# ZDANIE SPRAWY

Projekt indywidualny - NAT

Przez
Krystian Sereda


Wydział Elektryczny PW
Informatyka Stosowana

# Spis treści

# 1 Wstęp, założenia oraz cel projektu

Sprawozdanie ma na celu przedstawienie, lepsze zrozumienie oraz zobrazowanie działania Network Address Translation (NAT). Głównym celem tego sprawozdania będzie dostarczenie czytelnikom wiedzy na temat działania NAT oraz ukazanie jego praktycznego zastosowania w środowisku ZeroTier.

Część pierwsza sprawozdania skupi się na omówieniu teoretycznych podstaw NAT, gdzie zostanie wyjaśniony proces translacji adresów IP, który odbywa się za pomocą NAT. Przedstawione zostaną również główne zalety i zastosowania tej technologii.

Następnie, w dalszej części sprawozdania, zaprezentowane zostanie działanie NAT w środowisku ZeroTier. Aby umożliwić czytelnikom lepsze zrozumienie i zobrazowanie tego działania, przeprowadzony zostanie praktyczny eksperyment, wykorzystując dwie maszyny podłączone do wirtualnej sieci ZeroTier, co pozwoli zaobserwować zmiany w adresach źródłowych i docelowych podczas komunikacji między maszynami.

Przedstawione teoretyczne podstawy NAT oraz przeprowadzony praktyczny eksperyment w środowisku ZeroTier mają na celu umożliwienie czytelnikom lepszego zrozumienia działania i zastosowania tej technologii. Zachęcam do dalszego zgłębiania tematu NAT oraz eksploracji możliwości konfiguracji i dostosowywania translacji adresów IP w ostatnim dziale sprawozdania związanym ze źródłami.

# 2 Jak działa NAT?

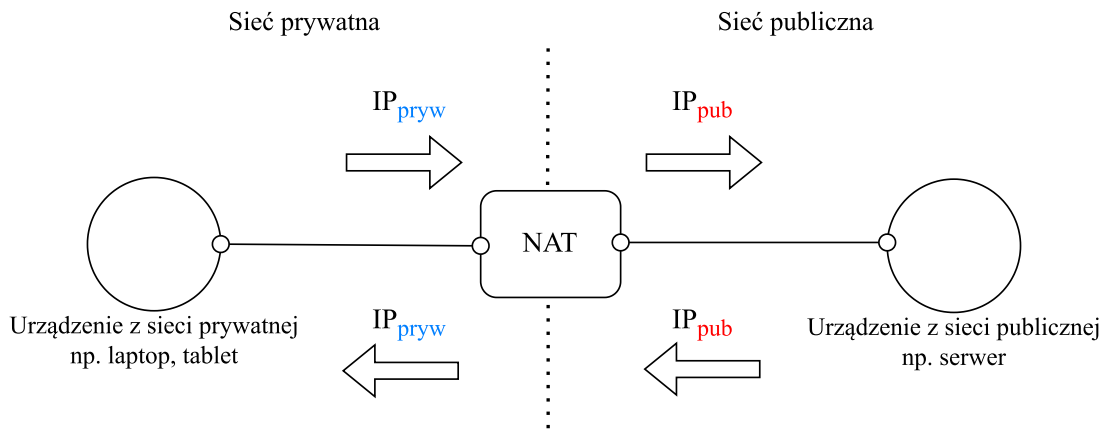
NAT (Network Address Translation) to mechanizm umożliwiający przekładanie adresów sieciowych między różnymi sieciami komputerowymi. Głównym celem NAT jest rozwiązanie problemu braku dostępnych publicznych adresów IP. NAT działa jako pośrednik pomiędzy sieciami prywatnymi a publicznymi, tłumacząc adresy IP i umożliwiając komunikację między nimi.

Dobrym przykładem zastosowania NAT jest jego wykorzystanie w naszych domach.

W domowym środowisku często posiadamy wiele urządzeń, takich jak komputery, telefony, tablety itp. które w celu swobodnego korzystania wymagają stałego podłączenia do internetu. Jednak dostawca usług internetowych często przypisuje nam tylko jeden publiczny adres IP. W tym miejscu NAT znajduje swoje zastosowanie.

W przypadku posiadania urządzenia umożliwiającego translację adresów, takiego jak trasownik, którego zadaniem jest pełnienie roli "tłumacza" adresów IP między naszą siecią prywatną a dostawcą internetu, jesteśmy w stanie korzystać z wielu urządzeń jednocześnie. Zaimplementowany w trasowniku NAT przekłada adresy IP urządzeń w naszej sieci prywatnej na publiczny adres IP.

Kiedy jedno z urządzeń w naszej sieci wysyła żądanie do serwera internetowego, trasownik zamienia adres źródłowy tego żądania na swój publiczny adres IP. Serwer internetowy wysyła odpowiedź na ten adres, a trasownik dokonuje odwrotnej translacji, przekierowując odpowiedź do właściwego urządzenia w sieci prywatnej.



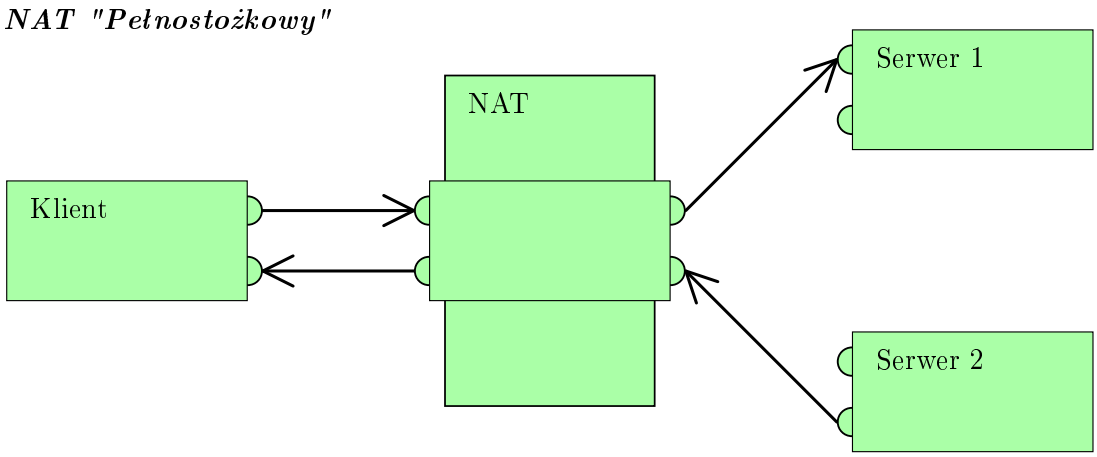

Rysunek 1: Uproszczony schemat zamiany adresów wykorzystywany podczas translacji NAT.

## 2.1 Rodzaje NAT ze względu na implementację

Pojedyncze określenie NAT nie klasyfikuje jednoznacznie implementacji jaka została dokonana. Istnieje kilka sposobów podziału NAT-ów, w zależności od różnych czynników i wymagań sieciowych. W przypadku różnych rodzajów NAT, mamy do czynienia z różnymi poziomami restrykcji i możliwościami dostępu zewnętrznego do wewnętrznej sieci.

### 2.1.1 NAT pełnostożkowy

W przypadku NAT pełnostożkowego (Full-cone NAT), po przypisaniu wewnętrznego adresu IP oraz portu (IPwew:PortWew) do zewnętrznego adresu IP oraz portu (IPpub:PortPub), wszystkie pakiety wysyłane z IPwew:PortWew są przekazywane przez IPpub:PortPub. Co istotne, dowolne urządzenie zewnętrzne może wysyłać pakiety do IPwew:PortWew, wysyłając je na IPpub:PortPub. Oznacza to, że istnieje otwarty dostęp dla urządzeń zewnętrznych do komunikacji z wewnętrznym urządzeniem. Ten rodzaj NAT charakteryzuje się niskim poziomem ograniczeń.

