

NAT64 begrijpen en configureren

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Waarom is NAT64 nodig?](#)

[Hoe kunnen we communicatie tussen IPv4 en IPv6 mogelijk maken?](#)

[Typen NAT64-vertaling](#)

[Scenario 1: Als we willen communiceren met IPv4-server \(gelegen in IPv4-netwerk\) vanaf host in IPv6-netwerk](#)

[Packet flow in geval van stateful NAT64](#)

[Gids voor het configureren van NAT64](#)

[Configuratie op NAT 46-router](#)

[Controleer NAT64-gegevens](#)

[Scenario 2: Verkeer geïnitieerd van IPv4-alleen-clients naar IPv6-alleen-servers](#)

[Gids voor het configureren van NAT46](#)

[Configuratie op NAT 46-router](#)

[Verificatie NAT46](#)

[Omzettingsscenario's en hun toepasbaarheid](#)

[Belangrijke opdrachten voor het oplossen van problemen voor het geval dat we tijdens de NAT64-implementatie een probleem tegenkomen](#)

Inleiding

NAT64 is een mechanisme voor IPv4-to-IPv6-transitie en IPv4-IPv6-coëxistentie. Samen met DNS64 is het primaire doel van NAT64 om een IPv6-only client in staat te stellen om communicatie naar een IPv4-only server te openen. NAT64 kan ook worden gebruikt voor IPv4-alleen-clients die communicatie met IPv6-servers openen met behulp van statische of handmatige verbindingen. Ik heb beide scenario's in dit document toegelicht.

Voorwaarden

Basiskennis van IPv6 en NAT

Waarom is NAT64 nodig?

- Bijna alle moderne IP-apparaten zijn IPv6-compatibel, maar nog veel oudere apparaten zijn IPv4-only. We hebben een manier nodig om deze apparaten via een IPv6-netwerk aan te sluiten.
- Sommige oudere toepassingen waarin IPv4-adressen in de bovenste lagen zijn opgenomen, zullen naar verwachting nog wel even rond zijn en moeten aan IPv6 worden aangepast.
- Aangezien IPv4-adressen niet beschikbaar worden, worden IPv6-adressen aan nieuwe apparaten toegewezen; de meeste bereikbare inhoud op het internet is echter nog IPv4. Deze nieuwe apparaten moeten die inhoud bereiken.

- Na enkele jaren zal het tegenovergestelde van toepassing zijn: Het grootste deel van de inhoud zal IPv6 zijn, maar het weinige resterende IPv4-only apparaat moet het nog bereiken.
- IPv4-apparaten moeten alleen met IPv6-apparaten spreken zonder dat de gebruiker hiervan op de hoogte is.

Hoe kunnen we communicatie tussen IPv4 en IPv6 mogelijk maken?

Aangezien IPv6 niet achterwaarts compatibel is met IPv4, moeten we over de noodzaak van overgangsmechanismen beschikken, die in een van de drie klassen vallen:

- **Dubbele gestapelde interfaces:** De eenvoudigste oplossing voor IPv4- en IPv6-coëxistentie (geen interoperabiliteit) is om interfaces "tweetalig" te maken, zodat ze IPv4-apparaten en IPv6-naar-IPv6-apparaten kunnen spreken. Welke versie zij gebruiken hangt af van de versie van pakketten die zij van een apparaat ontvangen of het type adres DNS geeft ze op wanneer ze een vraag naar een apparaatadres beantwoorden. Dubbele stapel was de bedoelde manier om van IPv4 naar IPv6 over te schakelen, maar de veronderstelling was dat de overgang voltooid zou zijn voordat IPv4 werd uitgeput. Dit is niet gebeurd, dus wordt het stapelen op twee manieren complexer: Hoe geeft u elke interface zowel een IPv4-adres als een IPv6-adres wanneer er niet genoeg IPv4-adressen beschikbaar zijn om rond te gaan?
- **Talen:** De tunnels gaan ook over coëxistentie, niet over interoperabiliteit. Ze staan apparaten of plaatsen van één versie toe om over een netwerksegment te communiceren - met inbegrip van Internet - van de andere versie. Dus twee IPv4 apparaten of plaatsen kunnen IPv4 pakketten over een IPv6 netwerk ruilen, of twee IPv6 apparaten of plaatsen kunnen IPv6 pakketten over een IPv4 netwerk ruilen.
- **Vertalers:** Vertalers maken interoperabiliteit tussen een IPv4-apparaat en een IPv6-apparaat door de kop van een pakket van een versie te veranderen in de header van de andere versie.

#Net als andere overgangsmethoden is vertaling geen langetermijnstrategie en het uiteindelijke doel zou inheemse IPv6 moeten zijn. Vertaling biedt echter twee belangrijke voordelen ten opzichte van tunneling:

- Vertaling biedt een middel om geleidelijk en naadloos over te stappen op IPv6.
- Contentproviders kunnen op transparante wijze diensten aanbieden aan IPv6-internetgebruikers.

Typen NAT64-vertaling

Stateless NAT64

In stateless NAT64 wordt de status niet bewaard, wat betekent dat voor elke IPv6-gebruiker een specifiek IPv4-adres vereist is. Omdat we in de IPv4-uitputtingsfase zijn, is het zeer moeilijk om deze modus van NAT64 over te nemen. Het enige voordeel van het gebruik van stateless NAT64 wanneer u weinig nummers van IPv6-adressen hebt (NAT46).

