

TECHNIKI ROUTINGU W SIECIACH KOMPUTEROWYCH

ACL, NAT, PAT, DHCP

opracowanie na podstawie materiałów Cisco

Marcin Raniszewski

Roman Krzeszewski

Łukasz Sturgulewski

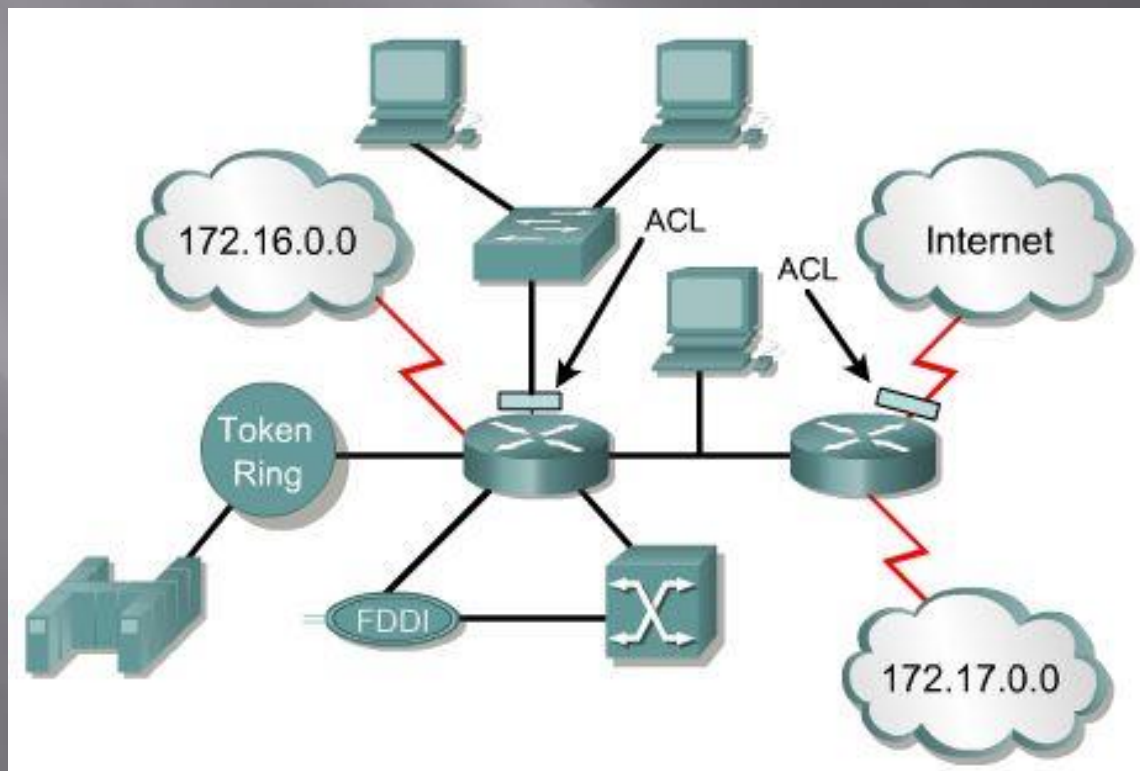
Grzegorz Nowak

Plan wykładu

- ▣ ACL (*Access Control List*)
- ▣ Funkcje maski blankietowej
- ▣ Standardowe listy ACL
- ▣ Rozszerzone listy ACL
- ▣ Rozmieszczanie list ACL
- ▣ Weryfikowanie list ACL
- ▣ NAT i PAT
- ▣ NAT - konfiguracja - translacja statyczna
- ▣ NAT i PAT- konfiguracja - translacja dynamiczna
- ▣ Sprawdzanie konfiguracji mechanizmów NAT i PAT
- ▣ DHCP (*Dynamic Host Configuration Protocol*)
- ▣ Dynamiczne konfigurowanie adresów IP routera za pomocą DHCP
- ▣ Dynamiczne przydzielanie adresów IP klientom za pomocą DHCP
- ▣ Sprawdzanie działania usługi DHCP

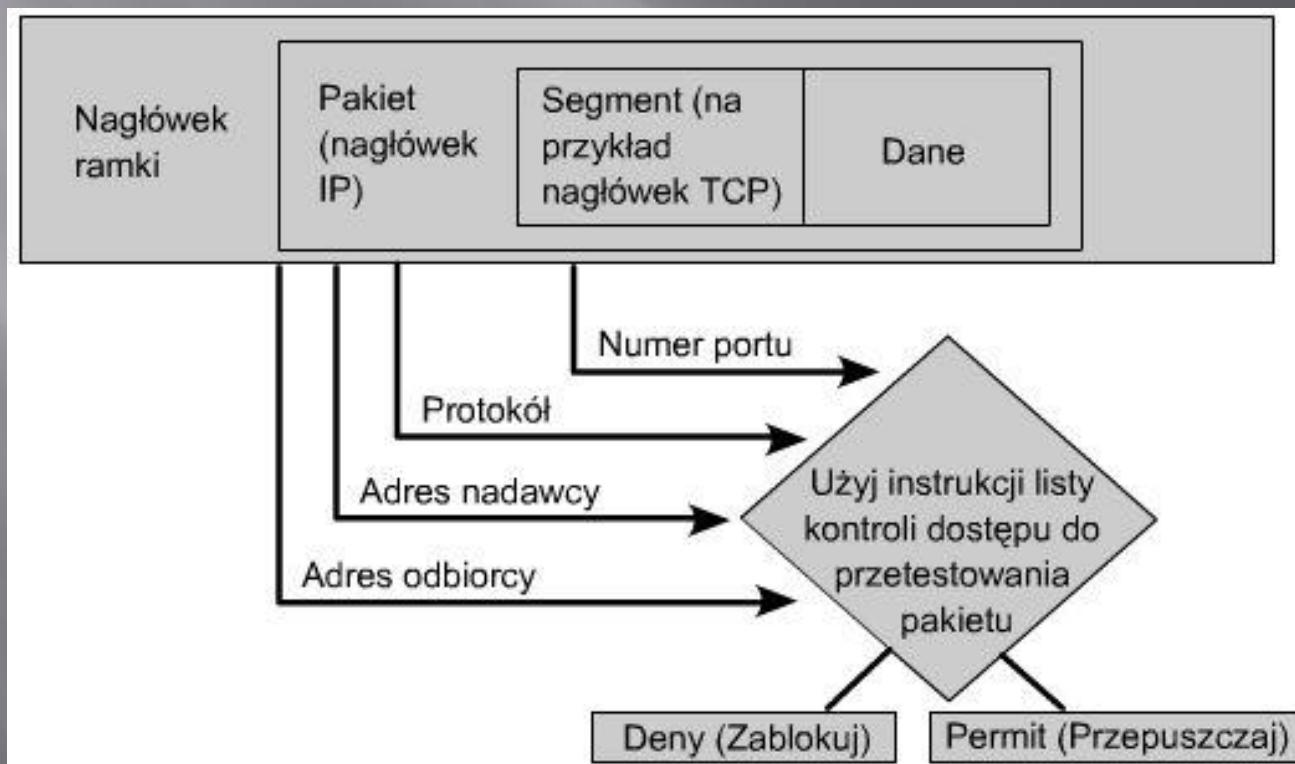
ACL (Access Control List)

- Listy ACL są **listami warunków** używanych do sprawdzania ruchu w sieci, który jest kierowany przez **interfejs** routera.
- Listy te informują router, jakie rodzaje pakietów **zaakceptować**, a jakie **odrzuć**.
- Akceptacja i odrzucenie zależą od spełnienia konkretnych warunków.
- Listy ACL umożliwiają zarządzanie ruchem oraz bezpieczny **dostęp do i z sieci**.
- Mogą one zostać utworzone **dla różnych sieciowych protokołów routowanych**.



ACL (Access Control List)

- Aby możliwe było filtrowanie ruchu, listy ACL muszą określić, czy **pakiety** mają być **przekazywane**, czy **blokowane** na interfejsach routera.
- Router sprawdza każdy pakiet i albo go przekaże, albo odrzuci, w zależności od warunków określonych na liście ACL.
- Lista ACL umożliwia podejmowanie decyzji o routingu **na podstawie adresu źródłowego, adresów docelowych, protokołów** oraz **numerów portów** wyższych warstw.



ACL (Access Control List)

- ▣ Listy ACL muszą być definiowane osobno dla każdego **protokołu**, **kierunku** oraz **interfejsu**.
- ▣ Na każdym interfejsie można zdefiniować dwa kierunki i wiele protokołów.
- ▣ Jeśli router ma dwa interfejsy skonfigurowane dla protokołów IP, AppleTalk i IPX, potrzebnych będzie 12 oddzielnych list ACL. Dla każdego protokołu powstanie jedna lista, co należy pomnożyć przez dwa dla każdego kierunku i jeszcze raz przez dwa dla każdego interfejsu.



ACL (Access Control List)

- ▣ Listy ACL mogą wykonywać następujące zadania:
 - Ograniczenie ruchu w sieci i zwiększenie wydajności sieci;
 - Umożliwienie kontroli ruchu w sieci;
 - Zapewnienie podstawowych zabezpieczeń podczas dostępu do sieci. Listy ACL mogą umożliwić jednemu hostowi dostęp do części sieci, uniemożliwiając jednocześnie dostęp do tej samej części innemu hostowi;
 - Decydowanie o typie ruchu przenoszonego lub blokowanego na poziomie interfejsów routera. Listy ACL mogą zezwalać na routing ruchu pocztowego (e-mail), ale blokować ruch Telnet;
 - Określanie, do których obszarów sieci może mieć dostęp użytkownik;
 - Klasyfikowanie ruchu z hostów w celu udostępnienia lub zakazu dostępu do segmentu sieci. Listy ACL mogą być wykorzystywane do umożliwiania lub zakazywania dostępu użytkownika do plików przy użyciu protokołów FTP lub HTTP.