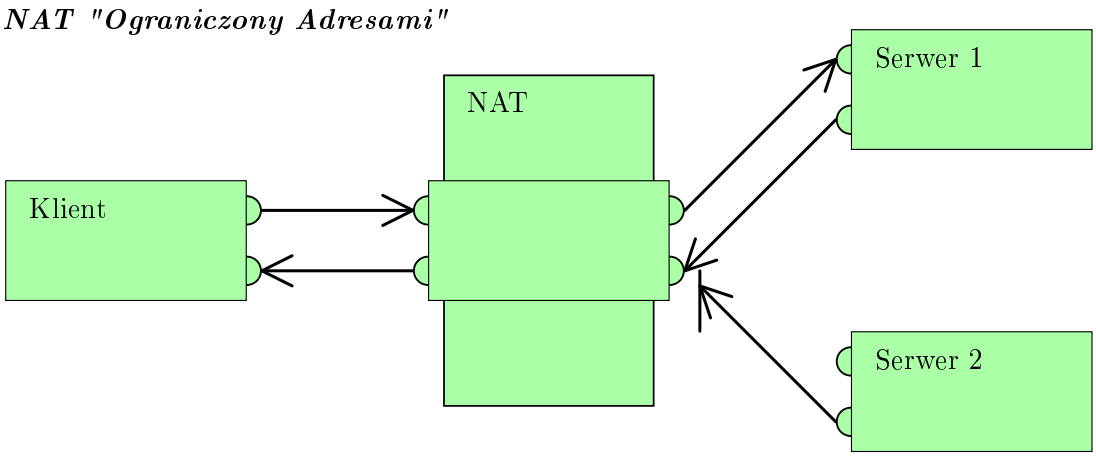



Rysunek 2: NAT pełnostożkowy

- Dowolny host zewnętrzny może wysyłać pakiety do wewnętrznego adresu IP i portu poprzez zewnętrzny adres IP i port.

### 2.1.2 NAT ograniczony adresami

W przypadku NAT ograniczonego adresem (Address-restricted cone NAT), po przypisaniu wewnętrznego adresu IP oraz portu (IPwew:PortWew) do zewnętrznego adresu IP oraz portu (IPpub:PortPub), wszystkie pakiety wysyłane z IPwew:PortWew są przekazywane przez IPpub:PortPub. Jednakże, aby zewnętrzne urządzenie (IPzew:Dowolny) mogło wysyłać pakiety do IPwew:PortWew, musi wcześniej otrzymać pakiet od IPwew:PortWew i wysyłać pakiety na IPpub:PortPub. "Dowolny" oznacza, że numer portu nie ma znaczenia. To wprowadza pewne ograniczenia dotyczące komunikacji między wewnętrznym a zewnętrznym urządzeniem, co zwiększa poziom bezpieczeństwa sieci.

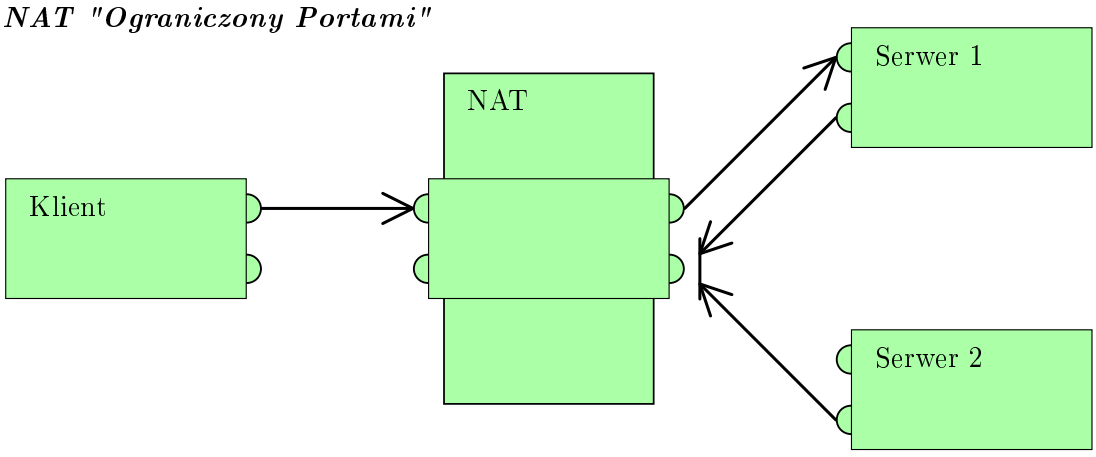



Rysunek 3: NAT ograniczony adresami

- Zewnętrzny host może wysyłać pakiety do wewnętrznego adresu IP i portu tylko wtedy, gdy wewnętrzny adres IP i port wcześniej wysłał pakiet do danego zewnętrznego hosta.
- Ograniczenie nie dotyczy numerów portów.

### 2.1.3 NAT ograniczony portami

NAT ograniczony portami, po przypisaniu wewnętrznego adresu (IPwew:PortWew) do zewnętrznego (IPpub:PortPub), wszystkie pakiety wysyłane z IPwew:PortWew są przekazywane przez IPpub:PortPub. Jednak, aby zewnętrzne urządzenie (IPzew:PORTzew) mogło wysyłać pakiety do IPwew:PortWew, musi wcześniej otrzymać pakiet od IPwew:PortWew i wysyłać pakiety na IPpub:PortPub, uwzględniając przy tym numer portu.




Rysunek 4: NAT ograniczony portami

- Zewnętrzny host może wysyłać pakiety do wewnętrznego adresu IP i portu tylko wtedy, gdy wewnętrzny adres IP i port wcześniej wysłał pakiet do danego zewnętrznego hosta.
- Ograniczenie dotyczy również numerów portów.

### 2.1.4 NAT symetryczny

NAT symetrycznym, zwany również jako Symmetric NAT, jest kolejnym rodzajem NAT stosowanym w sieciach komputerowych. Różni się on od wcześniej opisanych rodzajów NAT pod względem sposobu mapowania adresów i portów.

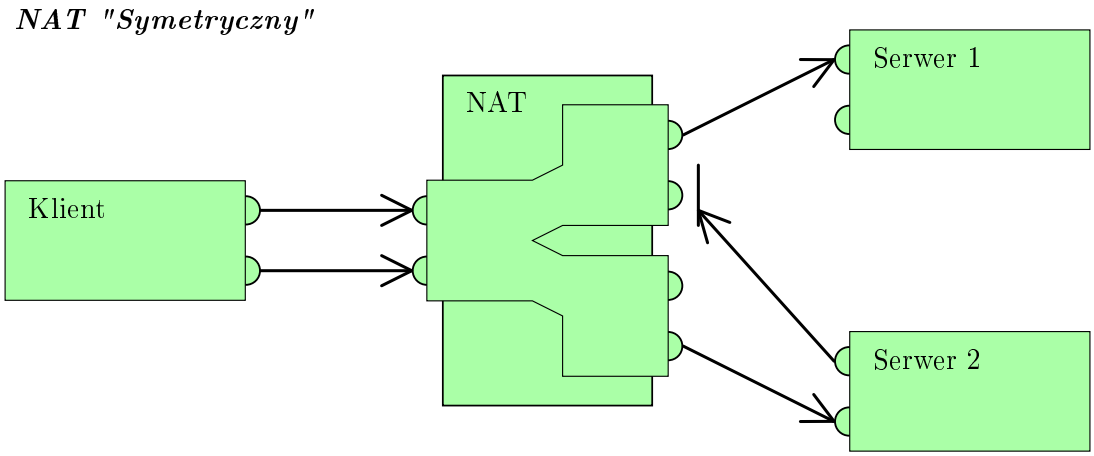
W przypadku NAT symetrycznego, kombinacja jednego wewnętrznego adresu IP oraz adresu i portu docelowego jest mapowana na unikalny zewnętrzny adres źródłowy IP oraz port. Oznacza to, że nawet jeśli ta

sama wewnętrzna maszyna wysyła pakiet z tym samym adresem źródłowym i portem, ale do innej lokalizacji, zostanie użyte inne mapowanie.

Co istotne, jedynie zewnętrzne urządzenie, które otrzymało pakiet od wewnętrznej maszyny, może wysłać pakiet z powrotem do tej maszyny. Oznacza to, że inicjowanie połączenia z zewnątrz jest możliwe tylko w odpowiedzi na pakiet otrzymany od wewnętrznej maszyny.

