Packet Tracer

Podstawowy kurs

Blog Karol221-10

Informacje ogólne

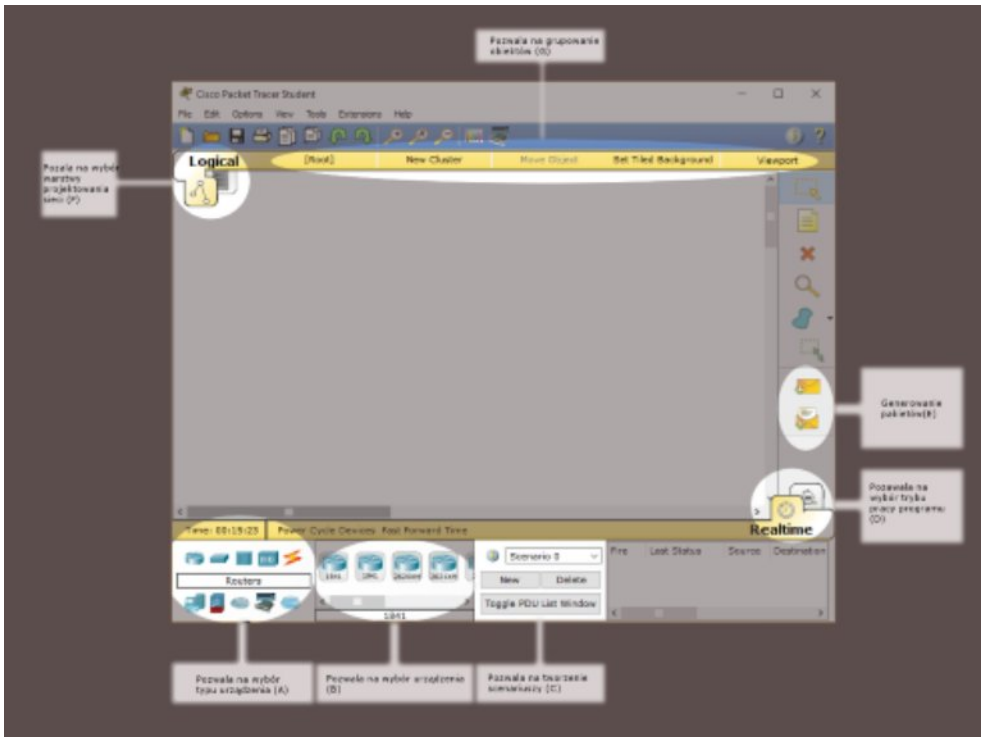
Aplikacja Cisco pozwala zająć się dość szerokim wachlarzem urządzeń sieciowych, takich jak routery (dostępne są: 1841, 1941, 2620XM, 2621XM, 2811, 2901, 2911, 819HGW) czy przełączniki (2950-24, 2950T-24, 2960-24TT). Mamy dostępną dużą ilość urządzeń klienckich (m.in. PC, laptopy, serwery, drukarki, telefony VoIP/analogowe, tablety, telefony itd.). Możemy symulować działanie Internetu za pomocą emulacji WAN. Można także tworzyć sieci bezprzewodowe (mamy dostępny jeden router WI-FI – WRT300N). Możliwe jest tworzenie własnych konfiguracji urządzeń. Do tego służą tzw. urządzenia „Generic" Generic AccessPoint-PT, Generic-Router-PT itd. Możemy projektować sieć zarówno logicznie, jak i fizycznie (ten drugi aspekt jest wg mnie trochę zubożony, ale ta opcja się czasem przydaje). Program posiada także wiele innych elementów, o których nie będę w tej chwili wspominał, ale być może napiszę o nich w którymś z moich przyszłych wpisów.

Packet Tracer tworzy w pełni wirtualne środowisko. Ma to swoje zalety, ale posiada także wady. Przede wszystkim, dzięki temu Packet Tracer nie obciąża zbyt bardzo naszego, czasem już trochę leciwego komputera. Możemy ustawić nawet kilkanaście routerów i kilkaset stacji roboczych, i nie odbije się to na wydajności całego systemu. Jednakże, jak podałem wcześniej, program był tworzony głównie pod studentów CCNA. Routery zawarte w Packet Tracerze nie udostępniają pełni swoich możliwości konfiguracyjnych. Aby to jakoś zobrazować, podam prosty przykład (nie bój się, że do końca nie rozumiesz o co w nim chodzi, to zagadnienie omówię dokładnie później) – każdy router Cisco może pracować jako serwer DHCP. Jak wiemy, przydziela on adresy IP na pewien okres czasu. Proces ten to dzierżawa adresu IP. Podczas konfiguracji podajemy polecenie lease 5 12 45. Co się dzieje? Błąd? Dlaczego? Po prostu Packet Tracer nie wspiera tego, jak i kilkunastu innych poleceń i trzeba się z tym pogodzić.

Ogólnie, w tej serii będę chciał omówić podstawowe zasady korzystania z tego programu i umiejętności, które są potrzebne do zdobycia dobrych ocen (6 ?) z takich przedmiotów jak Sieci komputerowe czy Projektowanie i montaż lokalnych sieci komputerowych. Nie będą to zagadnienia specjalnie skomplikowane, ale ich zrozumienie często stanowi problem na lekcjach, co skutkuje potem słabymi ocenami z zaliczeń. Dodatkowo, dość ciężko znaleźć jakieś informacje/tutoriale w języku polskim, które pozwoliłyby łatwiej

zrozumieć zasadę działania sieci i nie wpajałyby błędnych zasad i założeń. No to do dzieła!

Podstawowe elementy interfejsu programu Packet Tracer 6.2

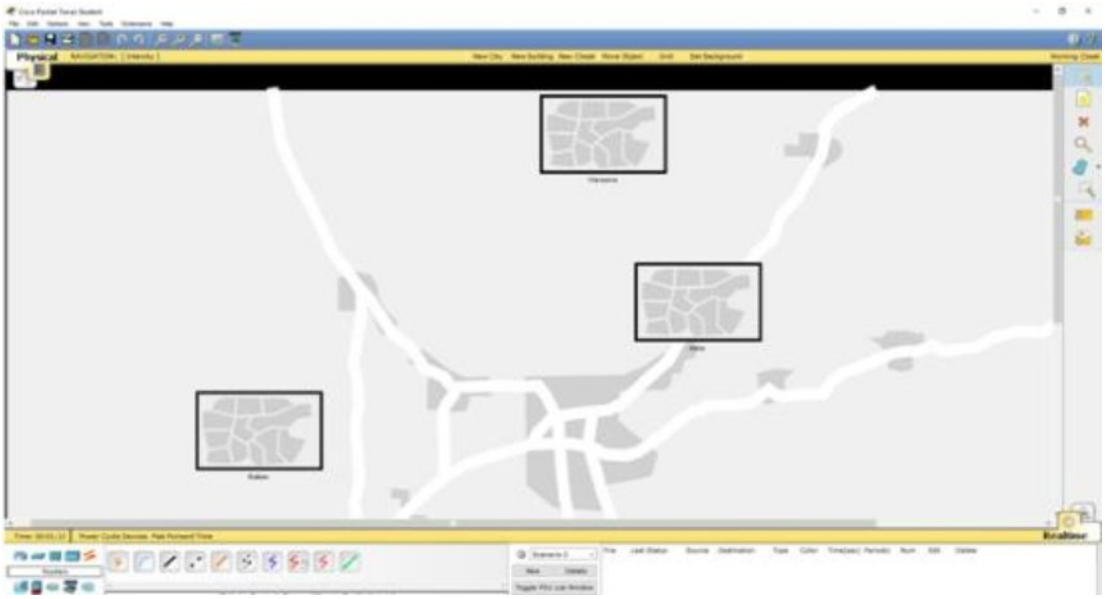


Na schemacie powyżej zaznaczyłem najważniejsze (wg mnie ?) elementy interfejsu programu. Poznajmy teraz dokładniej poszczególne grupy opcji. Dla uproszczenia do oznaczenia grup opcji będę posługiwał się oznaczeniami z rysunku:

- A – w tym miejscu możemy wybrać rodzaj urządzenia. Np.: jeśli będziemy chcieli wybrać router Cisco 1841, to z panelu A wybieramy ikonę routera, a z panelu B właściwy router. Urządzenia możemy wybierać spośród następujących kategorii: (wymieniam od lewej strony pierwszego rzędu) routery, przełączniki, huby, urządzenia bezprzewodowe (czyli po prostu punkty dostępowe ?), przewody do łączenia urządzeń, urządzenia końcowe, firewalle, emulacja WAN, urządzenia „własnej” specyfikacji oraz „Multiuser Connection” (o tej opcji powiem kiedy indziej).
- B – jak wcześniej wspominałem, tu wybieramy konkretne urządzenie. Jak widać, nie ma tu żadnej filozofii (…)
- C – tu możemy tworzyć tzw. scenariusze. Pokrótce, służą one do sprawdzania zaprojektowanej sieci pod różnymi względami. Możemy np.: w

jednym scenariuszu sprawdzać połączenie między jedną grupą komputerów, a w drugim scenariuszu między inną. Testy najłatwiej przeprowadzać za pomocą narzędzi z grupy E, a wyniki danego scenariusza przedstawiane są w tabelce po prawej stronie.

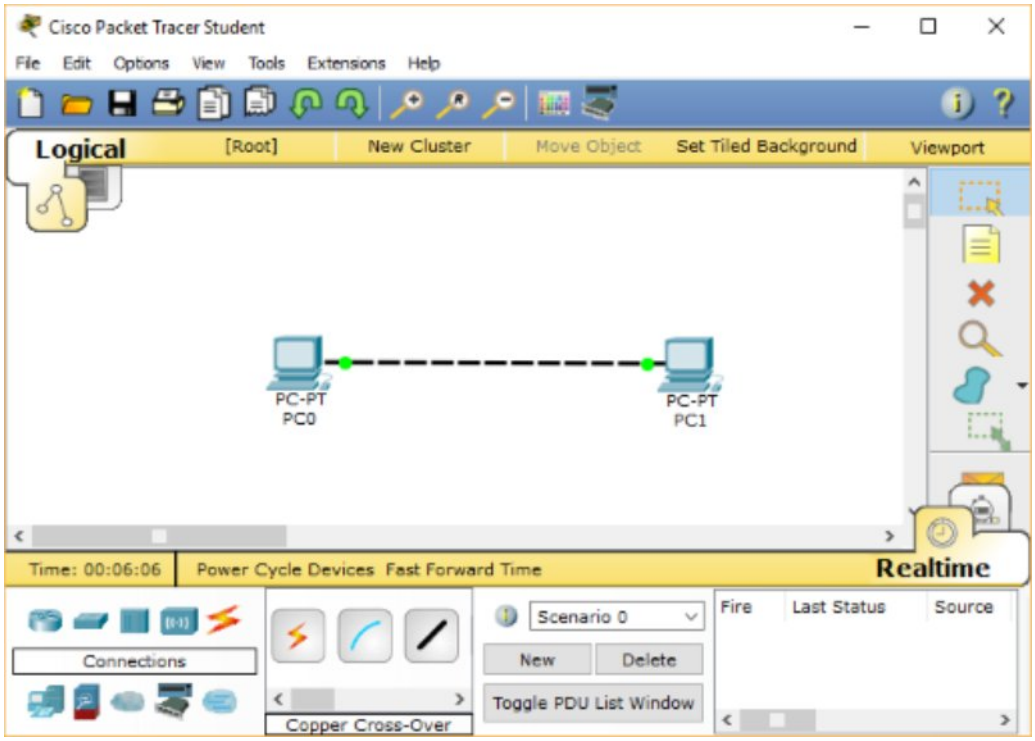
- D – możemy tu przełączać się między trybem symulacji. Do wyboru są dwa tryby: czasu rzeczywistego oraz symulacji. W pierwszym trybie sieć działa na bieżąco. Np.: jeśli wydamy polecenie ping na którymś z komputerów, jest ono wykonywane tak, jakby działo się to w normalnej sieci. Natomiast w trybie symulacji możemy obserwować pakiet po pakiecie pracę sieci. Możemy obserwować także trasę pakietu.
- E – te narzędzia służą do testowania naszych nowo zaprojektowanych, lśniących od nowości sieci. „Zamknięta" koperta pozwala nam na sprawdzenie, czy istnieje poprawne połączenie między dwoma elementami sieci. Jest to odpowiednik polecenia ping. Aby użyć tego narzędzia, klikamy na tę kopertę, a potem na urządzenie źródłowe i docelowe. Wyniki pojawiają się w podsumowaniu danego scenariusza (jeśli nie wiesz o co chodzi, odsyłam do C).
  „Otwarta" koperta posiada trochę więcej możliwości. Zasada działania jest taka sama, jak „zamkniętej" koperty, ale po wybraniu punktu źródłowego możemy podać szczegółowe informacje na temat rodzaju przesyłanego pakietu (m.in. protokół, rozmiar pakietu, numer następny, interwał, jeśli wysyłamy pakiety seryjnie)
- F – ten przełącznik pozwala przełączać się między projektowaniem „fizycznym" a „logicznym". Domyślnie jesteśmy w tym drugim trybie. Kiedy przejdziemy do projektowania fizycznego, możemy dodatkowo rozstawiać sprzęt sieciowy w różnych pomieszczeniach, np.: biura. Możemy nawet projektować połączenia międzymiastowe!. Aczkolwiek, to jest bardziej ciekawostka niż użyteczny moduł programu.



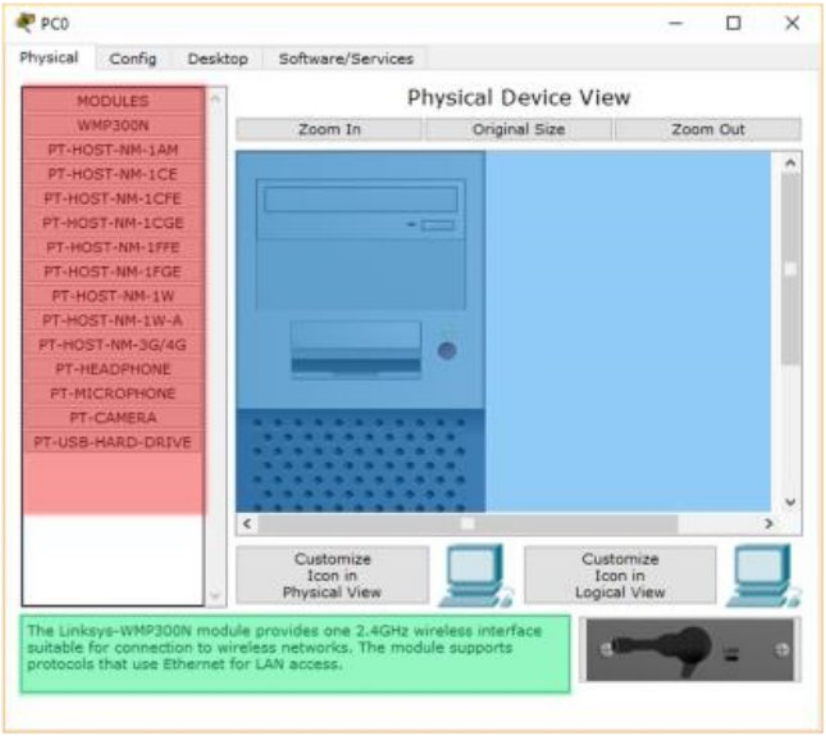
- G – ten panel w logicznym projektowaniu sieci pozwala nam „grupować” urządzenia. Co to znaczy? Otóż, jeśli brakuje nam przestrzeni roboczej na całą naszą sieć, wtedy zaznaczamy urządzenia, które mają należeć do grupy i klikamy „New Cluster”. Wszystkie te urządzenia są wtedy widoczne pod jedną nazwą. Jeśli ustawimy teraz na przykład drugi router i będziemy chcieli go połączyć z którymś urządzeniem z „klastra” to po prostu łączymy go z grupą. Wyświetli nam się wtedy lista wolnych gniazd. Wybieramy jedno z nich i voila.

Pierwsza prosta sieć – dodawanie urządzeń i wymiana modułów

Powoli zbliżamy się do końca niniejszego wpisu. Stworzymy prostą sieć, złożoną z dwóch komputerów (wiem, to banał ale każdy kurs należy zacząć od podstaw ?). Więc zaczynamy:
Aby dodać komputer, klikamy kategorię End Devices w obszarze A. Potem przeciągamy komputer (PC-PT) na obszar roboczy znajdujący się pośrodku. Tak samo dodajemy drugi komputer. Jak sobie poradziłeś, możemy je połączyć kablem. Urządzenia tego samego typu (komputer-komputer, switch-switch, router-router) łączy się tzw. kablem krosowym (więcej o tym w następnej części). Przechodzimy więc do kategorii Connections i wybieramy Copper Cross-Over. Najpierw klikamy pierwszy komputer, wybieramy FastEthernet0, potem klikamy na drugi komputer i też wybieramy gniazdo FastEthernet0. Tak powinna wyglądać w tym momencie nasza sieć:

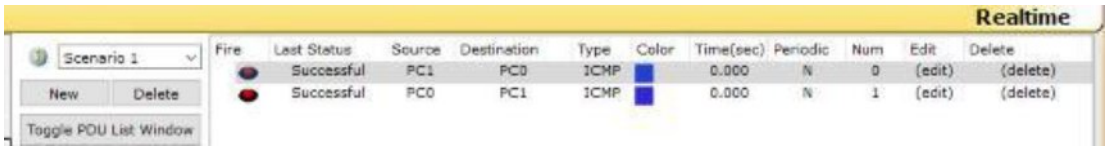


Połączenie powinno zaświecić się na zielono, ale to jeszcze nie znaczy, że odnieśliśmy sukces!. Należy jeszcze skonfigurować adresy IP.
Klikamy dwukrotnie na komputer PC0. Powinno otworzyć się okno konfiguracji, wyglądające tak:



Pozwoliłem sobie dodać trochę kolorków, gdyż czas na kolejną porcję teorii ?
Obszar zaznaczony kolorem czerwonym – mamy tutaj listę modułów, które

możemy dołączyć do danego urządzenia. Po wybraniu modułu, jego opis pojawia się w miejscu, gdzie widzisz zielony kolor. Okienko zaznaczone na niebiesko przedstawia nasz bajerancki komputer?.
Zakładka Config pozwala nam m.in. na konfigurację adresów IP czy bramy domyślnej. Możemy także zmienić np.: MAC adres karty sieciowej obecnej w danym komputerze.
Zakładka Desktop to istny system operacyjny, powalający możliwościami wszystko, co obecnie istnieje ?. Mamy tam dostępne różne „aplikacje" które pozwalają wykonywać różne operacje. Możemy między innymi przeprowadzić konfigurację IP, VPN, przeglądać wirtualny Internet za pomocą przeglądarki internetowej, pobawić się w wirtualnym cmd (konsoli znakowej) czy wysyłać/odbierać e-maile.
Wykonajmy więc konfigurację sieci na obydwóch komputerach. Przejdźmy na zakładkę Desktop, a potem uruchommy IP Configuration. Wybieramy statyczną konfigurację sieci (Static). Jako adres IP podaj 192.168.1.1. Kliknij pole Subnet Mask, które powinno się teraz uzupełnić samo. Pozostałych parametrów na razie nie ruszamy. Ich znaczenie opiszę następnym razem. Tak samo robimy na PC1, z tym, że tam ustalamy adres 192.168.1.2. Po zakończonej konfiguracji przydałoby się sprawdzić, czy nasza nowa „sieć" działa poprawnie. W tym celu kliknij ikonę „zamkniętej koperty" z menu po prawej stronie. Potem kliknij na PC0, a następnie na PC1. Powinien zostać wysłany pakiet ICMP. (czyli po prostu ping). W menu scenariuszy pojawi się wynik naszych działań. Jeśli w polu Status widzimy „Successful" to znaczy, że wszystko się udało i możesz być z siebie dumny. Jeśli nie, to znaczy, że popełniłeś gdzieś błąd. Dla 100% pewności sprawdźmy też ruch w drugą stronę. Wybieramy ikonę „zamkniętej koperty", potem klikamy na PC1 a następnie na PC0. W tym wypadku również wynikiem powinno być słynne Successful.



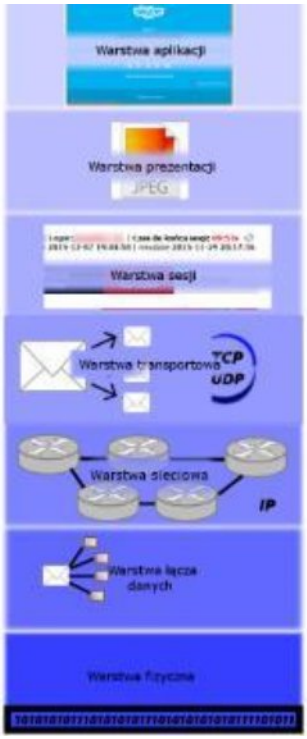
To by było na tyle jeśli chodzi o interfejs Packet Tracera. Zapraszam na kolejne części, w których będzie między innymi o:

- Adresowaniu sieci, adresach MAC, IPv4, IPv6
- Protokołach routingu statycznego, RIP, OSPF
- NAT

- Access-Listy
- VLAN-y
- Protokół PPP
- Emulacja WAN
- (w ograniczonym stopniu) VPN

Model OSI – podstawa przepływu danych w sieci

Wyobraź sobie następującą sytuację: Rozmawiasz ze swoją ukochaną dziewczyną przez Skype. Wszystko chodzi ładnie, płynnie, pięknie itd. Zastanawiałeś się kiedyś, jak to się dzieje? Jak dane z kamerki internetowej twojego komputera są przesyłane do innego komputera? W komunikacji internetowej najpopularniejszy jest model OSI. Składa się on z 7 warstw. Będę je omawiał na przykładzie programu Skype.



Najwyższą z warstw jest warstwa aplikacji. Jest to po prostu program, którego używamy na komputerze. Może to być Skype czy inny komunikator internetowy, arkusz kalkulacyjny, czy procesor tekstu. Gdy podejmiemy jakąś decyzję, która wymaga przesłania danych do innej stacji roboczej, warstwa aplikacji rozpoczyna cały proces i przesyła dane do warstw niższych. W naszym przykładzie, proces komunikacji może się rozpocząć np.: gdy będziemy chcieli rozpocząć z kimś wideorozmowę.

Kolejną warstwą jest warstwa prezentacji. Jej zadanie jest proste. Pilnuje, aby dane wysyłane przez warstwę aplikacji na pierwszej stacji roboczej były możliwe do odczytania na drugim komputerze. Koduje i dekoduje informacje, zapisuje je w różnych formatach. Ważnym zadaniem tej warstwy jest także szyfrowanie danych. Formaty danych używane w tej warstwie to najpopularniejsze formaty graficzne, takie jak JPG czy BMP. Natomiast jeśli chodzi o dźwięk czy obraz, to są to MIDI i MPEG. Wracając do naszego „skejpowego" przykładu, warstwa prezentacji odpowiedzialna byłaby za zapisanie surowego obrazu z kamerki internetowej w odpowiednim formacie wideo, a także za zaszyfrowanie tych danych.

