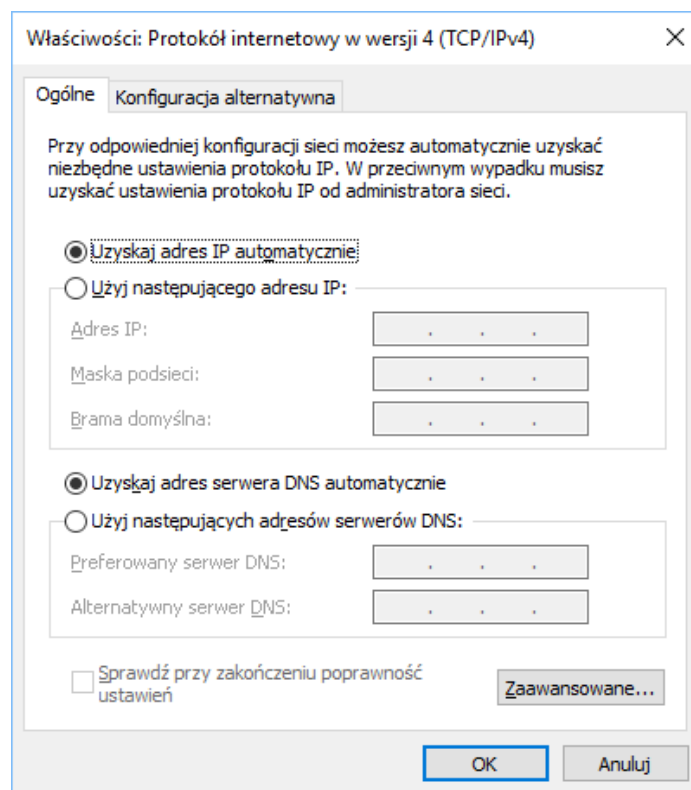
Kiedy dane przesyłane są pomiędzy hostami w sieci lokalnej to komunikacja między nimi odbywa się **bez udziału urządzeń pośredniczących**, wykorzystywane są oczywiście przełączniki, czyli switch'e, ale ich, szczególnie tych prostych, niezarządzanych przełączników, pracujących tylko w warstwie 2, nie nazywamy urządzeniami pośredniczącymi. W momencie, kiedy jednak nasz komputer chciałby wysłać dane do komputera z innej sieci, no to już takie **urządzenie pośredniczące jest potrzebne**. W sieci komputerowej urządzenie służące do przesyłania pakietów pomiędzy różnymi sieciami to **ruter**, natomiast proces określania trasy przesyłu pakietów nazywany jest **procesem rutowania**.



Aby dany host był w stanie wysłać pakiet do hosta znajdującego się w innej sieci, w ramach swojej konfiguracji musi mieć zdefiniowany adres tak zwanej **bramy domyślnej**. Brama to interfejs routera, bądź też serwera, który jest podłączony do tej samej sieci co host.



Konfiguracja bramy domyślnej na komputerach jest bardzo prosta, jeśli korzystasz z usługi DHCP, która przydziela na automatycznie adresy IP, w ogóle nie musisz się tym przejmować, DHCP zrobi to za Ciebie, jeśli natomiast statycznie, ręcznie przydzielasz adresy komputerom w sieci, to w systemie Windows bramę domyślną skonfigurujesz, zmieniając ustawienia karty sieciowej.



Jak widać, ten komputer korzysta z **serwera DHCP**, dlatego adres bramy jest tutaj niewidoczny. Można go sprawdzić **używając konsoli Windows**, należy w niej wykonać polecenie **ipconfig**. Wyświetli się wtedy konfiguracja IP komputera i można z niej odczytać adres bramy.

```
C:\WINDOWS\system32\cmd.exe

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Ethernet adapter Połączenie lokalne:

    Connection-specific DNS Suffix . :
    Link-local IPv6 Address . . . . . : fe80::49d3:f0b4:9dda:8c59%16
    IPv4 Address. . . . . : 192.168.0.101
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Tunnel adapter isatap.{B95CC550-1AD0-42F0-819F-AC499921B983}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :
Tunnel adapter Teredo Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix . :
    IPv6 Address. . . . . : 2001:0:5ef5:79fd:3cd2:30f4:3f57:ff9a
    Link-local IPv6 Address . . . . . : fe80::3cd2:30f4:3f57:ff9a%15
    Default Gateway . . . . . :

C:\Users\damian>
```

Każdy host w sieci, zarówno komputer, jak i ruter – przypominam, że interfejs routera ma przypisany adres IP, tak więc również jest hostem w sieci, posiada tzw. **tablicę routingu**, w której zapisywane są informacje o trasach do sieci docelowych, zarówno tych podłączonych, jak i odległych. Przykładowa **tablica routingu**, dla routerów marki CISCO widoczna jest poniżej.