Stateful NAT64

In stateful NAT64 blijven staten behouden. Een enkel IP-adres wordt gebruikt voor alle privé-

gebruikers met verschillende poortnummers. In het bovenstaande diagram wordt één IPv4-adres gebruikt met verschillende poortnummers voor alle gebruikers van IPv6 die in dat LAN aanwezig zijn om toegang te krijgen tot een openbare IPv4-server.

Hier vindt u meer informatie over het verschil tussen Stateful en Stateless NAT64-vertaling:

Stateless NAT64

1:1 vertaling

Geen behoud van IPv4-adres

Zorgt voor end-to-end adrestransparantie en schaalbaarheid

Geen staat of bindingen die op de vertaling zijn gemaakt

Vereist IPv4-vertaalbare IPv6-adrestoewijzing (verplichte vereiste)

Vereist of handmatige of op DHCPv6 gebaseerde adrestoewijzing voor IPv6-hosts

Stateful NAT64

1:N-vertaling

Hiermee behoudt u IPv4-adres.

Gebruikt adresoverlading, vandaar gebrek in end-to-end adrestransparantie

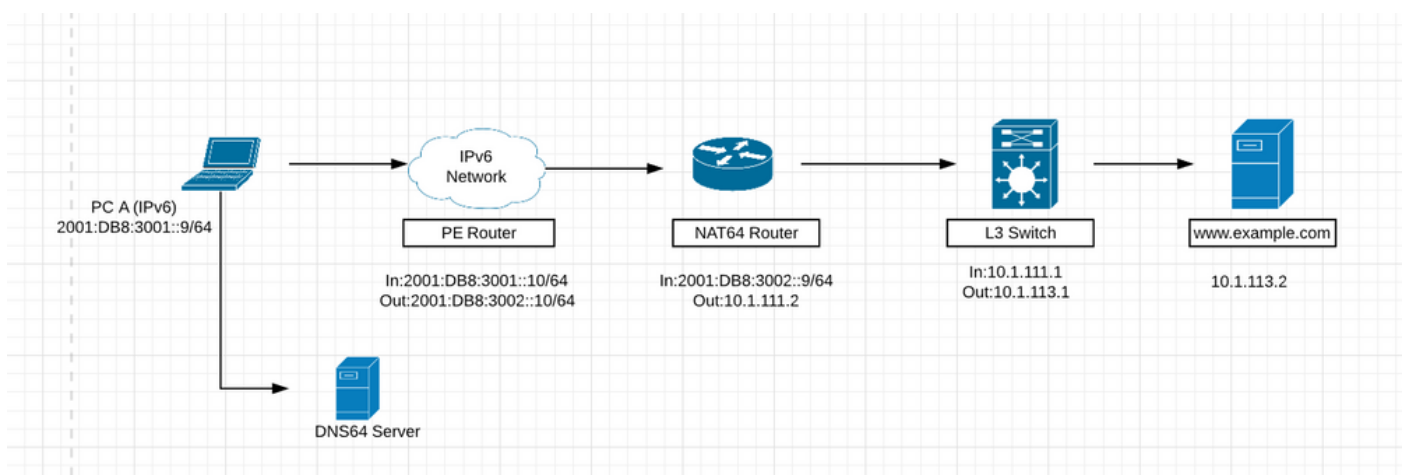
Staat of bindingen worden gemaakt op elke unieke vertaling

Geen vereiste met betrekking tot de aard van IPv6-adrestoewijzing

Gratis om elke modus van IPv6-adrestoewijzing te kiezen, bijv. Handmatig, DHCPv6, SLAAC

- In dit document heb ik stateful NAT64 aangetoond met LAB oefening waar IPv6 host naar IPv4 server wil communiceren. Ook, ik heb stateless NAT64 aangetoond waar IPv4-hosts naar IPv6-server willen bereiken, wordt dit scenario ook NAT46 genoemd.

Scenario 1: Als we willen communiceren met IPv4-server (gelegen in IPv4-netwerk) vanaf host in IPv6-netwerk