Warstwa sesji odpowiedzialna jest za utrzymanie połączenia między dwoma komputerami. Synchronizuje także „rozmowę" między warstwami prezentacji dwóch hostów. Zabezpiecza komunikację przed błędami, o których ewentualnym wystąpieniu informuje wyższe warstwy. W naszym przykładzie warstwa sesji odpowiedzialna byłaby za synchronizację głosu/dźwięku między rozmówcami, oraz za rozpoczęcie/zakończenie rozmowy.

Dane przechodzące przez warstwy wyższe (aplikacji->prezentacji->sesji) tworzą pakiet danych. Realizacja funkcji tych warstw zasadniczo należy do aplikacji i zależy od jej programistów. Warstwy niższe są już realizowane głównie przez system operacyjny. Najwyższa z „niższych" warstw to warstwa transportowa. Dzieli ona otrzymany od wyższych warstw pakiet danych na segmenty. Do każdego segmentu dodawany jest nagłówek określający jego właściwości. Warstwa transportowa dba o niezawodność „transportu" danych między hostami. Dba wykrywanie i ewentualne naprawianie błędów komunikacji. Protokołami, których używa warstwa transportowa są TCP i UDP. Gdy dane zostaną już podzielone na segmenty, są one przekazywane do niższej, warstwy sieci.

Zadanie warstwy sieci jest proste – znajduje najkrótszą drogę między komputerem źródłowym i docelowym, jaką może być wysłany segment. Dla każdego segmentu wyznaczona trasa może być inna. Adresuje logicznie segmenty (najczęściej posługując się adresami IP). Warstwa sieci także dokłada swój nagłówek do każdego segmentu. Typowym i najczęściej używanym protokołem tej warstwy jest protokół IP. Tak opakowany segment jest wysyłany do warstwy łącza danych.

Warstwa łącza danych dane otrzymane od warstwy sieci dzieli na jeszcze mniejsze części – ramki. Troszczy się o najbardziej podstawowe elementy pracy sieci, takie jak adresowanie fizyczne (po adresach MAC), topologie sieci, kontrolę przepływu danych i wysyłanie informacji do warstw wyższych o błędach występujących na tym poziomie.

Najniższą warstwą, która jest odpowiedzialna za rzeczywiste wysłanie/odebranie danych jest warstwa fizyczna. Zamienia ona ramki otrzymane od warstwy wyższej na fizyczne strumienie bitów, które są wysyłane przewodem elektrycznym, czy sygnałem radiowym do innego hosta. Ramki są przesyłane szeregowo (bit po bicie). To jest jedyne zadanie tej warstwy.

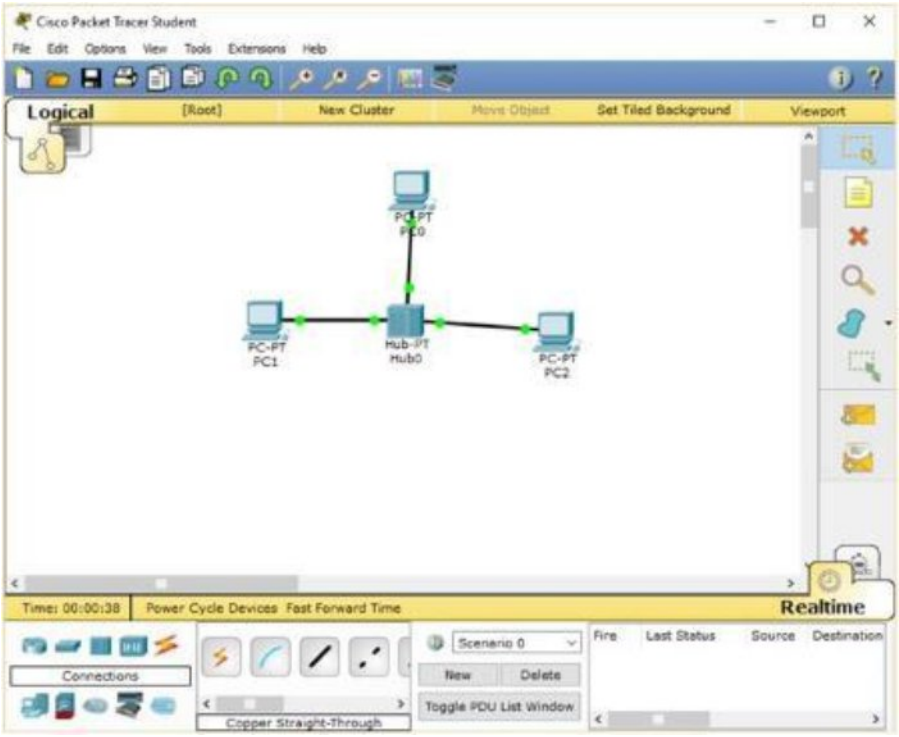
Opisałem, jak wygląda ten proces w wypadku wysyłania danych. W przypadku ich odbierania wszystko się dzieje po prostu w odwrotnej kolejności. Pierwszą warstwą jest wtedy warstwa łącza danych, która wysyła odebrane strumienie bitów do warstw wyższych, aż w końcu warstwa aplikacji wyświetla je użytkownikowi.

Podstawowe urządzenia sieciowe budujące sieć lokalną

Mamy ogólnie trzy typy urządzeń łączących elementy sieci komputerowej, z czego dwa z nich mają dzisiaj właściwie znaczenie historyczne. Są to:

- Koncentrator (hub)
- Most (bridge)
- Przełącznik (switch)

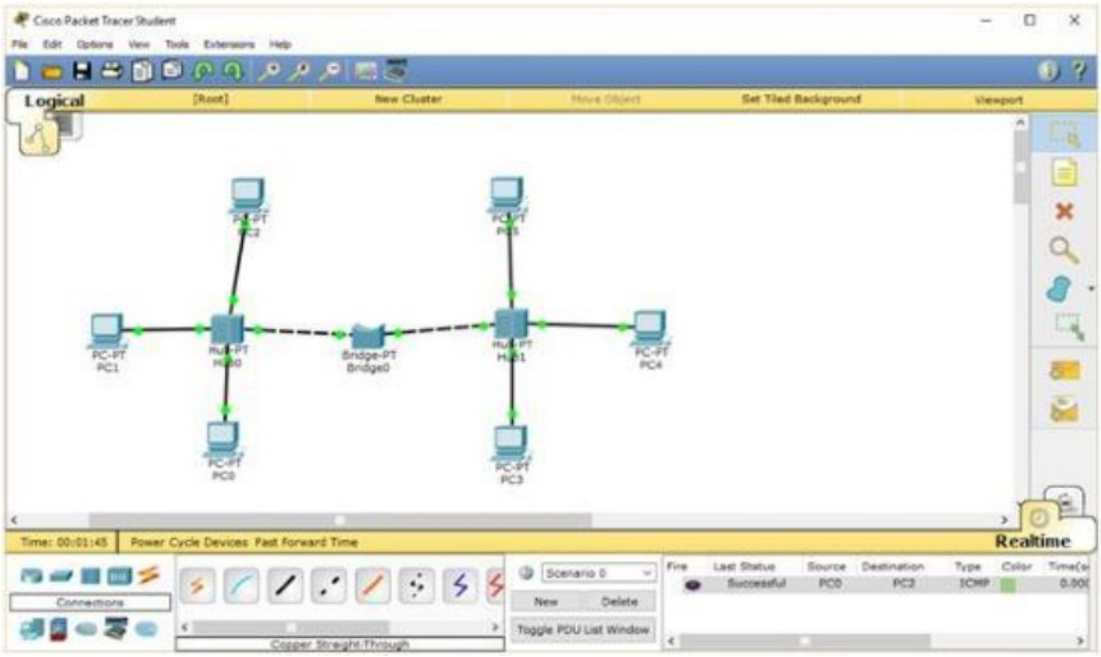
Zacznijmy od najprostszego zagadnienia. Mamy trzy komputery i chcemy, aby istniało połączenie i możliwość przesyłania danych między nimi. Najłatwiejszym w budowie urządzeniem spełniającym takie zadanie jest Hub. Pracuje on w warstwie pierwszej modelu OSI. Oznacza to, że ramkę, którą otrzyma od jednego z podłączonych urządzeń wysyła do wszystkich innych. Nie ma znaczenia fakt, do kogo ta ramka jest adresowana. Hub nie analizuje ich budowy. Niesie to ze sobą kilka wad. Przede wszystkim, sieć jest „zasypywana" zbędnym ruchem, którego wcale nie musi być. Poza tym, stwarza to idealne pole dla działań hackerów. Aby podsłuchać transmisję, wystarczy po prostu wpiąć się do jednego z portów huba i ustawić kartę sieciową w tryb nasłuchu. Ze względu na sposób pracy tego urządzenia będziemy widzieli wszystkie pakiety, jakie krążą w sieci. Zbudujmy teraz prostą sieć w programie Packet Tracer, aby zobaczyć, jak to wygląda na żywo. Powinna ona wyglądać mniej więcej tak:



Komputerom nadaj adresy IP z zakresu 192.168.1.1 (dla PC0) do 192.168.1.3 (dla PC2). Następnie przejdź w tryb symulacji. Wyślij pakiet PING z komputera PC0 na komputer PC2 (pamiętasz? Wystarczy wybrać ikonę „zamkniętej koperty" i kliknąć najpierw na źródłowym, a potem na docelowym komputerze). Potem kliknij przycisk Auto Capture/Play i spokojnie obserwuj, jak sobie te pakiety krążą. Po wstępnej wymianie pakietów ARP (co to jest, dowiesz się kiedy indziej) zobaczysz upragniony pakiet ICMP. Gdy koncentrator (czyli hub) otrzyma pakiet ICMP, wyśle go nie tylko do komputera PC2, ale także do PC1. Sprawdza się więc to, o czym mówiliśmy chwilę wcześniej ? i potwierdza, jak przydatnym narzędziem jest ten program.

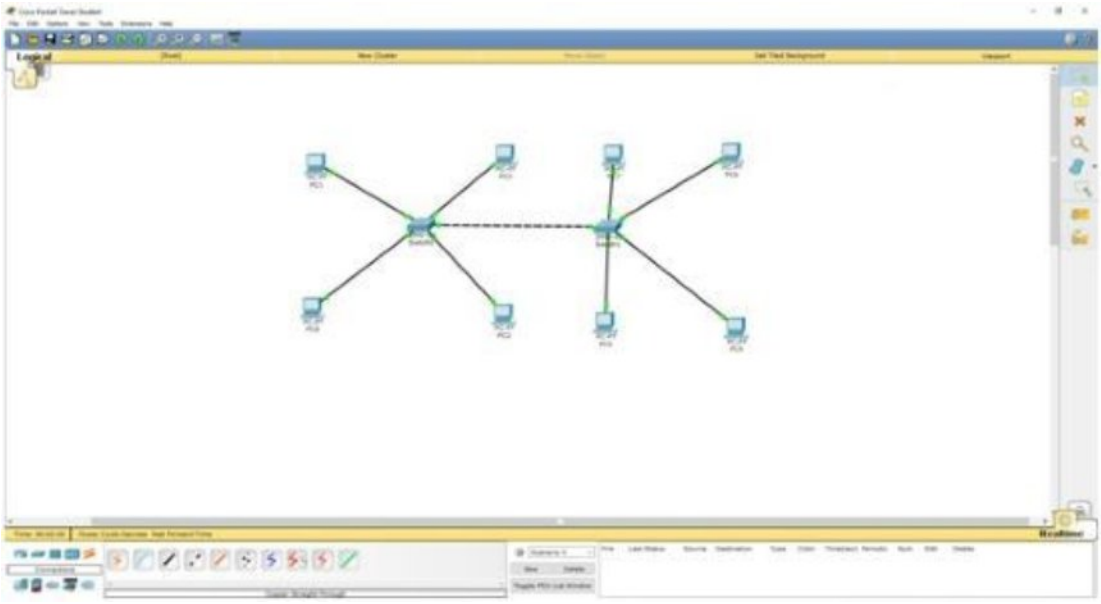
Gdyby cała sieć internetowa była oparta na koncentratorach, z pewnością nie byłaby to szybka ani bezpieczna sieć. Musiano więc opracować jakieś inne urządzenie, które służyłoby do dzielenia sieci na mniejsze segmenty, tak, aby ograniczyć wady koncentratorów. Tym urządzeniem jest most (bridge).

Most sieciowy pracuje w drugiej warstwie modelu OSI (a więc w warstwie łącza danych). Pozwala mu to (w przeciwieństwie do koncentratora) analizować ramki przepływające przez niego m.in. pod kątem adresu fizycznego hosta docelowego. Zasada działania jest prosta – jeśli pakiet jest adresowany do któregoś z komputerów znajdujących się w innym segmencie sieci, należy go puścić dalej. W przeciwnym wypadku należy go odrzucić. Zmniejsza to istotnie obciążenie sieci. Zmniejsza także liczbę kolizji pakietów w sieci (kolizja? – pewnie dla ciebie nowe pojęcie. Nie martw się, zostanie wyjaśnione za chwilę), a także zwiększa liczbę domen kolizyjnych.
Zobaczmy więc w praktyce, jak działa most sieciowy. Stwórz następującą sieć:



Komputerom nadaj adresy od 192.168.1.1 (dla PC0) do 192.168.1.6 (dla PC5). Przejdź teraz w tryb symulacji i wyślij pakiet PING z komputera PC0 do PC2. Kliknij Auto Capture/Play i obserwuj ruch pakietu ICMP. Jak zauważysz, pakiet zostanie wysłany przez hub do mostu sieciowego, ale ten nie prześle go dalej, ponieważ cel tego pakietu nie znajduje się w drugim segmencie sieci tylko w tym, z którego pakiet pochodzi.
Teraz spróbujmy wysłać pakiet PING z komputera PC0 na jakiś dowolny komputer w drugim segmencie sieci. Obserwuj ruch pakietu. Tym razem most sieciowy przepuści pakiet dalej, a więc wszystko się dzieje tak, jak być powinno.

Został nam do poznania jedynie przełącznik – ostatnie z urządzeń, które służy do budowania sieci LAN. Przełącznik, czyli switch jest urządzeniem warstwy drugiej – tak samo jak most. Różnica polega na tym, że most ma zwykle dwa porty, które służą do łączenia dwóch segmentów sieci. Natomiast przełącznik może mieć tych portów kilkanaście. Przełącznik w odróżnieniu od mostu, jest sprzętowo przystosowany do analizy ramek, tak więc wykonuje te zadanie o wiele szybciej. Każdy port przełącznika jest jak gdyby oddzielnym mostem, więc przy użyciu tego urządzenia ilość domen kolizyjnych jest maksymalna (w zasadzie każdy komputer znajduje się w innej domenie kolizyjnej) a niepotrzebne obciążenie sieci jest minimalne.
Istnieją przełączniki, które pracują nie tylko w warstwie drugiej, ale także w warstwach wyższych, ale nimi zajmiemy się później.
Aby zobaczyć na żywo zasadę działania przełącznika, zbudujmy prostą sieć:

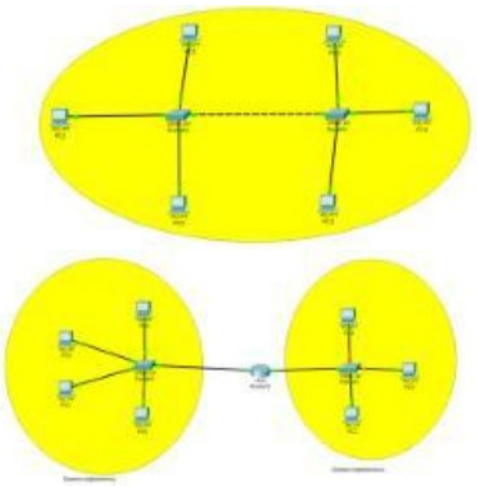


Komputerom nadaj adresy IP od 192.168.1.1 (dla PC0) do 192.168.1.8) (dla PC7). Standardowo przejdź w tryb symulacji. Wyślij pakiet PING z komputera PC0 do komputera PC2. Obserwuj ruch pakietu ICMP. Jak widzisz, nie jest on rozsyłany do wszystkich komputerów. Trafia jedynie do komputera docelowego. Tak samo będzie się działo, jak wyślesz pakiet ICMP z komputera PC0 na komputer np.: PC5. Switch1, po otrzymaniu pakietu od razu wyśle go do komputera PC5, a inne komputery nie zobaczą tej wiadomości. Te wszystkie zalety przesądziły o tym, że dzisiaj właśnie przełącznik jest najpopularniejszym urządzeniem służącym do łączenia komputerów.
Jak pewnie zauważysz, nie wspomniałem ani razu o urządzeniu zwanym routerem. Będzie o tym w następnej części mojego kursu.

Domeny kolizyjne i rozgłoszeniowe

Pojęcia domeny kolizyjnej użyłem już raz przy opisie zasad pracy huba i mostu sieciowego. Już wtedy pewnie zastanawiałeś się, co to jest? No to teraz się dowiesz :). Ogólnie przy budowie sieci wyróżniamy dwa typy domen. Zacznijmy od domeny kolizyjnej. Aby lepiej zobrazować zasadę jej działania, posłużę się przykładem:
Wyobraź sobie, że jedziesz samochodem po drodze. Póki nie ma żadnych innych samochodów, możesz bez problemu skręcić w którąkolwiek uliczkę i nie musisz czekać. Problem pojawia się, gdy wokół ciebie jeżdżą także inne pojazdy. Wtedy na skrzyżowaniu może powstać ogromny korek, w którym możesz spędzić dobrych kilkanaście minut, zanim pojedziesz dalej. Tak samo jest w sieciach. Kolizja powstaje, gdy w tym samym czasie, w tej samej sieci dwa różne urządzenia próbują wysłać wiadomość. Jak wiadomo, jest to niemożliwe, gdyż bit, to najprościej mówiąc, poziom sygnału (gdzie 1 oznacza, że jest prąd, a 0, że go nie ma). Nie jest więc możliwe wykonanie tej czynności. Jedno z urządzeń musi poczekać, aż „medium” (czyli kabel) się zwolni. Problem domen kolizyjnych ujawnia się zwłaszcza wtedy, gdy używamy koncentratorów. Jak widziałeś w pierwszym przykładzie, koncentrator po otrzymaniu pakietu wysyła go na wszystkie swoje porty. A więc w tym czasie żaden z komputerów nie może wysłać swojej wiadomości. Możemy więc wywnioskować, że wszystkie komputery podłączone do koncentratora znajdują się w jednej domenie kolizyjnej. Most sieciowy nie pozwala na przesyłanie pakietów do drugiego segmentu sieci, kiedy ten nie jest ich celem. A więc most dzieli jedną dużą domenę kolizyjną na dwie mniejsze. Przy przełączniku każdy komputer jest w „swojej” domenie kolizyjnej, a więc w tym wypadku kolizje praktycznie nie występują.

Warto w tym miejscu wspomnieć o takich pojęciach jak segment i segmentacja, których zresztą też już używałem w tym artykule. Segment sieci to innymi słowy jedna domena kolizyjna. Natomiast segmentacja polega na takim podziale sieci, aby zmniejszyć ilość kolizji pakietów, a zwiększyć liczbę domen kolizyjnych. Domena rozgłoszeniowa jest drugim rodzajem domen. Jest to grupa domen kolizyjnych. Aby zrozumieć ich działanie, wyobraź sobie następującą sytuację: Mieszkasz w pewnym domu (komputer). Chcesz dostarczyć wiadomość do jednego ze swoich przyjaciół, ale wiesz jedynie, że mieszka pod jednym z czterech adresów (inne stacje robocze). Przyjmijmy, że na drogach nie ma innych pojazdów (przełącznik jako urządzenie łączące stacje robocze). W najgorszym wypadku musisz odwiedzić wszystkie cztery domy, aby znaleźć ten właściwy. To właśnie istota domeny rozgłoszeniowej. Komputer, gdy chce wysłać pakiet na jakiś adres IP, do którego nie zna adresu fizycznego, musi wysłać pakiet ARP, który jest następnie rozgłaszany przez przełącznik. Wszystkie komputery podłączone do przełącznika do jedna domena rozgłoszeniowa. Duża liczba rozgłoszeń może istotnie obniżyć wydajność pracy sieci. Aby podzielić jedną domenę rozgłoszeniową na dwie mniejsze, należy użyć routera (gdyż pracuje on w warstwie 3 modelu OSI, dzięki czemu może przeprowadzić segmentację domen rozgłoszeniowych).



Router – cóż to takiego?

Prawie każdy z nas posiada w domu plastikową puszkę zwaną routerem. Ale czy wiemy, jak ona działa i za co odpowiada (poza lakonicznym „dzieleniem Internetu między wiele komputerów:)) Spróbujmy się nad tym zastanowić.

Router jest przede wszystkim urządzeniem międzysieciowym, czyli służy do łączenia dwóch lub większej ilości różnych sieci. Pracuje w warstwie trzeciej modelu OSI, co pozwala mu operować na logicznych adresach sieciowych. Router odpowiada za znalezienie najkrótszych ścieżek dla pakietów. Może łączyć sieci różnych standardów (Ethernet co prawda najpopularniejszym, ale istnieją także inne, takie jak TokenRing czy ATM).

Patrząc z perspektywy, mógłbyś powiedzieć, że routing (to, co robi router) i przełączanie (to, co robi przełącznik) niewiele się od siebie różnią, gdyż i to, i to zarządza ruchem danych w sieci. Podstawowe funkcje są podobne – fakt, ale istnieją znaczne różnice w zasadzie działania między routingiem a przełączaniem. Przede wszystkim obydwa procesy przy wykonywaniu swoich zadań korzystają z różnych danych – trasowanie opiera się głównie o adresy IP, natomiast switching o adresy MAC. Poza tym najprostsze przełączniki nie blokują rozgłoszeń (pracują w jednej domenie rozgłoszeniowej), co może doprowadzić do tzw. burzy rozgłoszeń.

Aby dogłębniej zrozumieć różnicę między routerem a przełącznikiem spróbujmy odnieść się do innego rodzaju sieci – sieci telefonicznych. Wyobraźmy sobie np.: firmę, która posiada lokalną centralę telefoniczną. Zawiera w pamięci wszystkie lokalne numery telefonów. Co jednak w przypadku, kiedy zapragniemy zadzwonić poza lokalną sieć? Wtedy lokalna centralka komunikuje się z inną, która jest wyżej w hierarchii (np.: do wojewódzkiej centralki, rozpoznającej numery kierunkowe) itd. Router można porównać właśnie do tej centralki wyższego rzędu, gdyż pełni podobną funkcję.

Powinniśmy w tym momencie domyślać się, do czego np.: w konfiguracji karty sieciowej służy brama domyślna. Otóż jest to po prostu najbliższe „wyjście na świat” – router pozwalający wyjść pakietom poza naszą sieć lokalną.