NAT symetryczny wprowadza znaczne ograniczenia w komunikacji między wewnętrznymi i zewnętrznymi maszynami. Może to znacznie utrudnić niektórym aplikacjom i usługom inicjowaniu połączeń z zewnątrz.

Jednak ten rodzaj NAT również przyczynia się do zwiększenia poziomu bezpieczeństwa sieci. Działa jako rodzaj zapory ogniowej, który blokuje nieautoryzowane połączenia z zewnątrz, ponieważ tylko urządzenia, które wcześniej otrzymały pakiety od wewnętrznej maszyny, mogą wysyłać pakiety z powrotem.




Rysunek 5: NAT symetryczny

- Kombinacja wewnętrznego adresu IP oraz adresu i portu docelowego jest mapowana na unikalny zewnętrzny adres źródłowy IP oraz port.
- Tylko zewnętrzne urządzenie, które otrzymało pakiet od wewnętrznej maszyny, może wysłać pakiet z powrotem.

### 2.1.5 Podsumowanie

Wybór odpowiedniego rodzaju NAT zależy od konkretnych wymagań sieciowych i priorytetów dotyczących bezpieczeństwa. W przypadku, gdy priorytetem jest wyższy poziom bezpieczeństwa, warto rozważyć zastosowanie ograniczonego adresu NAT, ograniczonego portem NAT lub symetrycznego NAT, które wprowadzają dodatkowe restrykcje i kontrolę dostępu. Natomiast w sytuacjach, gdzie wygodny i otwarty dostęp z zewnątrz jest kluczowy, pełnostożkowy NAT może być odpowiednim wyborem, mimo to należy pamiętać, że wiążę się to z pewnego rodzaju ryzykiem dla bezpieczeństwa sieci.

## 2.2 NAT jeden do wielu

Podstawowa zasada działania NAT została opisana w dziale `Jak działa NAT?`, natomiast w tej sekcji przyjrzę się najpopularniejszej jego odmianie czyli NAT jeden do wielu, która zyskała swój synonim w powszechnym użyciu jako ogólnie NAT.

Powszechnie pakiety IP posiadają źródłowy oraz docelowy adres IP. Pakiety te przechodzące z sieci prywatnej do sieci publicznej będą miały zmodyfikowany adres źródłowy, natomiast pakiety wracające z sieci publicznej do sieci prywatnej będą miały zmodyfikowany adres docelowy. Zdecydowana większość ruchu internetowego korzysta z protokołów TCP oraz UDP, które wykorzystują numery portów jako element identyfikacji. W celu uniknięcia niejasności podczas komunikacji, w tych protokołach numery portów są zmieniane, przez trasownik w taki sposób, że kombinacja adresu IP i numeru portu jednoznacznie określa odpowiednie miejsce docelowe w sieci.

Rozwiązując ten problem, wprowadzając zmianę numerów portów źródłowych NAT jeden do wielu nazywany jest również często jako PAT - Port Address Translation.

### 2.2.1 Zamiana adresu oraz portu

Równoczesne korzystanie z przeglądarek internetowych na wielu komputerach w jednej sieci prywatnej, na przykład w naszych domach, stanowi dobry przykład wykorzystania zmiany portów w NAT.

Każdy wychodzący pakiet IP z danego komputera zawierał będzie swój prywatny adres IP oraz losowo dobrany port. Czyli zapytania generowane przez przeglądarkę zawierać będą:

- Adres oraz port `docelowy` (np: 172.2.146.156:443)
- Adres oraz port `źródłowy` (np: 192.168.0.10:1111)

Podczas nawiązywania połączenia, system operacyjny przypisuje losowy numer portu źródłowego z puli dostępnych portów. Ten losowo wygenerowany numer portu źródłowego zapewnia unikalność połączenia i umożliwia rozróżnienie między różnymi połączeniami sieciowymi, które są równocześnie aktywne na tym samym urządzeniu.

W celach demonstracyjnych przyjmę, że dostępny port źródłowy przyjmuje wartość 1111, natomiast port nadany podczas translacji równy będzie 4444.

W dalszym ciągu, pakiet trafia do urządzenia wykonującego translację adresów i portów (PAT). Pakiet z adresem IP prywatnym i portem źródłowym 1111 zostaje przetłumaczony przez PAT na adres IP publiczny oraz, dostępny z punktu widzenia translatora adresów, port 4444.

| **Adres IP źródłowy** | **Port źródłowy** | **Adres IP docelowy** | **Port docelowy** |
|---|---|---|---|
| ... | ... | ... | ... |
| IP prywatne | 1111 | IP publiczne | 4444 |
| ... | ... | ... | ... |

Tabela 1: Tabela połączeń dla translacji adresów i portów.

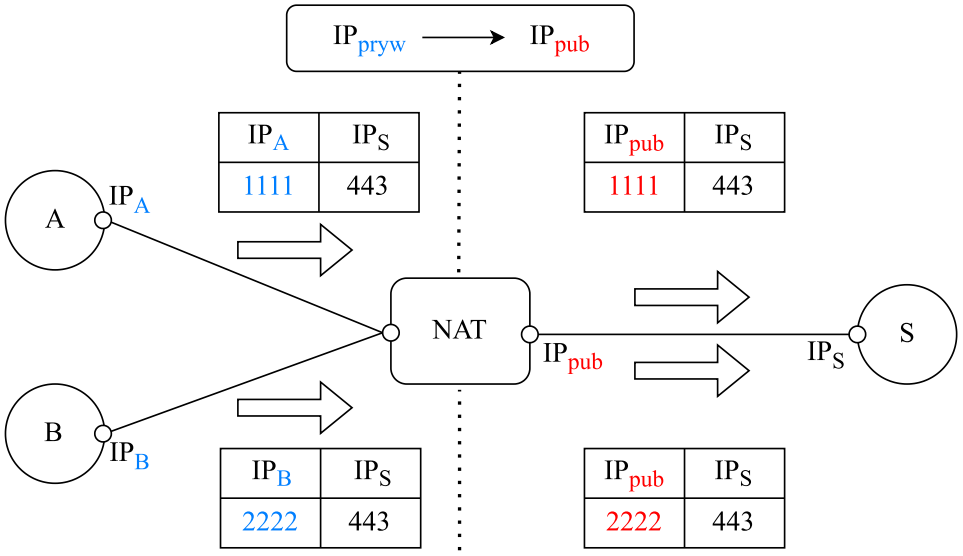
Wraz ze zamianą adresu oraz portu urządzenie dodaje odpowiedni wpis w tabeli translacji co umożliwia mu jednoznaczne odwzorowywanie odpowiedzi ze świata zewnętrznego do urządzeń prywatnych. W ten sposób odpowiedz z zewnątrz, "zaadresowana" adresem publicznym oraz portem 4444, trafi znowu do odpowiedniego nadawcy.

W naturalny sposób nasuwa się pytanie, czy zmiana portu jest konieczna? W dalszej częsci postaram się zobrazować istotę tego rozwiązania.

### 2.2.2 Co gdyby nie zmiana portów?

Rozważmy schemat blokowy wcześniej wspomnianej sytuacji w której to dwa urządzenia z sieci prywatnej (A oraz B takie jak komputer, laptop, tablet itd.) próbują skomunikować się z urządzeniem z zewnątrz (S) za pośrednictwem PAT.

W idealnym przypadku wygenerowane porty dwóch urządzeń są różne, a odpowiedzi od gospodarza zewnętrznego mogą zostać zwrócone jednoznacznie do odpowiednich urządzeń w sieci prywatnej.




Rysunek 6: Schemat komunikacji przy optymistycznym założeniu różnych portów.