ACL (*Access Control List*)

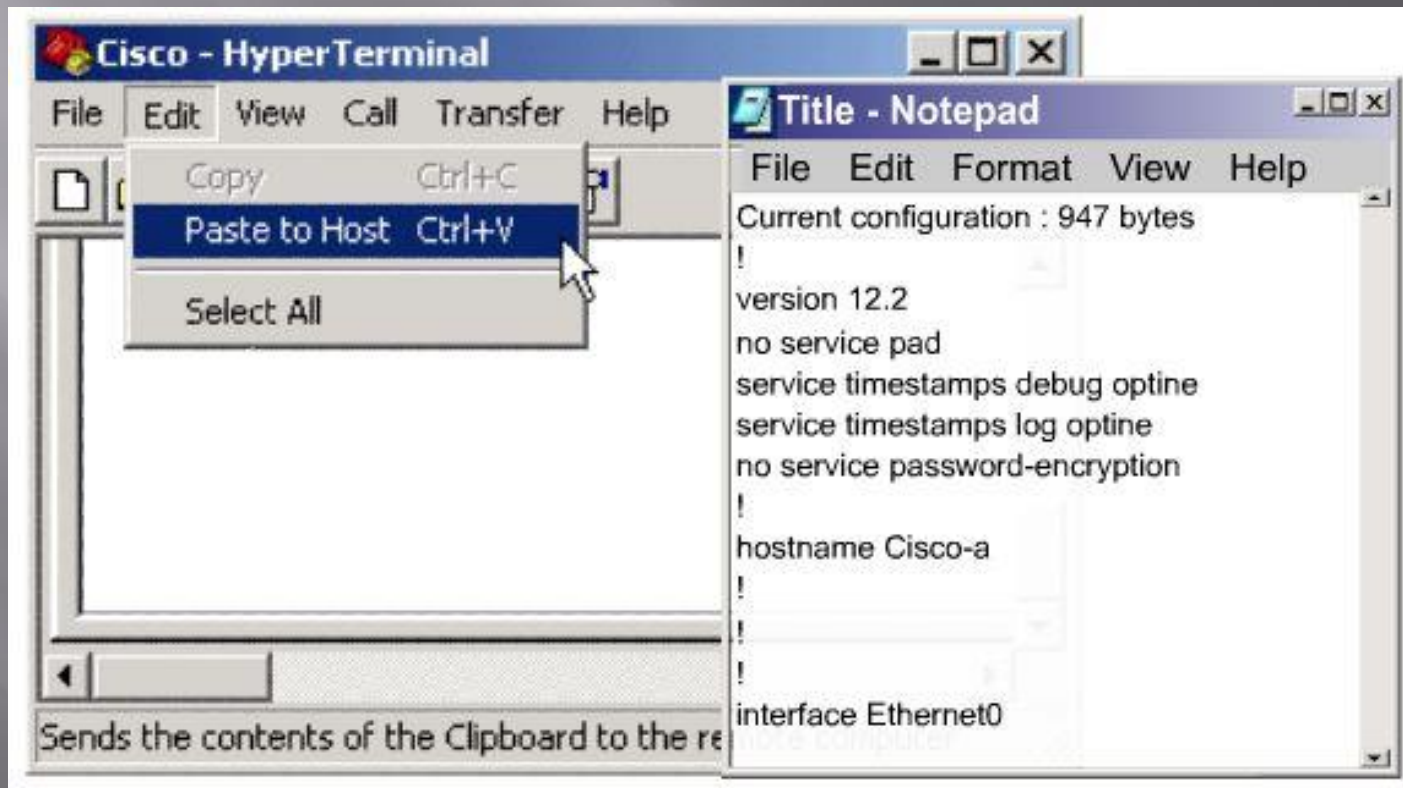
- ▣ Lista ACL składa się z instrukcji, które określają, czy przychodzące lub wychodzące na dane interfejsy pakiety są **akceptowane**, czy **odrzućane**.
- ▣ **Kolejność instrukcji na liście ACL jest istotna**. System Cisco IOS sprawdza pakiet, porównując go z każdą z instrukcji warunkowych w kolejności od początku do końca listy.

Po znalezieniu na liście pasującej pozycji jest przeprowadzana czynność akceptacji lub odrzucenia i nie są wykonywane żadne inne instrukcje z listy ACL.

Jeśli na początku listy znajduje się instrukcja warunkowa dopuszczająca cały ruch, żadna z instrukcji znajdujących się niżej nie będzie nigdy sprawdzana.

ACL (Access Control List)

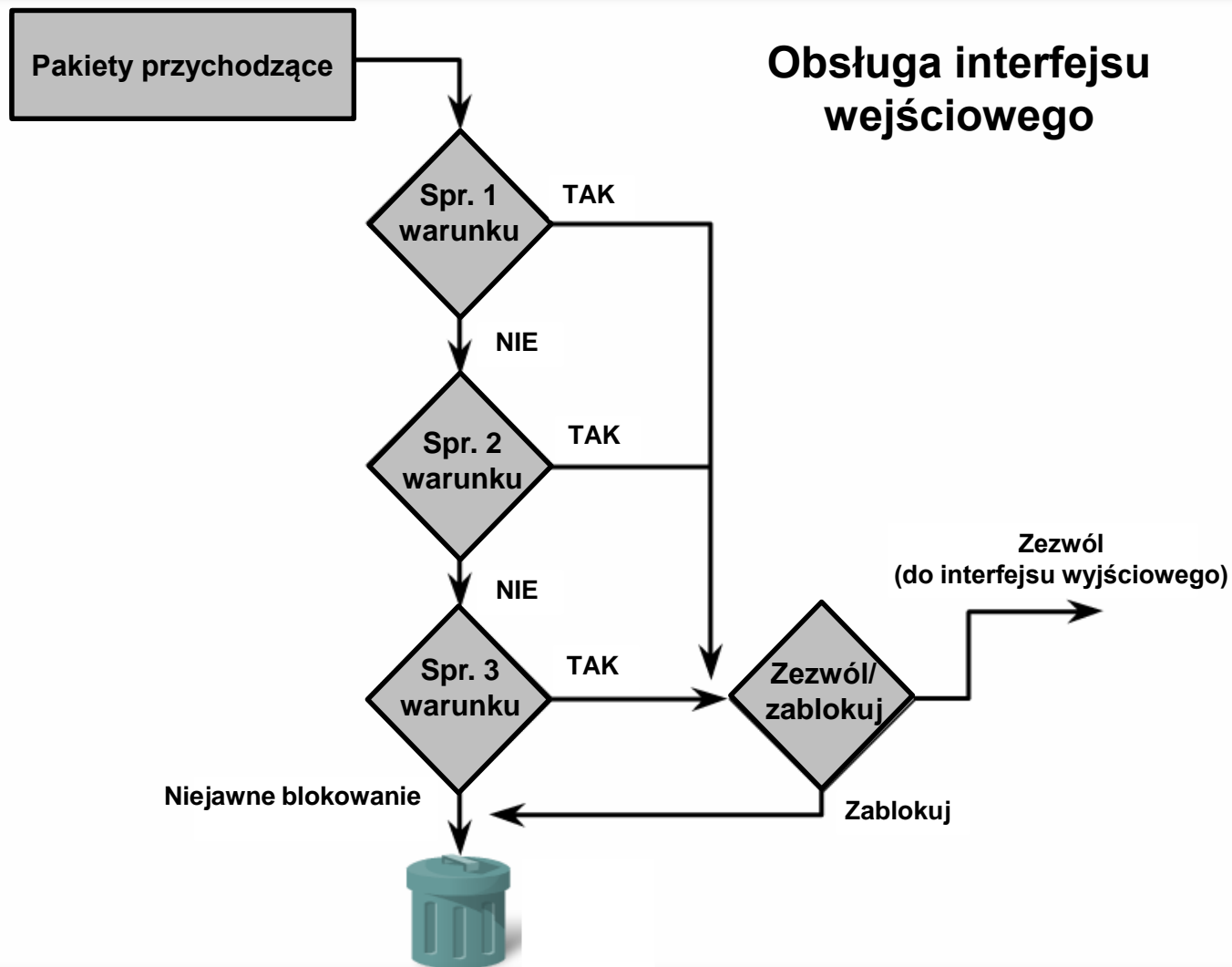
- ❑ Jeśli niezbędne jest dodanie nowych instrukcji warunkowych do listy kontroli dostępu, należy **usunąć całą listę i utworzyć ją na nowo**, tym razem z nową instrukcją warunku.
- ❑ Aby ułatwić modyfikację listy ACL, dobrze jest **użyć edytora tekstu** i wkleić listę do konfiguracji routera.