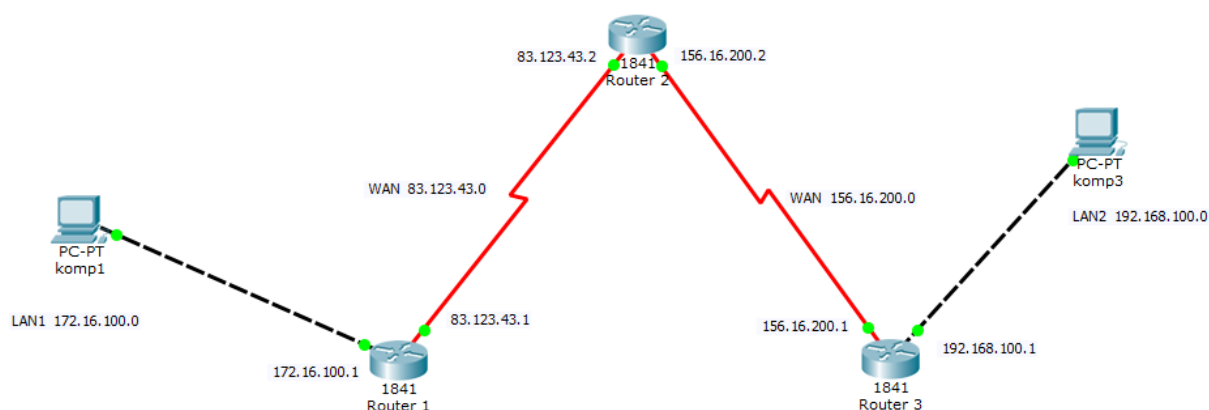
```
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

      83.0.0.0/24 is subnetted, 1 subnets
C       83.123.43.0 is directly connected, Serial0/1/0
R       156.16.0.0/16 [120/1] via 83.123.43.2, 00:00:27, Serial0/1/0
      172.16.0.0/24 is subnetted, 1 subnets
C       172.16.100.0 is directly connected, FastEthernet0/0
R       192.168.100.0/24 [120/2] via 83.123.43.2, 00:00:27, Serial0/1/0
```

Literką C oznaczone są sieci **bezpośrednio połączone**, czyli takie, do których **ruter** podłączony jest za pomocą medium transmisyjnego. **Literka R**, z kolei oznacza, że ruter otrzymał informację o tych sieciach od innych routerów z wykorzystaniem **protokołu RIP**. Dzięki temu ruter jest w stanie wysłać pakiet do sieci, nawet jeśli nie jest do niej bezpośrednio podłączony.

Poniżej podstawiony został proces przesyłania danych, w przykładowej sieci.



1. **Komputer 1**, chce wysłać jakieś dane do **komputera 3**.
2. Pakiet trafia najpierw do interfejsu routera stanowiącego **bramę** dla **komputera 1**.
3. **Ruter 1**, sprawdza w nagłówku pakietu **adres IP hosta docelowego** i porównuje go ze wpisami w swojej **tablicy routingu**.

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
 * - candidate default, U - per-user static route, o - ODR
 P - periodic downloaded static route

Gateway of last resort is not set

```

83.0.0.0/24 is subnetted, 1 subnets
C    83.123.43.0 is directly connected, Serial0/1/0
R    156.16.0.0/16 [120/1] via 83.123.43.2, 00:00:07, Serial0/1/0
    172.16.0.0/24 is subnetted, 1 subnets
C    172.16.100.0 is directly connected, FastEthernet0/0
R    192.168.100.0/24 [120/2] via 83.123.43.2, 00:00:07, Serial0/1/0
  
```

Ruter 1 ma wpisaną trasę do sieci docelowej, która przechodzi przez interfejs **rutera 2**

4. **Ruter 1** wysyła pakiet do **rutera 2**, będący **następnym skokiem** na trasie (ang. next hop).
5. Teraz **ruter 2** odczytuje pakiet, i sprawdza w swojej **tablicy routingu** czy ma trasę do sieci, do której ma zostać wysłany pakiet.