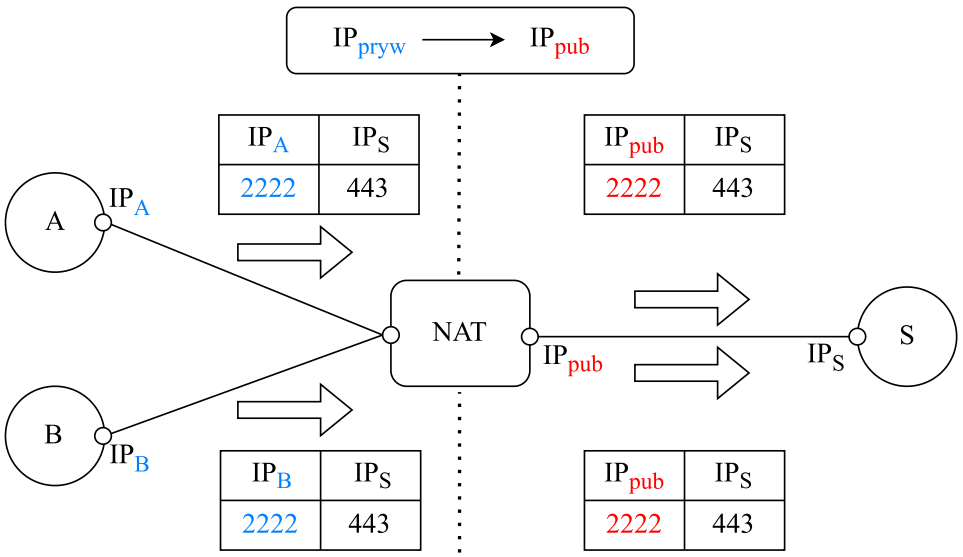
Jednoznaczność w tym przypadku polega na przypisaniu w tabeli translacji dokładnie jednego urządzenia do którego ma być kierowana odpowiedź z zewnątrz.

| **Adres IP źródłowy** | **Port źródłowy** | **Adres IP docelowy** | **Port docelowy** |
|---|---|---|---|
| ... | ... | ... | ... |
| IP prywatne A | 1111 | IP publiczne | 1111 |
| IP prywatne B | 2222 | IP publiczne | 2222 |
| ... | ... | ... | ... |

Tabela 2: Tabela połączeń dla translacji adresów i portów.

Odpowiedź kierowana na adres publiczny z odpowiednim portem tj. 1111 lub 2222 jednoznacznie wskazuje urządzenie w sieci prywatnej do którego kierowane mają być pakiety przychodzące, analogicznie urządzenia A lub B.

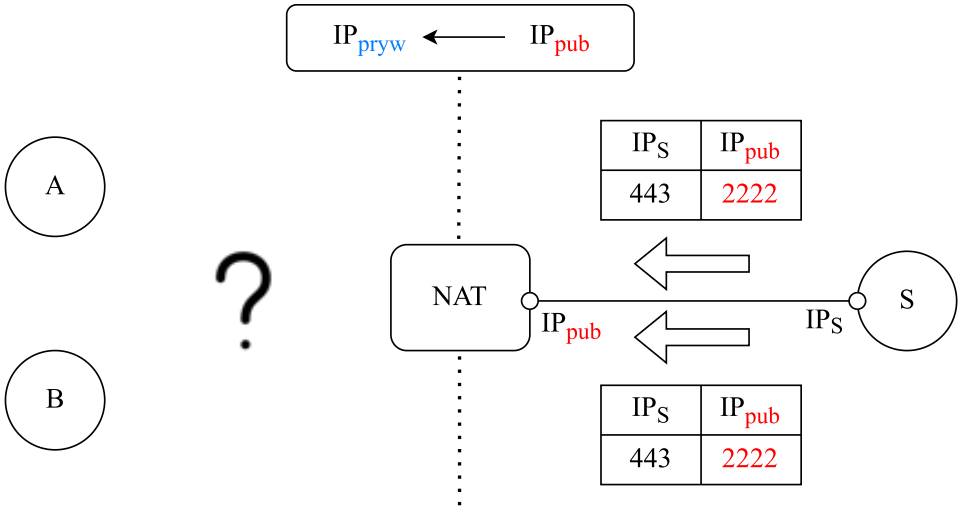
W związku z losowością przydzielania portów źródłowych nasuwa się pytanie, co gdy dwa urządzenia lokalne chcące porozumieć się z tym samym gospodarzem zewnętrznym otrzymają dwa identyczne numery portów. Urządzenia w sieci prywatnej nie wiedzą wzajemnie o wykorzystywanych przez nie portach, zatem taka sytuacja może istotnie nastąpić.




Rysunek 7: Schemat komunikacji przy losowym doborze portów.

W poniższym przykładzie przyjmijmy, że oba urządzenia wybrały 2222 jako port źródłowy.

W tym przypadku urządzenie korzystając z tabeli translacji nie byłoby w stanie określić do którego urządzenia kierowana jest odpowiedź. Trasownik odbierając odzew na adres publiczny wraz z portem 2222 nie mógłby jednoznaczanie podjąć decyzji do którego urządzenia powinien przekazywać pakiety.




Rysunek 8: Schemat w którym PAT nie jest w stanie jednoznacznie określić odbiorcy

| Adres IP źródłowy | Port źródłowy | Adres IP docelowy | Port docelowy |
|---|---|---|---|
| ... | ... | ... | ... |
| IP publiczne | 2222 | IP prywatne A | 2222 |
| IP publiczne | 2222 | IP prywatne B | 2222 |
| ... | ... | ... | ... |

Tabela 3: Tabela połączeń dla translacji adresów i portów.

Posiada on do wyboru dwie pozycje do których przypisany jest ten sam adres oraz port, przy takich okolicznościach pakiety wracające zostałyby odrzucone, co jest sytuacją zdecydowanie nieporządaną.

## 2.3 NAT i adresacja IPv6

NAT od dłuższego czasu pełnił istotną rolę w zarządzaniu brakiem dostępnych publicznych adresów IPv4. Jednak z postępującym wprowadzeniem protokołu IPv6, który oferuje nieporównywanie większą przestrzeń adresową, translacja adresów sieciowych nie jest powszechnie stosowana, ponieważ jednym z celów projektowych protokołu IPv6 jest przywrócenie pełnej łączności sieciowej. Duża przestrzeń adresowa IPv6 eliminuje potrzebę oszczędzania adresów, a każdemu urządzeniu można przypisać unikalny globalnie rutowalny adres.

Adresy IPv4 składają się z 32-bitowych liczb binarnych, co umożliwia maksymalnie

$$4\ 294\ 967\ 296$$

unikalnych adresów. Jednak z uwagi na rosnące zapotrzebowanie na adresy IP, puli adresów IPv4 szybko się wyczerpała. By temu zapobiec, adresy IPv6 składają się z 128-bitowych liczb binarnych, co daje ogromną przestrzeń adresową wynoszącą

$$340\ 282\ 366\ 920\ 938\ 463\ 463\ 374\ 607\ 431\ 768\ 211\ 456$$

Ponadto w przypadku IPv6 zrezygnowanie ze stosowania translacji adresów może przynieść wiele korzyści:

1. Uproszczona konfiguracja sieci: W IPv6, każde urządzenie może mieć publicznie dostępny adres IP, co upraszcza konfigurację sieci. Urządzenia mogą bezpośrednio komunikować się z innymi urządzeniami w Internecie bez konieczności konfiguracji skomplikowanych reguł NAT. To ułatwia tworzenie połączeń P2P i eliminuje potrzebę zarządzania ograniczeniami NAT.

2. Unikanie problemów z NAT traversal: NAT może wprowadzać trudności w przypadku niektórych aplikacji, takich jak połączenia peer-to-peer, VoIP, transmisje strumieniowe itp. W przypadku IPv6, brak NAT eliminuje te problemy, ponieważ każde urządzenie ma publiczny adres IP i jest bezpośrednio dostępne z zewnątrz.

3. Ułatwiony rozwój sieci: Bez NAT w IPv6, wiele protokołów i aplikacji może działać bez konieczności modyfikacji w celu obsługi NAT. Dzięki temu rozwój sieci i wdrażanie nowych technologii staje się prostsze i bardziej elastyczne.

**Zrezygnowanie z IPv4 i NAT: Przyszłość IoT z IPv6**

Przyszłość Internetu Rzeczy (IoT) oparta na protokole IPv6 zapowiada rewolucję w sposobie, w jaki urządzenia wymieniają dane i komunikują się w sieci. Rezygnacja z IPv4 i NAT otwiera drzwi do pełnej łączności i współpracy w IoT, dzięki ogromnej puli adresowej dostępnej w IPv6.