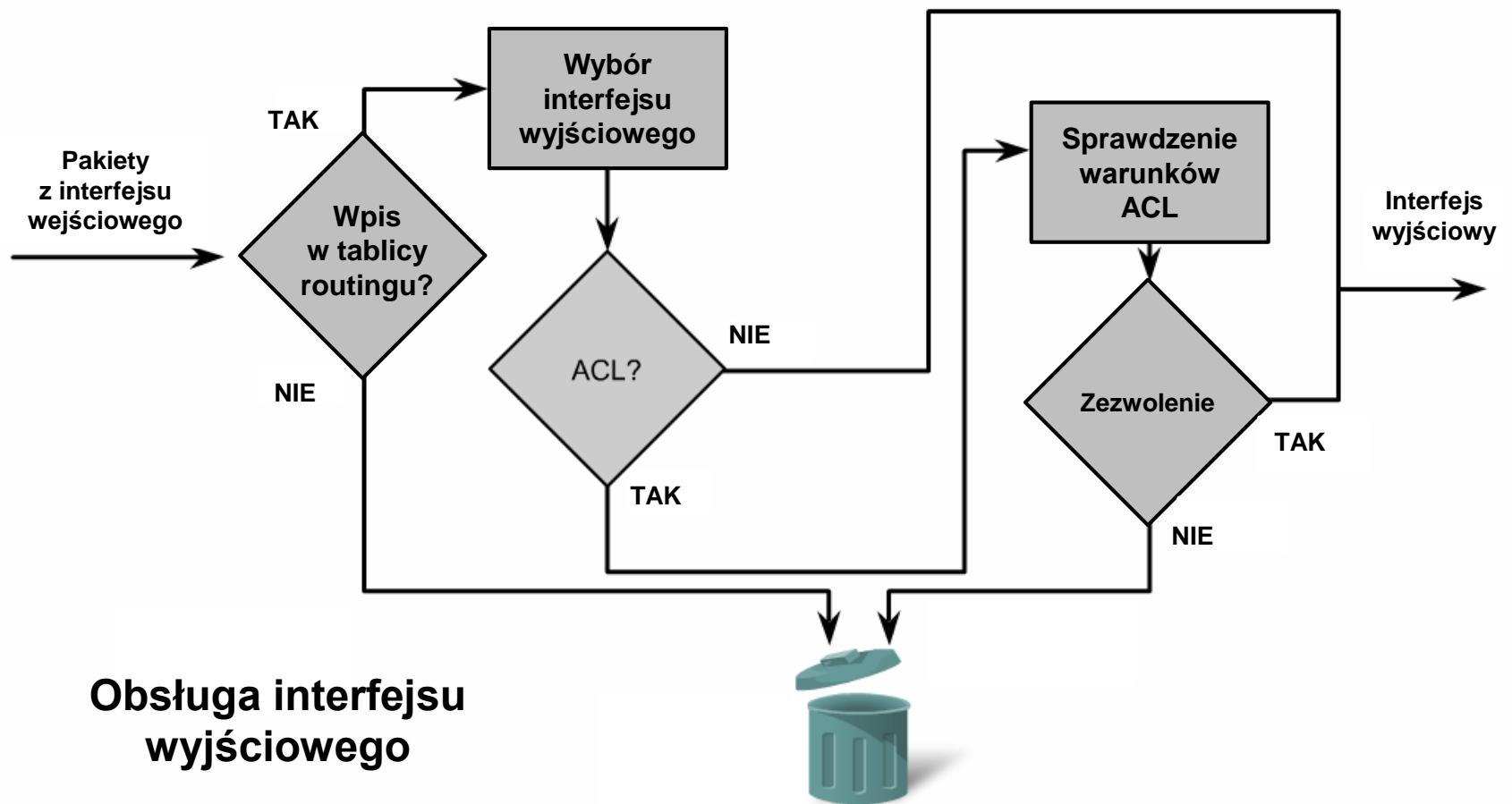
ACL (Access Control List)

- ▣ Proces działania routera rozpoczyna się tak samo, niezależnie od tego, czy używane są listy ACL, czy też nie.
- ▣ Po nadejściu ramki do interfejsu, router sprawdza, czy adres warstwy 2 odpowiada jego adresowi lub czy ramka jest ramką rozgłoszeniową.
- ▣ Jeśli adres ramki został zaakceptowany, informacja o ramce jest usuwana, a router sprawdza, czy istnieje lista ACL interfejsu wejściowego.
- ▣ Jeśli lista ACL istnieje, dla pakietu sprawdzane są instrukcje w niej zawarte. Pakiet, który spełnia warunek instrukcji, zostaje zaakceptowany lub odrzucony.
- ▣ Jeśli pakiet został zaakceptowany na interfejsie, sprawdzone dla niego zostaną pozycje tablicy routingu, aby określić interfejs docelowy i przenieść tam pakiet.
- ▣ Następnie router sprawdza, czy interfejs docelowy zawiera listę ACL. Jeśli lista ACL istnieje, dla pakietu sprawdzane są jej instrukcje. Jeśli pakiet spełnia warunek instrukcji, wykonywana jest czynność akceptacji lub odrzucenia.
- ▣ Jeśli nie ma listy ACL lub pakiet został zaakceptowany, zostanie on poddany enkapsulacji w nową ramkę warstwy 2 i przekazany przez interfejs do następnego urządzenia.

ACL (Access Control List)



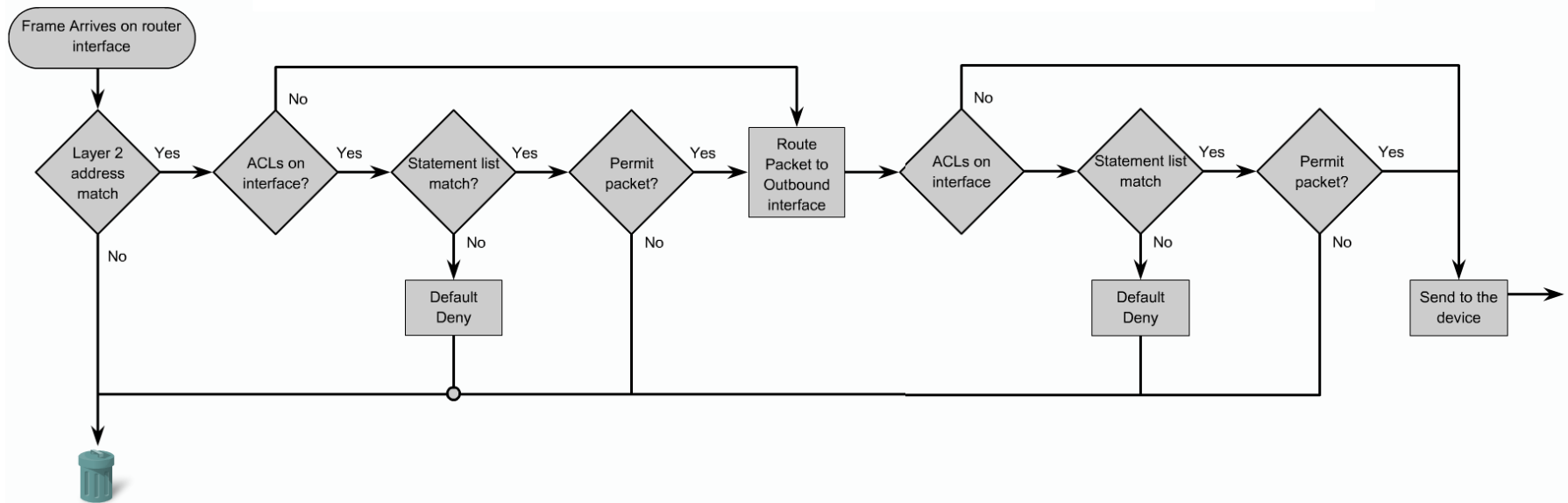
ACL (Access Control List)



ACL (*Access Control List*)

- ▣ Jeśli nie udało się dopasować żadnej instrukcji ACL, domyślnie na końcu listy jest umieszczana niejawna instrukcja **deny any**.
- ▣ Niejawna instrukcja **deny any** na końcu listy ACL zapobiega akceptacji pakietów, które nie zostały dopasowane.

ACL (Access Control List)



Funkcje maski blankietowej

- ▣ Maska blankietowa jest 32-bitową wielkością podzieloną na cztery oktety.
- ▣ Maska blankietowa jest nakładana na adres IP.
- ▣ Zera i jedynki w masce opisują sposób postępowania z odpowiadającymi im bitami adresu IP.
- ▣ Pojęcie maski blankietowej odnosi się do procesu dopasowywania bitów masek na liście ACL.
- ▣ Maska podsieci i maska blankietowa zastosowane w odniesieniu do adresu IP oznaczają dwie różne koncepcje.
- ▣ Maski podsieci używają binarnych zer i jedynek do identyfikacji części adresu IP oznaczających sieć, podsieć i hosta.
- ▣ Maski blankietowe używają binarnych zer i jedynek do filtrowania pojedynczych adresów IP lub ich grup w celu umożliwienia lub zabronienia dostępu do zasobów na podstawie adresu IP.
- ▣ Podobieństwo między maską blankietową a maską podsieci polega jedynie na tym, że obie mają długość 32 bitów i składają się z binarnych jedynek i zer.
- ▣ Binarne jedynki wskazują bity, które mają być ignorowane podczas dopasowania, zaś binarne zera - bity, które mają być dopasowane.

Funkcje maski blankietowej

- ▣ W procesie przetwarzania maski blankietowej jest ona **nakładana na adres IP** zawarty w instrukcji listy kontroli dostępu. W ten sposób tworzona jest wartość dopasowania.
- ▣ Druga część procesu ACL polega na tym, że maska blankietowa zostaje zastosowana w odniesieniu do **adresu IP sprawdzanego** przez daną instrukcję ACL.
- ▣ Wynik złożenia adresu IP i maski blankietowej musi być równy wartości dopasowania instrukcji ACL, aby dana instrukcja została zastosowana do pakietu.