Protokoły routowalne a protokoły routingu

W tym dziale często można spotkać się z wyżej wymienionymi pojęciami. Można by pomyśleć, że oznaczają to samo – wszak – różni się tylko końcówka :). Jednak tak nie jest.
Protokoły routingu służą routerom między innymi do dzielenia się informacjami na temat sąsiadujących sieci. Przykładami są m.in. OSPF, RIP, IGRP, EIGRP,

BGP.
Protokoły routowalne to protokoły, które „przenoszą dane” poprzez sieć. Muszą zawierać wszelkie niezbędne informacje, które pozwolą routerowi na dostarczenie danych do miejsca docelowego. Przykładem może być sławny i znany protokół IP.

Protokoły routingu

Protokołów trasowania jest dużo, więc muszą być jakoś uporządkowane, jakoś podzielone, aby można było się w tym połapać. Zajmijmy się teraz tymi podziałami.

Najpierw zastanówmy się ogólnie nad podziałem routingu. Protokoły pozwalają na w miarę automatyczne poznawanie kolejnych sieci przez router. Jednakże, musi istnieć także możliwość samodzielnego ustalenia trasy z punktu A do punktu B. Tu możemy zauważyć już pierwszy podział, więc podsumujmy:

1. Routing statyczny – sami ustalamy trasę pakietów z punktu A do punktu B.
2. Routing dynamiczny – ustalaniem trasy zajmują się protokoły routingu

Zajmijmy się teraz rodzajami protokołów routingu dynamicznego. Przede wszystkim inne protokoły używane są w małych sieciach (np.: biurowych, szkolnych itd.) a inne ogólnie w Internecie. Tu wyłania nam się kolejna metoda podziału protokołów - ze względu na zależności między routerami:

1. Wewnętrzne protokoły trasowania – używane są do wymiany informacji w pojedynczym systemie autonomicznym, np.: w obrębie biur jednej firmy, w obrębie kilku budynków szkoły itd.
2. Zewnętrzne protokoły trasowania – używane są do wymiany informacji pomiędzy różnymi systemami autonomicznymi. (np.: pomiędzy twoim komputerem a portalem Dobreprogramy :)).

Następnym kryterium, wg których możemy podzielić protokoły routingu jest sposób ich działania. Wyróżniamy tu trzy kategorie:

1. Protokoły wektora odległości – podczas wyznaczania trasy sugerują się odległością między routerami (czyli „liczbą skoków”). Router operujący protokołem wektora odległości wysyła do wszystkich sąsiadujących routerów uaktualnienia (np.: co 30s)..
2. Protokoły stanu łącza – podczas wyznaczania trasy sugerują się obciążeniem łącza, jego prędkością itd. Router operujący protokołem stanu łącza przekazuje do sąsiadujących routerów informację, wtedy, gdy zmieni się topologia sieci. Uaktualnienia przesyłane są także okresowo (np.: co 30min).

3. Protokoły hybrydowe – łączą cechy obydwu wcześniejszych rodzajów.

Podstawowe protokoły routingu i ważne pojęcia

Znamy już podział protokołów routingu. Powinniśmy teraz zapoznać się z ogólną charakterystyką poszczególnych protokołów, ale przedtem wykształćmy się w zakresie ważnych pojęć, gdyż bez nich ciężko będzie zrozumieć niektóre zagadnienia:

- Metryka – określa tak jakby „jakość" trasy. W przypadku RIPv1 jest to po prostu liczba skoków. Bardziej zaawansowane protokoły (takie jak IGRP) biorą także pod uwagę obciążenie łącza, prędkość, niezawodność, opóźnienie i wiele innych czynników.
- Tablica routingu – jest to miejsce, w którym router przechowuje informacje o znanych sieciach. Innymi słowy, jeśli router nie będzie miał danej trasy w swojej tablicy routingu, to nie będzie mógł z niej skorzystać, chyba że zdefiniowana jest trasa domyślna.
- Trasa domyślna – routery musiałyby mieć potężne pamięci masowe, aby przechowywać informacje o trasie do każdego miejsca na świecie. Byłoby to po prostu nieopłacalne. Dlatego też wymyślono pojęcie trasy domyślnej. Jeśli nasz router nie może w swojej tablicy routingu znaleźć docelowej sieci, kieruje pakiet trasą domyślną. Wtedy ten drugi router musi się martwić, co zrobić z pakietem a nie nasz :)
- System autonomiczny – zacytuję za Wikipedią, gdyż nie ma tu wiele do wyjaśniania – „to zbiór adresów sieci IP pod wspólną administracyjną kontrolą, w którym utrzymywany jest spójny schemat routingu".

Najprostszym protokołem routingu jest RIP. Używa on liczby skoków do określenia odległości do każdego routera w sieci. Jeżeli istnieje kilka ścieżek do tego samego miejsca, wybierana jest ta, która wymaga najmniejszej ilości skoków. W tym momencie nie ma znaczenia obciążenie łącza, jego przepustowość itd. Liczy się tylko liczba skoków. Trasy oparte o protokół RIP mogą przebiegać przez maksymalnie 15 routerów. Istnieją dwie wersje protokołu RIP

- RIPv1 – stara wersja protokołu. Specyfikacja RIPv1 została opublikowana w 1988 r. w dokumencie RFC 1058. (To było aż 28 lat temu!). Nie wspiera adresowania bezklasowego i masek podsieci..
- RIPv2 – najczęściej wykorzystywana wersja protokołu RIP. Jej specyfikacja zawarta jest w dokumencie RFC1723 z roku 1994. RIPv2 wspiera tzw. routing z prefiksem, czyli adresowanie bezklasowe.

IGRP – jest to protokół hybrydowy. Przeznaczony jest do dużych sieci, gdyż posiada o wiele większy limit skoków niż RIP. W przeciwieństwie do RIP, o

wyborze najlepszej ścieżki decyduje nie tylko liczba skoków, ale także opóźnienie, szerokość pasma, niezawodność. Podobnie jak RIPv1 używa tylko routingu klasowego. Protokół własnościowy firmy Cisco.

EIGRP – zawiera wszystkie pozytywne cechy IGRP, natomiast nie posiada jego wad, gdyż pozwala na korzystanie z adresowania bezklasowego. Protokół własnościowy firmy Cisco.

OSPF – jest to protokół stanu łącza. Został opracowany w 1988 roku. Standardy protokołu OSPF zostały zapisane w dokumencie RFC2328. Stworzony w tym samym celu co protokoły IGRP i EIGRP, czyli do obsługi dużych sieci. Jednakże jest, w przeciwieństwie do wyżej wymienionych, otwartym protokołem.

Podstawy obsługi Cisco IOS

Jeśli wcześniej nie słyszałeś o produktach Cisco, nazwa IOS pewnie kojarzy ci się z systemem firmy Apple. Nic bardziej mylnego, gdyż te dwa systemy nie mają ze sobą nic wspólnego.

Aby wejść do systemu IOS w programie Packet Tracer, kliknij dwa razy na routerze a potem przejdź na zakładkę CLI.

Jeśli nie używałeś wcześniej zakładki Config, zostaniesz pewnie powitany pytaniem „Continue with configuration dialog? (yes/no)”. Jeśli odpowiesz: yes – system zada kilka pytań o podstawową konfigurację, czyli m.in. o nazwę hosta, adresy IP interfejsów, hasła itd. W naszym przypadku najczęściej będziemy odpowiadali no.

Router Cisco posiada dwa poziomy bezpieczeństwa:

- Tryb użytkownika (rozpoznamy go, jeśli po nazwie hosta występuje znak >, np.: Router>) – wykorzystywany jest do typowych zadań związanych ze sprawdzeniem statusu routera. W tym trybie nie jest dozwolona zmiana konfiguracji.
- Tryb uprzywilejowany – rozpoznamy go po znaku # (np.: Router#). W tym trybie dozwolona jest zmiana konfiguracji routera.

Aby uzyskać dostęp do trybu uprzywilejowanego, musimy wpisać polecenie enable. Jeśli ustawilibyśmy hasło, w tym momencie nastąpiłaby prośba o jego podanie.
Aby wejść do trybu konfiguracji routera, musimy podać polecenie config terminal. Istnieją również inne przełączniki dla tego polecenia (np.: config network) ale Packet Tracer nie wspiera pozostałych opcji.

Warto dodać, że polecenia możemy skracać. Np.: polecenie en zostanie rozpoznane jako enable. Jeśli nie znamy całego polecenia, możemy nacisnąć znak ? podczas wpisywania. Zostaną wtedy wyświetlone wszystkie możliwe

zakończenia tego, co wpisaliśmy.
Warto przyswoić sobie jeszcze jedno ważne polecenie, a mianowicie:

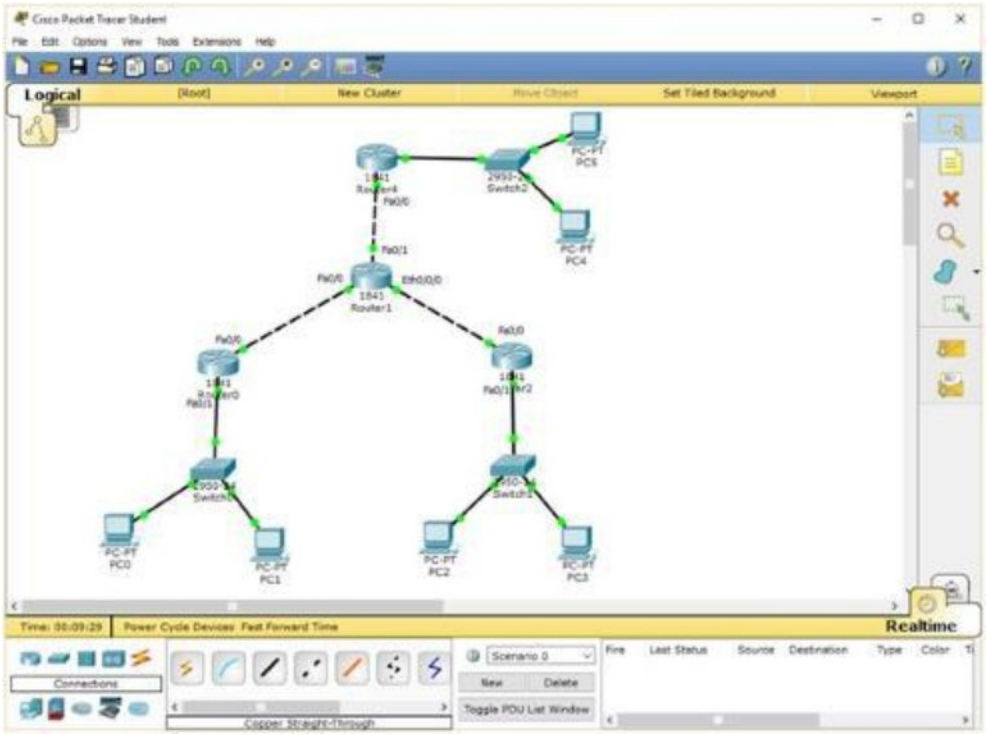
```
copy running config startup-config
```

Cóż ono robi? Otóż, wszystkie zmiany w konfiguracji routera, które wprowadzamy, zapisywane są w pamięci RAM. Stąd też są tracone, jeśli wyłączymy router (np.: w celu dodania jakiegoś modułu). Polecenie to zapisuje konfigurację routera do pamięci NVRAM, która jest nieulotna.

Routing statyczny

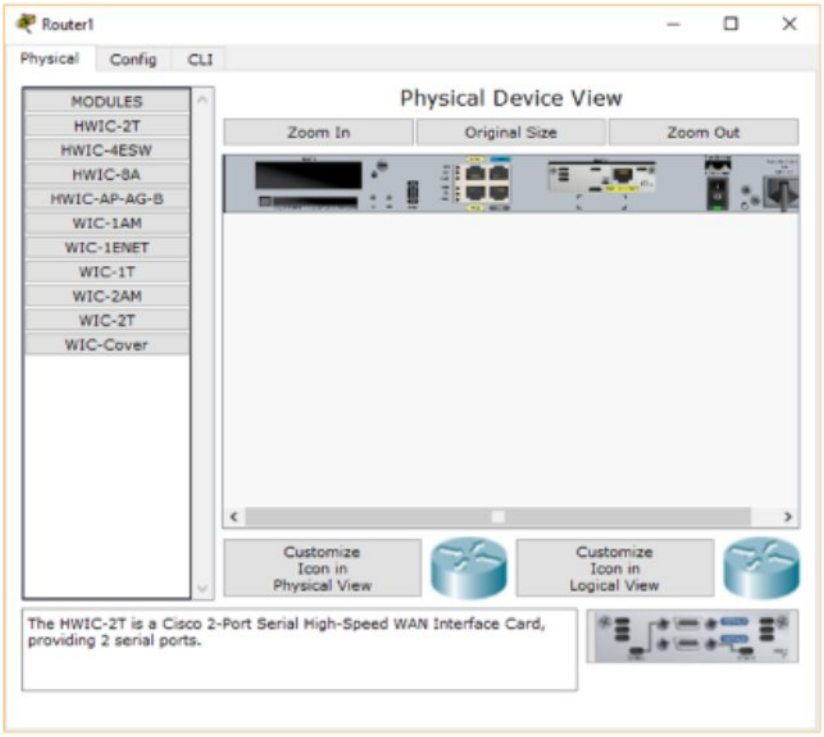
Rozpoczniemy od najprostszego rodzaju routingu – trasowania statycznego. Na czym ono polega, było wytłumaczone wcześniej. Bierzemy się więc za praktykę:

Utwórz topologię taka jak na rysunku poniżej: (nie przejmuj się na razie, że u ciebie połączenia świecą się na czerwono. Interfejsy nie są jeszcze po prostu włączone i skonfigurowane). Ważna uwaga: zwracaj uwagę na interfejsy, które ze sobą łączysz. Są one opisane przy połączeniach na rysunku poniżej jako np.: fa0/0, fa0/1. Aby moje instrukcje sprawdziły się w twoim przypadku, musisz połączyć routery dokładnie tak, jak na schemacie.



Router 1841 ma domyślnie dwa interfejsy FastEthernet. Urządzeniu Router1 potrzebny jest dodatkowy, trzeci. Aby go dodać, musimy otworzyć okno konfiguracji routera. Na pierwszej zakładce (Physical) widzimy przybliżony wygląd tego routera w rzeczywistości. Musimy do niego dołożyć moduł WIC-1ENET, który dodaje jeden port Ethernet 10Mbps. Aby to zrobić, wyłączamy

router, przeciągamy moduł na jego miejsce, a potem włączamy router. Efekt tego działania jest widoczny na rysunku poniżej.

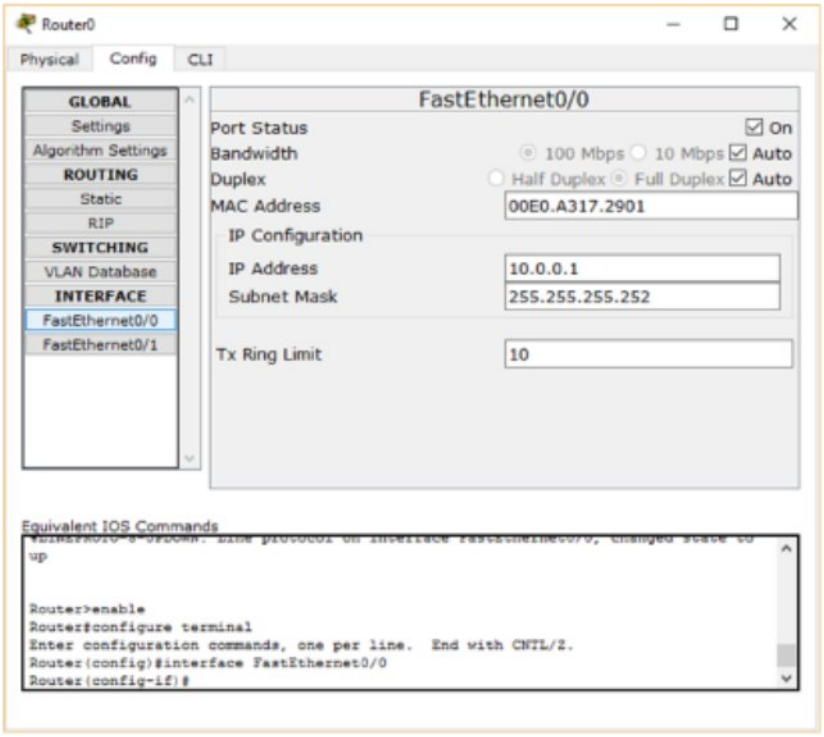


Nadaj komputerom i urządzeniom adresy IP wg poniższej tabelki:

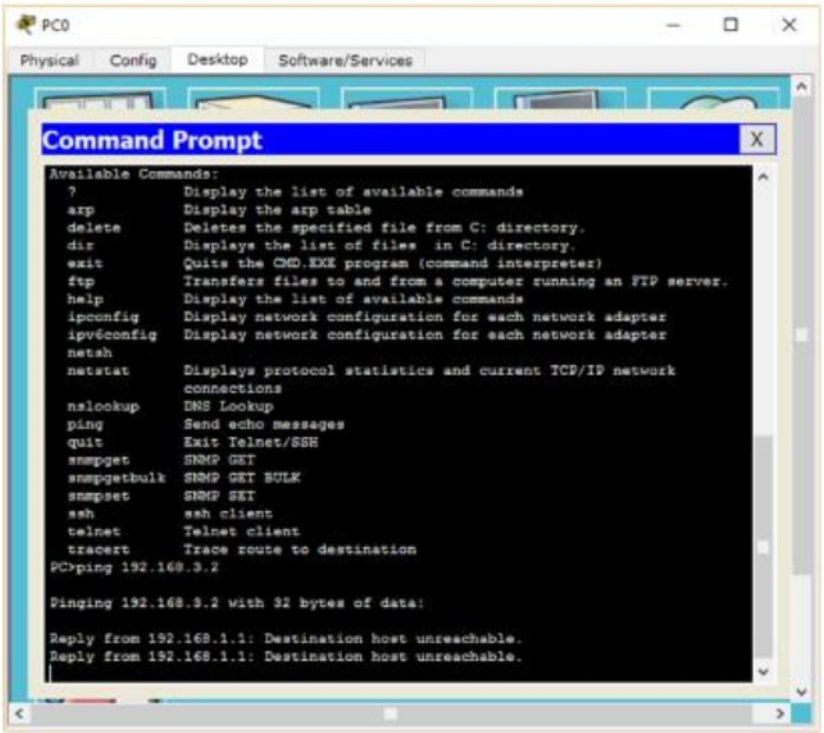
Urządzenie	Interfejs	Adres IP	Maska podsieci
Router0	Fa0/0	10.0.0.1	255.255.255.252
	Fa0/1	192.168.1.1	255.255.255.0
Router1	Fa0/0	10.0.0.2	255.255.255.252
	Fa0/1	10.0.0.5	255.255.255.252
	Eth0/0/0	10.0.0.9	255.255.255.252
Router2	Fa0/0	10.0.0.10	255.255.255.252
	Fa0/1	192.168.2.1	255.255.255.0
Router4	Fa0/0	10.0.0.6	255.255.255.252
	Fa0/1	192.168.3.1	255.255.255.0

Komputer	Adres IP	Brama sieciowa
PC0	192.168.1.2	192.168.1.1
PC1	192.168.1.3	
PC2	192.168.2.2	192.168.2.1
PC3	192.168.2.3	
PC4	192.168.3.2	192.168.3.1
PC5	192.168.3.3	

Nie zapomnij o włączeniu interfejsów na routerach!. Aby to zrobić, zaznacz ptaszkiem polecenie On w oknie konfiguracji interfejsu, podobnie jak na poniższym screenie.



Najpierw spróbujmy przetestować połączenie bez konfiguracji routingu. Wejdź w terminal komputera PC0 i wpisz polecenie ping 192.168.3.2. Jak widzisz na screenie poniżej, od bramy sieciowej przyszła odpowiedź, że lokalizacja docelowa jest niedostępna. Dzieje się tak, ponieważ router nie ma żadnych informacji o pozostałych sieciach. Spróbujemy rozwiązać ten problem.



Zacznijmy od Router0. Musimy mu wyjaśnić, że aby pakiet trafił do sieci 192.168.3.0, musi go wysłać na interfejs Fa0/0 urządzenia Router1. Wejdźmy

więc do CLI na Router0. Jeśli widzisz znak zachęty >. Np.: Router>, to wpisz polecenie en, a następnie config t. Informacja, że jesteś w trybie konfiguracji, będzie przedstawiona w postaci znaku zachęty wyglądającego np.: tak: Router(config)#.

Zapoznajmy się z poleceniem ip route. Jego podstawowa składnia jest następująca:

```
ip route adres_ip_docelowej_sieci maska_docelowej_sieci adres_nastepnego_skoku
```

Gdy jesteśmy w trybie konfiguracji, podajmy polecenie:

```
ip route 192.168.3.0 255.255.255.0 10.0.0.2.
```

Oznacza ono, że jeśli pakiet ma trafić do sieci 192.168.3.0, musi zostać wysłany na adres 10.0.0.2, czyli na interfejs Fa0/0 urządzenia Router1.

Pakiet dociera do Router1, ale głupi dalej nie wie, co z tym pakietem zrobić. Poinstruujmy go więc, aby wysłał to co otrzyma na interfejs Fa0/0 urządzenia Router4. Po zalogowaniu się i przejściu w tryb konfiguracji podaj polecenie:

```
ip route 192.168.3.0 255.255.255.0 10.0.0.6
```

.
Spróbuj teraz ponownie spingować komputer PC5 z PC0. Co, dalej niepowodzenie? Jeśli zastanawiasz się, dlaczego, przejdź w tryb symulacji i obserwuj drogę pakietu ICMP. Zauważysz, że dociera poprawnie do PC5 i nawet następuje odpowiedź, ale nie jest skonfigurowana trasa w drugą stronę. Router4 nie wie, gdzie ma przesłać ten pakiet. Musimy więc wytłumaczyć mu, gdzie leży sieć 192.168.1.0. Robimy to w ten sam sposób, co poprzednio: Zaloguj się na Router4 i wpisz polecenie:

```
ip route 192.168.1.0 255.255.255.0 10.0.0.5.
```

Podobnie na Router1:

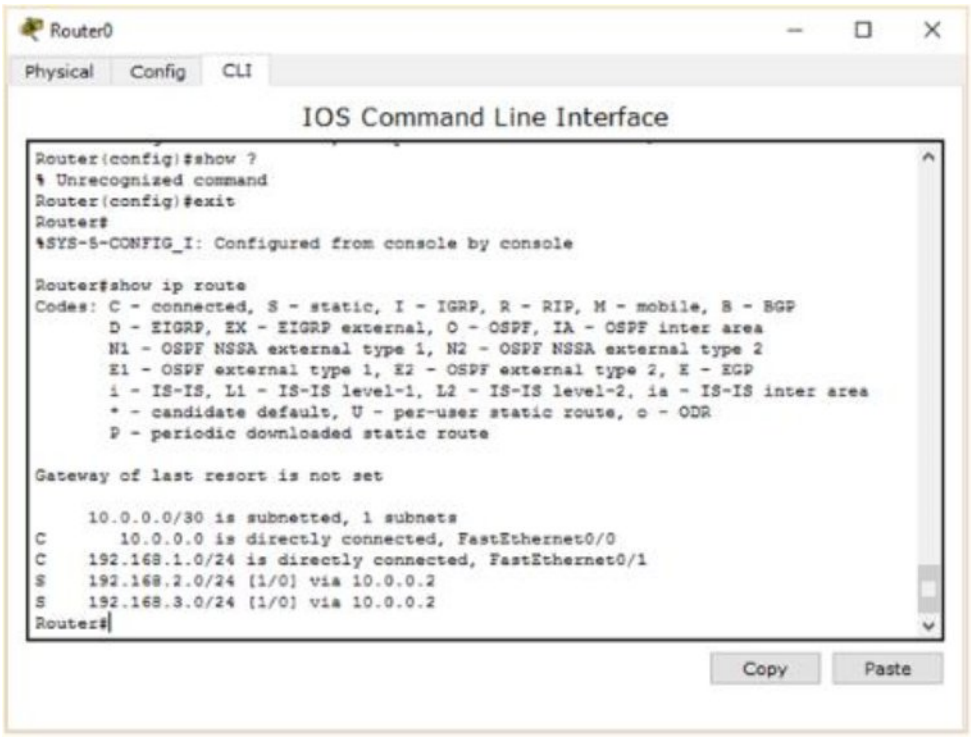
```
ip route 192.168.1.0 255.255.255.0 10.0.0.1
```

Teraz powinniśmy uzyskać pełną komunikację. Ping z PC0 lub PC1 na PC4 lub PC5 powinien zakończyć się sukcesem.
Została nam jeszcze komunikacja z siecią 192.168.2.0. Rozwiązanie tego problemu pozostawiam jako ćwiczenie dla czytelnika. (Jeśli nie chce ci się tego robić, pod koniec artykułu znajdują się linki do gotowych plików pkt).