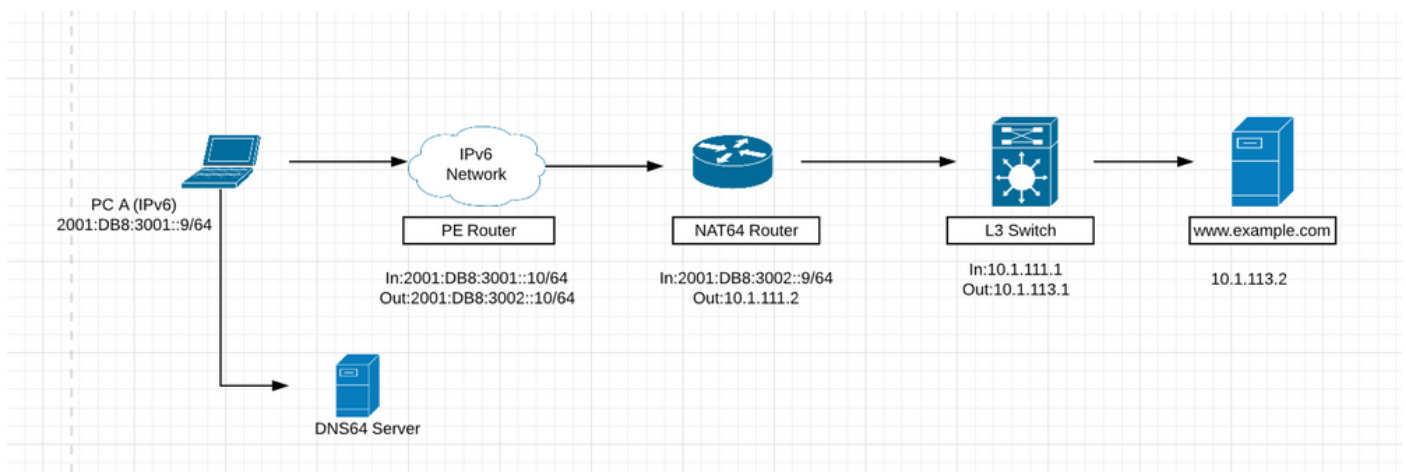


- In het bovenstaande beeld wil Host in het IPv6-netwerk naar een webserver(www.example.com) bereiken, met ip 10.1.113.2 in een ipv4-netwerk.
- Als we het ipv4-adres (10.1.113.2) direct vanaf host in ipv6-netwerk zullen typen, begrijpt het apparaat dit ipv4-adres niet, omdat het alleen ipv6-adressen begrijpt. Dus het pakje wordt op de host gelegd.
- Op dezelfde manier zal het apparaat, als we ipv6-adres van ipv4-netwerk pingelen, dat ip niet begrijpen en het fouten werpen omdat het standaard is ingesteld voor alleen ipv4-netwerk.
- Bovendien kan een ipv4-pakket niet door een ipv6-netwerk worden routeerd en vice versa. Daarom is er behoefte aan vertaling zodat we de pakketten op randapparatuur naar ipv4 of ipv6 kunnen vertalen, afhankelijk van de behoefte.

Er zijn drie hoofdcomponenten voor NAT64

- **prefix van NAT64:** Om het even welke prefix/32,/40,/48,/56,/64, of /96 gebruikt met een geconverteerd IPv4-adres om het pakket via het IPv6-only netwerk te verzenden. Het voorvoegsel NAT64 kan een netwerkspecifiek voorvoegsel (NSP) of een bekend voorvoegsel (WKP) zijn. Een NSP wordt toegewezen door een organisatie en is gewoonlijk een netto van het IPv6 prefix van de organisatie. WKP voor NAT64 is 64:ff9b::/96. Als een NSP niet gespecificeerd of geconfigureerd is, zal NAT64 WKP gebruiken om het geconverteerde IPv4-adres voor te bereiden. Het voorvoegsel NAT64 wordt ook Pref64 genoemd::/n.
- **DNS64-server:** De DNS64-server werkt als een normale DNS-server voor IPv6-AAA-records, maar zal ook proberen een IPv4 A-record te lokaliseren wanneer er geen AAA-record beschikbaar is. Als een record is gevestigd, converteert DNS64 de IPv4 A-record naar een IPv6 AAA-record met behulp van het NAT64-prefix. Dit geeft de indruk op de IPv6-only host dat het met een server kan communiceren met IPv6.
- **NAT64-router:** De NAT64-router adverteert met het NAT64-prefix in het IPv6-only netwerk, samen met het uitvoeren van de vertaling tussen de IPv6-only en IPv4-only netwerken.

Packet flow in geval van stateful NAT64



1. Stel dat de hierboven vermelde beeldhost in IPv6-netwerk wil communiceren met de webserver www.example.com (10.1.113.2), die alleen IPv4-server is.
2. Om deze communicatie mogelijk te maken, moeten we DNS64-server in ons IPv6-netwerk hebben geïnstalleerd die DNS voor IPv4-verzoeken kan begrijpen en oplossen.
3. De DNS-server werkt als een normale DNS-server voor IPv6-AAA-records, maar zal ook proberen een IPv4 A-record te lokaliseren wanneer er geen AAA-record beschikbaar is. Als een record is gevestigd, converteert DNS64 de IPv4 A-record naar een IPv6 AAA-record met behulp van het NAT64-prefix. Dit geeft de indruk op de IPv6-only host dat het met een server kan communiceren met IPv6.
4. Nu wordt de DNS-afwikkelingsaanvraag voor www.example.com naar DNS64-server verzonden. Het kijkt eerst op in zijn IPv6 AAA-opnametabel maar er vindt geen IPv6 AAA-record omdat deze webserver aan IPv4-adres toebehoort. Daarna kijkt het in zijn IPv4 database en vindt het IPv4 adres dat op deze website is afgestemd. Nu zal DNS64-server dit IPv4-adres converteren naar IPv6-adres door dit IPv4-adres om te zetten in hex- en voorvoegsel NAT64 erop te zetten.

Door dit te doen zal dit indruk op IPv6 slechts host geven dat het met een webserver kan communiceren met IPv6.

5. De pakketten worden in het IPv6 enige netwerk naar apparaat routeerd dat NAT64 doet met de hulp van het prefix NAT64 dat werd voorbereid op de hexuitdraai waarde van IPv4 adres.

6. De NAT64-router adverteert met het NAT64-prefix in het IPv6-only netwerk, samen met het uitvoeren van de vertaling tussen de IPv6-only en IPv4-only netwerken.