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
 * - candidate default, U - per-user static route, o - ODR
 P - periodic downloaded static route

Gateway of last resort is not set

```

83.0.0.0/24 is subnetted, 1 subnets
C    83.123.43.0 is directly connected, Serial0/1/0
    156.16.0.0/24 is subnetted, 1 subnets
C    156.16.200.0 is directly connected, Serial0/1/1
R    172.16.0.0/16 [120/1] via 83.123.43.1, 00:00:16, Serial0/1/0
R    192.168.100.0/24 [120/1] via 156.16.200.1, 00:00:13, Serial0/1/1
  
```

Ruter 2 ma wpisaną trasę do sieci docelowej, która przechodzi przez interfejs **rutera 3**

6. **Ruter 2** przesyła pakiet do **rutera 3**.
7. **Ruter 3** również odczytuje adres IP i sprawdza **tablicę routingu**.

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
 * - candidate default, U - per-user static route, o - ODR
 P - periodic downloaded static route

Gateway of last resort is not set

```

R    83.0.0.0/8 [120/1] via 156.16.200.1, 00:00:09, Serial0/1/0
    156.16.0.0/24 is subnetted, 1 subnets
C    156.16.200.0 is directly connected, Serial0/1/0
R    172.16.0.0/16 [120/2] via 156.16.200.1, 00:00:09, Serial0/1/0
C    192.168.100.0/24 is directly connected, FastEthernet0/0
  
```

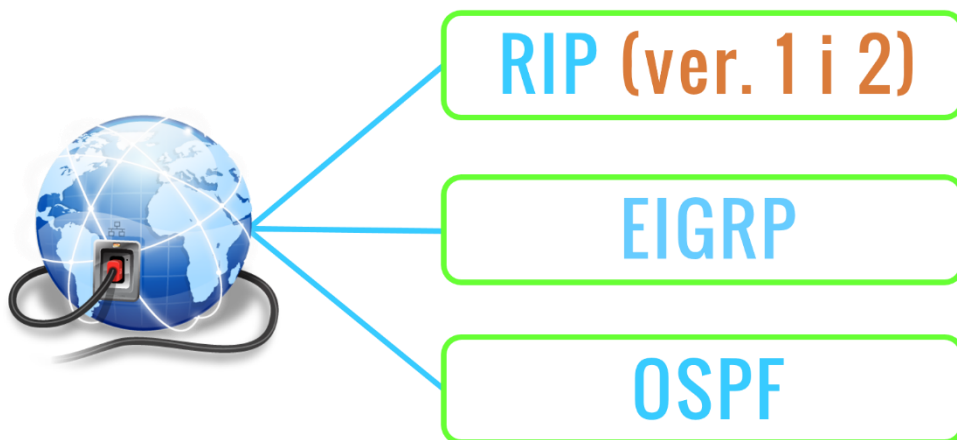
Sieć docelowa, jest siecią bezpośrednio podłączoną **rutera 3**

8. **Ruter 3**, jest ruterem lokalnej sieci, do której adresowany jest pakiet zatem przekazuje go do hosta docelowego, czyli do **komputera 3**.

Na sam koniec kilka słów o tym, w jaki sposób **rutery zdobywają informacje o trasach**. Właściwie to sposoby na zdobycie tych informacji są dwa. Pierwszy sposób to **ruting statyczny**, a drugi to **ruting dynamiczny**. Zarówno jeden, jak i drugi ma swoje wady i zalety. W przypadku routingu statycznego, tras statycznych, główną rolę odgrywa **administrator sieci**. To on odpowiedzialny jest za bieżącą aktualizację tras na routerze, podobnie jak odpowiedzialny jest za statyczne przypisywanie **adresów IP** na komputerach użytkowników. **Zaletą stosowania tras statycznych** jest ich niezawodność oraz znacznie mniejsze wykorzystanie mocy obliczeniowej routerów, wymaganej do przetwarzania danych. Z drugiej jednak strony pojawia się kwestia stałego nadzoru administratora i **ciągłego, ręcznego aktualizowania tras**, bo musisz zdawać sobie sprawę, że w dobie błyskawicznego rozwoju nie tylko informatyki, ale technologii w ogóle, pojawianie się nowych tras, czy znikanie starych wcale nie jest rzadkością. Jeśli administrator nie będzie nadążał z aktualizacją tras, to w pewnym momencie może to skutkować **sporymi opóźnieniami lub nawet utratą pakietów**.