Wyobraźmy sobie świat, w którym wszystkie urządzenia IoT mają unikalne adresy IP w pełni obsługiwane przez protokół IPv6. W takiej przyszłości, inteligentne domy, inteligentne miasta, przemysł czy rolnictwo precyzyjne mogą wykorzystać potencjał IPv6 do bezpośredniej wymiany danych między sobą.

Zamiast polegać na mechanizmach NAT, które wprowadzają ograniczenia i komplikacje w komunikacji między urządzeniami, IPv6 oferuje prostą i bezpośrednią łączność. Każde urządzenie IoT może mieć swoje własne globalnie routowalne IP, co umożliwia natychmiastową wymianę informacji bez pośrednictwa.

Przykładowo, inteligentne miasto zbudowane w oparciu o IPv6 może zbierać dane z setek tysięcy czujników, które monitorują ruch drogowy, jakość powietrza, oświetlenie ulicznego i wiele innych parametrów. Dzięki pełnej łączności w IPv6, te urządzenia mogą komunikować się ze sobą bez przeszkód, umożliwiając dynamiczną optymalizację systemów zarządzania miastem.

Taka przyszłość IoT z IPv6 zapowiada nie tylko zbieranie i analizę danych, ale także integrację różnych dziedzin życia w harmonijną i efektywną całość. Inteligentne domy, samochody autonomiczne, systemy monitoringu zdrowia - wszystkie te dziedziny mogą korzystać z pełnej łączności i bezpośredniej komunikacji, co stwarza ogromne możliwości dla innowacji i poprawy jakości życia.

## 2.4 NAT - dodatkowe zabezpieczenie

NAT mimo swojego pierwotnego założenia może również pełnić rolę w zapewnianiu dodatkowego bezpieczeństwa w sieci.

Ukrycie adresów IP sieci wewnętrznej: NAT umożliwia maskowanie prawdziwych adresów IP urządzeń znajdujących się w sieci wewnętrznej (prywatnej) za jednym adresem publicznym.

Filtrowanie pakietów: NAT działa jako punkt kontrolny dla ruchu sieciowego. Może być skonfigurowany do filtrowania pakietów na podstawie adresów źródłowych i docelowych, numerów portów oraz stanu połączenia.

Separacja sieci: NAT pozwala na podział sieci na podsieci, co pomaga w utrzymaniu izolacji urządzeń. Można skonfigurować różne podsieci wewnętrzne z różnymi adresami IP, a NAT zapewni komunikację między nimi.

Ograniczenie dostępu do usług: NAT może być skonfigurowany w taki sposób, aby przekierowywać tylko określone usługi lub porty do odpowiednich urządzeń w sieci wewnętrznej. Pozwala to na ograniczenie dostępu do usług lub zasobów wewnętrznych tylko do uprawnionych urządzeń lub adresów IP.

## 2.5 Perforacja NAT

W dzisiejszych czasach coraz więcej aplikacji i usług wymaga bezpośredniej komunikacji między urządzeniami w sieci. W przypadku, gdy te urządzenia znajdują się za trasownikami z zastosowanym mechanizmem NAT, ustanowienie takiego bezpośredniego połączenia może stanowić wyzwanie.

Komunikacja P2P, która odbywa się bez udziału serwera pośredniczącego, wymaga bezpośredniego połączenia między urządzeniami, co pozwala na szybką i efektywną wymianę danych. Jednak NAT może wprowadzać restrykcje, uniemożliwiające bezpośrednią komunikację między nimi, na przykład poprzez blokowanie przychodzących połączeń. W takiej sytuacji konieczne jest zastosowanie odpowiednich technik, które umożliwią "perforację" NAT i umożliwią bezpośrednie połączenie.

Istnieje kilka popularnych technik, które są stosowane w celu pokonania ograniczeń NAT i ustanowienia bezpośredniego połączenia między hostami. Protokoły takie jak STUN, TURN, UPnP oraz ICE są powszechnie wykorzystywane w aplikacjach komunikacyjnych, takich jak wideokonferencje, czat wideo czy transmisje strumieniowe. Każda z tych technik ma swoje własne cechy, wady i zalety, które należy uwzględnić przy wyborze odpowiedniego rozwiązania dla konkretnego przypadku.

Perforacja NAT, znana również jako NAT traversal lub NAT piercing, to technika umożliwiająca bezpośrednią komunikację między maszybanu znajdującymi się za różnymi NAT-ami. NAT wprowadza pewne ograniczenia i utrudnienia w komunikacji sieciowej, zwłaszcza w przypadku protokołów peer-to-peer i niektórych aplikacji, które wymagają bezpośredniego połączenia.

"W celu obejścia tych ograniczeń istnieje wiele technik, takich jak STUN (Session Traversal Utilities for NAT), TURN (Traversal Using Relays around NAT), UPnP (Universal Plug and Play) oraz ICE (Interactive Connectivity Establishment). Każda z tych technik ma swoje unikalne cechy, zalety i wady. Jeśli interesuje Cię pogłębienie lub lepsze zrozumienie tych technik, zachęcam do odwiedzenia bardzo dobrego artykułu pod tytułem "How NAT traversal works"[1]"

### 2.5.1 STUN (Session Traversal Utilities for NAT)

STUN (Session Traversal Utilities for NAT) jest protokołem, który służy do pokonywania ograniczeń sieciowych, takich jak NAT (Network Address Translation) i umożliwiania bezpośredniej komunikacji między urządzeniami w sieci. STUN jest szczególnie przydatny, gdy chcemy zestawić połączenie P2P (peer-to-peer), gdzie bezpośrednią komunikację między urządzeniami utrudniają NATy.

W przypadku NAT, gdzie wiele urządzeń wewnętrznych jest ukryte za jednym adresem publicznym, STUN umożliwia odkrycie zewnętrznego adresu IP i portu, które są niezbędne do nawiązania bezpośredniego połączenia. Urządzenie korzystające z protokołu STUN wysyła zapytanie do serwera STUN, który zwraca odpowiedź zawierającą informacje o zewnętrznym adresie IP i porcie. Te informacje mogą być następnie przekazane innym urządzeniom w celu nawiązania bezpośredniego połączenia.

- Wymaga serwera STUN do inicjalizacji połączenia.
- Działa jako pośrednik przy inicjalizacji komunikacji między urządzeniami za NAT.
- Pozwala na odkrycie zewnętrznego adresu IP i portu urządzenia.
- Umożliwia przekazanie informacji o zewnętrznym adresie innym urządzeniom w celu bezpośredniej komunikacji.
- Zależność od dostępności publicznych serwerów STUN, co może wprowadzać pewne ograniczenia związane z niezawodnością i wydajnością.

### 2.5.2 TURN (Traversal Using Relays around NAT)

TURN działa jako pośrednik, który przekierowuje pakiety między urządzeniami znajdującymi się za NAT-ami. Gospodarz nawiązuje połączenie z serwerem TURN i przesyła do niego pakiety, które są następnie przekazywane do odpowiedniego odbiorcy.

- TURN działa jako pełnoprawny pośrednik, który przekierowuje pakiety między urządzeniami znajdującymi się za NAT-ami.
- Protokół TURN wymaga dedykowanego serwera, który pełni rolę pośrednika w komunikacji.
- Przekierowywanie pakietów przez serwer TURN może wprowadzać opóźnienia w transmisji.

### 2.5.3 UPnP (Universal Plug and Play)

UPnP (Universal Plug and Play) to protokół, który umożliwia urządzeniom w sieci automatyczną konfigurację portów na urządzeniach obsługujących NAT. Dzięki UPnP, maszyny w sieci mogą komunikować się z trasownikiem, który obsługuje ten protokół, w celu otwarcia odpowiednich portów i przekierowania ruchu do urządzeń, które go potrzebują.