Funkcje maski blankietowej

- ▣ Na listach ACL są stosowane dwa specjalne słowa kluczowe: opcje **any** i **host**.
- ▣ Opcja **any** zastępuje **0.0.0.0** dla adresu IP i **255.255.255.255** dla maski blankietowej.

Opcja ta **pasuje do dowolnego adresu**, z którym jest porównywana.

- ▣ Opcja **host** zastępuje **0.0.0.0** dla maski blankietowej. Maską tą wymaga, aby wszystkie bity w adresie ACL i adresie pakietu pasowały do siebie.

Opcja ta będzie wymuszała **dopasowanie do jednego konkretnego adresu**.

Listy ACL

1. **Standardowe listy ACL** sprawdzają adres źródłowy routowanych pakietów IP.

(1-99) oraz (1300-1999) – Standardowe listy ACL

Ponieważ standardowe listy ACL nie zawierają adresu przeznaczenia, umieszcza się je jak **najbliżej miejsca przeznaczenia**.

```
access-list 10 permit 192.168.30.0 0.0.0.255
```

2. **Rozszerzone listy ACL** sprawdzają źródłowe i docelowe adresy pakietów oraz mogą sprawdzać protokoły i numery portów.

(100-199) oraz (2000-2699) – Rozszerzone listy ACL

Rozszerzone listy ACL umieszcza się **jak najbliżej źródła** zabronionego ruchu.

```
access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
```

Istnieje możliwość przypisywania **nazw** listom ACL (nazwa powinna zaczynać się od litery, zawierać wyłącznie znaki alfanumeryczne (bez spacji lub znaków przystankowych), sugerowane jest użycie wielkich liter).

Standardowe listy ACL

- Standardowe listy ACL sprawdzają adres źródłowy routowanych pakietów IP.
- Pakiety, które zostały dopuszczone, są kierowane przez router do interfejsu wyjściowego. Jeśli nie zostały dopuszczone, są odrzucane na interfejsie wejściowym.
- Standardowa wersja polecenia konfiguracji globalnej **access-list** służy do definiowania standardowej listy ACL o numerze z przedziału od 1 do 99. Od wersji 12.0.1 systemu Cisco IOS, standardowe listy ACL korzystają z dodatkowych numerów (od 1300 do 1999), co łącznie daje maksymalną liczbę 798 standardowych list ACL. Te dodatkowe numery są znane pod nazwą dodatkowych list kontroli dostępu IP.

```
access-list 2 deny    172.16.1.1
access-list 2 permit 172.16.1.0 0.0.0.255
access-list 2 deny    172.16.0.0 0.0.255.255
access-list 2 permit 172.0.0.0  0.255.255.255
```

Należy zauważyć, że w pierwszej instrukcji ACL nie została użyta maska blankietowa. Jest wtedy stosowana maska domyślna 0.0.0.0.

Standardowe listy ACL

- ▣ Pełna składnia polecenia standardowej listy ACL wygląda następująco:

```
Router(config) #access-list numer-listy-  
dostepu {deny | permit | remark} źródło [maska-  
blankietowa-zródła]
```

- ▣ Słowo kluczowe **remark** służy do wprowadzania wyjaśnień opisujących listę kontroli dostępu. Długość wyjaśnienia jest ograniczona do 100 znaków:
 - Router(config) #**access-list** 1 **remark** Umożliwia wyłącznie dostęp stacji roboczej administratora
 - Router(config) #**access-list** 1 **permit** 171.69.2.88

Standardowe listy ACL

- ▣ Aby usunąć standardową listę ACL, należy użyć odmiany polecenia z wyrazem **no**. Składnia jest następująca:

```
Router(config) #no access-list numer-listy-dostępu
```

- ▣ Polecenie **ip access-group** łączy istniejącą standardową listę ACL z interfejsem:

```
Router(config-if) #ip access-group numer-listy-dostępu {in | out}
```

- ▣ Słowo kluczowe **in** stosuje się by przypisać listę dla ruchu wchodzącego do interfejsu routera, zaś **out** - do ruchu wychodzącego z interfejsu routera.

Rozszerzone listy ACL

- ▣ Rozszerzone listy ACL sprawdzają źródłowe i docelowe adresy pakietów oraz mogą sprawdzać protokoły i numery portów.
- ▣ Zwiększa to elastyczność opisu elementów sprawdzanych przez listę ACL.
- ▣ Rozszerzone listy ACL mogą jednocześnie zezwalać na ruch e-mail z interfejsu Fa0/0 do określonych miejsc dostępnych poprzez interfejs S0/0 oraz uniemożliwiać przesyłanie plików i przeglądanie sieci WWW.

Rozszerzone listy ACL

- ❑ Na pojedynczej liście ACL można skonfigurować **wiele instrukcji**. Wszystkie instrukcje powinny mieć ten sam numer listy kontroli dostępu, aby łączyć je z tą samą listą ACL.
- ❑ Rozszerzone listy ACL używają numerów list kontroli dostępu z zakresu od 100 do 199 i od 2000 do 2699.
- ❑ **Liczba instrukcji warunków jest dowolna** i ograniczona wyłącznie przez dostępną pamięć routera. Oczywiście im więcej instrukcji zawiera lista ACL, tym trudniej ją zrozumieć i nią zarządzać.
- ❑ Na końcu instrukcji rozszerzonej listy ACL można podać numer portu **TCP** lub **UDP**, aby zwiększyć precyzję warunku.
- ❑ Można używać do nich **operatorów logicznych**, takich jak równy (eq), nierówny (neq), większy (gt) lub mniejszy (lt).

Rozszerzone listy ACL

```
Router(config)#access-list access-list-number  
    {deny | permit | remark} protocol source  
    source-wildcard destination destination-  
    wildcard [operator] [port number or name]  
    [established]
```

- Router(config)#**access-list** 151 **permit** tcp
192.168.2.0 0.0.0.255 **any eq** www
- Router(config)#**access-list** 151 deny ip **any host**
192.168.3.2 **gt** 1023
- Router(config)#**access-list** 151 **permit** icmp **any**
any

- ▣ Jeżeli użyjemy w warunku słowa **established** dopasowane będą tylko takie pakiety, które należą do już nawiązanej sesji (bity ACK lub RST są ustawione, tylko dla tcp).
- ▣ Jeżeli użyjemy słowa **ip** w miejscu protokołu, każdy pakiet tcp, udp lub icmp zostanie dopasowany.

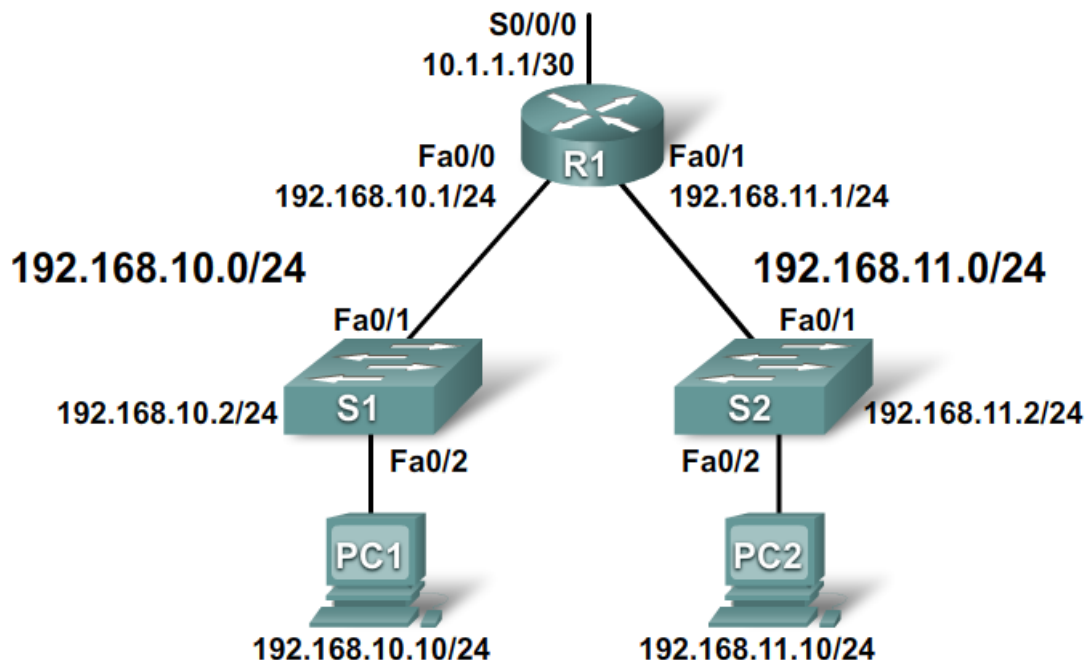
Rozszerzone listy ACL

- ▣ Polecenie **ip access-group** służy do łączenia istniejącej rozszerzonej listy ACL z interfejsem.
- ▣ Należy pamiętać, że dla danego interfejsu, kierunku i protokołu dopuszczalna jest tylko jedna lista ACL.
- ▣ Format polecenia jest następujący:

```
Router(config-if) #ip access-group numer-listy-  
dostepu {in | out}
```