7. Zodra pakketjes apparaat treffend apparaat dat NAT64-vertaling doet, worden de pakketten afgestemd op ACL die we hebben geconfigureerd voor NAT64. Als pakketten overeenkomen met dit ACL-pakket, wordt het pakket verder vertaald met NAT64 als pakket niet overeenkomt met geconfigureerde ACL, wordt het gerouteerd met normale IPv6-routing naar zijn bestemming.

8. Stateful NAT64 maakt gebruik van geconfigureerde toegangscontrolelijsten (ACL's) en prefix-lijsten om IPv6-geïnitieerde verkeersstromen te filteren die de NAT64-status mogen creëren. Filtering van IPv6-pakketten wordt uitgevoerd in de IPv6-naar-IPv4-richting, omdat dynamische toewijzing van mapping tussen een IPv6-host en een IPv4-adres alleen in deze richting kan worden uitgevoerd. Stateful NAT64 ondersteunt endpointafhankelijke filtering voor de IPv4-to-IPv6-pakketstroom met PAT-configuratie.

9. In een stateful NAT64 PAT-configuratie, moet de pakketstroom uit het IPv6-rijk zijn ontstaan en de staatsinformatie in NAT64-staatstabellen hebben gemaakt. Packets van IPv4-zijde die geen eerder gemaakte toestand hebben, worden opnieuw geworpen. Endpoint-onafhankelijke filtering wordt ondersteund door statische netwerkadresomzetting (NAT) en niet-PAT-configuraties.

Het eerste IPv6-pakket wordt naar de NAT virtuele interface (NVI) geleid, gebaseerd op de automatische routeringsinstelling die is ingesteld voor het stateful prefix. Stateful NAT64 voert een reeks lookups uit om te bepalen of het IPv6-pakket overeenkomt met een van de geconfigureerde toewijzingen op basis van een ACL-raadpleging (toegangscontrolelijst). Gebaseerd op de mapping, wordt een IPv4-adres (en poort) gekoppeld aan het IPv6-doeladres.

Het IPv6-pakket wordt vertaald en het IPv4-pakket wordt gevormd door de volgende methoden te gebruiken:

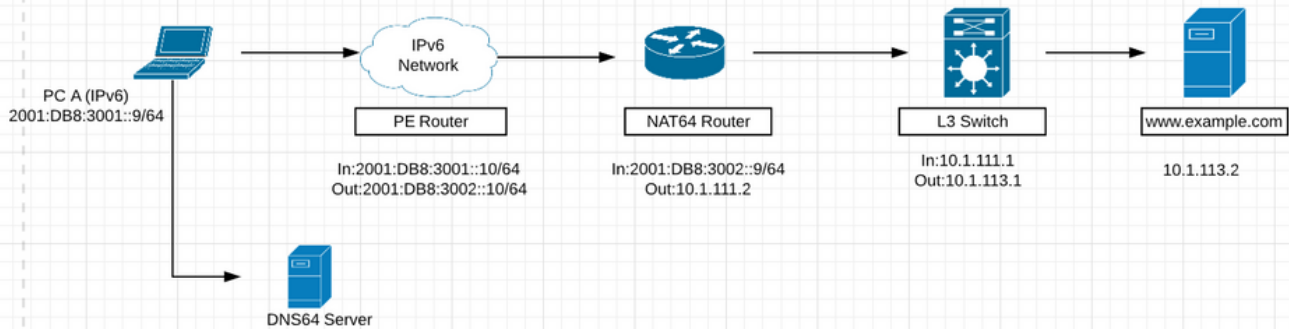
1. Het IPv4-adres van de bestemming verwijderen door het prefix van het IPv6-adres te verwijderen. Het bronadres wordt vervangen door het toegewezen IPv4-adres (en poort).
2. De rest van de velden wordt vertaald van IPv6-naar-IPv4 om een geldig IPv4-pakket te vormen.

10. Een nieuwe NAT64-vertaling wordt gemaakt in de sessiedatabank en in de bind-database. De pool- en poortdatabases worden aangepast naar gelang de configuratie.

11. Het retourverkeer en het daaropvolgende verkeer van de IPv6-pakketstroom gebruiken deze sessiedatabase voor vertaling.

- Om NAT64 te laten werken moet er bereikbaarheid zijn om ipv6-adres van een interface te gebruiken dat zich in ipv6-netwerk bevindt vanuit ipv6 en de bereikbaarheid moet ook van NAT64-router naar ipv4-adres van de server zijn.

Gids voor het configureren van NAT64



Stap 1. Host A is een IPv6-only gastheer die met de server www.example.com wil communiceren. Dit leidt tot een DNS query (AAA): www.example.com op de DNS64-server. De DNS64 is een belangrijk onderdeel van dit proces. Een DNS64-server is zowel een DNS-server voor IPv6 als IPv4. Het maakt de illusie voor de client dat IPv4-servers kunnen worden bereikt met een IPv6-adres.

Host A stuurt een DNS-zoekopdracht (AAA): www.example.com op de DNS64-server. Wat host A betreft, is dit een normale DNS AAA-zoekopdracht voor een IPv6-server.

Stap 2. De DNS-server ontvangt de DNS-AAA-zoekopdracht van host A. In een poging om de domeinnaam op te lossen, stuurt de DNS-64-server een zoekopdracht naar de DNS-AAA-gezaghebbende server voor www.example.com.

Stap 3. De IPv6 DNS AAA-gezaghebbende server keert een reactie terug die erop wijst dat deze geen AAA-resource record heeft voor www.example.com.