- Brak potrzeby zewnętrznego serwera.
- Bezpieczeństwo UPnP zależne od jakości implementacji w urządzeniach i routerach. Słabe lub niewłaściwie skonfigurowane mechanizmu UPnP mogą stworzyć potencjalne luki w zabezpieczeniach sieciowych.
- Dynamiczne mapowanie portów: UPnP umożliwia dynamiczne mapowanie portów w urządzeniach sieciowych, co pozwala na skuteczne przekierowanie ruchu do odpowiednich urządzeń w sieci. Dzięki temu, komunikacja między urządzeniami w sieci staje się możliwa, nawet jeśli są one chronione przez NAT.

### 2.5.4 ICE (Interactive Connectivity Establishment)

ICE to protokół, który wykorzystuje kombinację technik, takich jak STUN, TURN i negocjacji połączeń, w celu znalezienia optymalnej ścieżki komunikacji między urządzeniami znajdującymi się za NAT-ami. ICE dokonuje wyboru najlepszej dostępnej metody, aby zapewnić bezpośrednie połączenie, jeśli to możliwe, lub skorzystać z serwera TURN jako rozwiązania awaryjnego.

- Wieloetapowe podejście: ICE korzysta z wieloetapowego procesu negocjacji połączenia. Urządzenia wymieniają informacje na temat swojej lokalizacji, adresów IP oraz portów, a następnie próbują ustalić najlepszą dostępną ścieżkę komunikacyjną.
- Szyfrowanie: Protokół ICE wspiera szyfrowanie połączeń
- Kompatybilność z różnymi protokołami: ICE jest zaprojektowany tak, aby działał z różnymi protokołami komunikacyjnymi
- Adaptacyjność: Protokół ICE jest elastyczny i adaptuje się do zmieniających się warunków sieciowych. Jeśli występują zmiany w infrastrukturze sieciowej, na przykład zmiana adresu IP lub portów, ICE jest w stanie dostosować się do tych zmian i utrzymać stabilne połączenie.

## 2.6 Dostępne rozwiązania NAT

Istnieje wiele możliwości skorzystania i zaobserwowania translacji adresów przy użyciu NAT w różnych środowiskach wirtualizacyjnych oraz na systemie Linux z wykorzystaniem nftables/iptables.

W dalszej części tego sprawozdania zdecydowałem się skorzystać z opcji implementacji NAT przy użyciu nftables na mojej maszynie z systemem Linux. Samodzielnie skonfigurowałem odpowiednie reguły przekładania adresów, aby umożliwić wirtualnym maszynom korzystanie z mojej sieci lokalnej i uzyskiwanie dostępu do zewnętrznych zasobów sieciowych.

### 2.6.1 Hyper-V

W środowisku Hyper-V, dostępny jest wbudowany mechanizm NAT, który umożliwia przekładanie adresów sieciowych wewnątrz wirtualnej sieci. Dzięki temu wirtualne maszyny mogą korzystać z jednego adresu IP routera, aby nawiązywać połączenia z zewnętrznymi zasobami sieciowymi. NAT w Hyper-V można skonfigurować za pomocą narzędzi zarządzania Hyper-V lub przy użyciu PowerShell'a. Ogólny przegląd wraz z szczegółowym opisem znaleźć można w dokumentacji Hyper-V [3].

### 2.6.2 VirtualBox

Podobnie jak w przypadku Hyper-V, wirtualizator VirtualBox udostępnia opcję translacji adresów sieciowych, która umożliwia wirtualnym maszynom komunikację z zewnętrzną siecią. Konfiguracja tej funkcji jest możliwa poprzez ustawienia sieciowe dla każdej maszyny wirtualnej.

Dokumentacja VirtualBox'a zawiera pełen zestaw komend i opcji, które umożliwiają elastyczną konfigurację usługi NAT. Poniżej przedstawiam kilka przykładów komend, które mogą być w tej kewsti przydatne:

- Aby utworzyć sieć NAT o nazwie `natnet1` z określonym adresem sieci (np. "192.168.15.0/24") i ją włączyć, możesz skorzystać z komendy:

  ```
  VBoxManage natnetwork add -netname natnet1 -network "192.168.15.0/24" -enable
  ```

- Jeśli chcesz dodać serwer DHCP do istniejącej sieci NAT o nazwie `natnet1`, co umożliwi dynamiczną konfigurację adresów IP w maszynach wirtualnych, skorzystaj z polecenia:

  ```
  VBoxManage natnetwork modify -netname natnet1 -dhcp on
  ```

- Aby uruchomić usługę sieci NAT o nazwie `natnet1` wraz z przypisanym serwerem DHCP, co umożliwi wirtualnym maszynom korzystanie z sieci zewnętrznej, użyj komendy:

  ```
  VBoxManage natnetwork start -netname natnet1
  ```

Należy jednak pamiętać, że powyższe przykłady stanowią jedynie niewielki fragment możliwości konfiguracyjnych wirtualnej sieci NAT w VirtualBox. Aby uzyskać więcej informacji i szczegółów, warto zapoznać się z oficjalną dokumentacją VirtualBox'a [4].

### 2.6.3 Linux z nftables

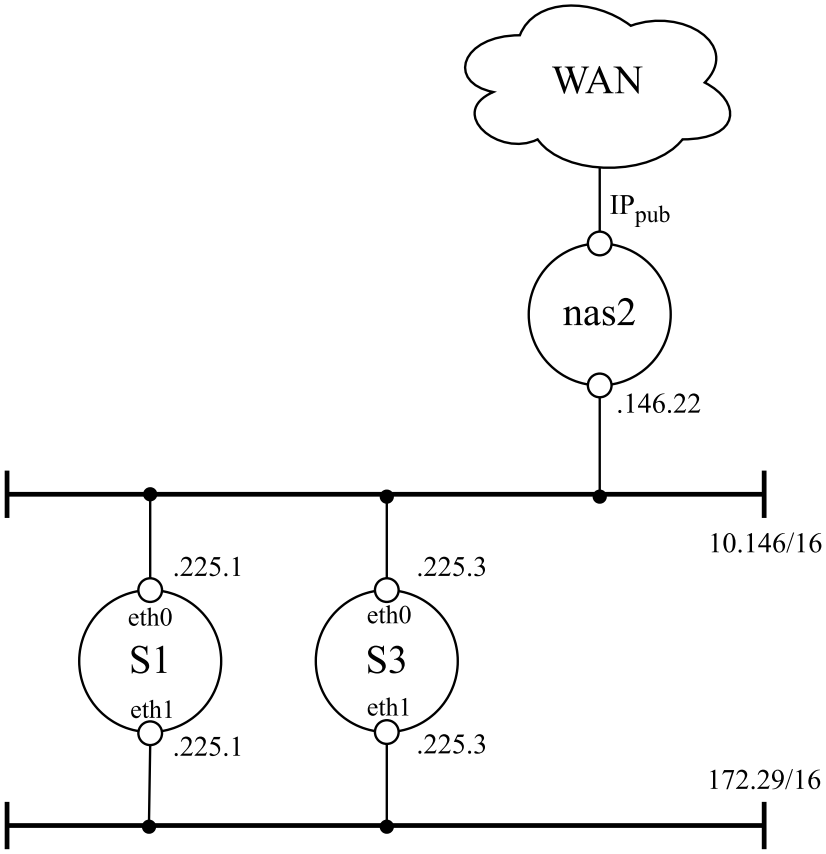
Alternatywnie, na systemie Linux możemy skorzystać z narzędzi takich jak nftables lub iptables do implementacji NAT. Te narzędzia umożliwiają tworzenie reguł przekładania adresów i translacji portów, co daje nam pełną kontrolę nad procesem translacji adresów sieciowych.

# 3 Prezentowanie działania NAT

## 3.1 Założenia oraz plan działania

W ramach prezentacji działania mechanizmu NAT w środowisku ZeroTier, wykorzystałem dwie maszyny: S1 oraz S3 z system operacyjnym ArchLinux. Celem było przedstawienie procesu translacji adresów oraz analiza komunikacji między nimi. W tej sekcji opiszę założenia oraz plan działania, które przygotowałem przed przystąpieniem do prezentacji.