Sieć skonfigurowana. Uzyskaliśmy komunikację między wszystkimi komputerami. Zerknijmy jeszcze na tablice routingu w poszczególnych routerach. Zacznijmy od Router0. Aby to zrobić, musimy wyjść z trybu konfiguracji poleceniem exit (znak zachęty powinien wyglądać np.: tak: Router#). Wydajemy polecenie

```
show ip route
```

.



Na początku wyniku tego polecenia mamy opis poszczególnych znaków – co one znaczą. Później mamy wyświetloną właściwą tablicę routingu. Zacznijmy jej analizę:

```
10.0.0.0/30 is subnetted, 1 subnets
```

```
C 10.0.0.0 is directly connected, FastEthernet0/0
```

Oznacza, że sieć 10.0.0.0 jest podzielona na podsieci, a do routera bezpośrednio podłączona jest jedna podsieć – zgadza się, gdyż adresów z tej podsieci używaliśmy do adresowania interfejsów służących do komunikacji między routerami.

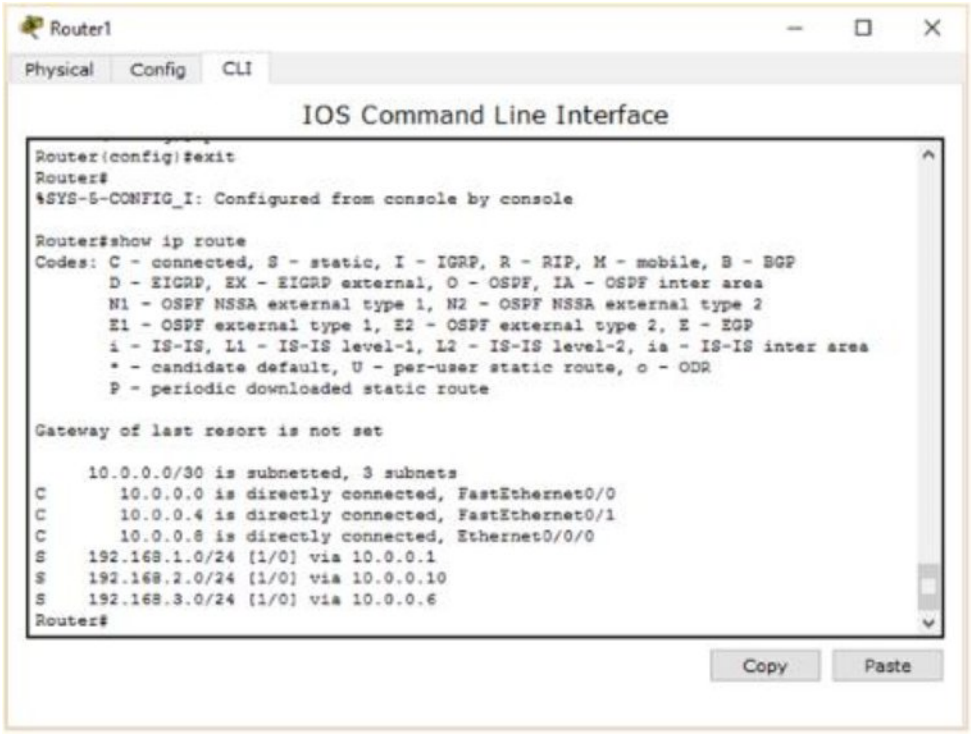
```
C 192.168.1.0/24 is directly connected, FastEthernet0/1
```

kolejna sieć przyłączona bezpośrednio routera.

```
S 192.168.2.0/24 [1/0] via 10.0.0.2.
```

tu zapisane zostało to, co wcześniej konfigurowaliśmy. S oznacza, że mamy do czynienia z routingiem statycznym. Potem mamy podany adres sieci docelowej 192.168.2.0/24. To, co jest w nawiasie kwadratowym to nic innego jak metryka – opłacalność poruszania się daną trasą. Po słówku via mamy adres następnego skoku – czyli aby pakiet dotarł do sieci 192.168.2.0, jego

trasa musi przebiegać przez router z interfejsem 10.0.0.2.
Spójrzmy jeszcze na Router1, który łączy te 3 sieci.

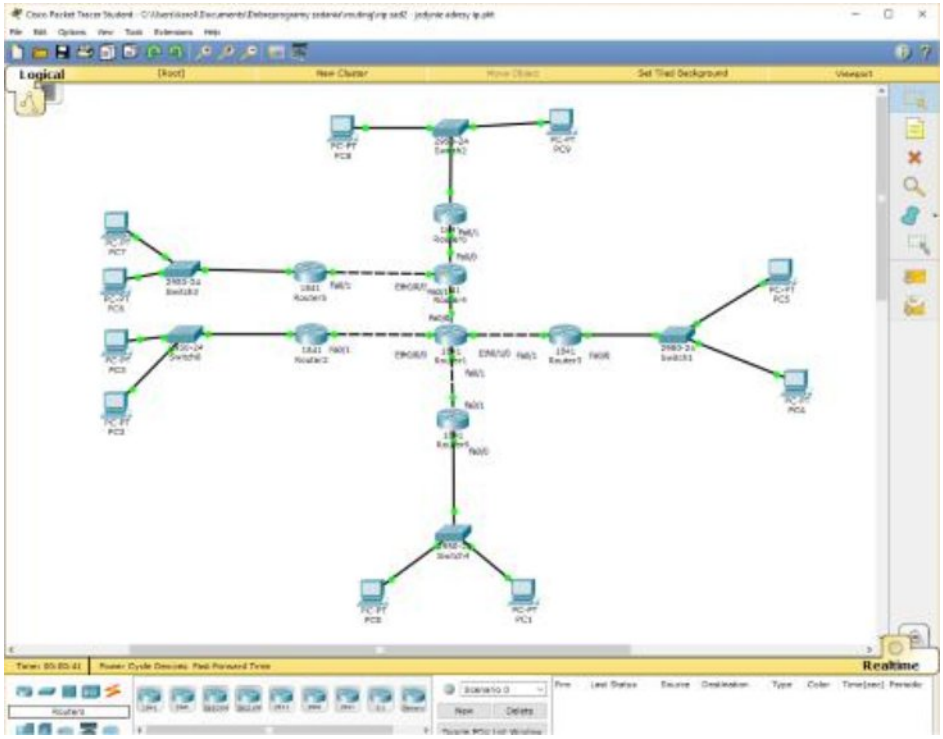


Widzimy tu, że router ten wie o sieciach 192.168.1.0, 192.168.2.0 i 192.168.3.0 tylko dlatego, że mu o tym powiedzieliśmy. Inaczej nie wiedziałby, że takie sieci w ogóle istnieją. Do routera są przyłączone bezpośrednio sieci 10.0.0.0 przez fa0/0, 10.0.0.4 przez fa0/1 i 10.0.0.9 przez eth0/0/0.

Jaka jest podstawowa wada routingu statycznego, którym się przed chwilą zajmowaliśmy? Jak pewnie zauważyłeś, aby dodać nowy router, musimy zmieniać konfigurację na wszystkich pozostałych routerach znajdujących się w sieci. Często jest to nieporęczne, niewygodne, a nawet niemożliwe. Routing dynamiczny nie posiada tych wad, o czym się za chwilę przekonasz.

Routing dynamiczny – RIP

Na początku zajmiemy się najprostszym z protokołów routingu dynamicznego – RIP. Zbudujmy topologię taką jak na schemacie poniżej:



Naszym celem jest to, aby każda sieć mogła komunikować się z każdą. Ustalmy następujące adresy IP:

Urządzenie	Interfejs	Adres
Router0	Fa0/0	192.168.1.1
	Fa0/1	10.0.0.1
Router1	Fa0/0	13.0.0.1
	Fa0/1	10.0.0.2
	Eth0/0/0	11.0.0.1
	Eth0/1/0	12.0.0.1
Router2	Fa0/0	192.168.2.1
	Fa0/1	11.0.0.2
Router3	Fa0/0	192.168.3.1
	Fa0/1	12.0.0.2
Router4	Fa0/0	15.0.0.1
	Fa0/1	13.0.0.2
	Eth0/0/0	14.0.0.1
Router5	Fa0/0	192.168.4.1
	Fa0/1	14.0.0.2
Router6	Fa0/0	192.168.5.1
	Fa0/1	15.0.0.2

Urządzenie	Adres IP	Brama
PC0	192.168.1.2	192.168.1.1
PC1	192.168.1.3	
PC2	192.168.2.2	192.168.2.1
PC3	192.168.2.3	
PC4	192.168.3.2	192.168.3.1
PC5	192.168.3.3	
PC6	192.168.4.2	192.168.4.1
PC7	192.168.4.3	
PC8	192.168.5.2	192.168.5.1
PC9	192.168.5.3	

Zacznijmy konfigurację od routera Router0. Musimy poinformować router, jakie sieci ma rozgłaszać do swoich sąsiadów. W tym celu wchodzimy w CLI, a potem w tryb konfiguracji. Pierwszym poleceniem, które należy wydać, jest:

```
router rip
```

Znak zachęty zmieni się na np.: Router(config-router)# co oznacza, że modyfikujemy ustawienia protokołów routingu. Teraz wpisujemy polececenia:

```
network 192.168.1.0

network 10.0.0.0
```

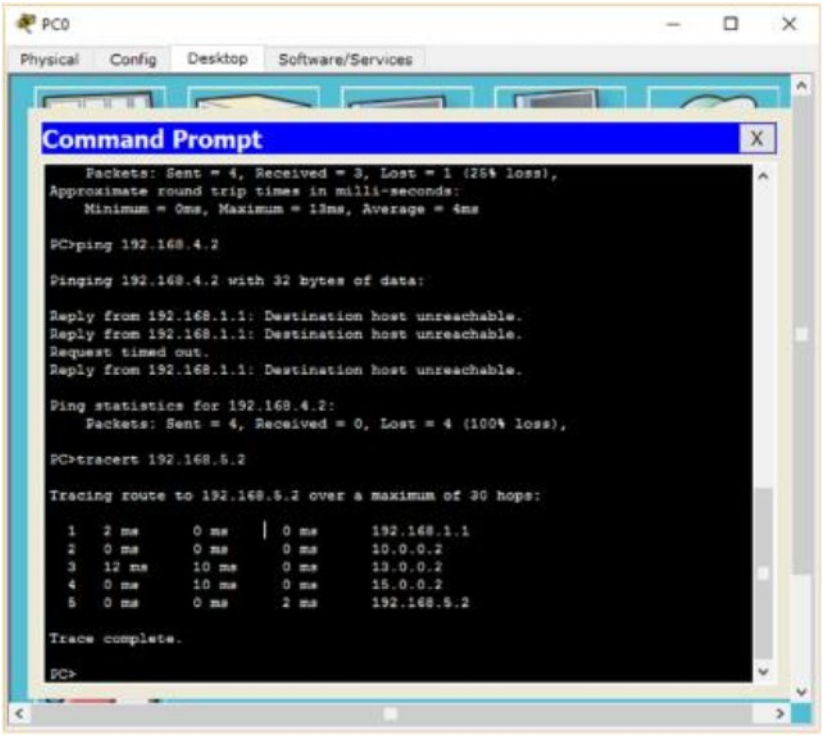
Jak myślisz, co wprowadziliśmy? No właśnie, są to adresy sieci bezpośrednio przyłączonych do naszego urządzenia Router0. Router rozgłosi tę informację do wszystkich sąsiednich routerów, dzięki temu np.: Router1 będzie mógł dowiedzieć się o istnieniu sieci 192.168.1.0. Podobną konfigurację przeprowadzamy na pozostałych routerach, np.: na router1 należy wydać następujące polecenia:

```
router rip

network 10.0.0.0

network 11.0.0.0

network 12.0.0.0

network 13.0.0.0
```

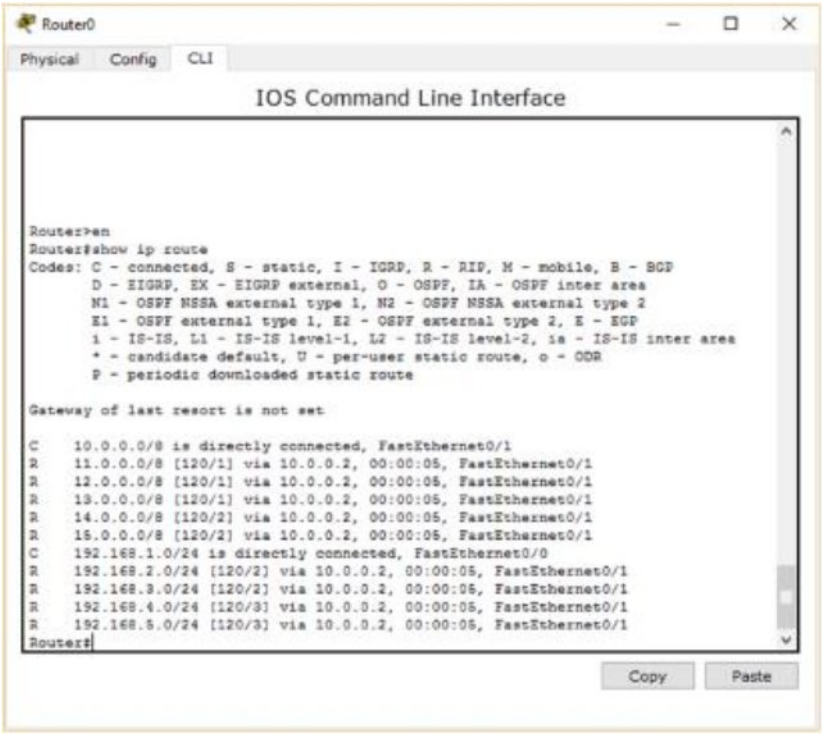
Konfigurację pozostałych routerów pozostawiam w gestii czytelnika, gdyż nie jest to zbytnio skomplikowane.

Poznaliśmy już polecenie ping. Jeśli skonfigurowałeś już protokół RIP na wszystkich routerach, możemy zapoznać się z poleceniem tracert, które w przyszłości będziemy często wykorzystywali. Wejdź na komputer PC0 i wydaj polecenie:

```
tracert 192.168.5.2
```



Polecenie to wyświetli ci trasę, jaką przebył pakiet. Jest to bardzo przydatne narzędzie diagnostyczne, ułatwiające znalezienie błędów w konfiguracji sieci. Zajmijmy się teraz ostatnią kwestią, a mianowicie poddajmy analizie tablicę routingu z urządzenia np.: Router0.

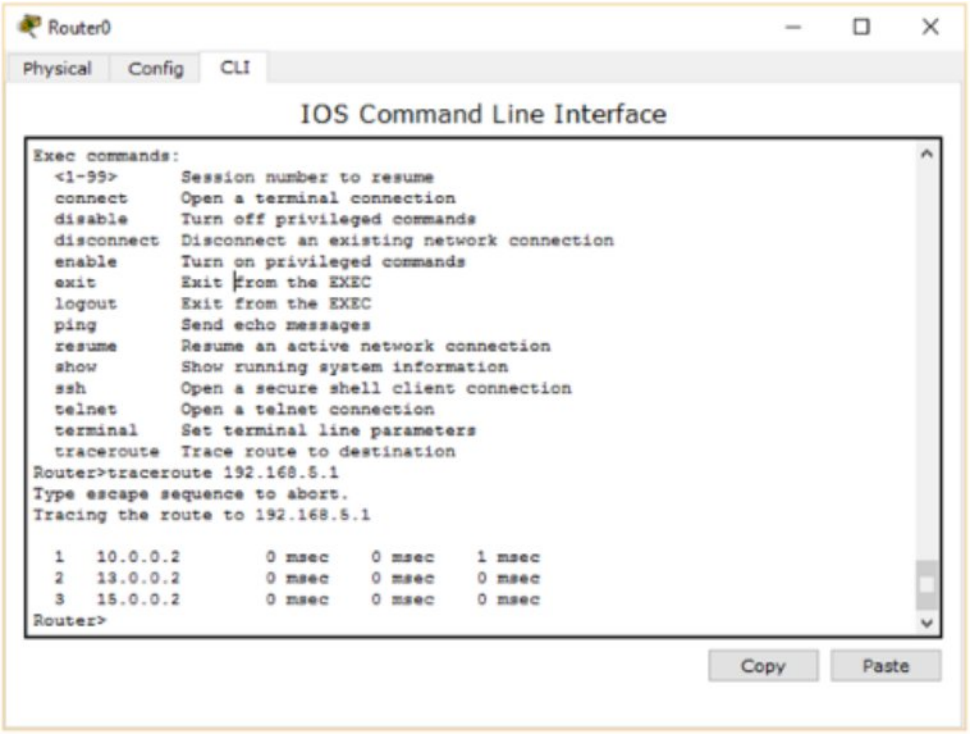


Widzimy nowe oznaczenie – R – co oznacza, że sieci te rozgłaszane są za pomocą protokołu RIP. Zwróćmy szczególną uwagę na metrykę, która składa

się z dwóch części: Pierwsza – 120 to tzw. domyślny dystans administracyjny (default administrative distance). Jego wartość zależy od protokołu routingu, którego użyjemy. Wartość tego pola dla różnych protokołów przedstawia poniższa tabelka:

Protokół	Domyślna wartość
Interfejs podłączony bezpośrednio	0
Routing statyczny	1
Internal EIGRP	90
IGRP	100
OSPF	110
RIP	120

Zajmijmy się teraz wartością po znaku /. Przypomnij sobie to, co mówiliśmy we wstępie o protokole RIP. Dla RIP metryką, oznaką wydajności danej trasy jest jedynie liczba skoków. I właśnie to jest znaczenie tej liczby. Np.: aby dostać się do sieci 192.168.5.0, należy wykonać 3 skoki. Możemy to zresztą sprawdzić. Wyjdź z trybu administracyjnego i wydaj polecenie traceroute 192.168.5.1 Co widzimy?



Rzeczywiście, aby pakiet dotarł do sieci 192.168.5.0, należy wykonać 3 skoki. Pozostałej części tablicy routingu nie będę omawiał, gdyż wygląda to analogicznie jak w przypadku routingu statycznego.

Link do plików pkt

Tu możesz pobrać gotowe pliki pkt z sieciami omawianymi w tym wpisie.

Liczniki RIP

Wcześniej dowiedzieliśmy się, że RIP wysyła aktualizację tras co pewien czas. Skąd router wie, w którym momencie ma wysłać taką aktualizację? Otóż, protokół RIP posiada cztery liczniki: Update Timer, Invalid Timer, Flush Timer i Hold-down Timer. One sterują takimi czynnościami. Czym dokładnie zajmują się te liczniki? Zobaczmy:

- Update Timer – jest to jeden z najprostszych liczników do zrozumienia. Jego wartość to po prostu interwał między dwoma rozgłoszeniami. Domyślna wartość tego licznika to 30. Oznacza to, że aktualizacje tras routingu są rozsyłane co 30 sekund.
- Invalid Timer – załóżmy, że router R1 dostaje od routera R2 cykliczne aktualizacje co 30s. Nagle komunikacja urywa się. Mija pewien czas i nadal nie ma żadnego znaku życia od R2. Co się wtedy dzieje? Router R1 uznaje, że R2 kopnął w kalendarz i oznacza ten router jako nieosiągalny. Za to właśnie odpowiada Invalid Timer. Jest maksymalny czas, przez który wpis w tablicy routingu może istnieć bez aktualizacji. Po upływie czasu zawartego w Invalid Timer router docelowy jest uznawany za nieosiągalny. Domyślna wartość tego licznika to 180s.
- Flush timer – Załóżmy, że trasa do routera R2 została uznana za nieosiągalną przez poprzedni licznik. Router jest nieosiągalny, jednakże wpis cały czas wisi w tablicy routingu. Czy tak będzie w nieskończoność? Otóż nie. Po upływie czasu zawartego w Flush timer wpis jest usuwany z tablicy routingu. Liczniki Flush timer i Invalid Timer są startowane automatycznie – w momencie, kiedy nie występują aktualizację ze strony routera R2. Stąd też wartość licznika Flush timer musi być większa od wartości licznika Invalid timer. Domyślna wartość licznika Flush timer wynosi 240 sekund. Przykład: Jeśli od routera R2 nie docierają informacje aktualizacyjne, to po 180s trasa do tego routera jest oznaczana jako nieosiągalna, a po kolejnych 60s wpis jest usuwany z tablicy routingu.
- Hold-down timer – ten licznik ma najbardziej skomplikowane zadanie. Wróćmy jeszcze raz do wcześniej wspominanego przykładu. Router R2 staje się nieosiągalny. Trasa do niego jest oznaczana jako nieosiągalna. Co jednak stanie się, jeśli w tym momencie od innego routera (np.: R3) dostaniemy informację o tym, że jednak istnieje trasa do R2? Mogłoby wtedy powstać zjawisko pętli routingu, o którym więcej będzie w następnym rozdziale. Aby do tego nie dopuścić, opracowano właśnie ten licznik. Jeśli od routera, do którego biegnie dana trasa (w tym wypadku R2) nie otrzymano informacji aktualizacyjnych, to trasa ta, po oznaczeniu jako nieosiągalna, nie może być nadpisana przez informacje aktualizacyjne otrzymywane od innych

routerów. Domyślna wartość tego licznika to 180 sekund. Pomyślmy, jak to by wyglądało w naszym przykładzie. Otóż, jeśli trasa do R2 zostanie oznaczona jako nieosiągalna, to przez 180 sekund informacja o „nieosiągalności” tego routera nie może zostać zmieniona. Nawet jeśli nasz router dostanie informację o dostępnej trasie, np.: od R3 to informacja ta nie zostanie wpisana do tablicy routingu naszego routera.