Stap 4. Na het ontvangen van een leeg antwoord (name fout) op de AAA-toets, leidt dit tot de DNS64-server om een query (A:) te verzenden www.example.com naar de IPv4 DNS-A-gezaghebbende server.

Stap 5. De IPv4 DNS-A-gezaghebbende server heeft geen resource record voor www.example.com en geeft een reactie op het IPv4-adres voor de server terug (A: www.example.com 10.1.113.2).

Stap 6. De DNS64-server ontvangt het IPv4-adres van de DNS A-gezaghebbende server en synthetiseert een AAA-record door het adres vooraf te bevestigen met het NAT64-prefix, 2800:1503:2000:1:1:/96, en zet het IPv4-adres om in hexadecimaal, 0a01:7102. Dit adres wordt gebruikt door host A als het IPv6-adres van de bestemming voor het bereiken van de www.example.com-server.

Stap 8. Het gesynthetiseerde AAA-record is volledig transparant om A te host. Om A te ontvangen, lijkt het alsof www.example.com bereikbaar is via het IPv6-netwerk en internet. Host A heeft nu de adresinformatie die nodig is om IPv6-pakketten naar www.example.com te verzenden met de volgende informatie:

- IPv6-doeladres: 2800:1503:2000:13:01:01
- IPv6-bronadres: 2001:DB8:3001:

Stap 9. De router NAT64 ontvangt het IPv6-pakket dat door host A wordt verzonden op zijn NAT64-enabled interface. Het komt overeen met de inkomende pakketten in geconfigureerde

ACL. Als de overeenkomst niet gevonden is dan wordt het pakket onvertaald doorgestuurd met behulp van de normale IPv6-routing. Als match wordt gevonden, dan ondergaat het pakket de volgende vertaling:

- De IPv6-header wordt vertaald in een IPv4-header.
- Het IPv6-doeladres wordt vertaald in een IPv4-adres door het IPv6-stateful NAT64-voorvoegsel 2800:1503:2000:1:1:1:/96. te verwijderen. De lagere 32 bits van het IPv6-adres, 0a01:7102, worden weergegeven als het decimale IPv4-adres 10.1.13.2.
- Het IPv6-bronadres wordt vertaald in een IPv4-adres met behulp van de geconfigureerde IPv4-adrespool. Afhankelijk van de NAT64-configuratie kan dit een 1:1-adresvertaling zijn of IPv4-adresoverlading. Dit is vergelijkbaar met NAT voor IPv4. In dit scenario wordt het bron-IPv6-adres van host A vertaald naar het IPv4-adres van 50.50.50.50.
- Stateful NAT64 IP-adresvertaalstatus wordt gecreëerd voor zowel de bron- als de doeladressen. Deze staten worden gecreëerd de eerste keer dat de vertaling op het pakket wordt uitgevoerd. Deze status wordt gehandhaafd voor volgende pakketten in de stroom. De staat eindigt als het verkeer en de staats-onderhoudstimer verlopen.

```
HUB-BR-1#sh nat64 translations
Proto  Original IPv4      Translated IPv4
       Translated IPv6  Original IPv6
-----
icmp   10.1.113.2:2654    [2800:1503:2000:1:1:0:a01:7102]:2654
       50.50.50.50:2654  [2001:db8:3001::9]:2654
Total number of translations: 1
```

Stap 10. Na de vertaling van NAT64 wordt het vertaalde IPv4-pakket verzonden met behulp van het normale IPv4-routeswitchproces. In dit scenario wordt het IPv4-doeladres 10.1.13.2 gebruikt om het pakket door te sturen.

Stap 1. De www.example.com server op 10.1.113.2 antwoordt, die uiteindelijk door de NAT64-router wordt ontvangen.

Stap 12. De NAT64-router ontvangt het IPv4-pakket van de www.example.com-server op een van de NAT64-enabled interfaces. De router onderzoekt het IPv4-pakket om te bepalen of een NAT64-vertaalstatus voor het IPv4-doeladres bestaat. Als er geen vertaalstatus bestaat, wordt het pakje weggegooid. Als er een vertaalstatus bestaat voor het IPv4-doeladres, voert de NAT64-router de volgende taken uit:

- De IPv4-header wordt vertaald in een IPv6-header.
- Het IPv4-bronadres wordt vertaald in een IPv6-bronadres met behulp van de bestaande NAT64-vertaalstatus. In dit scenario wordt het bronadres vertaald van een IPv4-adres van 10.1.13.2 naar het IPv6-adres 2800:1503:2000:1:1:1:01:7102. Het doeladres wordt vertaald vanaf een IPv4-adres 50.50.50 tot 2001:DB8:3001:9.

Stap 13. Na de vertaling wordt het IPv6-pakket verzonden met behulp van het normale IPv6-routeswitchproces.