1. Wybór maszyn:
   - S1: Maszyna S1 została wykorzystana jako urządzenie w sieci, które generuje ruch i symuluje urządzenie podłączone do sieci korzystającej z mechanizmu NAT.
   - S3: Maszyna S3 została wykorzystana jako maszyna odpowiedzialna za wykonanie NAT i przekierowanie ruchu z S1 (oraz opcjonalnie innych maszyn) na zewnętrzną sieć.
2. Analiza schematu sieci: Przed przystąpieniem do prezentacji działania NAT, przeprowadziłem szczegółową analizę schematu sieci, aby zrozumieć, jak maszyny są połączone i jak ruch jest przekierowywany między nimi. Analiza ta umożliwiła mi również odpowiednie skonfigurowanie interfejsów sieciowych oraz ustalenie bram domyślnych.




Rysunek 9: Schemat sieci ZeroTier wykorzystany podczas prezentowania działania NAT

3. Konfiguracja maszyny S1: Aby zapewnić poprawne działanie prezentacji, dokonałem konfiguracji maszyny S1, w tym ustawienie bramy domyślnej na adres IP maszyny S3. Dzięki temu ruch generowany przez S1 został przekierowany do S3, gdzie została wykonana translacja adresów NAT.
4. Konfiguracja maszyny S3: Na maszynie S3 wgrałem poniżej omawiany skrypt, który odpowiada za realizację mechanizmu NAT. Dodatkowo, ustawiłem bramę domyślną na adres IP .146.22 (adres IP nas2), co zapewniło dostęp do internetu z maszyny S3.

## 3.2 Konfiguracja S1

Podczas prezentacji wykonano konfigurację maszyny S1, której głównym celem było ustawienie bramy domyślnej. Przed przystąpieniem do konfiguracji, status trasy domyślnej na maszynie S1 był następujący:

Początkowe ustawienie bram domyślnych dla interfejsó eth1 oraz eth0:

```
seredak@s1 ~ % ip route
default via 172.29.146.22 dev eth1 proto dhcp src 172.29.225.1 metric 100
default via 10.146.146.22 dev eth0 proto dhcp src 10.146.225.1 metric 100
```

Aby ustawić nową bramę domyślną, wykonano następujące polecenie:

```
seredak@s1 ~ % ip route del default
seredak@s1 ~ % sudo ip route add default via 172.29.225.3 dev eth1 metric 50
```

Finalne ustawienie bram domyślnych:

```
seredak@s1 ~ % ip route
default via 172.29.225.3 dev eth1 metric 50
```

W wyniku tych poleceń, trasa domyślna została zmieniona na adres IP 172.29.225.3 z interfejsem eth1, a wartość metryki została ustawiona na 50. Wartość metryki wskazuje na priorytet trasy - im mniejsza wartość metryki, tym wyższy priorytet. W tym przypadku, wartość 50 została wybrana jako niższa od wartości 100, aby nadać nowej trasie domyślnej wyższy priorytet w porównaniu do poprzednich tras.

## 3.3 Konfiguracja S3

W celu zaprezentowania działania NAT, na maszynie S3 został wykonany wcześniej przygotowany skrypt. Poniżej przedstawiam cztery kroki konfiguracyjne oraz wyniki poszczególnych poleceń.

### 3.3.1 Krok 1: Uruchomienie skryptu

Na początku wykonujemy skrypt, który skonfiguruje odpowiednie reguły NAT. Omówienie skryptu znajduje się w dalszej części sprawozdania.

Użyte reguły:

```
table ip NAT
table ip NAT {
  chain NAT {
    type nat hook postrouting priority srcnat; policy accept;
    oifname "eth0" masquerade
  }
}
```

### 3.3.2 Krok 2: Testowanie połączenia (brak dostępu do internetu)

Po wykonaniu skryptu, testujemy połączenie zewnętrzne, na przykład z serwerem Google.com, za pomocą polecenia `ping google.com`.

```
seredak@s3 ~ % ip route list
10.146.0.0/16 dev eth0 proto kernel scope link src 10.146.225.3
172.29.0.0/16 dev eth1 proto kernel scope link src 172.29.225.3
seredak@s3 ~ % ping google.com
ping: connect: Sieć jest niedostępna
```

W wyniku testu otrzymujemy komunikat "Sieć jest niedostępna", co wskazuje na brak połączenia z internetem.

### 3.3.3 Krok 3: Ustawienie bramy domyślnej

Aby umożliwić dostęp do internetu, ustawiamy maszynie S3 bramę domyślną dla eth0 taką jak przed wykonaniem skryptu (`10.146.146.22`). Wykonując polecenie:

```
seredak@s3 ~ % sudo ip route add 0.0.0.0/0 via 10.146.146.22 dev eth0 metric 100
```

### 3.3.4 Krok 4: Testowanie połączenia (dostępny internet)

Po ustawieniu bramy domyślnej, ponownie testuję połączenie zewnętrzne z serwerem Google.com. Poniżej przedstawiamy wynik tego kroku.

```
seredak@s3 ~ % ping google.com
PING google.com (216.58.215.110) 56(84) bytes of data.
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=1 ttl=118 time=5.51 ms
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=2 ttl=118 time=5.63 ms
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=3 ttl=118 time=5.42 ms
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=4 ttl=118 time=6.11 ms
```

W wyniku testu otrzymujemy odpowiedzi z serwera Google.com, co potwierdza dostępność internetu na maszynie S3.

Warto zauważyć, że brama domyślna została ustawiona na adres IP 10.146.146.22, co umożliwiło dostęp do internetu. Bez ustawienia bramy domyślnej, translacja adresów NAT nadal by działała, ale brak dostępu do internetu uniemożliwiłby komunikację z serwerami spoza sieci lokalnej.

### 3.4 Analiza skryptu

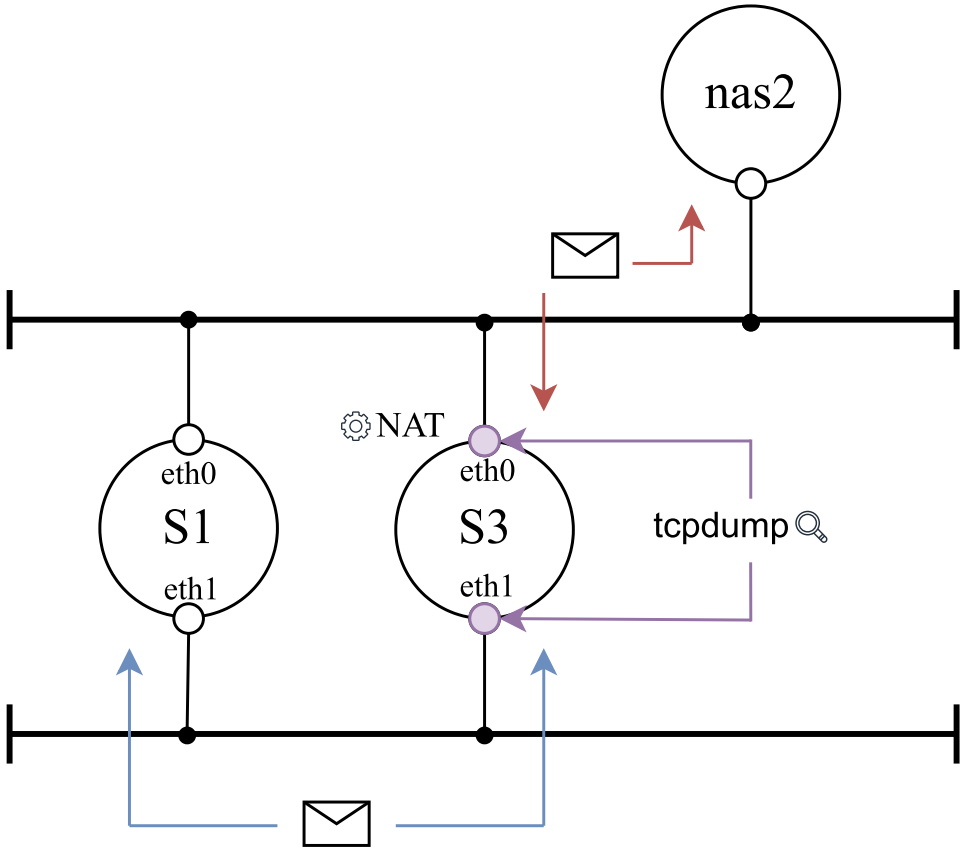
1. Usunięcie wszystkich istniejących adresów z interfejsu WAN.
2. Przypisanie nowego adresu IP do interfejsu WAN.
3. Usunięcie wszystkich istniejących adresów z interfejsu LAN.
4. Przypisanie nowego adresu IP do interfejsu LAN.
5. Włączenie przekazywania pakietów w jądrze.
6. Tworzenie kopii zapasowej oryginalnego pliku konfiguracyjnego `nftables.conf` o nazwie `nftables.conf-org`.
7. Tworzenie nowego pliku `nftables.conf` zawierającego reguły NAT dla interfejsu WAN.
8. Kopiowanie pliku `nftables.conf` do odpowiedniego katalogu konfiguracyjnego.
9. Uruchomienie usługi `nftables`.