Pętla routingu – czyli „korek w sieci”.

Jak wiemy z poprzedniej części kursu, routery zapisują trasy do poszczególnych sieci w swoich tablicach routingu. Dzięki nim router wie, do kogo wysłać dany pakiet tak, aby dotarł do miejsca docelowego. Co się jednak stanie, jeśli przez przypadek do tablicy routingu zostanie wprowadzona jakaś błędna informacja? Może to zagrozić spójności sieci, a także doprowadzić to tzw. „pętli routingu”. Cóż to takiego jest? Zjawisko to może wystąpić na przykład w sytuacji, kiedy jedna z sieci przyłączonych bezpośrednio do routera stała się niedostępna. Zjawisko to polega na tym, że pakiet krąży między dwoma routerami (np.: R2 wysyła go do R1, a R1 odsyła do R2). Zajmijmy się analizą przykładowej sytuacji, w której może wystąpić pętla routingu.

Załóżmy, że wszystkie funkcje zapobiegające pętlom routingu (takie jak podzielony horyzont, o czym mowa będzie później) są wyłączone. W pewnym momencie połączenie między R1 a Switch1 się urywa …

R1 bezpośrednio przyłączony do tej sieci wymaże nieaktualną informację z tablicy routingu. Lecz jeśli w tym momencie od innego routera (np.: R2) otrzyma kopię tablicy routingu zawierającą feralną sieć, to R1 wpisze sobie do swojej tablicy, że trasa do niedostępnej sieci biegnie właśnie przez R2, od którego otrzymano informację. Jeśli w tym momencie jakiś komputer będzie chciał połączyć się z niedostępną siecią, to przebieg trasy pakietu będzie następujący:

1. R2 otrzymuje pakiet. W swojej tablicy routingu ma zapisane, że aby pakiet dotarł do docelowej sieci, musi zostać wysłany do R1. Tak też się dzieje.
2. Pakiet dociera do R1. Przez to, że R1 otrzymał błędną informację od R2, sądzi, że należy pakiet wysłać do R2. Tak też się dzieje.
3. R2 otrzymuje pakiet. Powtarza się sytuacja z punktu pierwszego.

Występowanie w sieci pętli routingu niesie ze sobą wiele przykrych konsekwencji. Pakiety, które były adresowane do konkretnej sieci, mogą „ugrzązć w korku” między dwoma routerami i nigdy nie dotrzeć do celu. Pętle routingu powodują obciążenie procesora routera przez co sieć staje się mniej wydajna.

Na szczęście, projektanci protokołów trasowania przewidzieli taką sytuację i opracowali kilka sposobów na ominięcie tego typu problemów. Są to:

- Czas „wstrzymania” (hold-down timer) – został omówiony w poprzednim rozdziale, więc nie będę do tego wracał.
- Podzielony horyzont (Split-horizon) – bardzo ciekawa funkcja. Jeśli ta funkcja nie jest włączona, to R2, po otrzymaniu informacji aktualizacyjnej od R1, natychmiastowo wyśle do R1 kopię własnej tablicy routingu. Zwykle nie ma takiej potrzeby. Taka funkcjonalność nie przynosząc wielkich zalet, może natomiast powodować wiele problemów, takich jak nieprawidłowe wpisy w tablicy routingu..
- Uaktualnienia zatrucia zwrotnego (poison reverse updates) – jeśli w sieci istnieje pętla routingu, to zjawisko takie najłatwiej rozpoznać po metryce, która zwiększa się w niekontrolowany sposób. W takiej sytuacji są wysyłane uaktualnienia zatrucia zwrotnego, które mają na celu usunięcie nieprawidłowej trasy z tablic routingu routerów. Podobną funkcję w protokole RIP pełnią tzw. „triggered updates”. W skrócie, jeśli sieć przyłączona do routera okaże się nieosiągalna, to natychmiastowo, bez czekania na Update Timer wysyłana jest informacja do pozostałych routerów o nieosiągalności trasy.

Zaawansowana konfiguracja protokołu RIP

Wiemy już, co to jest pętla routingu. Zastanówmy się teraz, jak skonfigurować protokół RIP, aby uniknąć takich sytuacji. Skupimy się głównie na drugiej wersji protokołu RIP, umożliwiającą korzystanie z VLSM. Poznamy także inną przyczynę powstawania pętli routingu, która wynika ze źle skonfigurowanej funkcji „auto summary”. Tak więc zaczynajmy!

W poprzednim rozdziale poznaliśmy rodzaje liczników RIP. Spróbujmy teraz w praktyce zmienić wartość poszczególnych liczników i ustalić, jak to wpływa na sieć w rzeczywistości. Przyjmijmy, że otrzymałeś od pewnej firmy następujące zadanie:

„Chcemy stworzyć sieć składającą się z czterech routerów. Wszystkie sieci muszą się ze sobą łączyć. Do łączenia routerów należy użyć adresów IP z sieci 10.0.0.0/31. Natomiast komputery powinny znajdować się w podsieciach 172.20.0.0/29”. Należy użyć protokołu RIPv2. Konwergencja powinna być sześciokrotnie mniejsza niż przy tradycyjnej konfiguracji. Komputer z dowolnej

podsieci powinien mieć możliwość połączenia z dowolnym innym. Po to cię wezwaliśmy.”

Link do zadania znajduje się tutaj. (plik: zad1.pka).
Poniższa tabelka zawiera adresy IP interfejsów, które należy przydzielić routerom abyś wykonał zadanie zgodnie z zaleceniami.

Router	Fa0/0	Fa1/0	Fa1/1
R1	10.0.0.5/31	10.0.0.2/31	172.20.0.9/29
R2	10.0.0.14/31	10.0.0.1/31	172.20.0.1/29
R3	10.0.0.13/31	10.0.0.10/31	172.20.0.25/29
R4	10.0.0.6/31	10.0.0.9/31	172.20.0.17/29

Adresy IP komputerów i ich bramy domyślne przedstawia poniższa tabelka:

Komputer	Adres IP	Brama domyślna
PC0	172.20.0.2/29	172.20.0.1
PC1	172.20.0.10/29	172.20.0.9
PC2	172.20.0.18/29	172.20.0.17
PC3	172.20.0.26/29	172.20.0.25

Musimy osiągnąć łączność między wszystkimi komputerami. Aby to uczynić, musimy skonfigurować protokół RIP w wersji drugiej. Dzieje się to podobnie, jak w wypadku wersji pierwszej. Jedyną różnicą jest to, że musimy poinformować router o tym, że chcemy użyć właśnie wersji drugiej protokołu RIP.

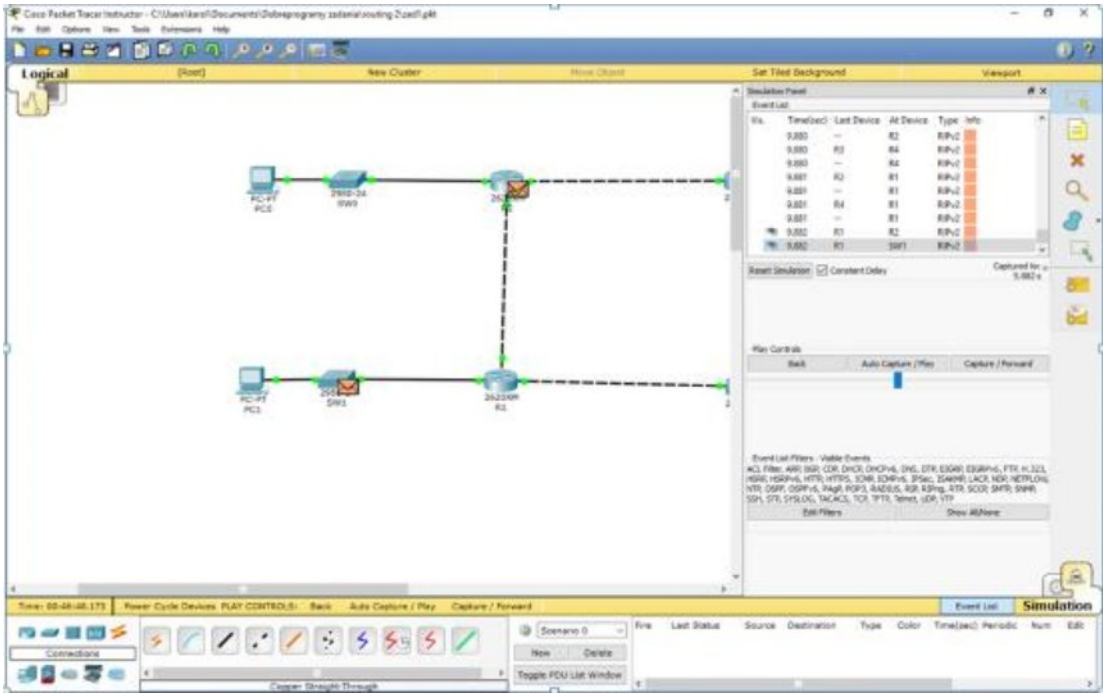
Pamiętasz pewnie jak wejść do trybu konfiguracyjnego routera. Będąc w CLI wydajemy polecenie config t. Potem należy wydać polecenie router rip, aby przejść do konfiguracji protokołu routingu. Tu poznamy jedno z nowych poleceń. Aby zadeklarować, że chcemy użyć drugiej wersji protokołu RIP, musimy wydać polecenie version 2. Dzięki temu sieci VLSM będą rozgłaszane poprawnie. Jak pamiętasz, informacje o rozgłaszanych sieciach wpisywaliśmy za pomocą polecenia network. Polecenie to nie przyjmuje jako argumentu maski podsieci. Więc np.: jeśli mamy podsieci 10.0.0.4/31 i 10.0.0.8/31 to wystarczy, że wydamy polecenie network 10.0.0.0. Podsieci należące do sieci klasy A 10.0.0.0 zostaną rozpoznane automatycznie.

Dla R1 ciąg wydanych poleceń powinien być następujący:

```
version 2
network 10.0.0.0
network 172.20.0.0
```

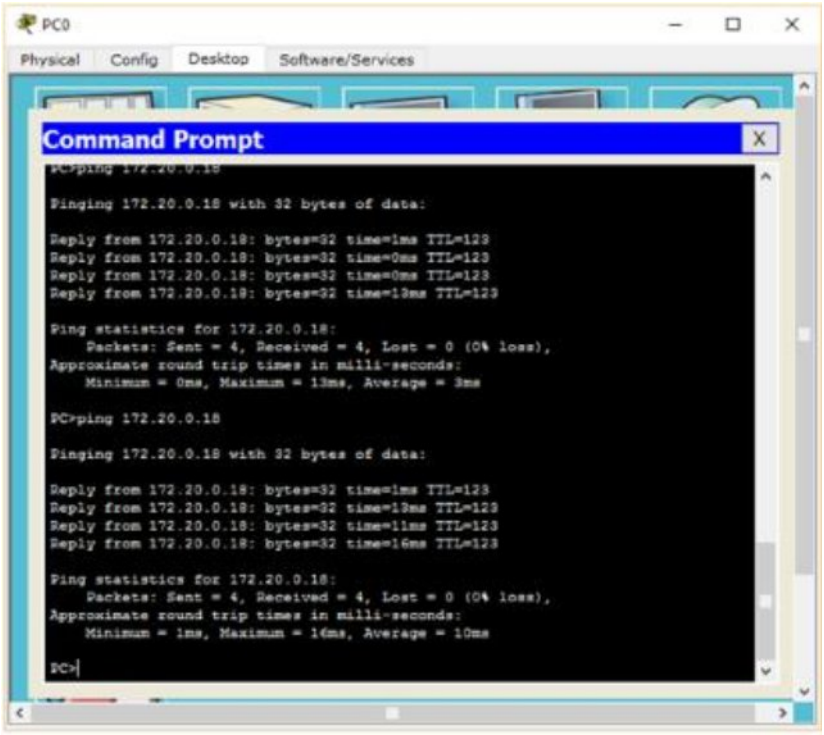
Dla pozostałych routerów (R2-R4) polecenia będą analogiczne.

Skonfigurowaliśmy podstawowy routing. Zanim przejdziemy do testów, zwróćmy uwagę na miejsca docelowe pakietów rozgłoszeniowych RIP. Przejdźmy w tryb symulacji i poobserwujmy chwilę (możemy klikać przycisk Capture/Forward aby szybciej dojść do tego momentu) do jakich urządzeń docierają pakiety rozgłoszeniowe RIP.

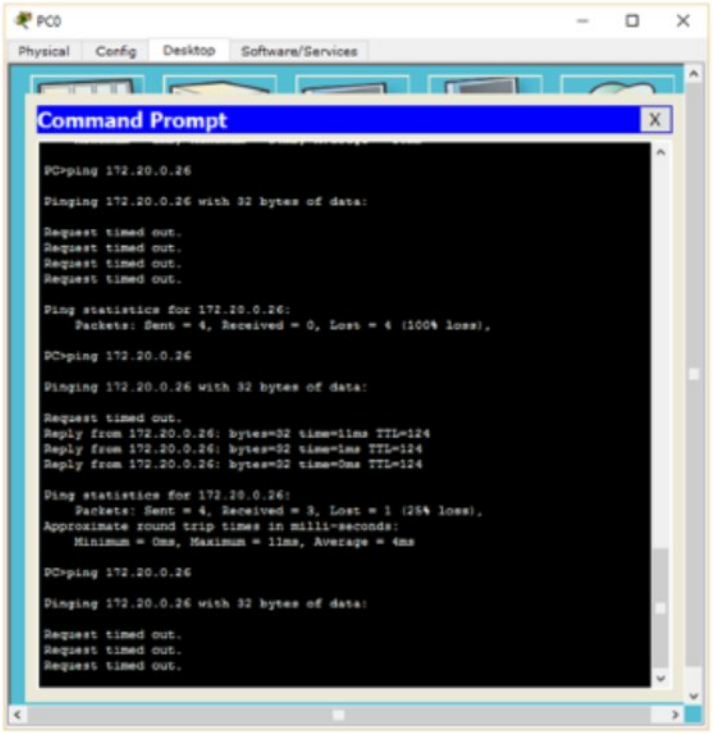


Zauważymy, że R1 poinformował nie tylko sąsiednie R2 i R4 o podsieciach. Informacja dotarła także do SW1. Nie stanowi to wielkiego problemu, ale mimo wszystko w takiej sieci generowany jest zbędny ruch. Aby to naprawić, musimy użyć kolejnego nowego polecenia: passive-interface. Jako argument polecenie to przyjmuje nazwę interfejsu. Jeśli ustawimy jakiś interfejs jako pasywny, to ten interfejs nie będzie wysyłał pakietów routingu. Na każdym z routerów ustawmy więc jako pasywny interfejs fa1/1, gdyż właśnie ten interfejs w każdym wypadku jest podłączony do podsieci zawierającej komputery.

Routing został skonfigurowany. Wejdź teraz na PC0 i spróbuj wysłać pakiet ICMP (czyli spingować) do komputera PC2. Wpisz polecenie ping 172.20.0.18.

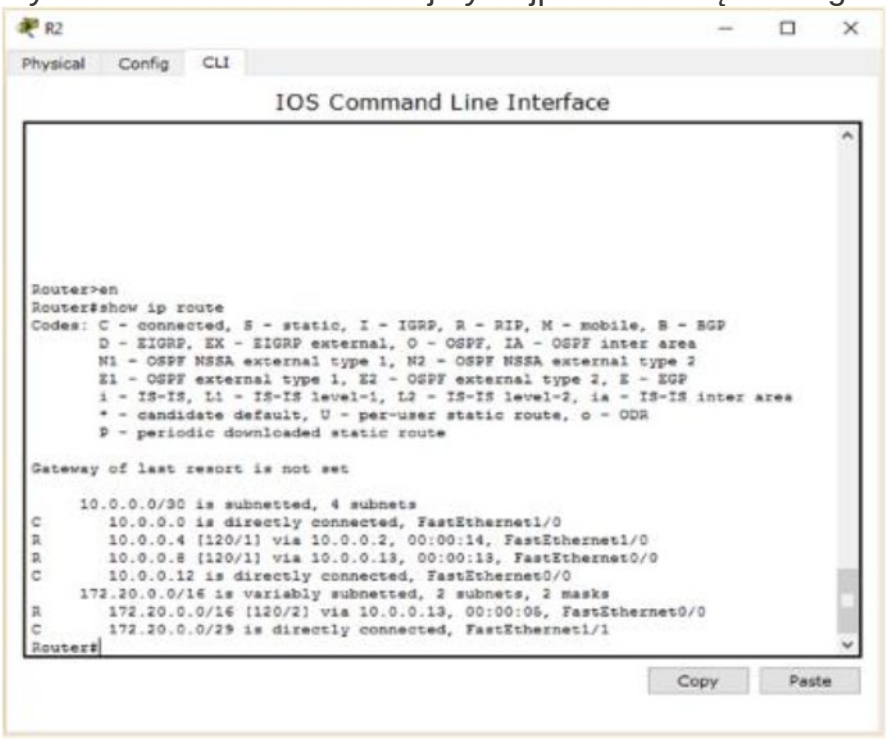


Jak widzimy, wszystko działa poprawnie. Komputer docelowy odpowiada. No to teraz spróbujmy spingować PC3 z PC0 (ping 172.20.0.26).



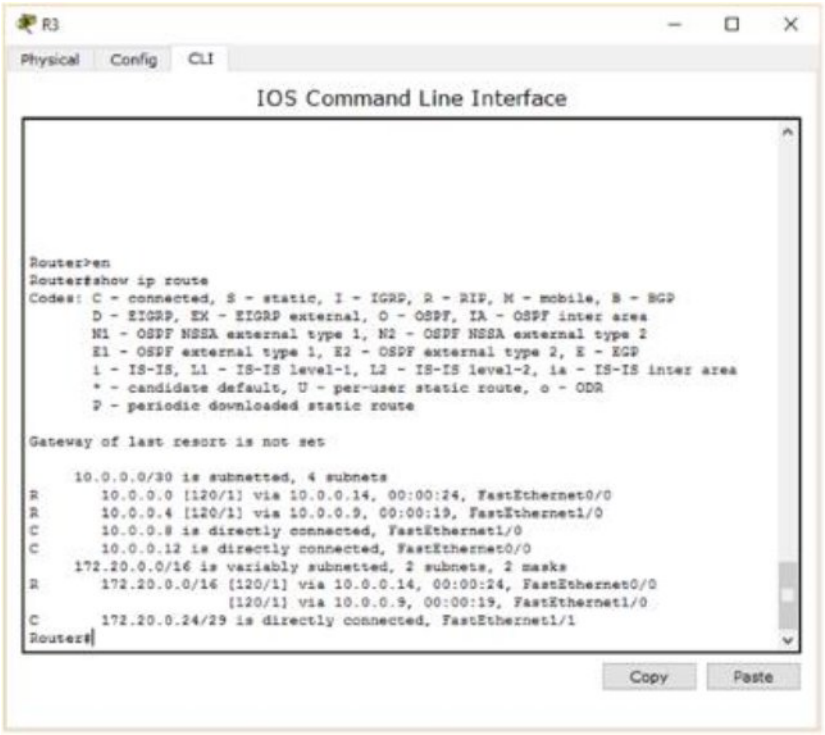
Coś jest nie tak. Bardzo często czas odpowiedzi został przekroczony. W moim wypadku, z 12 wysłanych pakietów jedynie 3 pozwoliły na odebranie odpowiedniego pakietu zwrotnego. Taki problem nie może występować w korporacyjnej sieci, gdyż byłaby po prostu niewydajna. Zastanówmy się, co poszło nie tak? Najprostszym sposobem jest przejście w tryb symulacji i obserwowanie drogi, jaką pokonują pakiety przez sieć. Zróbmy więc to. Tym

razem spingujmy komputer PC3 z PC0 za pomocą „zamkniętej koperty”. Jeśli pakiet przebył poprawną trasę (PC0->SW0->R2->R3->SW3->PC3) to ponów próbę. Przy kolejnych próbach (prędzej czy później) zauważysz, że pakiet „zbacza” z optymalnej trasy. Np.: z R3 zamiast do SW3 trafia do R4, który z kolei wysyła pakiet do R1 a ten znowu odsyła go do R4. Dochodzimy do wniosku, że w tej sieci występuje typowa pętla routingu. Jednakże, czym jest ona spowodowana? Aby poznać przyczynę, musimy zajrzeć do tablic routingu poszczególnych routerów. Przeanalizujmy najpierw tablicę routingu R2.

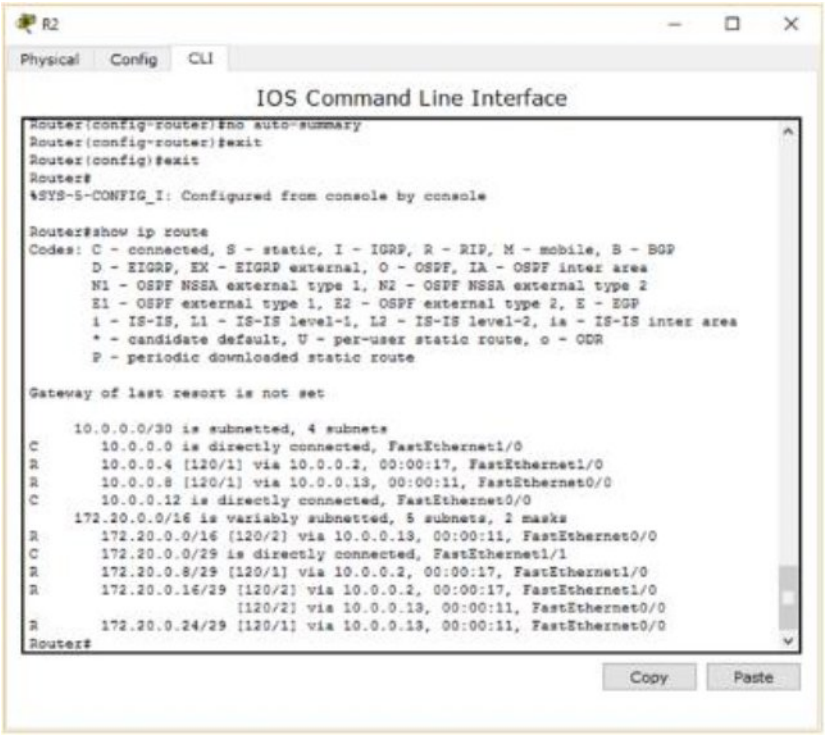


Niby wszystko jest w porządku. Zastanawiający jest jednak poniższy wpis: 172.20.0.0/16 [120/1] via 10.0.0.13 00:00:3 FastEthernet0/0

Zauważyłeś co się nie zgadza? Tak, to właśnie maska podsieci. W tablicy routingu mamy zapisane, że podsieć 172.20.0.0 ma maskę 255.255.0.0, co jest niezgodne z rzeczywistością. Nie widzimy za to prawdziwych podsieci, które zostały przez nas utworzone (poza tymi, które są bezpośrednio podłączone do R2). Spójrzmy więc na kolejny punkt trasy, czyli na R3:



Tu widzimy jeszcze gorsze zamieszanie. Mamy powiedziane, że do podsieci 172.20.0.0 możemy dostać się aż przez dwa routery – o adresach 10.0.0.14 lub przez 10.0.0.9. Po raz kolejny nie widzimy wpisów odpowiedzialnych za trasę do utworzonych wcześniej podsieci (172.20.0.0/29, 172.20.0.8/29 172.20.0.16/29 i 127.20.0.24/29). Może więc wystąpić nawet taka sytuacja, że pakiet otrzymany od PC0 R2 wysyła do R3, R3 odsyła do R2 i tak w kółko. Routery powinny rozgłaszać adresy podsieci bezklasowo, a rozgłaszają klasowo. Jest to wynikiem działania funkcji „auto-summary”. W skrócie odpowiada ona za to, że wszystkie podsieci w tablicy routingu są sprowadzane do ich klasowego odpowiednika. Funkcja ta działa poprawnie jeśli do routingu używamy adresów klasowych. Problem pojawia się gdy zaczynamy korzystać z VLSM. Aby rozwiązać ten problem, na każdym z routerów musimy wydać polecenie: no auto-summary. Spójrzmy więc jeszcze raz na tablicę routingu R2, tym razem po wprowadzeniu zmiany:



Widzimy ogromną różnicę. Teraz router posiada informację o każdej z istniejących podsieci. W takim wypadku nie powstanie już pętla routingu.