Configuratie op NAT 46-router

1.IPv6-interface met twee kanten:

```
HUB-BR-1#sh run int gig0/0/1
Building configuration...

Current configuration : 131 bytes
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
 nat64 enable
 cdp enable
 ipv6 address 2001:DB8:3002::9/64
end
```

2. IPv4-interface:

```
HUB-BR-1#sh run int gig0/0/0
Building configuration...

Current configuration : 119 bytes
!
interface GigabitEthernet0/0/0
 ip address 10.1.111.2 255.255.255.0
 negotiation auto
 nat64 enable
 cdp enable
end
```

3. ACL-passend IPv6-verkeer maken

```
HUB-BR-1#sh ipv6 access-list nat64acl
IPv6 access list nat64acl
    permit ipv6 2001:DB8:3001::/64 any sequence 10
HUB-BR-1#
```

4. Stel NAT64 IPv6-to-IPv4-adresomzetting in:

#nat64 voorvoegsel stateful 2800:1503:2000:1:1:/96 → IP-server wordt in kaart gebracht aan dit IP-adres van ipv6. U kunt elk ipv6-netwerkadres hier configureren, maar dit ipv6-netwerkadres moet bereikbaar zijn vanaf uw ipv6-netwerk. Bovendien moet DNS64-server het IPv6-netwerkadres in kaart brengen naar server ipv4-adres.

5. #nat64 v4 pool pool1 50.50.50.50 50.50.50 —> Het oorspronkelijke ipv6-bronadres wordt vertaald naar ips van deze pool terwijl het pakket in ipv4-netwerk wordt ingevoerd.
6. #nat64 v6v4 lijst nat64acl pool1 overload —>Dit vertaalt ipv6-adressen die overeenkomen met nat64acl-adres ipv4-adres uit de pool
7. Hex-waarde van 10.1.13.2 is 0a01:7102.Zodra deze configuratie is uitgevoerd, ping 2800:1503:2000:1:1:1:0a01:7102 adres van PC A.

```
#ping 2800:1503:2000:1:10:01:710
```

Controleer NAT64-gegevens

```
#show nat64 vertaling
```

```
HUB-BR-1#sh nat64 translations
Proto  Original IPv4      Translated IPv4
       Translated IPv6  Original IPv6
-----
icmp   10.1.113.2:7749   [2800:1503:2000:1:1:0:a01:7102]:7749
       50.50.50.50:7749 [2001:db8:3001::9]:7749
Total number of translations: 1
```

```
#show nat64 statistieken
```

```

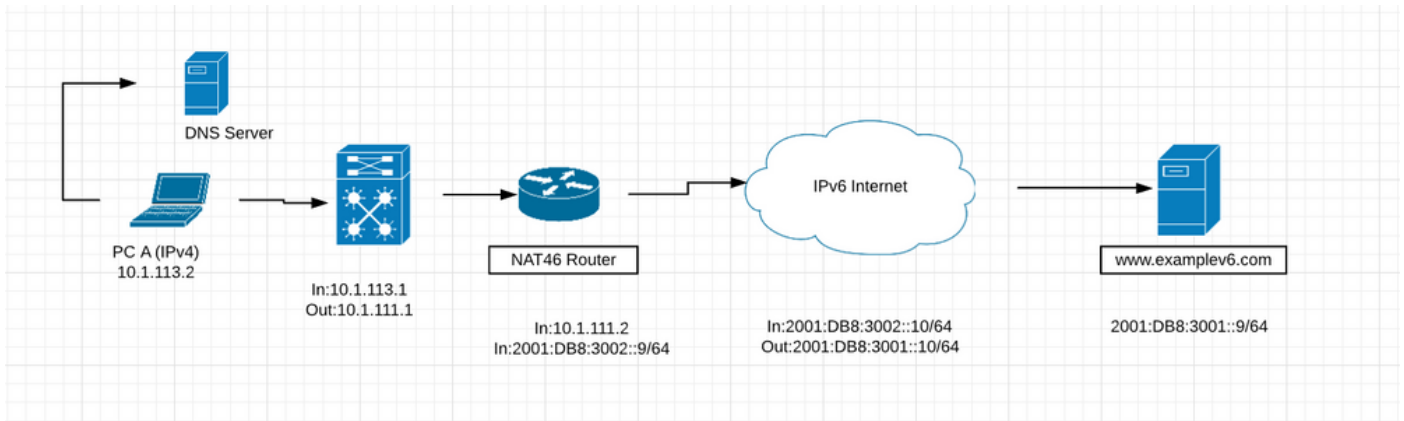
HUB-BR-1#sh nat64 statistics
NAT64 Statistics

Total active translations: 1 (0 static, 1 dynamic; 1 extended)
Sessions found: 33
Sessions created: 4
Expired translations: 4
Global Stats:
  Packets translated (IPv4 -> IPv6)
    Stateless: 0
    Stateful: 18
    MAP-T: 0
  Packets translated (IPv6 -> IPv4)
    Stateless: 0
    Stateful: 20
    MAP-T: 0

Interface Statistics
GigabitEthernet0/0/0 (IPv4 configured, IPv6 not configured):
  Packets translated (IPv4 -> IPv6)
    Stateless: 0
    Stateful: 15
    MAP-T: 0
  Packets translated (IPv6 -> IPv4)
    Stateless: 0
    Stateful: 0
    MAP-T: 0
  Packets dropped: 5
GigabitEthernet0/0/1 (IPv4 not configured, IPv6 configured):
  Packets translated (IPv4 -> IPv6)
    Stateless: 0
    Stateful: 0
    MAP-T: 0
  Packets translated (IPv6 -> IPv4)
    Stateless: 0
    Stateful: 20
    MAP-T: 0
  Packets dropped: 0
Dynamic Mapping Statistics
v6v4
  access-list nat64acl pool pool1 refcount 1
  pool pool1:
    start 50.50.50.50 end 50.50.50.50
    total addresses 1, allocated 1 (100%)
    address exhaustion packet count 0
Limit Statistics