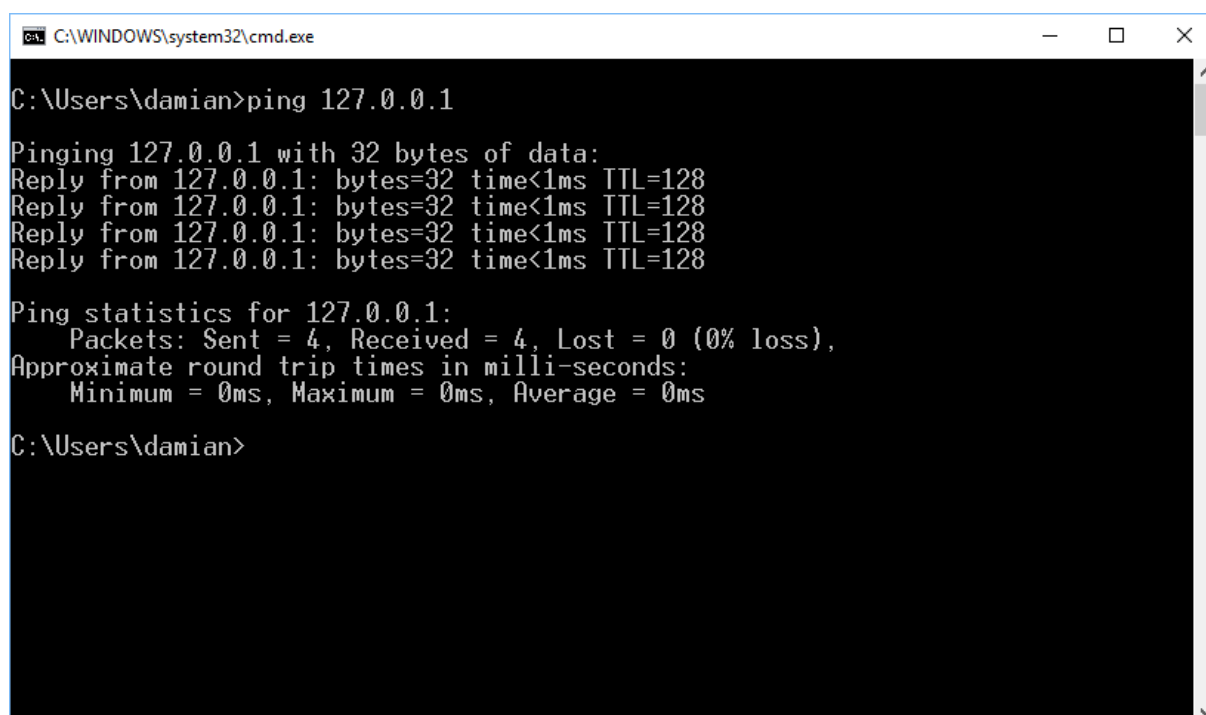
Po drugiej stronie rzeki mamy **ruting dynamiczny i protokoły routingu**. Dzięki routingowi dynamicznemu routery mogą **automatycznie uczyć się tras**, zdobywają o nich informacje od innych routerów, **bez ingerencji administratora**. Oczywiście konieczna jest ich **początkowa konfiguracja** protokołów routingu. Jest to wygodne rozwiązanie, gdyż zmiana topologii sieci czy pojawienie się nowych tras nie wymaga ręcznego ich uaktualniania. Oczywiście kija ma dwa końce i ta dynamiczność niesie za sobą również pewne problemy. Ruting dynamiczny **powoduje większe obciążenie** routerów związane z przetwarzaniem przez nie danych. Każda zmiana w tablicy wymaga **wykonywania skomplikowanych obliczeń**, dlatego też routery muszą być wyposażone w sporą moc obliczeniową. Ponadto przesyłanie informacji o sieciach przez routery powoduje **obciążenie w sieci**. Pomimo tych niedogodności, ruting dynamiczny jest **chętnie stosowany przez administratorów**, gdyż jest to zdecydowanie skuteczniejszy sposób na utrzymanie aktualnych informacji o trasach niż ruting statyczny. Jednymi z najczęściej spotykanych protokołów routingu są protokoły **RIP**, w wersji 1 i 2, **EIGRP** oraz **OSPF**.