Został nam ostatni punkt do wykonania. Musimy sześciokrotnie zmniejszyć czas konwergencji sieci. Domyślasz się pewnie, że powinniśmy zmniejszyć wartość pierwszego licznika, odpowiadającego za interwał między rozgłoszeniami (Update Timer). Skoro domyślną wartością tego licznika jest 30s. To nowa wartość musi być sześciokrotnie mniejsza i wynosić 5 sekund. Aby zmienić wartość liczników, musimy użyć polecenia timers basic. Jako argumenty podajemy kolejne wartości liczników wg takiej kolejności, w jakiej je omawialiśmy (Update,Invalid,Holddown,Flush). My jesteśmy zainteresowani tylko licznikiem Update. Musimy więc na każdym routerze wydać polecenie:

```
timers basic 5 180 180 240
```

Hurra! Wykonaliśmy zadanie poprawnie. Właściciel tej sieci może być teraz zadowolony z naszej pracy :)

EIGRP – Extended Interior Gateway Routing Protocol

EIGRP jest protokołem wektora odległości. Stworzony został jako następca IGRP. Wielką zaletą w porównaniu z IGRP jest obsługa VLSM, co pozwala na znaczne zaoszczędzenie ilości adresów IP.

Podobnie jak RIP, EIGRP swoją wiedzę na temat sąsiadujących sieci opiera na uaktualnieniach, które są rozsyłane domyślnie co 90 sekund. Protokół IGRP/EIGRP posługuje się trzema rodzajami tras:

- Trasy wewnętrzne – trasy między podsieciami sieci powiązanych z interfejsem routera. (np.: między siecią podłączoną do fa0/0 i fa0/1 routera). Jeśli sieci powiązanej z interfejsem routera nie podzieliliśmy na podsieci, to trasy wewnętrzne nie są rozgłaszane.
- Trasy systemowe – trasy do innych sieci znajdujące się w obrębie jednego systemu autonomicznego
- Trasy zewnętrzne – trasy do innych sieci, znajdujących się w innym systemie autonomicznym.

Konfiguracja EIGRP jest banalnie prosta i bardzo podobna do konfiguracji wcześniej poznanego protokołu routingu – RIP.

Aby skonfigurować EIGRP na routerze, musimy w trybie konfiguracyjnym wydać polecenie router eigrp x. Zamiast x podstawiamy numer systemu autonomicznego. Potem należy dodać listę rozgłaszanych sieci. Robimy to tak samo, jak przy protokole RIP. Jednakże tym razem musimy także podać maskę podsieci w postaci wildcard (odwrotności maski tradycyjnej). Przykładowy ciąg poleceń mógłby wyglądać tak:

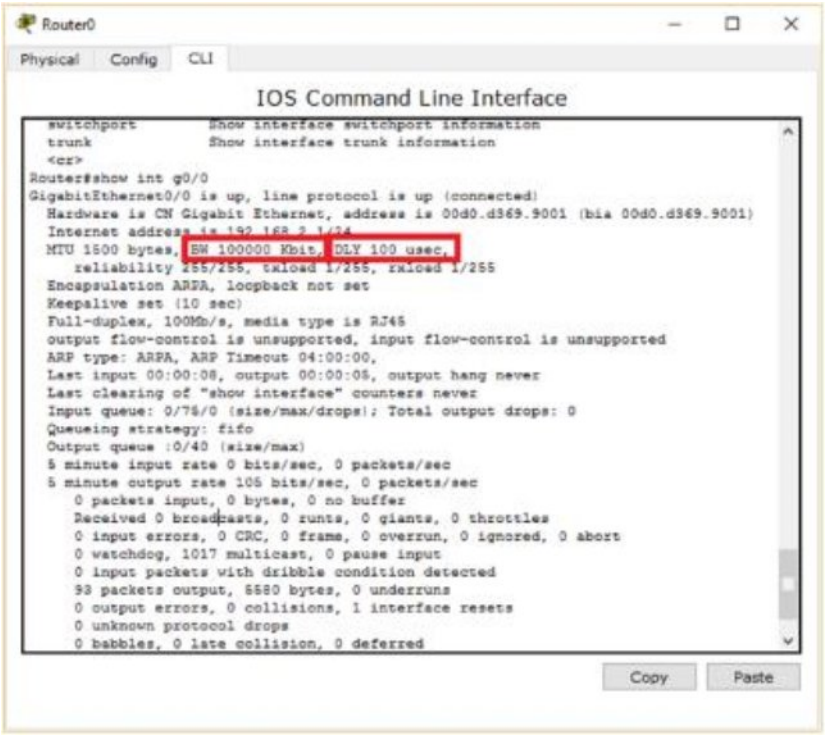
```
router eigrp 1
network 10.0.0.0 0.0.0.3
network 10.0.0.4 0.0.0.3
network 10.0.0.8 0.0.0.3
network 192.168.10 0.0.0.255
```

Również i w tym wypadku najlepiej wyłączyć funkcję auto-summary, gdyż może ona powodować problemy.

Metryka EIGRP

Wiesz już pewnie, co to jest metryka. Mówiliśmy o tym w poprzedniej części kursu. W przypadku RIP, jedynie liczba skoków określała jakość trasy. W EIGRP metryka jest o wiele bardziej skomplikowana i bierze pod uwagę większą ilość parametrów. Brane są pod uwagę następujące czynniki:

- Pasmo – do dalszych obliczeń wybierana jest najniższa przepustowość (np.: chcemy dostarczyć pakiet od R1 do R3 przez R2. Połączenie od R1 do R3 ma przepustowość 1 Gbit/s, natomiast połączenie od R2 do R3 jedynie 10Mbit/s. Aby obliczyć metrykę bierzemy więc pod uwagę szybkość 10Mbit/s.
- Opóźnienie – każdy interfejs powoduje jakieś opóźnienie. Aby je sprawdzić, w trybie uprzywilejowanym możemy wydać polecenie: show interface x (gdzie za X podstawiamy rodzaj i numer interfejsu, np.: fa0/0). Dla interfejsów Ethernet domyślnym opóźnieniem jest 100 mikrosekund.



- Niezawodność – określana jest na podstawie wymiany komunikatów podtrzymania. Domyślnie nie jest brana pod uwagę przy wyznaczaniu metryki.
- Obciążenie – aktualne wykorzystanie łącza prowadzącego do miejsca docelowego. Domyślnie nie jest brane pod uwagę przy wyznaczaniu metryki.

Dowiedzmy się teraz, jak ręcznie obliczyć metrykę EIGRP. Do jej wyliczenia musimy znać wartości pięciu stałych K1-K5. Domyślne wartości przypisują wartość 1 stałym K1 i K3. Natomiast metryki K2, K4, K5 są równe zero. Poznajmy teraz wzór na metrykę:

$$metryka = 256 * (K1 * Pasmo + \frac{K2 * Pasmo}{256 - obciążenie} + K3 * Opóźnienie * \frac{K5}{Niezawodność + K4})$$

Wzór jest nieco skomplikowany, prawda ?. Na szczęście w domyślnej konfiguracji jest on sporo uproszczony. Skoro wartości metryk K2, K4 i K5 domyślnie są równe zero a metryk K1 i K3 jeden, to zobaczmy co się stanie ze wzorem:

$$metryka = 256 * (1 * Pasmo + \frac{0 * Pasmo}{256 - Obciążenie} + 1 * Opóźnienie * \frac{0}{Niezawodność + 0})$$

$$metryka = 256 * (Pasmo + 0 + Opóźnienie * 0)$$

Widzimy, że w tym wypadku pod uwagę brane byłoby tylko pasmo. Byłoby to trochę niefunkcjonalne. Na szczęście twórcy protokołu EIGRP pomyśleli o tym i uznali, że przy domyślnych wartościach routingu nie powinniśmy brać pod uwagę ostatniego członu, który ma postać: K5/(Niezawodność+K4). W tym wypadku wzór ostateczny będzie miał postać:

$$metryka = 256 * (Pasmo + Opóźnienie)$$

Do wzoru niestety nie możemy podstawić parametrów bezpośrednio odczytanych z wyniku polecenia show interface. Musimy to odpowiednio przekształcić. Wzory będą następujące:

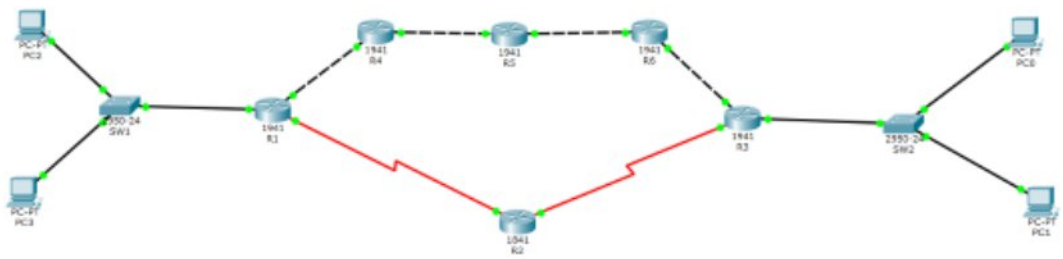
$$Pasmo = \frac{10\,000\,000}{Odczytane\ pasmo\ (\frac{kbit}{s})}$$

$$Opóźnienie = \frac{Suma\ opóźnień\ na\ trasie}{10}$$

Ostateczny wzór mógłby więc wyglądać tak:

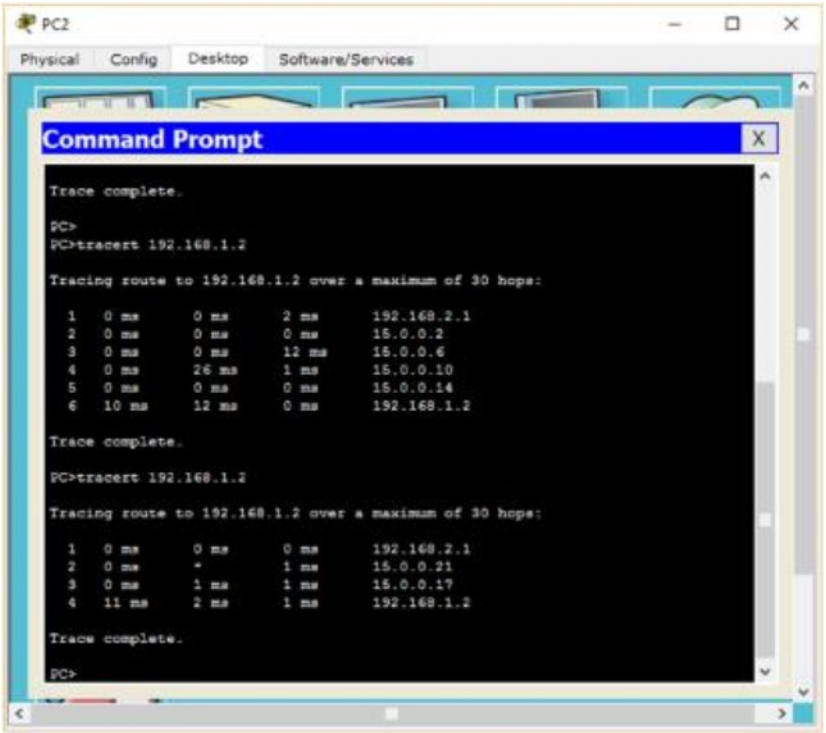
$$metryka = 256 * (\frac{10\,000\,000}{Odczytane\ pasmo\ (\frac{kbit}{s})} + \frac{Suma\ opóźnień\ na\ trasie}{10})$$

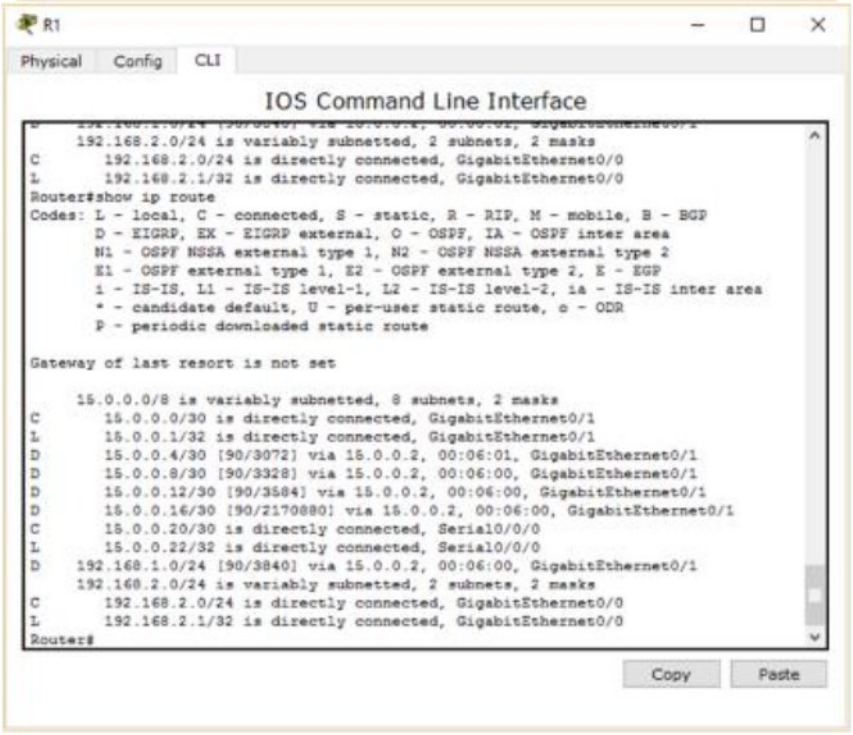
Aby zobaczyć metrykę EIGRP w praktyce, zbudujmy sieć o następującej topologii (możesz pobrać gotowy plik. Link znajduje się tutaj. (plik: przyk1.pkt).



Zwróciłeś pewnie uwagę na nowy typ połączenia. „Czerwone pioruny” to połączenie typu serial – szeregowe. Jego zasada działania podobna jest do RS-232. Szybkość zresztą też.

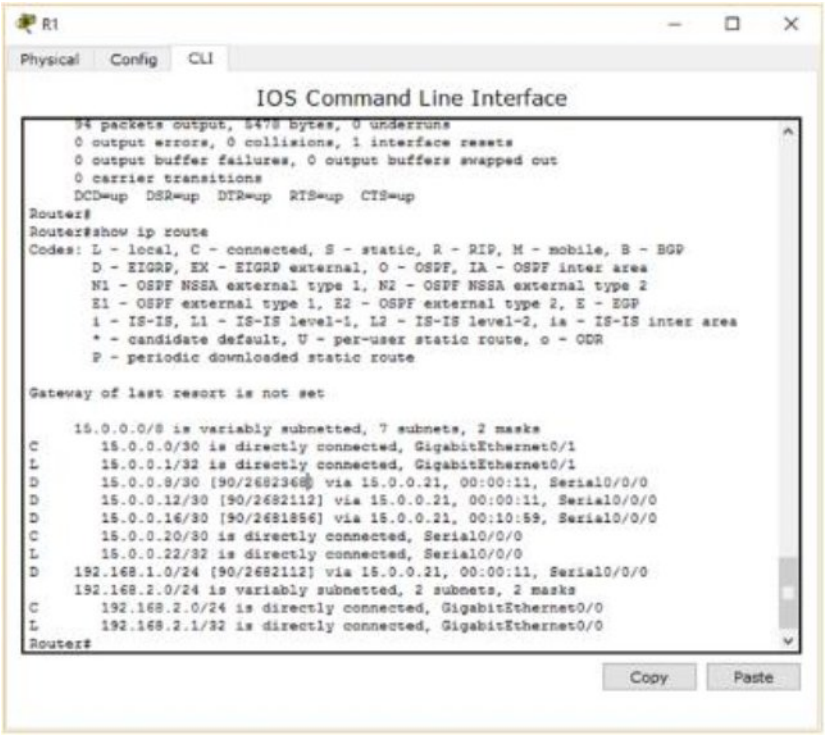
Spójrzmy dokładnie na wcześniej przedstawiony schemat sieci. Załóżmy, że naszym komputerem jest PC2 i chcemy wysłać pakiet do PC0. Istnieją dwie trasy do miejsca docelowego. Pakiet danych może zostać wysłany za pomocą łącza szeregowego przez R2 lub za pomocą Gigabit Ethernet przez R4, R5 i R6. Wydajniejszym rozwiązaniem na pewno będzie skorzystanie z drugiej opcji. W końcu łącze serial o przepustowości 1.5Mbit/s jest 666 razy wolniejsze od Gigabit Ethernet. Ale zastanów się, która opcja byłaby wybrana, gdyby sieć została oparta o protokół RIP? Aby dotrzeć z PC2 do PC0 przez łącze o większej przepustowości, musielibyśmy wykonać aż 6 skoków. Lecz jeśli skorzystamy z połączenia szeregowego będą to jedynie cztery skoki. Widzisz pewnie różnicę? Krótsza trasa nie zawsze oznacza lepszą trasę. Spójrzmy teraz na tablicę routingu R1.





Dystans administracyjny protokołu EIGRP przy trasach systemowych wynosi 90. To znaczy, że jeśli na routerze mielibyśmy skonfigurowany zarówno RIP jak i EIGRP, to preferowaną trasą byłaby ta dostarczana przez EIGRP. Działoby się tak dlatego, gdyż dystans administracyjny protokołu RIP wynosi 120. Jest większy niż dystans EIGRP (90).

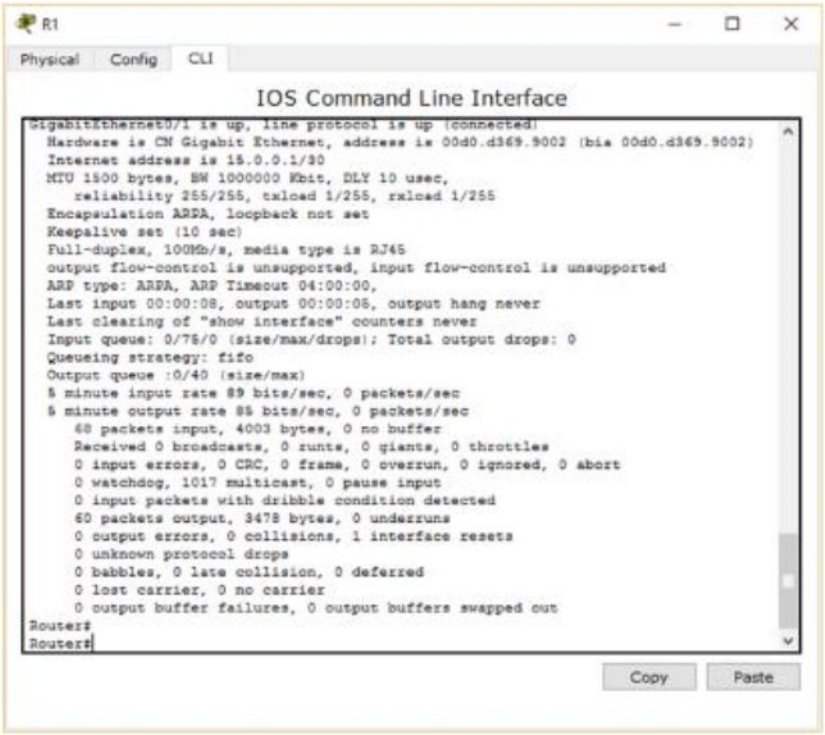
Metryka trasy do sieci 192.168.1.0 wynosi 6144. Zasymulujmy teraz „awarię” na łączu Gigabit Ethernet tak, aby jedyną dostępną opcją była podróż pakietu przez kabel Serial i spójrzmy, jak zmieni się tablica routingu.



Metryka trasy do 192.168.1.0 zwiększyła się masakrycznie! Teraz widzimy tam liczbę 2682212, co oznacza, że druga trasa wg protokołu EIGRP jest 436 razy gorsza niż wcześniejsza.

Poznaliśmy wzory, spróbujmy więc przeliczyć ręcznie te metryki. Zajmijmy się najpierw bardziej optymalną trasą, biegnącą przez łącze Gigabit Ethernet.

Musimy poznać dwa najważniejsze parametry tej trasy: szybkość i opóźnienie. Co do przepustowości, możemy się domyślić, jaka będzie jej wartość. Ale opóźnienia nie odgadniemy. Zresztą, po co się wysilać, skoro odpowiednich informacji może nam dostarczyć sam router? Użyjmy polecenia show interface, o którym wspominałem na samym początku niniejszego artykułu.