- Router(config) #**interface** Serial0/1
- Router(config-if) #**ip access-group** 151 **in**
- Router(config-if) #**end**

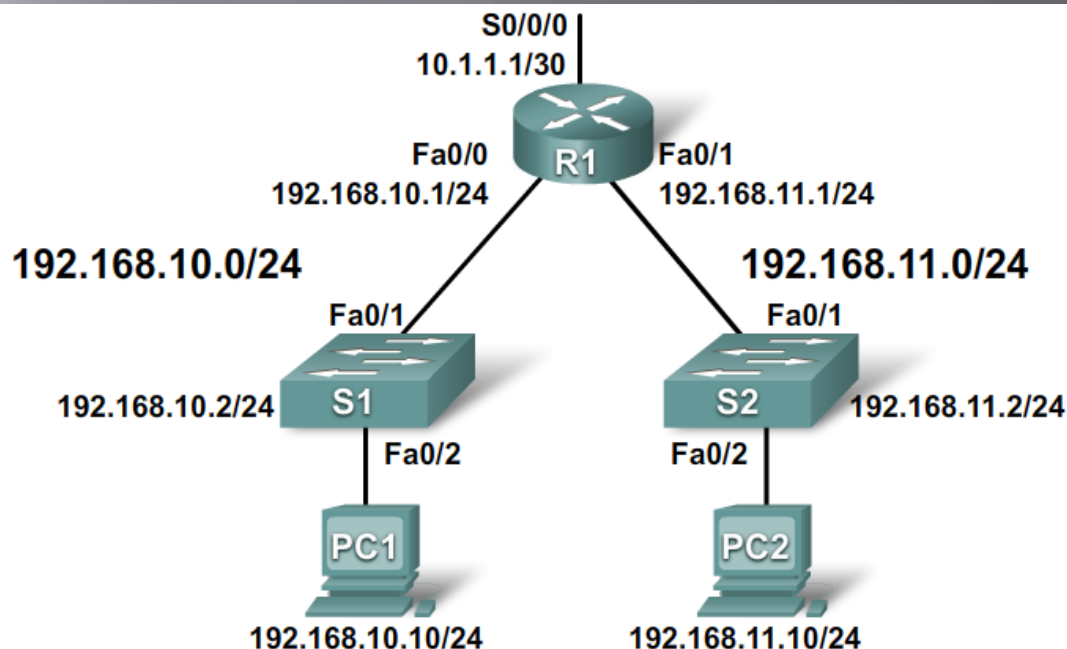
Przykłady list ACL



```
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)# interface S0/0/0
R1(config-if)# ip access-group 1 out
```

Przykład 1. ACL o numerze 1 zezwala na ruch z podanej sieci. Domyślnie blokowany jest ruch z innych lokalizacji sieciowych (**access-list 1 deny 0.0.0.0 255.255.255.255**). Lista ACL1 dowiązana jest do interfejsu szeregowego 0/0/0 i działa jako filtr wyjściowy.

Przykłady list ACL

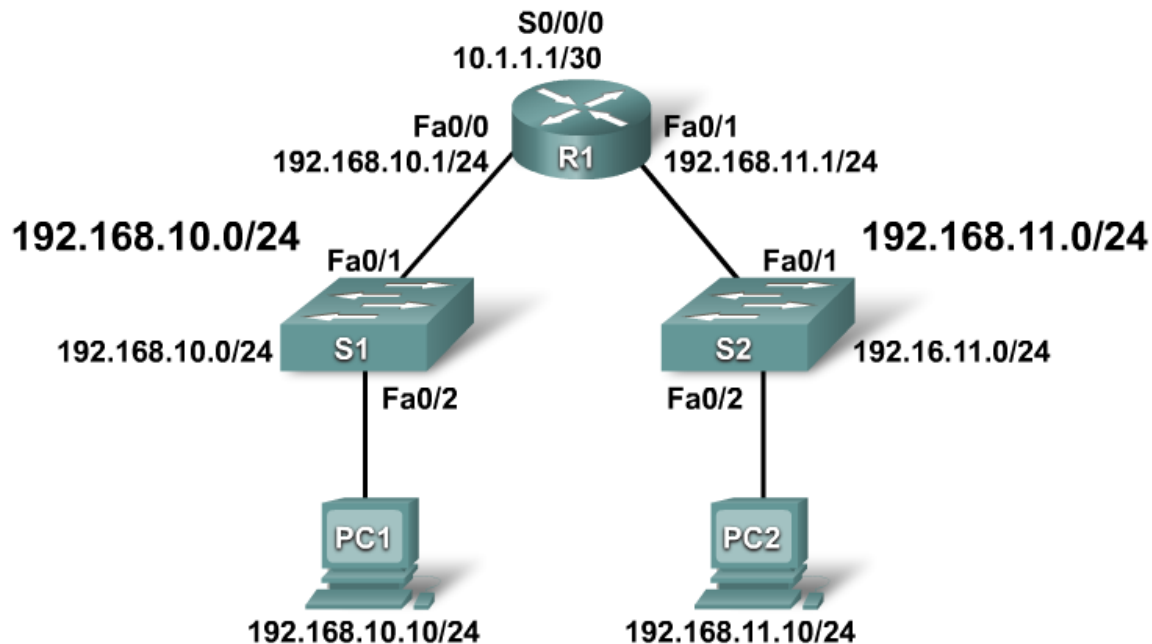


```
R1(config)#no access-list 1
R1(config)#access-list 1 deny 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#interface S0/0/0
R1(config-if)#ip access-group 1 out
```

Przykład 2. ACL o numerze 1 jest najpierw usunięta. Nowa lista ACL 1 blokuje konkretny host i zezwala na ruch z podanej sieci. Domyślnie blokowany jest ruch z innych lokalizacji sieciowych.

Lista ACL1 dołączana jest do interfejsu szeregowego 0/0/0 i działa jako filtr wyjściowy.

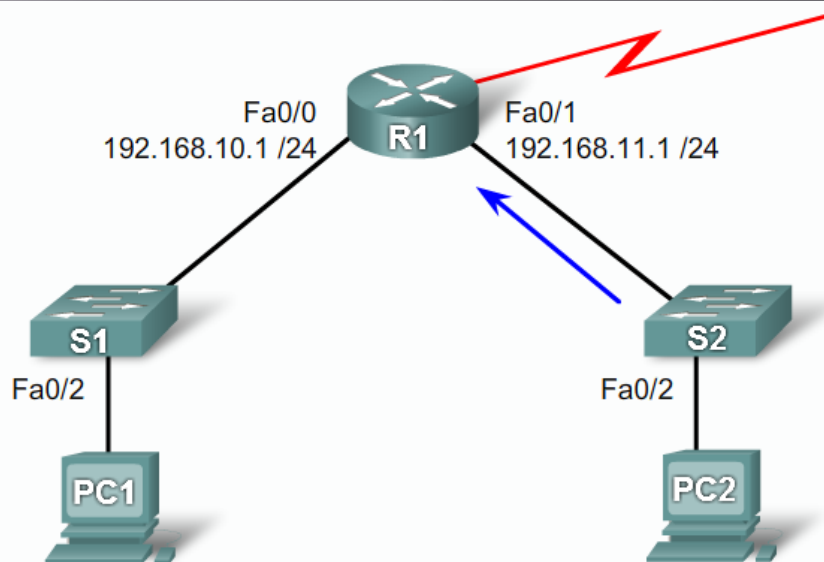
Przykłady list ACL



```
R1(config)#ip access-list standard NO_ACCESS
R1(config-std-nacl)#deny host 192.168.11.10
R1(config-std-nacl)#permit 192.168.11.0 0.0.0.255
R1(config-std-nacl)#interface Fa0/0
R1(config-if)#ip access-group NO_ACCESS out
```

Przykład 3. ACL o nazwie NO_ACCESS. Ponieważ nie używa się numerów, należy zadeklarować jaką lista jest tworzona (standardowa czy rozszerzona).

Przykłady list ACL



```
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0  
0.0.0.255 eq 21  
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0  
0.0.0.255 eq 20  
R1(config)# access-list 101 permit ip any any  
R1(config)# interface Fa0/1  
R1(config-if)# ip access-group 101 in
```

Przykład 4. Rozszerzona lista ACL 101. Blokowanie ruchu ftp z sieci 192.168.11.0. Blokowane są oba porty 20 i 21.

Rozmieszczanie list ACL

- ▣ **Rozszerzone** listy ACL umieszczają **jak najbliżej źródła** blokowanego ruchu.
- ▣ **Standardowe** listy ACL nie określają adresów docelowych, tak więc należy umieszczać je **jak najbliżej celu**.
- ▣ Administratorzy mogą umieszczać listy kontroli dostępu **jedynie na kontrolowanych przez siebie urządzeniach**. Z tego powodu rozmieszczenie list kontroli dostępu należy rozpatrywać w kontekście zakresu działania administratora sieci.