Ruting dynamiczny – protokoły routingu



W każdym systemie operacyjnym zaimplementowane są programy, która umożliwiają nam wykonanie testów warstwy sieciowej. Jednym z nich jest program **PING**, wykorzystywany do **testowania połączeń pomiędzy hostami**. Jest on dostępny pod tą nazwą zarówno w systemach Windows, jak i w różnych dystrybucjach Linuxa. Drugi natomiast to program **TRACERT**, służący do **testowania trasy pomiędzy hostem źródłowym i docelowym**. W systemach opartych na jądrze linuxa, ten sam program nosi nazwę **TRACEROUTE**. Program **PING**, korzystając z innego protokołu warstwy sieci, a mianowicie **protokołu ICMP**, wysyła on datagram z **żądaniem echo** i czeka na odpowiedź. Po uzyskaniu odpowiedzi wyświetla nam czas jak minął od **wysłania żądania do uzyskania informacji zwrotnej**. **PING** może być wykorzystany do testowania:

- tzw. **lokalnego stosu**, czyli do sprawdzenia poprawności instalacji **protokołu IP** na komputerze, wystarczy, że wprowadzimy w **konsoli Windows** polecenie **PING**, z jednym z adresów pętli zwrotnej, czyli z zakresu od **127.0.0.1** do **127.255.255.254**:



```
C:\WINDOWS\system32\cmd.exe

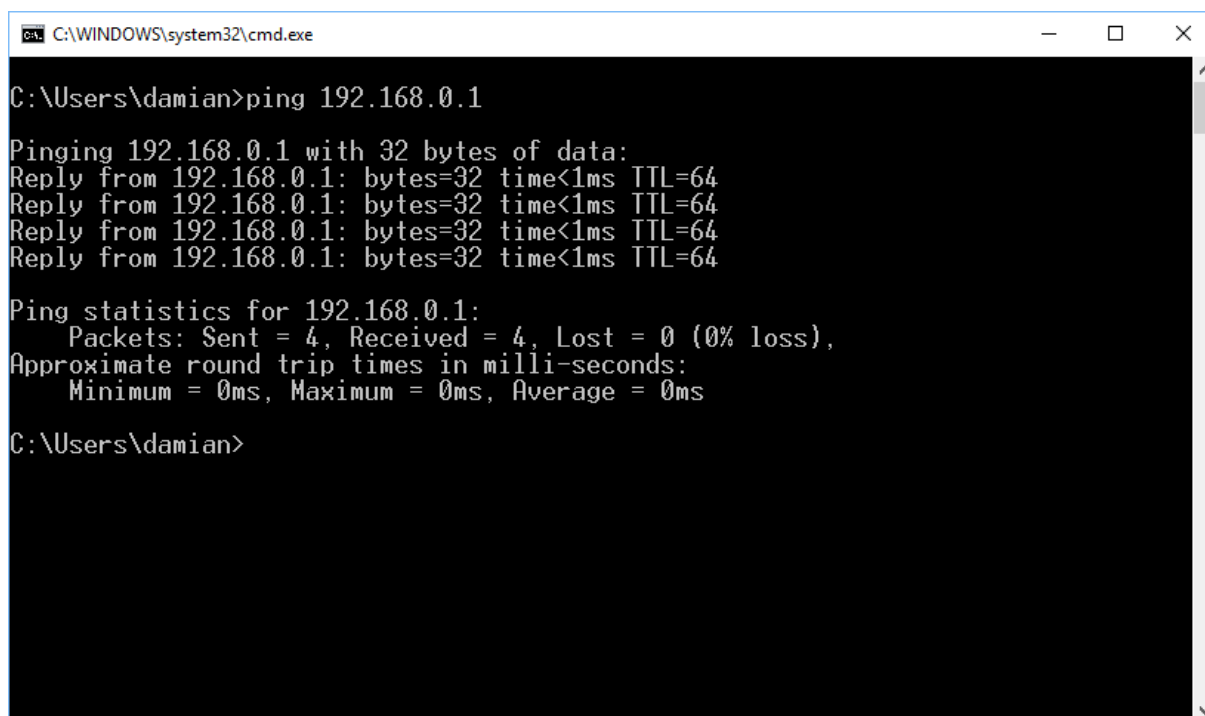
C:\Users\damian>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\damian>
```