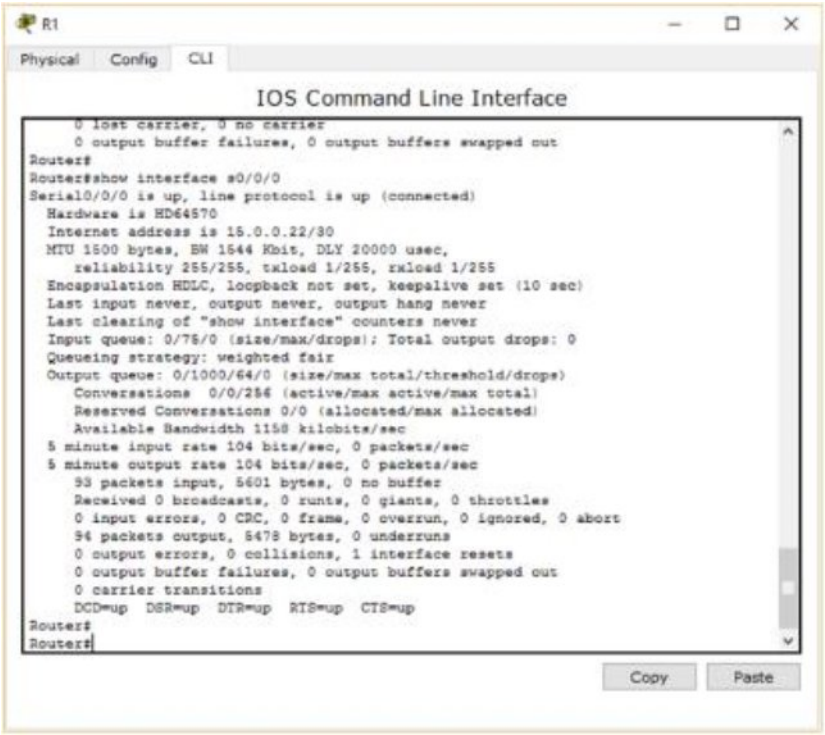


Parametry, na które musimy zwrócić uwagę to BW i DLY. W tym wypadku prędkość interfejsu to 1000000 kbit/s. Natomiast opóźnienie wynosi 10 mikrosekund. Policzmy więc:

$$metryka = 256 * (\frac{10\,000\,000}{1\,000\,000} + \frac{5 * 10}{10})$$

$$metryka = 256 * \left(10 + \frac{15}{10}\right) = 256 * 15 = 3840$$

Pomnożyliśmy 10 przez 5, gdyż mamy 5 skoków (R4->R5->R6->R3->sieć docelowa). Wynik zgadza się z tym, co automatycznie obliczył router, więc poprawnie wyliczyliśmy metrykę.
Obliczmy jeszcze metrykę drugiej, gorszej trasy. Zobaczymy, czy uzyskamy taki sam wynik jak router. Ponownie wydajemy polecenie show interface, tym razem jako argument podając interfejs szeregowy s0/0/0.



Odczytujemy parametry. Opóźnienie wynosi 20000 mikrosekund, czyli jest ponad 2000 razy większe niż w przypadku GigabitEthernet. Prędkość jest za to 650 razy mniejsza i wynosi 1544 kbit/s. Musimy wykonać dwa skoki, aby dostać się do routera R3 i jeszcze jeden, aby dostać się do docelowej sieci. Stąd nasz wzór będzie miał następującą postać:

$$metryka = 256 * (\frac{10\,000\,000}{1544} + \frac{2 * 20000 + 10}{10})$$

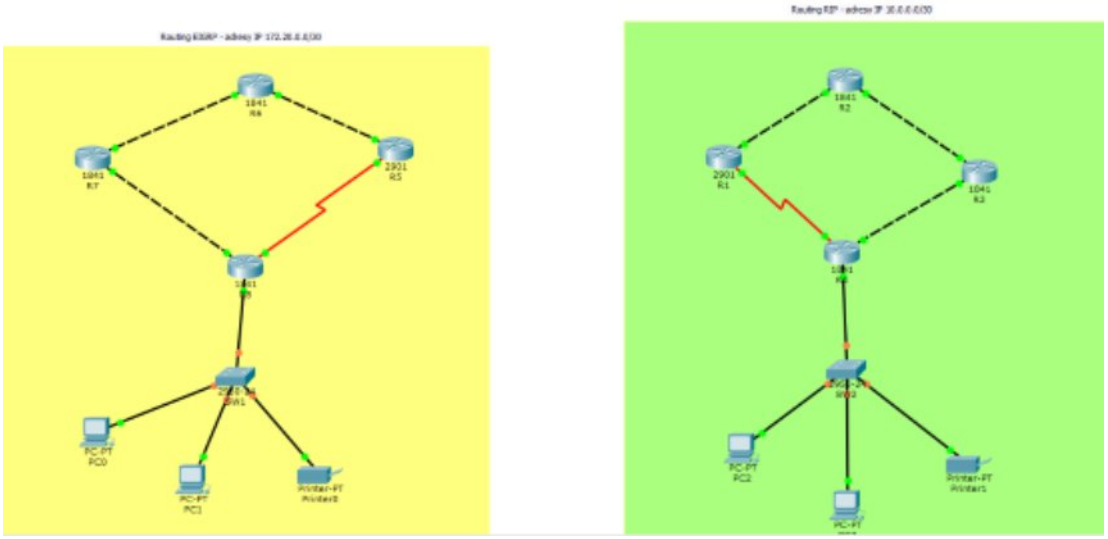
$$metryka = 256 * (6477 + 4000) = 256 * 10477 = 2682112$$

Po raz kolejny otrzymaliśmy poprawny wynik. Oznacza to, że bezbłędnie umiemy obliczać metrykę routingu EIGRP.

Warto jeszcze powiedzieć słówko na temat tej tajemniczej cyfry 256, przez którą mnożymy wynik dodawania. Bierze się to stąd, że EIGRP używa 32-bitowej liczby jako metryki, natomiast IGRP jedynie 24-bitowej. Aby zachować kompatybilność między tymi dwoma protokołami routingu, użyto swoistego „uzupełnienia” 24-bitowej liczby do 32-bitowej.

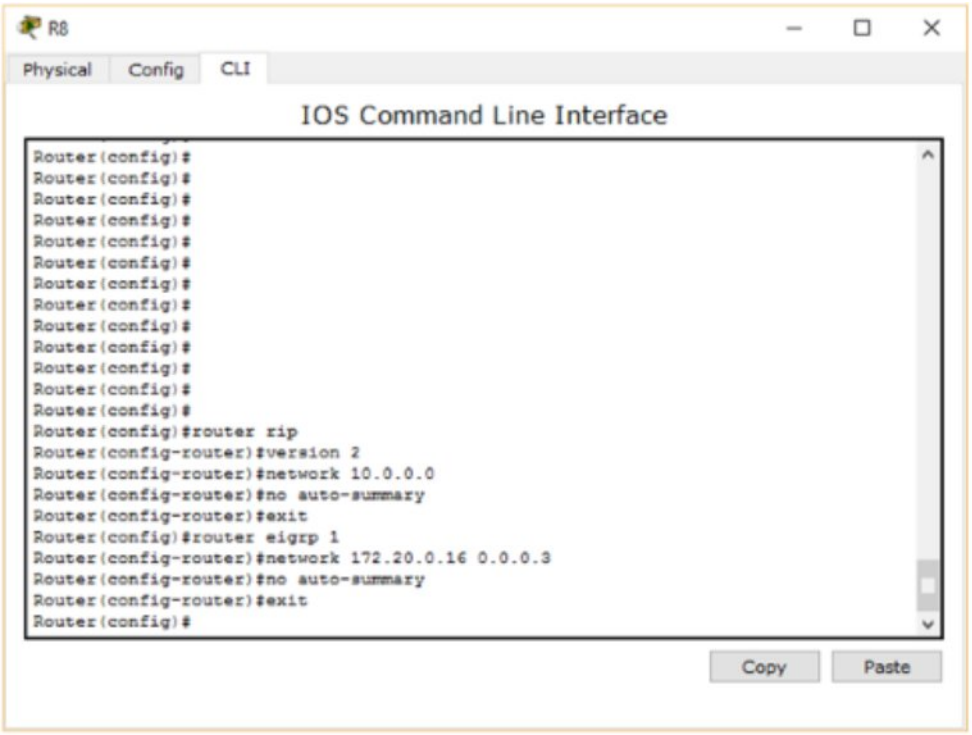
Podstawy redystrybucji – wymiana informacji między różnymi protokołami routingu

Załóżmy, że mamy sieć firmową, dobrze skonfigurowaną, działającą od kilku lat. Chcemy ją połączyć z nowo powstałym oddziałem naszej firmy. Sieć w drugim oddziale oparta jest o protokół EIGRP. Bez konfiguracji redystrybucji nie ma szans, aby te dwie sieci mogły się ze sobą skomunikować. Zobaczmy więc, jak wygląda najprostsza możliwa metoda konfiguracji na tym prostym przykładzie. Topologia jest dostępna do pobrania tutaj. (plik: zad2.pkt).). Spójrzmy na wstępną topologię:



Naszym zadaniem jest połączenie sieci 172.20.0.0/30, w której jest stosowany routing EIGRP i 10.0.0.0/30, w której użyty został RIPv2. Najprostszym zadaniem będzie dołożenie kolejnego routera, który będzie „pośredniczył” w komunikacji między tymi dwoma sieciami. Nazwijmy go R9. Najlepiej, aby był to router 2901. Łączymy interfejs Gig0/0 R4 z Gig0/0 R9 oraz Gig0/1 R9 z Gig0/0 R8. Na R8 i R4 są już ustawione odpowiednie adresy IP. Na R9 Gig0/0 ustawmy adres 172.20.0.18 z maską 255.255.255.252 a na drugim interfejsie 10.0.0.18 z taką samą maską jak wcześniej.

Adresy IP ustawione. Teraz musimy skonfigurować routing w taki sam sposób, jak robiliśmy wcześniej. Różnica polega na tym, że poprzednio na jednym routerze konfigurowaliśmy tylko jeden protokół routingu. Tym razem na R9 musimy skonfigurować aż dwa protokoły: EIGRP i RIP. Poprawną konfigurację zobaczysz na poniższym screenie:



Tym sposobem zarówno z sieci 10.0.0.0/30 jak i 172.20.0.0/30 powinniśmy móc dogadać się z routerem R9. To jednak nie wszystko. R9 dalej nie wie, jak ma „łączyć” ze sobą te dwie sieci. Musimy mu o tym powiedzieć za pomocą nowego polecenia. Tą komendą jest redistribute.

Przejdźmy najpierw do trybu konfiguracji protokołu RIP. Jako pierwszy argument tego polecenia podajemy rodzaj protokołu, którego pakiety chcemy „przerobić” na RIP. Kolejnymi będą numer systemu autonomicznego i metryka. Po co metryka? A no dlatego, gdyż RIP całkowicie inaczej liczy „jakość trasy” niż EIGRP. Router samodzielnie nie przerobi sobie metryki EIGRP na RIP. Musimy sami mu podać, jak ma ją rozumieć. Wydajmy więc następujące polecenie:

```
redistribute eigrp 1 metric 2
```

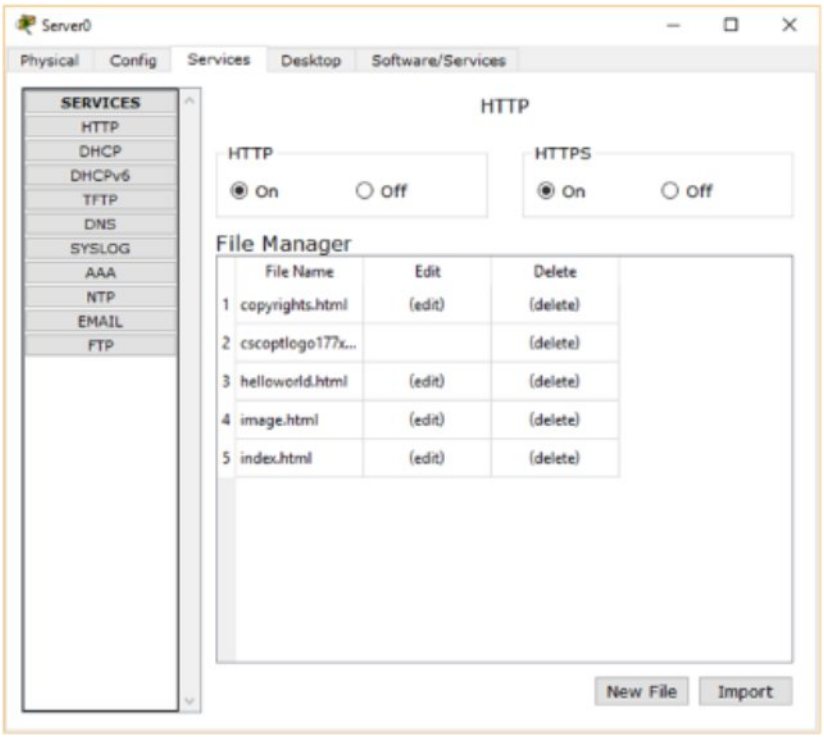
Ostatni parametr (2) oznacza, że potrzeba dwóch skoków do osiągnięcia tej sieci z poziomu routera R9.

Teraz wykonujemy identyczne czynności w konfiguracji protokołu EIGRP. Tym razem, jako protokół, który chcemy „redystrybuować” podajemy oczywiście RIP. Metrykę musimy zapisać w formacie protokołu EIGRP. Przykładowo:

```
redistribute rip metric 1000000000 255 1 1500
```

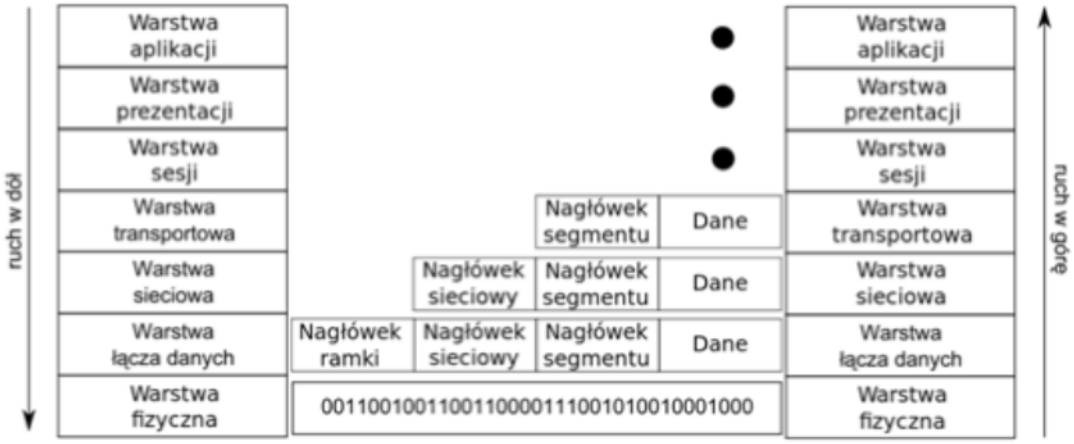
oznacza, że połączenie ma przepustowość 1000000000 kbit/s, jest w 100% dostępne (255) i minimalnie obciążone a MTU (będziemy o tym mówić później) wynosi 1500.
Po tej czynności powinniśmy móc bez problemu wysłać pakiet ICMP z komputera PC2 do komputera np.: PC1.
Gotowa topologia z pełną konfiguracją jest dostępna do pobrania tutaj. (plik: zad2_skoncz.pkt).

Podstawowe protokoły wykorzystywane w sieci Internet



Usługi udostępniane przez serwer w Packet Tracer

Otwórzmy Packet Tracera, ustawmy serwer i po dwukrotnym kliknięciu na nim przejdźmy na zakładkę Services. Właśnie tu możemy konfigurować usługi, które dany komputer ma dostarczać sieci. Jednakże, zanim przejdziemy do omawiania tych protokołów, wróćmy jeszcze na chwilę do modelu OSI, podstawy działania dzisiejszego Internetu.



Model OSI - po raz kolejny ...

Wszystkie protokoły, które widzimy pod napisem Services w Packet Tracer to protokoły warstwy aplikacji. Jak wiemy z wcześniejszych części kursu, są to aplikacje uruchomione na komputerze, które chcą przesyłać dane w sieci. Takie zadanie spełnia każdy z protokołów wymienionych wyżej: HTTP obsługuje żądania pobierania treści stron internetowych, FTP – pobieranie plików itd.

Jednym z najpopularniejszych protokołów niższej warstwy jest TLS – zajmujący się szyfrowaniem danych. TLS-em, SSL i innymi protokołami warstwy prezentacji zajmiemy się w późniejszych częściach kursu.

Warstwa sesji to protokoły służące głównie autoryzacji użytkowników. Najbardziej znanymi są SMB (Server Message Block, wykorzystywany, jeśli udostępniamy/pobieramy pliki w ramach sieci lokalnej systemów Windows). Również on nas na razie nie interesuje.

Doszliśmy wreszcie do warstwy transportowej. To właśnie w tym miejscu dzieje się najwięcej ciekawych rzeczy. Pracuje tutaj dwa protokoły, które są powszechnie używane w każdym zakamarku współczesnego Internetu. Są to Transmission Control Protocol (TCP) oraz User Datagram Protocol. Protokoły te zajmują się podziałem danych otrzymanych z wyższych warstw na części i przygotowaniu ich do wysłania. Jest to bardzo ciekawy proces, któremu warto się przyjrzeć.

W tej części zajmiemy się protokołem TCP. Jest połączeniowy. Cóż to oznacza? Otóż, przed wysłaniem lub odebraniem danych pomiędzy stacją źródłową i docelową musi zostać nawiązane połączenie. TCP gwarantuje, że wszystkie pakiety, które zostaną wysłane przez serwer, klient bezpiecznie i w całości odbierze. W przeciwnym wypadku klient może wysłać „reklamację” (żądanie retransmisji), którą serwer z reguły przyjmuje i retransmituje segmenty. Mimo zagwarantowania, że pakiety na 100% dotrą do celu, to nie jest powiedziane, że dotrą w odpowiedniej kolejności. Klient na podstawie danych zawartych w nagłówku musi je odpowiednio poskładać.

Nagłówek protokołu TCP

<table>
<tr><td>Offset</td><td>Oktet</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Oktet</td><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>0</td><td>0</td><td colspan="16">Port nadawcy</td><td colspan="16">Port odbiorcy</td></tr>
<tr><td>4</td><td>32</td><td colspan="32">Numer sekwencyjny</td></tr>
<tr><td>8</td><td>64</td><td colspan="32">Numer potwierdzenia</td></tr>
<tr><td>12</td><td>96</td><td colspan="4">Długość nagłówka</td><td colspan="6">Zarezerwowane</td><td colspan="6">Flagi</td><td colspan="16">Szerokość okna</td></tr>
<tr><td>16</td><td>128</td><td colspan="16">Suma kontrolna</td><td colspan="16">Wskaźnik priorytetu</td></tr>
<tr><td>20</td><td>160</td><td colspan="32">Opcje</td></tr>
</table>

Wygląda na skomplikowany? To tylko pozory. Dzięki analizie nagłówka wszystko stanie się jasne.

Port nadawcy i port odbiorcy

Pierwsze, co rzuca się w oczy to pola: port nadawcy i port odbiorcy. Cóż to jest ten port? Do tej pory pod pojęciem port rozumieliśmy co najwyżej fizyczny port przełącznika/routera. Jednakże tutaj znaczenie tego słowa jest całkowicie inne. Pole zajmuje 16 bitów, a więc mamy 2 do potęgi 16 kombinacji. Zastanówmy się do czego to może służyć… Otóż, załóżmy, że posiadamy jeden komputer. Chcemy, aby po połączeniu się z nim użytkownik mógł przeglądać stronę internetową. Bardzo proszę, uruchamiamy serwer WWW i wszystko hula. Jednakże, co się stanie, gdybyśmy chcieli na tym samym komputerze uruchomić jeszcze serwer FTP lub nawet uTorrenta? Przecież wszystkie te programy pracują na jednym komputerze i w tym samym czasie odbierają lub wysyłają pakiety! Skąd np.: uTorrent ma wiedzieć, jakie dane powinien odebrać dla siebie, a jakie powinny odczytać inne aplikacje? No właśnie. Aby uniknąć takich konfliktów, wprowadzono pojęcie portu. Jest to 16 bitowa liczba, która określa daną aplikację lub protokół. Przykładowo, serwery stron internetowych zwykle działają na porcie 80. Natomiast, jeśli chcemy pobrać plik przez serwer FTP, korzystamy z portu 21.

Porty od 1 do 1023 są tzw. portami systemowymi. Używane są przez usługi systemowe lub serwery różnych aplikacji, których używamy na co dzień w Internecie. Twórcy aplikacji sieciowych nie powinni używać portów z tego zakresu. Dla nich przeznaczone są liczby większe od 1023.

Numer sekwencyjny

Następnym polem jest numer sekwencyjny. Jego zastosowanie również jest bardzo logiczne. Pojedynczy segment danych, jak dobrze wiemy, ma ograniczoną wielkość. Może zdarzyć się tak, że dane, które chcemy przesłać nie zmieszczą się w jednym segmencie. Lecz, w takim razie w jaki sposób poskładać plik z kilku segmentów po stronie odbiorcy? Poszczególne części plików nie muszą przecież przyjść w takiej kolejności, w jakiej zostały wysłane. Za to właśnie odpowiada numer sekwencyjny. Na początku transmisji ustawiany jest tzw. inicjujący numer sekwencyjny. Strony transmisji umawiają się, że od takiej i takiej cyfry zaczną numerację bajtów w przesyłanych plikach. W kolejnych segmentach numer ten jest powiększany o liczbę bajtów wysłanych w poprzednim segmencie. Żebyśmy lepiej zrozumieli znaczenie tego pola, omówmy następujący przykład. Przyjmijmy, że mamy do wysłania 500 bajtów danych, które udało się upakować w czterech segmentach o rozmiarach kolejno: 92 bajtów, 130 bajtów, 143 bajtów i 135 bajtów. Przyjmijmy również, że strony dogadały się, że numerację zaczynamy od zera. Wtedy pole numer sekwencyjny będzie miało następujące wartości:

- Dla pierwszego segmentu: 0 (gdyż umówiliśmy się, że właśnie od 0 rozpoczynamy transmisję, a jest to właśnie pierwszy właściwy segment).
- Dla drugiego segmentu: 92 (gdyż właśnie na tym bajcie zakończyliśmy odbiór pliku.)
- Dla trzeciego segmentu: 222 (gdyż 92+130=222. Upraszczając, numer sekwencyjny to suma długości wszystkich wcześniejszych segmentów)
- Dla czwartego segmentu: 365.

W ten sposób, nawet jeśli jako pierwszy odbierzemy pakiet trzeci (o numerze sekwencyjnym 222) to będziemy wiedzieli, że należy poczekać, gdyż jest to jakaś część pliku. Gdy odbierzemy wszystkie części, będziemy wiedzieli, jak je uporządkować i jak z tych części z powrotem złożyć działający instrument.

Numer potwierdzenia

Co tam mamy dalej… numer potwierdzenia. Mówiliśmy wcześniej, że protokół TCP gwarantuje dotarcie pakietu do kresu jego podróży. Klient po odebraniu segmentu, powinien powiadomić serwer że dane zostały poprawnie wysłane. W takim wypadku ustawia się flagę ACK. W polu numer potwierdzenia wpisuje się natomiast sumę odebranych dotychczas danych. Dzięki temu serwer wie, które części np.: pliku zostały poprawnie wysłane, a które trzeba retransmitować.

Długość nagłówka

Pole, o którego zastosowaniu mówi sama nazwa. Jednakże, jedna rzecz może być zastanawiająca … Dlaczego to pole ma tylko 4 bity? Według najprostszej interpretacji nagłówek mógłby mieć wtedy zaledwie 2^4=16 bitów/bajtów. To nie jest prawdą.

Pole długość nagłówka określa liczbę słów 32 bitowych, które składają się na cały nagłówek. Więc jeśli nagłówek będzie miał 160 bitów (20 bajtów), to w tym polu ujrzymy cyfrę 5. W ten sposób możemy też obliczyć maksymalny rozmiar nagłówka segmentu TCP, który wynosi 2^4-1=16-1=15 trzydziesto-dwu bitowych słów. Maksymalny rozmiar nagłówka wynosi więc 15*32=480 bitów, czyli 60 bajtów. Dla informacji, nagłówek nie może mieć mniej niż 5 słów, co zresztą doskonale widać na załączonym schemacie.