```

Scenario 2: Verkeer geïnitieerd van IPv4-alleen-clients naar IPv6-alleen-servers



- Bovenstaande figuur laat een scenario zien waarin klanten in een IPv4-netwerk communiceren met een IPv6-only server die NAT64 gebruikt. Het doel is toegang te bieden tot IPv6-diensten die transparant zijn voor de IPv4-klanten. In dit scenario is de DNS64-server niet vereist. Statische mapping tussen het IPv6- en IPv4-adres wordt ingesteld op de NAT64-router.
- Houd er rekening mee dat dit scenario in de nabije toekomst onwaarschijnlijk is. De meeste servers die zijn ingeschakeld voor IPv6 kunnen ook IPv4-ondersteuning bieden. Het is waarschijnlijker dat IPv6-servers gedurende enige tijd twee-stapels zullen uitvoeren. IPv6-servers zullen uiteindelijk algemener worden, maar niet snel.

Gids voor het configureren van NAT46

Stap 1. De eerste stap is het configureren van IPv6-naar-IPv4 statische mapping op NAT46-router om toegang te bieden tot de IPv6-server 2001:DB8:3001::9/64 vanaf het IPv4-adres 10.1.113.2. Ook het IPv4-adres 50.50.50.50 moet worden geregistreerd als een DNS-bronrecord voor www.examplev6.com op de DNS-server. De statische NAT64-afbeelding wordt gemaakt met deze opdracht:

```
NAT64-router (configuratie)# nat64 v6v4 statisch 2001:DB8:3001::9 50.50.50.50
```

Stap 2. PC A is een IPv4-only host die met de server wil communiceren www.examplev6.com. Dit veroorzaakt een DNS query (A): www.examplev6.com) naar zijn IPv4 DNS-gezaghebbende server.

Stap 3. De DNS-server reageert met een resource record voor www.examplev6.com, 50.50.50.50.

Stap 4. Host A heeft nu de adresinformatie die nodig is om IPv4-pakketten te verzenden naar www.examplev6.com met

- IPv4-doeladres: 50.50.50.50
- IPv4-bronadres: 10.1.113.2

Stap 5. De NAT64-router ontvangt het IPv4-pakket op de NAT64-enabled-interface en voert de volgende taken uit:

- De IPv4-header wordt vertaald in een IPv6-header.
- Het IPv4-doeladres wordt vertaald in een IPv6-adres met behulp van de bestaande NAT64-vertaalstatus die is gecreëerd door de statische configuratie in Stap 1. Het IPv4-adres van het bestemming van 50.50.50.50 wordt vertaald naar het IPv6-doeladres 2001:DB8:3001:9.

- Het IPv4-bronadres wordt vertaald in een IPv6-adres door het stateful NAT64-prefix 2800:1503:2000:1:1:/96 toe te voegen aan het IPv4-adres. Dit resulteert in een IPv6-bronadres van 2800:1503:2000:1:1:0a01:7102. (0a01:7102 is het hexadecimale equivalent van 10.1.113.2.)

Stap 6. Na de vertaling wordt het IPv6-pakket routeerd met behulp van het normale IPv6-routingproces. Het pakket wordt uiteindelijk op 2001 naar de www.examplev6.com server gestuurd:DB8:3001::9.

Stap 7. De server www.examplev6.com antwoordt met een pakket dat voor host A is bestemd.

Stap 8. De NAT64-router ontvangt het IPv6-pakket dat door de IPv6-server op de NAT64-enabled-interface wordt verstuurd en voert de volgende taken uit:

- De IPv6-header wordt vertaald in een IPv4-header.
- Het IPv6-bronadres wordt vertaald naar 50.50.50.50 met behulp van een stateful vertaaltabel.
- Het IPv6-doeladres wordt vertaald in een IPv4-adres door het IPv6-stateful NAT64-prefix 2800:1503:2000:1:1:/96 te verwijderen. De lagere 32 bits van het IPv6-adres, 0a01:7102, worden weergegeven als het getal decimale IPv4-adres 10.1.13.2.

Stap 9. Na de vertaling wordt de NAT64-router het pakket met het normale IPv4-routingproces naar 10.1.13.2 verzonden.

- Overeenkomstig het vorige scenario wordt er een transparante communicatie tot stand gebracht tussen de IPv4-only client en de IPv6-only server met stateful NAT64. De configuraties zijn vergelijkbaar behalve de statische mapping opdracht die in Stap 1 wordt besproken.