- **połączeń z hostami w sieci lokalnej**, wówczas zamiast adresu pętli zwrotnej wpiszemy adres hosta w sieci lokalnej (np. **192.168.0.1**):



```
C:\WINDOWS\system32\cmd.exe

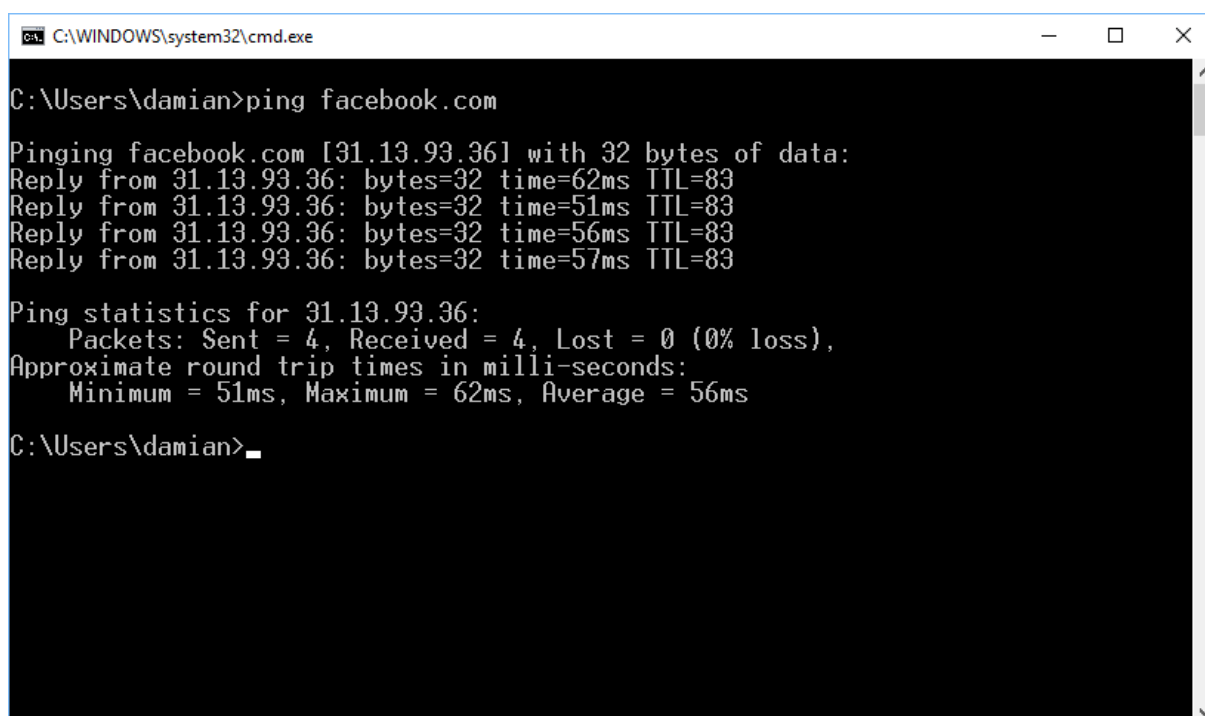
C:\Users\damian>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\damian>
```