Weryfikowanie list ACL

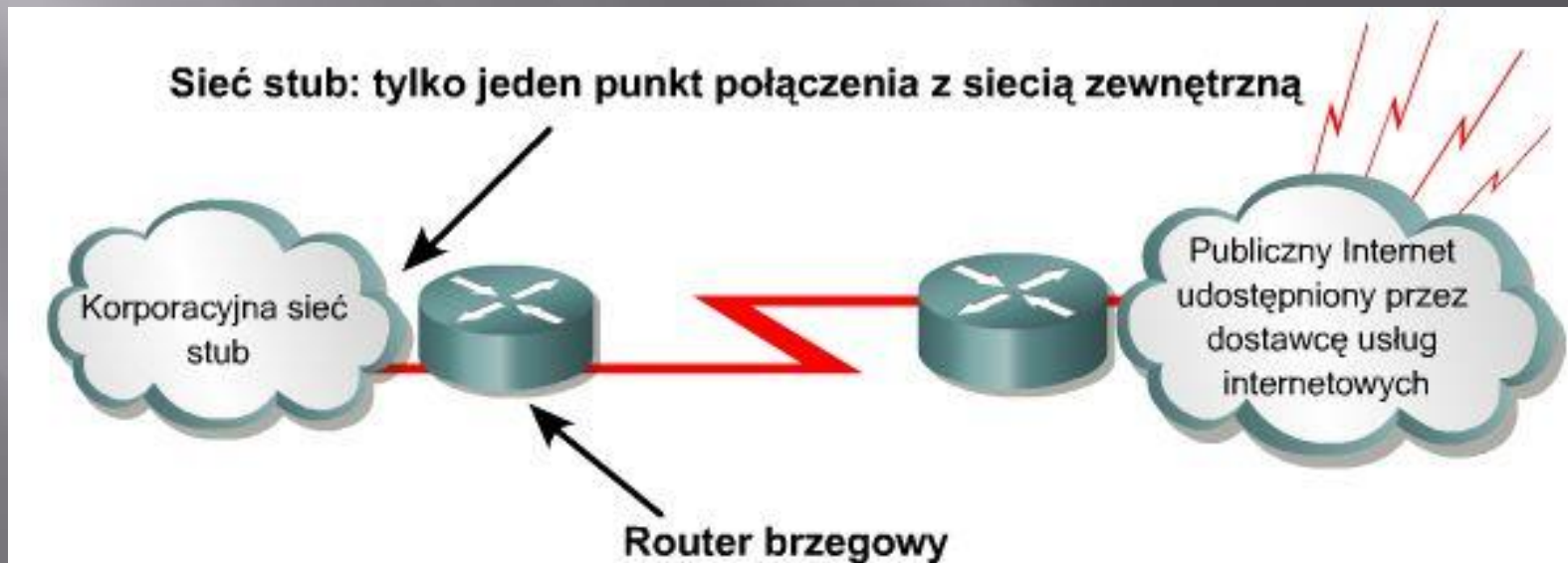
- ▣ Polecenie **show ip interface** służy do wyświetlania informacji o interfejsie IP oraz wskazania, czy do interfejsu przypisane są listy ACL.
- ▣ Polecenie **show access-lists** służy do wyświetlania zawartości wszystkich list ACL na routerze. Aby wyświetlić konkretną listę, jako opcję tego polecenia należy podać nazwę lub numer listy ACL.
- ▣ Polecenie **show running-config** służy również do wyświetlania listy kontroli dostępu na routerze oraz informacji o ich przypisaniu do interfejsów.

Weryfikowanie list ACL

```
Router#show access-lists  
Standard IP access list 2  
  deny   172.16.1.1  
  permit 172.16.1.0, wildcard bits 0.0.0.255  
  deny   172.16.0.0, wildcard bits 0.0.255.255  
  permit 172.0.0.0, wildcard bits 0.255.255.255  
Extended IP access list 101  
  permit tcp 192.168.6.0 0.0.0.255 any eq telnet  
  permit tcp 192.168.6.0 0.0.0.255 any eq ftp  
  permit tcp 192.168.0.0 0.0.0.255 any eq ftp-data  
Router#
```

NAT i PAT

- ▣ Technologia NAT (*Network Address Translation*) umożliwia **ograniczenie liczby publicznych adresów IP** i wykorzystanie prywatnych adresów IP w sieciach wewnętrznych.
- ▣ Te prywatne, wewnętrzne adresy są poddawane **translacji na adresy publiczne**, które mogą być routowane.
- ▣ Operacja ta wykonywana jest przez znajdujące się między sieciami urządzenia, na których działa wyspecjalizowane oprogramowanie obsługujące funkcję NAT, pozwalające na zwiększenie poziomu prywatności w sieci przez ukrycie wewnętrznych adresów IP.
- ▣ Urządzenie realizujące translację NAT zazwyczaj działa na granicy sieci szczytkowej (*ang. stub*). Sieć szczytkowa to sieć, która ma pojedyncze połączenie z sąsiednią siecią.



NAT i PAT

- ▣ Gdy host w sieci stub chce przesłać dane do hosta znajdującego się na zewnątrz, przekazuje pakiet do **routera brzegowego**.
- ▣ Router brzegowy realizuje proces NAT, czyli proces **translacji** prywatnego adresu wewnętrznego hosta na publiczny adres zewnętrzny, który może być routowany.
- ▣ W terminologii mechanizmu NAT sieć wewnętrzna to zbiór sieci, których adresy poddawane są translacji. Sieć zewnętrzna obejmuje wszystkie pozostałe adresy.
- ▣ Translacje NAT mogą być wykorzystywane do różnych celów, a przetłumaczone adresy mogą być **przydzielane dynamicznie lub statycznie**.

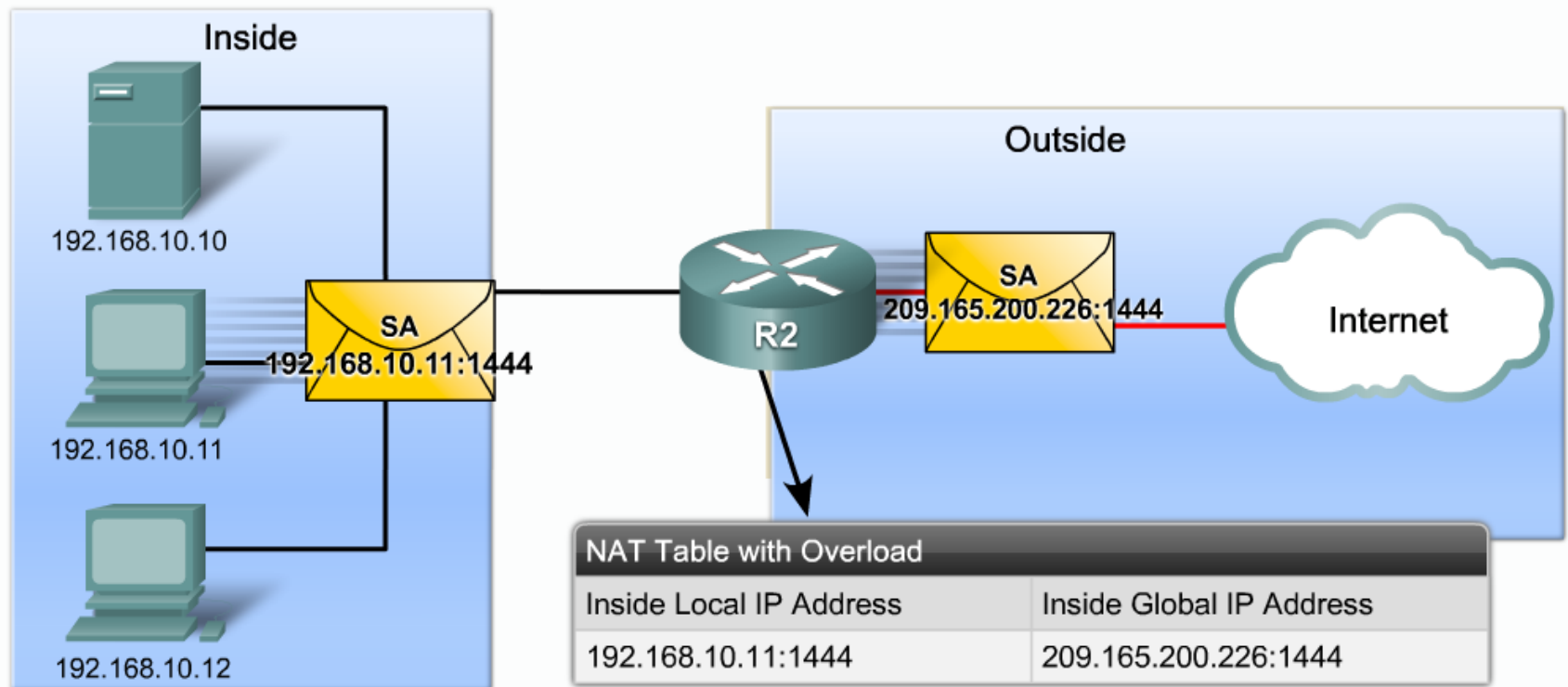
NAT i PAT

- ▣ **Statyczna translacja NAT** umożliwia utworzenie odwzorowania typu **jeden-do-jednego** pomiędzy adresami lokalnymi i globalnymi. Jest to szczególnie przydatne w wypadku hostów, które muszą mieć stały adres dostępny z Internetu. Takimi wewnętrznymi hostami mogą być serwery lub urządzenia sieciowe w przedsiębiorstwie.
- ▣ **Dynamiczna translacja NAT** służy do odwzorowania prywatnego adresu IP na adres publiczny. Hostowi w sieci jest przypisywany dowolny adres z **puli publicznych adresów IP**.