Configuratie op NAT 46-router

1. IPv4-interface:

```
HUB-BR-1#sh run int gig0/0/0
Building configuration...

Current configuration : 137 bytes
!
interface GigabitEthernet0/0/0
 ip address 10.1.111.2 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 nat64 enable
 cdp enable
end
```

2. IPv6-interface met twee kanten:

```

HUB-BR-1#sh run int gig0/0/1
Building configuration...

Current configuration : 131 bytes
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
 nat64 enable
 cdp enable
 ipv6 address 2001:DB8:3002::9/64
end

```

3. Andere formaten nodig op de router om verkeer succesvol van IPv4 naar IPv6 te vertalen:

```

nat64 prefix stateful 2800:1503:2000:1:1::/96
nat64 v6v4 static 2001:DB8:3001::9 50.50.50.50

```

Nadat de configuratie succesvol is, pingelt u 50.50.50.50 van IPv4 host.

```
#ping 50.50.50.50
```

Verificatie NAT46

```
#show nat64 vertalingen
```

```

HUB-BR-1#sh nat64 translations

Proto  Original IPv4      Translated IPv4
       Translated IPv6  Original IPv6
-----
illegal ---
       50.50.50.50   2001:db8:3001::9
icmp   10.1.113.2:11    [2800:1503:2000:1:1:0:a01:7102]:11
       50.50.50.50:11 [2001:db8:3001::9]:11

Total number of translations: 2

```

```
#show nat46 statistieken
```

```

HUB-BR-1#sh nat64 statistics
NAT64 Statistics

Total active translations: 2 (1 static, 1 dynamic; 1 extended)
Sessions found: 9967
Sessions created: 14
Expired translations: 14
Global Stats:
  Packets translated (IPv4 -> IPv6)
    Stateless: 0
    Stateful: 4990
    MAP-T: 0
  Packets translated (IPv6 -> IPv4)
    Stateless: 0
    Stateful: 4992
    MAP-T: 0

Interface Statistics
GigabitEthernet0/0/0 (IPv4 configured, IPv6 not configured):
  Packets translated (IPv4 -> IPv6)
    Stateless: 0
    Stateful: 1947
    MAP-T: 0
  Packets translated (IPv6 -> IPv4)
    Stateless: 0
    Stateful: 0
    MAP-T: 0
  Packets dropped: 58
GigabitEthernet0/0/1 (IPv4 not configured, IPv6 configured):
  Packets translated (IPv4 -> IPv6)
    Stateless: 0
    Stateful: 0
    MAP-T: 0
  Packets translated (IPv6 -> IPv4)
    Stateless: 0
    Stateful: 1947
    MAP-T: 0
  Packets dropped: 0
Dynamic Mapping Statistics
  v6v4
Limit Statistics

```

Omzettingsscenario's en hun toepasbaarheid

Scenario's voor IPv6/IPv4-omzetting	Toepasselijkheid	Voorbeeld
Scenario 1: Een IPv6-netwerk naar het IPv4-internet	<ul style="list-style-type: none"> • IPv6-alleen-netwerk dat op transparante wijze toegang wil hebben tot zowel IPv6- als bestaande IPv4-inhoud • Gearafeerd van IPv6-hosts en -netwerk 	<ul style="list-style-type: none"> • ISP's ontwikkelen nieuwe services en netwerken voor smartphones met IPv6-gebruikers (derde generatie [3G], LTE-evolutie op lange termijn, enz.) • Ondernemingen die IPv6-alleen-netwerk implementeren
Scenario 2: Het IPv4-internet naar een IPv6-netwerk	<ul style="list-style-type: none"> • servers in IPv6-alleen-netwerk die op transparante wijze zowel IPv4- als IPv6-gebruikers willen dienen • Initiatief van IPv4-hosts en -netwerk 	Opkomende of bestaande inhoudaanbieders die diensten uitrollen in IPv6-omgeving
Scenario 3: IPv6-internet naar een IPv4-netwerk	<ul style="list-style-type: none"> • servers in een bestaand IPv4-netwerk dat IPv6-internetgebruikers wil dienen • Gearafeerd van IPv6-hosts en -netwerk 	Bestaande inhoudaanbieders die naar IPv6 migreren en zo diensten willen aanbieden aan IPv6-internetgebruikers als

		onderdeel van de coëxistentiestrategie
Scenario 4: Een IPv4-netwerk naar het IPv6-internet	In de nabije toekomst geen haalbare zaak; dit scenario zal waarschijnlijk pas enige tijd na de vroege fase van de IPv6/IPv4-overgang plaatsvinden	None
Scenario 5: Een IPv6-netwerk naar een IPv4-netwerk	Zowel een IPv4-netwerk als een IPv6-netwerk bevinden zich binnen dezelfde organisatie	Gelijkaardig aan scenario 1, intranet in plaats van internet
Scenario 6: Een IPv4-netwerk naar een IPv6-netwerk	Hetzelfde als hierboven	Gelijkaardig aan scenario 2, intranet in plaats van internet
Scenario 7: Het IPv6-internet naar het IPv4-internet	Zou lijden onder een slechte doorvoersnelheid	None
Scenario 8: Het IPv4-internet naar het IPv6-internet	Geen levensvatbare vertaaltechniek om onbeperkte IPv6-adresomzetting aan te kunnen	None

Belangrijke opdrachten voor het oplossen van problemen voor het geval dat we tijdens de NAT64-implementatie een probleem tegenkomen

```
#show platform hardware qfp actieve statistics drop (om te zien of er nat64 drups zijn)
```

```
#show in werking stellen-configuratie | inclusief nat64 (om te zien of alles op IOS is ingesteld)
```

```
#show platform hardware qfp actieve optie nat64 datapath statistics (om de reden voor druppelteller te controleren)
```

```
#show platform hardware qfp actieve optie nat64 datapath pool (om pool te controleren is correct geconfigureerd)
```

```
#show platform hardware qfp actieve optie nat64 datapath map (om pool te controleren en te zien in mapping configuratie wordt correct uitgevoerd)
```

```
#show platform software object-manager F0 in behandeling-ack-update (om te controleren of er hangende objecten zijn)
```