Zarezerwowane

Pole zostawione dla przyszłych zastosowań. Domyślnie wypełnione jest samymi zerami. W nowszych implementacjach pełni pewną rolę (konkretnie

chodzi o przeciążenia) ale w naszych obecnych rozważaniach możemy bezpiecznie przyjąć, że ta część nagłówka nie pełni żadnej roli.

Flagi

Bardzo ciekawe pole. Flagi częściowo definiują, jak należy interpretować dany pakiet. Przykładowo, pakiety oznaczone flagą SYN służą do tworzenia połączenia, a pakiety oznaczone ACK to pakiety potwierdzenia. Te dwa rodzaje flag były wspominane we wcześniejszej części tego artykułu. Jednakże, to nie wszystko, co TCP ma w tym miejscu do zaoferowania:

- URG – pakiet oznaczony tą flagą musi być natychmiast przetworzony. Host docelowy może nawet zawiesić aktualnie wykonywane czynności w celu przetworzenia danych znajdujących się w pakiecie oznaczonych flagą Urgent.
- ACK – flaga, którą omówiliśmy już wcześniej. Oznacza, że segment jest potwierdzeniem poprawnie przesłanych danych. Wskazuje także na użycie pola numer potwierdzenia.
- PSH – dane, które aplikacja chce wysłać zwykle gromadzone są w buforze. Dzięki temu można optymalniej wykorzystać pasmo, wysyłając większą ilość danych w jednym segmencie. Nie zawsze jest to korzystne. Np.: w przypadku telnetu czy ssh chcemy, aby polecenie zostało natychmiastowo przesłane i wykonane przez zdalną maszynę. Właśnie w takich wypadkach używana jest flaga PSH. Powoduje ona natychmiastowe opróżnienie bufora i wysłanie danych.
- RST – rzadko używana. Mówi o konieczności zresetowania połączenia
- SYN – mówiliśmy o niej we wcześniejszej części artykułu. Wykorzystywana przy nawiązywaniu połączenia. Numer umieszczony w polu numer sekwencyjny, w segmencie oznaczonym flagą SYN to inicjujący numer sekwencyjny.
- FIN – flaga wykorzystywana przy zakończeniu połączenia. Informuje, że nadawca nie zamierza już wysyłać więcej danych.

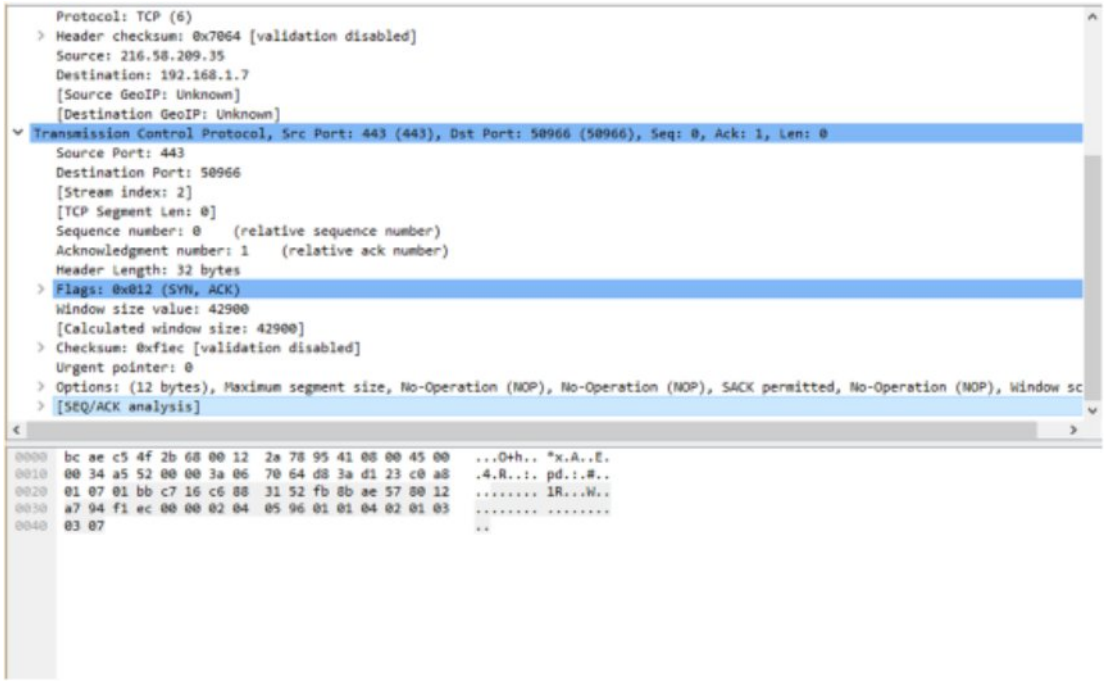
Szerokość okna

Powoli zbliżamy się do końca. Dotarliśmy do enigmatycznie brzmiącej „szerokości okna”. Cóż to oznacza? Jest to maksymalna ilość bajtów danych, jakie może odczytać odbiorca. Zapobiega to zbyt dużemu napływowi danych. Jeśli klient nie może sobie poradzić z natłokiem zer i jedynek, w pakiecie z flagą ACK ustalana jest wartość zero. Oznacza to, że strona wysyłająca powinna się na jakiś czas wstrzymać z wysyłaniem kolejnych danych.

Suma kontrolna

Powiedzieliśmy już sobie, że protokół TCP gwarantuje dotarcie bezbłędnych danych do komputera docelowego. Jeśli dane są błędne, możemy poprosić o retransmisję. Lecz, skąd mamy wiedzieć, czy rzeczywiście w przesłanych danych kryje się jakaś nieprawidłowość? Za to właśnie odpowiada suma kontrolna. Zmiana choćby jednego bitu nagłówka lub danych pociąga za sobą zmianę sumy kontrolnej, co pozwala wykryć że coś po drodze poszło nie tak. Na podstawie sumy kontrolnej zawartej w nagłówku TCP nie naprawimy segmentu. Nie ma takiej możliwości. Możemy jedynie zdefiniować czy dane są uszkodzone czy nie.

Suma kontrolna składa się z 16 bitów. Liczona jest dość prosto. Obliczymy sumę kontrolną dla następującego pakietu:



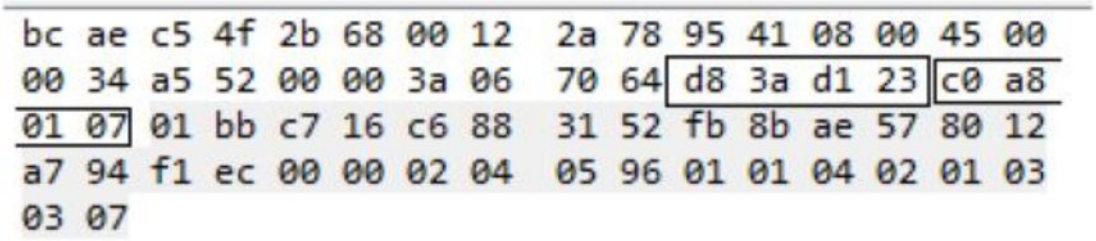
Przykładowy pakiet

Przede wszystkim musimy wziąć pod uwagę jedną bardzo ważną rzecz. W procesie liczenia sumy kontrolnej biorą udział dane nie tylko z nagłówka TCP. Komputer, w trakcie wykonywania tego procesu, tworzy sobie tymczasowo w pamięci tzw. pseudo nagłówek, którego schemat znajduje się poniżej:

0	4	8	12	16	20	24	28	32
Adres źródłowy IP (z nagłówka IP)								
Adres docelowy IP (z nagłówka IP)								
Zarezerwowane (zawsze zera)			Protokół (z nagłówka IP)		Długość segmentu TCP (liczona w locie)			

Pseudonagłówek TCP

Dzięki temu zabiegowi nawet uszkodzenie nagłówka IP zostanie wychwycone w sumie kontrolnej TCP. Liczenie sumy kontrolnej musimy więc zacząć od skonstruowania pseudo nagłówka. Wszystkie dane, które będą brały udział w operacji dzielimy na 16 bitowe bloki, które będą zapisane w systemie szesnastkowym. Jeśli czegoś do końca nie rozumiesz, wszystko wyjaśni się na poniższym przykładzie.



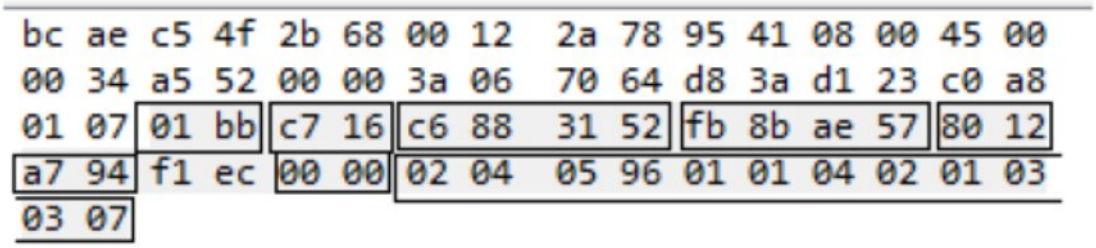
Dane, które pobieramy do pseudonagłówka - adres źródłowy i docelowy

Dana	Zapis szesnastkowy
Adres źródłowy IP	D83A
	D123
Adres docelowy IP	C0A8
	0107
Zarezerwowane + protokół[1]	0006
Długość segmentu TCP[2]	0020

Ad.1 – zauważ, że zarówno pole zarezerwowane jak i protokół mają po 8 bitów. Aby policzyć sumę kontrolną, wszystkie „słowa” muszą być 16-bitowe. Dlatego też połączyliśmy te dwa pola w zapisie w jedno.

Ad. 2 – Długość segmentu TCP musimy policzyć sami. W naszym segmencie nie są zapisane żadne dane, gdyż jest to pakiet z flagą synchronizacji i potwierdzenia. Wobec tego pod uwagę będziemy brali jedynie długość nagłówka TCP, która wynosi 32 bajty (spójrz na pozycję Header Length). Jeśli nasz segment przenosiłby jakieś dane, musielibyśmy do długości nagłówka doliczyć także długość danych.

Przeprowadźmy teraz zwykłe dodawanie. Należy zsumować wszystkie wiersze zapisane w powyższej tabeli, czyli D83A+D123+C0A8+0107+0006+0020. Wynik, który otrzymamy, będzie równy 26B32. Zapamiętajmy go.
Teraz przejdźmy do analizy samego nagłówka TCP. Podobnie jak wyżej, sporządźmy sobie tabelkę, którą skrzętnie wypełnimy danymi:



Dane, które potrzebne nam są do obliczenia sumy - nagłówek protokołu TCP

Dana	Zapis szesnastkowy
Port źródłowy	01BB
Port docelowy	C716
Numer sekwencyjny	C688
	3152
Numer potwierdzenia	FB8B
	AE57
Flagi	8012
Rozmiar okna	A794
Suma kontrolna[3]	0000
Wskaźnik pilności	0000
Opcje	0204
	0596
	0101
	0402
	0103
	0307
Dane	0000 (brak)

Ad. 3. Suma kontrolna jest obecna w nagłówku TCP, dlatego dałem ją w tabelce. Jednakże, w procesie jej pierwszego obliczania pole to przyjmuje wartość 0000.

Sumujemy przepisane do tabelki wartości. Wynik, jaki powinniśmy otrzymać, to: 4A2DA. Jesteśmy już bardzo blisko celu. Teraz musimy dodać ten wynik do otrzymanego wcześniej 26B32. Powinniśmy otrzymać 70E0C. Zastanówmy się, czy wszystko poszło tak jak trzeba? Wynik otrzymany przez nas jest coś za duży, czyż nie? To dlatego, że na samym wyniku musimy przeprowadzić jeszcze kilka operacji. Jeśli otrzymamy liczbę, która zajmuje więcej niż 16 bitów (4 pozycje w zapisie szesnastkowym) to pierwszą cyfrę usuwamy i dodajemy do otrzymanego wyniku. W naszym przypadku efekt obliczeń zostanie przekształcony następująco: 70E0C->E0C+7=E13.

Ostatni krok to przeprowadzenie operacji NOT. Na czym to polega? Po prostu w zapisie binarnym E13 musimy zamienić wszystkie zera na jedynki, a jedynki na zera. Zróbmy to, dla uproszczenia, w tabelce.

HEX	0				E				1				3			
BIN	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	1
(NOT)BIN	1	1	1	1	0	0	0	1	1	1	1	0	1	1	0	0
(NOT)HEX	F				1				E				C			

Voila! Otrzymaliśmy poprawny wynik. Właśnie taka suma kontrolna została zapisana w nagłówku segmentu TCP, nad którym pracowaliśmy. W podobny sposób przebiega sprawdzanie poprawności pakietu. Oblicza się sumę kontrolną, a potem porównuje z tą zapisaną w nagłówku. To jest najtrudniejszy element dzisiejszego artykułu. Dalej będzie tylko prościej :)

Wskaźnik priorytetu

Pole to oznacza koniec pilnych danych. W polu zawarty jest numer pierwszego bajtu następującego po pilnych danych (liczone względem numeru sekwencyjnego).

Opcje

Wykorzystywane najczęściej do przesłania dodatkowych informacji na temat połączenia. Parametrów, które można tu wpisać jest dość dużo. Pełna lista dostępna jest na stronie IANA. (http://www.iana.org/assignments/tcp-parameters/tcp-parameters.xhtml#tc...). Omówimy kilka najczęściej używanych:

- MSS – maksymalny rozmiar segmentu – opcja używana w trakcie nawiązywania połączenia. Przekazuje maksymalny rozmiar segmentu, jaki host może przyjąć.
- NAK – Potwierdzenie negatywne – ciekawa nazwa i ciekawa opcja. Przyjmijmy, że mamy 5 segmentów, które składają się na pojedynczy pakiet. Załóżmy, że odebraliśmy 1,2,3 oraz 5 segment. Brakuje nam jednej części, więc nie możemy złożyć całego pakietu w jedną całość. Nie możemy też poprosić o retransmisję tylko jednego segmentu, gdyż nie pozwala na to standardowa polityka TCP. W przypadku dużej ilości takich błędów generujemy spore dodatkowe obciążenie. Jeśli obydwie strony zadeklarują możliwość korzystania z NAK, to strona może poprosić o retransmisję jedynie brakującego fragmentu.
- SACK – Zastosowanie podobne do NAK. Z tym że tym razem odbiorca informuje, które segmenty udało się odebrać poprawnie. Żeby korzystać z tej funkcjonalności, najpierw obydwie strony komunikacji muszą wynegocjować w trakcie inicjacji połączenia możliwość korzystania z SACK.

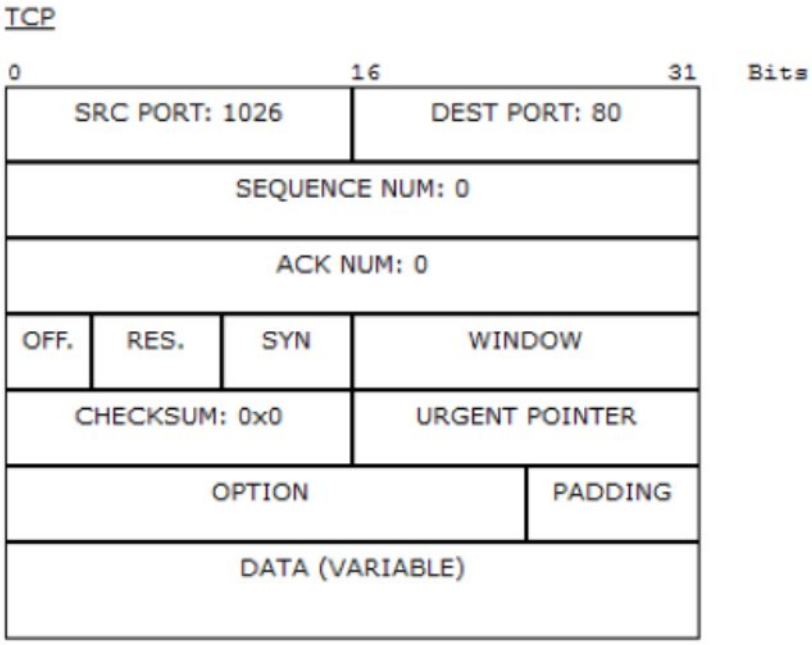
Nawiązywanie połączenia TCP

Protokół TCP jest protokołem połączeniowym. Wobec tego wymaga, aby ustalić połączenie oraz wszystkie jego parametry zanim zacznie się właściwa transmisja. Proces ten możemy zaobserwować w programie Packet Tracer. Stwórzmy prostą sieć, złożoną z klienta i serwera. Ustaw serwer (Server-PT) oraz klienta (PC-PT). Połącz je kablem skrosowanym. Adresy IP ustaw takie, jakie chcesz. Reguły poznałeś już w poprzednich artykułach.

Przejdź w tryb symulacji (przełączanie trybu pracy – między simulation a realtime znajduje się w dolnej prawej części ekranu.). Wejdź teraz na komputer klienta. W zakładce Desktop uruchom przeglądarkę internetową (Web Browser). W polu URL wpisz adres IP serwera.

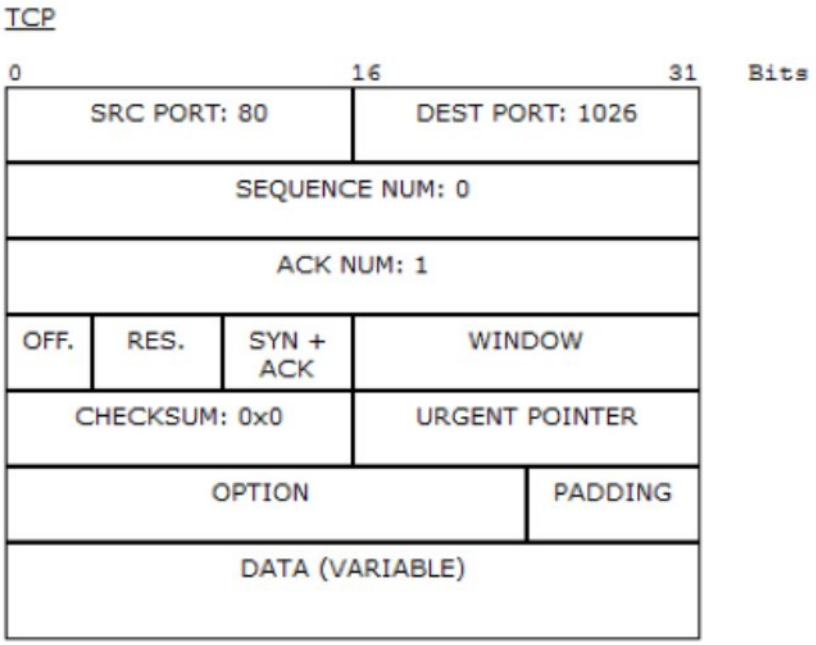
Przy PC0 pojawiła się koperta (najprawdopodobniej zielona). Kliknij na nią i przejdź na zakładkę Outbound PDU Details.

Klient nawiązuje połączenie



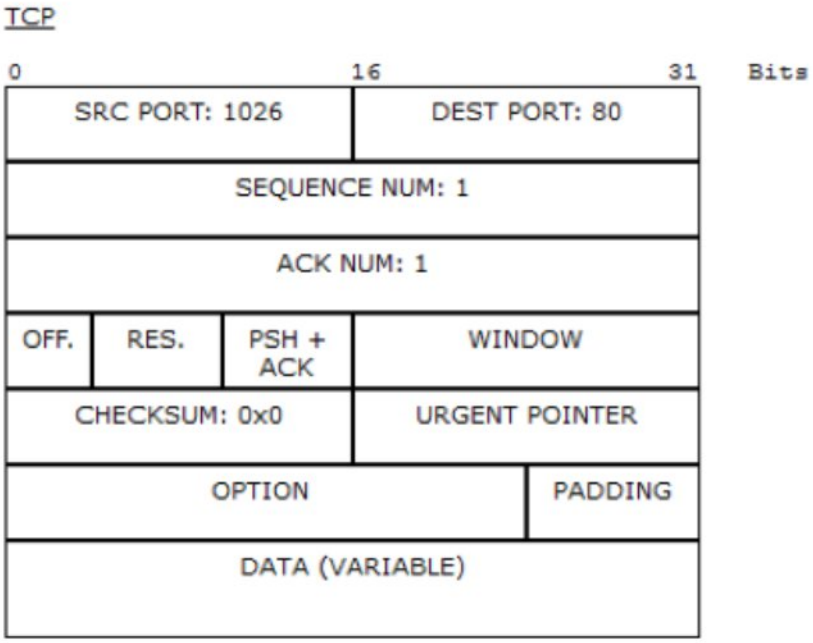
To jest pierwszy pakiet, który klient wysyła do serwera, chcąc nawiązać połączenie. Co my tu widzimy ciekawego? Przede wszystkim, ustalona jest flaga SYN. Klient informuje także, od jakiej cyfry rozpocznie numerację swoich segmentów (SEQUENCE NUM: 0). Zobaczmy, jaka będzie odpowiedź serwera.

Serwer akceptuje połączenie



Klient odpowiada na żądanie serwera. W pakiecie ustalone są dwie flagi: SYN – co oznacza, że ten pakiet służy do nawiązania połączenia oraz ACK – która informuje klienta, że pakiet wysłany przez niego został poprawnie odebrany. W pozostałych polach znajdują się dane. Podobnie jak wcześniej, SEQUENCE NUM zawiera numer, od którego rozpocznie się numerowanie danych.

Klient akceptuje warunki i rozpoczyna komunikację



W tym segmencie znajdują się już właściwe dane (w tym wypadku http). Numer sekwencyjny oraz numer potwierdzenia ustawione są na jeden. Ustawiona flaga ACK powoduje, że pakiet, oprócz przekazania danych ma uświadomić serwerowi, że pakiet wysłany przez niego został poprawnie odebrany.

Dalsza komunikacja -> zakończenie połączenia

Dalsza komunikacja przebiega już na prostej zasadzie – klient wysyła dane->serwer odbiera dane i wysyła potwierdzenie (flaga ACK) do klienta. Jednakże, co się dzieje w wypadku, gdy jedna lub obie strony chcą zakończyć pogawędkę?

Przyjmijmy, że najpierw klient chce zakończyć połączenie. Wysyła wtedy do serwera pakiet z flagą FIN. Warto zauważyć, że nawet ostatni pakiet może zawierać dane dla serwera. Serwer, po odebraniu i przeanalizowaniu pakietu odsyła do klienta segment TCP z flagą ACK. Warto zauważyć, że strona, która nie zgłosiła chęci zakończenia połączenia (w tym wypadku serwer) może dalej do drugiego rozmówcy wysyłać swoje wiadomości. Druga strona może oczywiście zakończyć połączenie w ten sam sposób, jaki opisany jest wyżej.

(na podstawie http://www.dobreprogramy.pl/karol221-10/Packet-Tracer-6.2-od-zera-do-tworzenia-sieci-cz.-0,68812.html – BLOG Karol221-10)