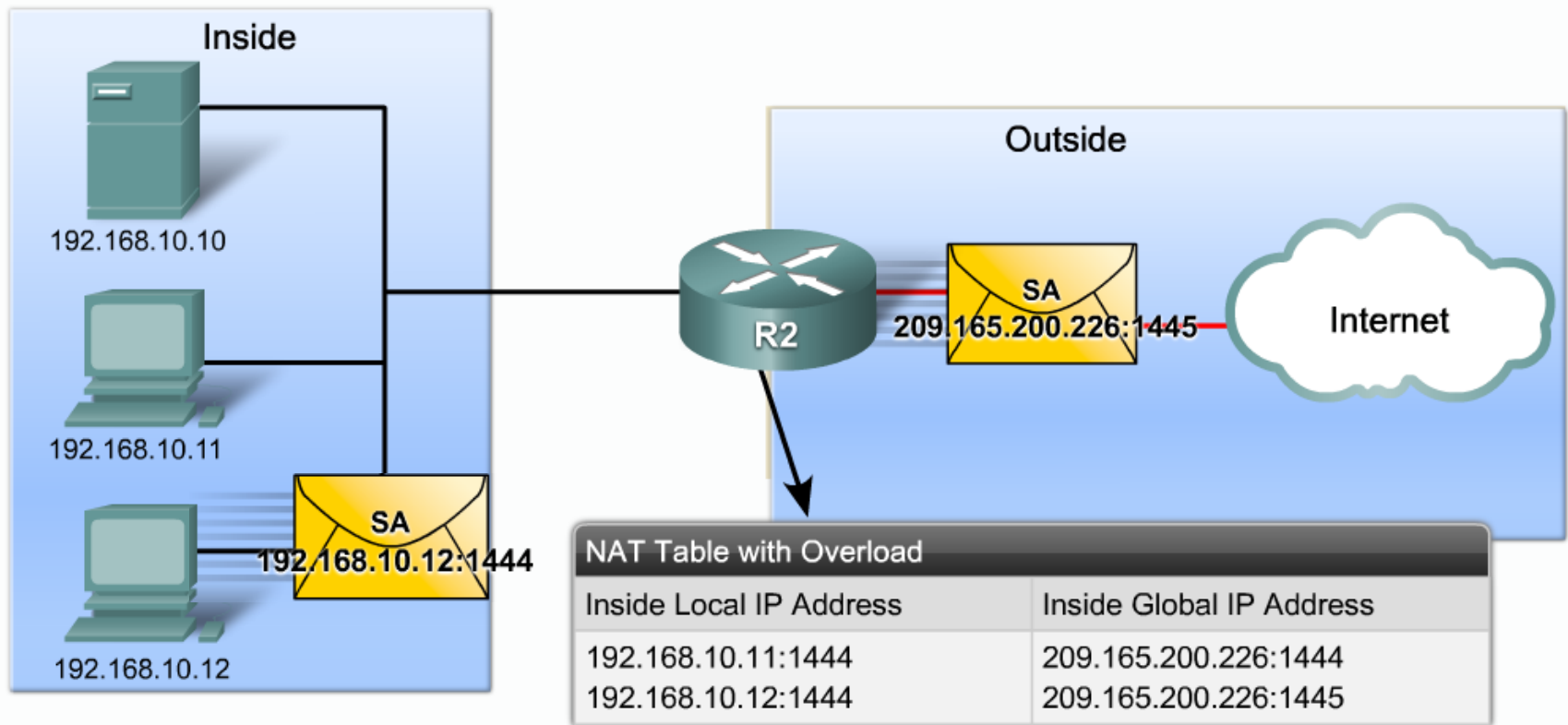
NAT i PAT

- ▣ Technika przeciążenia, lub inaczej translacji **PAT** (*Port Address Translation*), służy do odwzorowania **wielu prywatnych adresów IP na jeden publiczny adres IP**.
- ▣ Istnieje taka możliwość, ponieważ z każdym adresem prywatnym związany jest inny **numer portu**.
- ▣ W technologii PAT tłumaczone adresy są rozróżniane przy użyciu unikatowych numerów portów źródłowych powiązanych z globalnym adresem IP.
- ▣ Numer portu zakodowany jest na 16 bitach.
- ▣ Całkowita liczba adresów wewnętrznych, które mogą być przetłumaczone na jeden adres zewnętrzny, może teoretycznie wynosić nawet 65 536.
- ▣ W rzeczywistości do jednego adresu IP może zostać przypisanych około **4000** portów. W mechanizmie PAT podejmowana jest zawsze **próba zachowania pierwotnego portu źródłowego**. Jeśli określony port źródłowy jest już używany, funkcja PAT przypisuje pierwszy dostępny numer portu.
- ▣ Gdy zabraknie dostępnych portów, a skonfigurowanych jest wiele zewnętrznych adresów IP, mechanizm PAT przechodzi do następnego adresu IP w celu podjęcia kolejnej próby przydzielenia pierwotnego portu źródłowego. Ten proces jest kontynuowany aż do wyczerpania wszystkich dostępnych numerów portów i zewnętrznych adresów IP.

NAT i PAT



NAT i PAT



NAT i PAT

- ▣ Zastosowanie technologii NAT zapewnia następujące korzyści:
 - **Elastyczność** - eliminacja konieczności ponownego przypisania adresów IP do każdego hosta po zmianie dostawcy usług internetowych (ISP). Użycie mechanizmu NAT pozwala na uniknięcie zmiany adresów wszystkich hostów, dla których wymagany jest dostęp zewnętrzny, a to wiąże się z oszczędnościami czasowymi i finansowymi;
 - **Zmniejszenie liczby adresów** przy użyciu dostępnej w aplikacji funkcji multipleksowania na poziomie portów (PAT);
 - **Zwiększenie poziomu bezpieczeństwa** w sieci. Ponieważ w wypadku sieci prywatnej nie są rozgłaszane wewnętrzne adresy ani informacje o wewnętrznej topologii, sieć taka pozostaje wystarczająco zabezpieczona, gdy dostęp zewnętrzny odbywa się z wykorzystaniem translacji NAT.

NAT - konfiguracja - translacja statyczna

```
Router(config)#ip nat inside source static 192.168.1.15  
172.16.1.10  
Router(config)#ip nat inside source static 192.168.1.16  
172.16.1.11
```

Jest to konfiguracja statycznej translacji adresów dwóch urządzeń wewnętrznych. Wewnętrzny adres 192.168.1.15 będzie zawsze widoczny na zewnątrz jako 172.16.1.10, a adres 192.168.1.16 - jako 172.16.1.11. Ponieważ translacje te są statyczne, działają w obu kierunkach.

NAT - konfiguracja - translacja statyczna

```
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#interface Ethernet1/0
Router(config-if)#ip address 172.16.1.2 255.255.255.0
Router(config-if)#ip nat outside
Router(config-if)#exit
```

Przykład ten obejmuje **dwa interfejsy wewnętrzne i jeden zewnętrzny**. Interfejsy wewnętrzne wyznacza się za pomocą polecenia **ip nat inside**, zaś interfejsy zewnętrzne za pomocą **ip nat outside**. Przy konfigurowaniu NAT **trzeba** wyznaczyć przynajmniej jeden interfejs zewnętrzny.

NAT i PAT- konfiguracja - translacja dynamiczna

Zdefiniowanie listy ACL:

```
Router(config)#access-list 15 permit 192.168.0.0 0.0.255.255
```

Zdefiniowanie puli adresów:

```
Router(config)#ip nat pool PULANAT 172.16.1.100 172.16.1.150  
netmask 255.255.255.0
```

Powiązanie listy i puli adresów:

```
Router(config)#ip nat inside source list 15 pool PULANAT
```

Zdefiniowanie interfejsów wewnętrznych i jednego zewnętrznego

```
Router(config)#interface FastEthernet0/0  
Router(config-if)#ip address 192.168.1.1 255.255.255.0  
Router(config-if)#ip nat inside  
Router(config-if)#exit
```

```
Router(config)#interface FastEthernet0/1  
Router(config-if)#ip address 192.168.2.1 255.255.255.0  
Router(config-if)#ip nat inside  
Router(config-if)#exit
```

```
Router(config)#interface Ethernet1/0  
Router(config-if)#ip address 172.16.1.2 255.255.255.0  
Router(config-if)#ip nat outside  
Router(config-if)#exit
```

NAT i PAT- konfiguracja - translacja dynamiczna

- ❑ Urządzenia wewnętrzne będą reprezentowane na zewnątrz przez różne adresy globalne. Pierwsze urządzenie wewnętrzne, które nawiąże połączenie wychodzące, otrzyma pierwszy adres z zakresu (172.16.1.100), a następnie kolejne adresy. Zakres adresów konfiguruje się za pomocą polecenia **ip nat pool**.
- ❑ W tym przypadku w poleceniu **ip nat inside** nie ma słowa kluczowego **overload**. Bez tego słowa kluczowego po wyczerpaniu puli adresów router nie będzie mógł przydzielić nowego adresu. Urządzenia po prostu nie będą mogły nawiązywać nowych połączeń przez ten router. Jeśli jednak dołączymy słowo kluczowe **overload**, router wróci na początek zakresu i zacznie przydzielać wiele adresów wewnętrznych każdemu adresowi zewnętrznemu (PAT):

```
Router(config)#ip nat inside source list 15 pool  
PULANAT overload
```
- ❑ Każde urządzenie wykluczone przez listę dostępu nie będzie używać reguły NAT. Wykluczone urządzenia będą widoczne na zewnątrz pod prawdziwymi (wewnętrznymi lokalnymi) adresami IP.