- **połączeń z hostami znajdującymi się w odległych sieciach**. Można tutaj zamiast **adresu IP**, wprowadzić nazwę domenową, czyli np. **facebook.com**, jeśli chcemy sprawdzić komunikację z serwerem, na którym przechowywana jest dana strona WWW:



```
C:\WINDOWS\system32\cmd.exe

C:\Users\damian>ping facebook.com

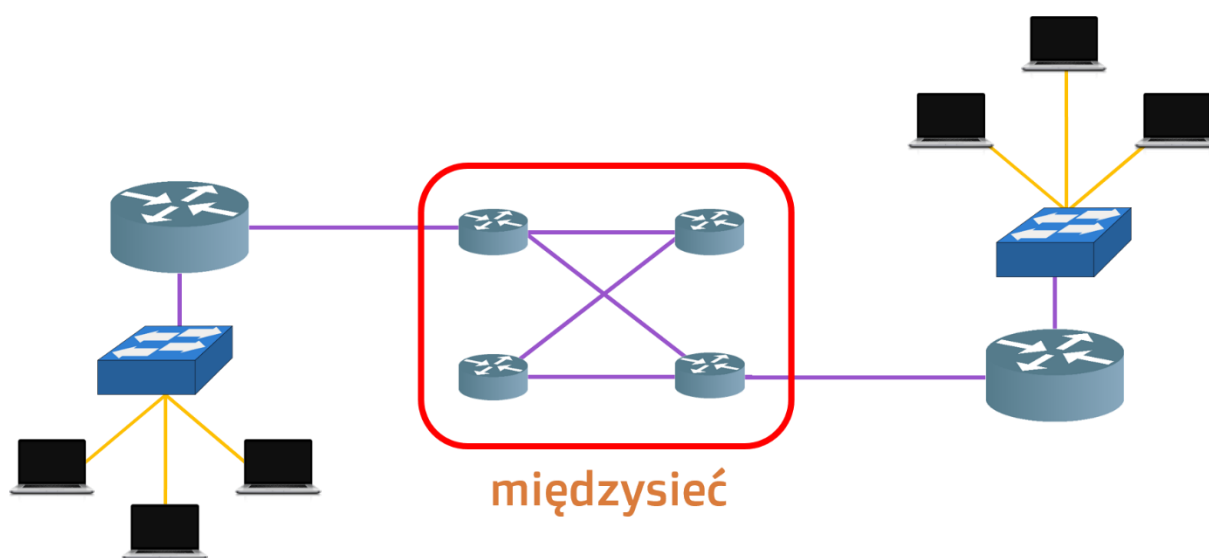
Pinging facebook.com [31.13.93.36] with 32 bytes of data:
Reply from 31.13.93.36: bytes=32 time=62ms TTL=83
Reply from 31.13.93.36: bytes=32 time=51ms TTL=83
Reply from 31.13.93.36: bytes=32 time=56ms TTL=83
Reply from 31.13.93.36: bytes=32 time=57ms TTL=83

Ping statistics for 31.13.93.36:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 51ms, Maximum = 62ms, Average = 56ms

C:\Users\damian>
```

Czasami, może się zdarzyć, że pomimo działania odległej sieci i poprawnej komunikacji, **nie uzyskamy odpowiedzi** na żądanie echo wysłane przez program **PING**. Wynika to z faktu, że niektórzy administratorzy sieci ograniczają, lub całkowicie uniemożliwiają wprowadzanie **datagramów ICMP** do swoich sieci, ze względów bezpieczeństwa.

Kolejnym elementem związanym z **testowaniem warstwy sieciowej** jest sprawdzenie **trasy przesyłania pakietów od hosta źródłowego do docelowego**. W sieć rozległą mogą pracować tysiące ruterów, które tworzą tak zwaną **międzysieć**, czyli połączenia pomiędzy **sieciami lokalnymi** rozszanymi po całym świecie.



Do tego, aby sprawdzić, przez **jakie routery przesyłany jest pakiet**, od **komputera** do np. **serwera WWW** wykorzystamy program **TRACERT**, dla systemów Windows, lub **TRACEROUTE** dla systemów Linux. Oba działają dokładnie tak samo i podobnie jak **PING** wykorzystują **protokół ICMP**, i **wiadomości echo**. Aby wykonać test wystarczy **wpisać polecenie TRACERT** w konsoli wraz z **adresem hosta docelowego**. Może to być **adres ip**, jeśli chcemy przetestować trasę do **konkretnego hosta**, może to być również **adres domenowy**, czyli np. **wp.pl**.

Poniżej widać **test trasy** do serwera na którym przechowywana jest **strona wirtualnej polski**.

```

C:\WINDOWS\system32\cmd.exe

C:\Users\damian>tracert wp.pl

Tracing route to wp.pl [212.77.98.9]
over a maximum of 30 hops:

  1  <1 ms    <1 ms    <1 ms    192.168.0.1
  2  34 ms     39 ms     27 ms     172.18.22.226
  3  46 ms     36 ms     39 ms     172.18.32.67
  4  32 ms     54 ms     27 ms     172.18.32.41
  5  43 ms     37 ms     36 ms     172.18.32.2
  6  39 ms     39 ms     27 ms     80.50.105.17
  7  54 ms     54 ms     103 ms    gda-r1.tpnet.pl [194.204.175.90]
  8  51 ms     39 ms     39 ms     z-wp-gda-ar1.tpnet.pl [213.76.0.166]
  9  39 ms     45 ms     48 ms     rtr-int-1.rtr1.adm.wp-sa.pl [212.77.96.42]
 10  46 ms     37 ms     40 ms     www.wp.pl [212.77.98.9]

Trace complete.

C:\Users\damian>_

```