Całość bazuje na skrypcie dostępnym w `/bin/config/nat-nftables` dla ArchLinux. Wykorzystany w sprawozdaniu skrypt dostępny jest w serwisie GitHub [5].

Istnieje wiele możliwości rozwoju i konfiguracji reguł bazujących na powyższym skrypcie. Możliwe jest jego dostosowywanie pod dany przypadek użycia konfigurując adresy IP czy zasady w tabeli. Możemy dostosować argument "IP" w linii "IP=ip" w zależności od naszych potrzeb. Przykładowo, możemy użyć "IP=ip" dla reguł dotyczących pakietów IPv4, "IP=inet" dla reguł dotyczących pakietów IPv4 i IPv6, a także innych argumentów opisanych w dokumentacji ArchLinux - nftables [2].

### 3.5 Nasłuchiwanie ruchu - tcpdump

Podczas eksperymentów z narzędziem tcpdump udało mi się zaobserwować ruch sieciowy oraz translację adresów NAT. Z urządzenia S1 został wysłany ping (google.com), który został skierowany na domyślną bramę S3, a następnie dokonana została translacja, co pokazuję i objaśniam poniżej.




Rysunek 10: Istotna część sieci wykorzystana przy nasłuchiwaniu ruchu i dalszej analizie

```
seredak@s3 ~ % sudo tcpdump -i any icmp -nn
```

W wyniku tego otrzymałem następujące zdarzenia ruchu sieciowego na maszynie S3 wykonującej NAT:

```
eth1 In  IP 172.29.225.1   > 192.178.25.174: ICMP echo request, id 10, seq 1, length 64
eth0 Out IP 10.146.225.3   > 192.178.25.174: ICMP echo request, id 10, seq 1, length 64
eth0 In  IP 192.178.25.174 > 10.146.225.3:   ICMP echo reply,   id 10, seq 1, length 64
eth1 Out IP 192.178.25.174 > 172.29.225.1:   ICMP echo reply,   id 10, seq 1, length 64
```

### Analiza ruchu

W trakcie analizy ruchu sieciowego można jednoznacznie stwierdzić, że miała miejsce translacja adresów sieciowych (NAT). Poniżej przedstawiam zaobserwowany wyżej ruch sieciowy wraz z wyjaśnieniem:

| Kolejność | Opis |
|---|---|
| 1 | Maszyna S1 wysyła pakiet ping (ICMP echo request) o adresie źródłowym S1 (172.29.225.1) i adresie docelowym serwera zewnętrznego (192.178.25.174). |
| 2 | Pakiet ping zostaje odebrany na interfejsie `eth1` urządzenia pośredniczącego (S3). |
| 3 | Następuje translacja adresów, gdzie adres źródłowy (S1) zostaje zamieniony na adres interfejsu `eth0` urządzenia pośredniczącego (10.146.225.3), a adres docelowy pozostaje niezmieniony (192.178.25.174). |
| 4 | Zmodyfikowany pakiet ping zostaje wysłany z interfejsu `eth0` urządzenia pośredniczącego do serwera zewnętrznego. |
| 5 | Serwer zewnętrzny odbiera pakiet ping od adresu źródłowego urządzenia pośredniczącego (192.178.25.174) i odpowiada pakietem echo reply (ICMP echo reply). |
| 6 | Otrzymany pakiet echo reply zostaje odebrany na interfejsie `eth0` urządzenia pośredniczącego. |
| 7 | Kolejny etap translacji adresów ma miejsce, gdzie adres źródłowy (serwer zewnętrzny) zostaje zamieniony na adres interfejsu `eth1` urządzenia pośredniczącego (192.178.25.174), a adres docelowy pozostaje niezmieniony (S1). |

Tabela 4: Ilustracja zdarzeń, związana z translacją adresów na S3

Podczas analizy ruchu sieciowego można jednoznacznie stwierdzić obecność translacji adresów sieciowych (NAT). Otrzymane pakiety ping na interfejsie `eth1` oraz wysłane pakiety ping z interfejsu `eth0` są jasnym dowodem na to zjawisko.

Kiedy maszyna S1 wysyłała pakiety ping na serwer zewnętrzny, adres źródłowy w pakietach był zamieniany na adres interfejsu `eth0` urządzenia S3. W rezultacie, z perspektywy maszyny S1, wydawało się, że komunikuje się bezpośrednio z urządzeniem S3, ponieważ pakiety opuszczały jej interfejs `eth0` z adresem źródłowym odpowiadającym temu interfejsowi.

Z drugiej strony, serwer zewnętrzny, będący celem komunikacji, odbierał pakiety ping od urządzenia S3. Dzięki translacji NAT, adres źródłowy w przychodzących pakietach był zamieniany na adres interfejsu `eth1` urządzenia S3. Z perspektywy serwera zewnętrznego, wszystkie otrzymane pakiety wydawały się pochodzić bezpośrednio od urządzenia S3, co sugerowało, że komunikuje się on wyłącznie z tym urządzeniem.

W rezultacie, z punktu widzenia maszyny S1, wydawało się, że komunikuje się bezpośrednio z urządzeniem S3, podczas gdy serwer zewnętrzny był przekonany, że prowadzi komunikację wyłącznie z urządzeniem S3. Dzięki translacji NAT, osiągnięto skuteczną komunikację między maszyną S1 a serwerem zewnętrznym, jednocześnie ukrywając prawdziwą topologię sieciową przed serwerem zewnętrznym.

## 4 Podsumowanie

Podsumowując, mam nadzieję, że niniejsze sprawozdanie wniosło istotny wkład w zrozumienie działania Network Address Translation (NAT) oraz ukazało jego praktyczne zastosowanie. Przez omówienie teoretycznych podstaw NAT, procesu translacji adresów IP oraz prezentację głównych zalet i zastosowań tej technologii, sprawozdanie powinno dostarczyć czytelnikowi solidnej wiedzy na ten temat.

W końcowej części sprawozdania przedstawiłem praktyczne zastosowanie NAT, obserwując działanie na dwóch maszynach. Poprzez analizę ruchu między nimi, zaprezentowałem, jak NAT przekształca adresy IP, umożliwiając komunikację między sieciami o różnych adresach.

Wnioskiem płynącym z tego sprawozdania jest to, że NAT jest fascynującym rozwiązaniem, które warto dobrze zrozumieć w dzisiejszym otaczającym nas świecie. Translacja adresów IP odgrywa istotną rolę w rozwiązaniu problemu z brakującą ilością adresów IPv4. Dlatego zrozumienie działania i możliwości konfiguracji tej technologii jest niezwykle ważne dla pełnego wykorzystania jej potencjału.

## Literatura


[1] David Anderson. "How NAT traversal works". In: (2020). URL: https://tailscale.com/blog/how-nat-traversal-works/.

[2] *Dokumentacja ArchLinux - nftables*. URL: https://wiki.archlinux.org/title/nftables.

[3] *Dokumentacja Hyper-V - artykuł o konfiguracja NAT*. URL: https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.

[4] *Dokumentacja VirtualBox - konfiguracja NAT*. URL: https://www.virtualbox.org/manual/ch06.html#network_nat_service.

[5] *Implementacja wykorzystanego skryptu - GitHub*. URL: https://github.com/seredak319/NAT/blob/main/nat-nftables.

[6] W. Richard Stevens. *TCP/IP Illustrated*. Reading, MA: Addison-Wesley, 1994, pp. 303–346.