Sprawdzanie konfiguracji mechanizmów NAT i PAT

- ▣ Należy użyć polecenia **debug ip nat**, aby sprawdzić działanie funkcji NAT, wyświetlając informacje o każdym pakiecie poddawanym translacji na routerze.
- ▣ Użycie polecenia **debug ip nat detailed** powoduje wygenerowanie opisu każdego pakietu branego pod uwagę przez mechanizm translacji. Wyświetlane są także informacje dotyczące konkretnych błędów lub wyjątków, takich jak brak możliwości przydzielenia adresu globalnego.

Polecenie	Opis
<code>show ip nat translations</code>	Wyświetla aktywne translacje
<code>show ip nat statistics</code>	Wyświetla statystykę translacji

Sprawdzanie konfiguracji mechanizmów NAT i PAT

```
R2#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	209.165.200.225:16642	192.168.10.10:16642	209.165.200.254:80	209.165.200.254:80
tcp	209.165.200.225:62452	192.168.11.10:62452	209.165.200.254:80	209.165.200.254:80

```
R2#show ip nat translations verbose
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	209.165.200.225:16642	192.168.10.10:16642	209.165.200.254:80	209.165.200.254:80
create 00:01:45, use 00:01:43 timeout:86400000, left 23:58:16, Map-Id(In): 1,				
flags:				
extended, use_count: 0, entry-id: 4, lc_entries: 0				
tcp	209.165.200.225:62452	192.168.11.10:62452	209.165.200.254:80	209.165.200.254:80
create 00:00:37, use 00:00:35 timeout:86400000, left 23:59:24, Map-Id(In): 1,				
flags:				
extended, use_count: 0, entry-id: 5, lc_entries: 0				

```
R2#
```

Sprawdzanie konfiguracji mechanizmów NAT i PAT

```
R2#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	209.165.200.225:16642	192.168.10.10:16642	209.165.200.254:80	209.165.200.254:80
tcp	209.165.200.225:62452	192.168.11.10:62452	209.165.200.254:80	209.165.200.254:80

```
R2#show ip nat translations verbose
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	209.165.200.225:16642	192.168.10.10:16642	209.165.200.254:80	209.165.200.254:80
tcp	209.165.200.225:62452	192.168.11.10:62452	209.165.200.254:80	209.165.200.254:80

```
R2#show ip nat statistics
```

```
Total active translations: 3 (0 static, 3 dynamic; 3 extended)
Map-Id(In): 1,
Outside interfaces:
  Serial0/1/0
Inside interfaces:
  Serial0/0/0, Serial0/0/1
Hits: 173 Misses: 9
CEF Translated packets: 182, CEF Punted packets: 0
Expired translations: 6
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 interface Serial0/1/0 refcount 3
Queued Packets: 0
R2#
```


Sprawdzanie konfiguracji mechanizmów NAT i PAT

```
R2#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	209.165.200.225:16642	192.168.10.10:16642	209.165.200.254:80	209.165.200.254:80
tcp	209.165.200.225:62452	192.168.11.10:62452	209.165.200.254:80	209.165.200.254:80

```
R2#show ip nat translations verbose
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	209.165.200.225:16642	192.168.10.10:16642	209.165.200.254:80	209.165.200.254:80
tcp	209.165.200.225:62452	192.168.11.10:62452	209.165.200.254:80	209.165.200.254:80

```
R2#show ip nat statistics
```

```
Total active translations: 3 (0 static, 3 dynamic; 3 extended)
Map-Id(In): 1,
Outside interfaces:
  Serial0/1/0
Inside interfaces:
  Serial0/0/0, Serial0/0/1
Hits: 173 Misses: 9
CEF Translated packets: 182, CEF Punted packets: 0
Expired translations: 6
```

```
Dynamic mappings:
```

```
-- Inside Source
```

```
[Id: 1] access-list 1 interface Se
```

```
Queued Packets: 0
```

```
R2#
```

```
R2#clear ip nat translation *
```

```
R2#show ip nat translations
```

```
R2#
```


DHCP (*Dynamic Host Configuration Protocol*)

- ▣ Protokół DHCP działa w trybie klient-serwer.
- ▣ Protokół DHCP pozwala klientom DHCP w sieciach IP na uzyskiwanie informacji o ich konfiguracji z serwera DHCP.
- ▣ Użycie protokołu DHCP zmniejsza nakład pracy wymagany przy zarządzaniu siecią IP.
- ▣ Najważniejszym elementem konfiguracji odbieranym przez klienta od serwera jest adres IP klienta.
- ▣ Protokół DHCP jest opisany w dokumencie RFC 2131.
- ▣ Klient DHCP wchodzi w skład większości nowoczesnych systemów operacyjnych, takich jak systemy Windows, Novell Netware, Sun Solaris, Linux i MAC OS.
- ▣ Klient żąda uzyskania danych adresowych z sieciowego serwera DHCP.
- ▣ Serwer ten zarządza przydzielaniem adresów IP i odpowiada na żądania konfiguracyjne klientów.

DHCP (*Dynamic Host Configuration Protocol*)

- ❑ Protokół DHCP działa jako proces serwera służący do przydzielania danych adresowych IP dla klientów. Klienci **dzierżawią informacje pobrane z serwera** na czas ustalony przez administratora. Gdy okres ten dobiega końca, klient musi zażądać nowego adresu. Zazwyczaj klient uzyskuje ten sam adres.
- ❑ Administratorzy na ogół preferują serwery sieciowe z usługą DHCP, ponieważ takie rozwiązanie jest skalowalne i łatwo nim zarządzać.
- ❑ Routery Cisco mogą wykorzystywać specjalny zestaw funkcji systemu IOS firmy Cisco – Easy IP – w celu udostępnienia opcjonalnego, w pełni funkcjonalnego serwera DHCP.
- ❑ Domyślny okres dzierżawy ustawień konfiguracyjnych w wypadku oprogramowania Easy IP to 24 godziny. Rozwiązanie takie jest przydatne w małych firmach i biurach domowych, gdzie użytkownicy mogą wykorzystać funkcje protokołu DHCP i mechanizmu NAT bez konieczności używania serwera NT lub UNIX.
- ❑ Administratorzy konfiguruje serwery DHCP tak, aby przydzielane były adresy ze **zdefiniowanych pul adresów**. Na serwerach DHCP mogą być dostępne także inne informacje, takie jak adresy serwerów DNS, adresy serwerów WINS i nazwy domen.
- ❑ W wypadku większości serwerów DHCP administratorzy mogą także zdefiniować adresy MAC obsługiwanych klientów i automatycznie przypisywać dla tych klientów zawsze te same adresy IP.

Dynamiczne konfigurowanie adresów IP routera za pomocą DHCP

- ▣ Polecenie konfiguracyjne **ip address dhcp** pozwala routerowi dynamicznie pobierać informacje adresowe dotyczące określonego interfejsu:
 - Router(config)#**interface** Ethernet0
 - Router(config-if)#**ip address dhcp client-id** Ethernet0
 - Router(config-if)#end

Dynamiczne przydzielanie adresów IP klientom za pomocą DHCP

- ▣ Poniższy zbiór poleceń konfiguracyjnych pozwala routerowi dynamicznie przydzielić adresy IP klienckim stacjom roboczym:
 - Router(config) #**service dhcp**
 - Router(config) #**ip dhcp pool** POOL_172.25.1.0/24
 - Router(dhcp-config) #**network** 172.25.1.0
255.255.255.0
 - Router(dhcp-config) #**default-router** 172.25.1.1
 - Router(dhcp-config) #**dns-server** 172.25.1.1
 - Router(dhcp-config) #**exit**
 - Router(config) #**ip dhcp excluded-address**
172.25.1.1 172.25.1.50
 - Router(config) #**ip dhcp excluded-address**
172.25.1.200 172.25.1.255
 - Router(config) #**end**

Sprawdzanie działania usługi DHCP

- ▣ Aby sprawdzić działanie usługi DHCP, można użyć polecenia **show ip dhcp binding**. Polecenie to służy do wyświetlenia listy wszystkich powiązań utworzonych przez usługę DHCP.
- ▣ Aby sprawdzić, czy komunikaty są odbierane lub wysyłane przez router, należy użyć polecenia **show ip dhcp server statistics**. Użycie tego polecenia spowoduje wyświetlenie informacji o liczbie wysłanych i odebranych komunikatów DHCP.
- ▣ Do rozwiązywania problemów dotyczących działania serwera DHCP można używać polecenia **debug ip dhcp server events**. Użycie tego polecenia spowoduje wyświetlenie informacji o tym, czy serwer okresowo sprawdza wygaśnięcia dzierżawy adresów. Wyświetlone zostaną także procesy związane z adresami zwracanymi oraz przypisywanymi.

Podsumowanie

- ▣ ACL (*Access Control List*)
- ▣ Funkcje maski blankietowej
- ▣ Standardowe listy ACL
- ▣ Rozszerzone listy ACL
- ▣ Rozmieszczanie list ACL
- ▣ Weryfikowanie list ACL
- ▣ NAT i PAT
- ▣ NAT - konfiguracja - translacja statyczna
- ▣ NAT i PAT- konfiguracja - translacja dynamiczna
- ▣ Sprawdzanie konfiguracji mechanizmów NAT i PAT
- ▣ DHCP (*Dynamic Host Configuration Protocol*)
- ▣ Dynamiczne konfigurowanie adresów IP routera za pomocą DHCP
- ▣ Dynamiczne przydzielanie adresów IP klientom za pomocą DHCP
- ▣ Sprawdzanie działania usługi DHCP