# Netwerkadresomzetting op een tick

## Inhoud



## Inleiding

Wat bedoelen we met netwerkadresomzetting (NAT) op een stok? Het begrip "op een stok" impliceert gewoonlijk het gebruik van één enkel fysiek interface van een router voor een taak. Net zoals we subinterfaces van dezelfde fysieke interface kunnen gebruiken om Inter-Switch Link (ISL) trunking uit te voeren, kunnen we één fysieke interface op een router gebruiken om NAT te realiseren.

**Opmerking:** de router moet switch van elk pakje verwerken vanwege de loopback-interface. Dit degradeert de prestaties van de router.

## Voorwaarden

### Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

### Gebruikte componenten

Deze optie vereist dat u een versie van Cisco IOS® Software gebruikt die NAT ondersteunt. Gebruik Cisco Functie Navigator II (alleen geregistreerde klanten) om te bepalen welke IOS-versies u met deze functie kunt gebruiken.

### Conventies

Raadpleeg voor meer informatie over documentconventies de technische Tips van Cisco.

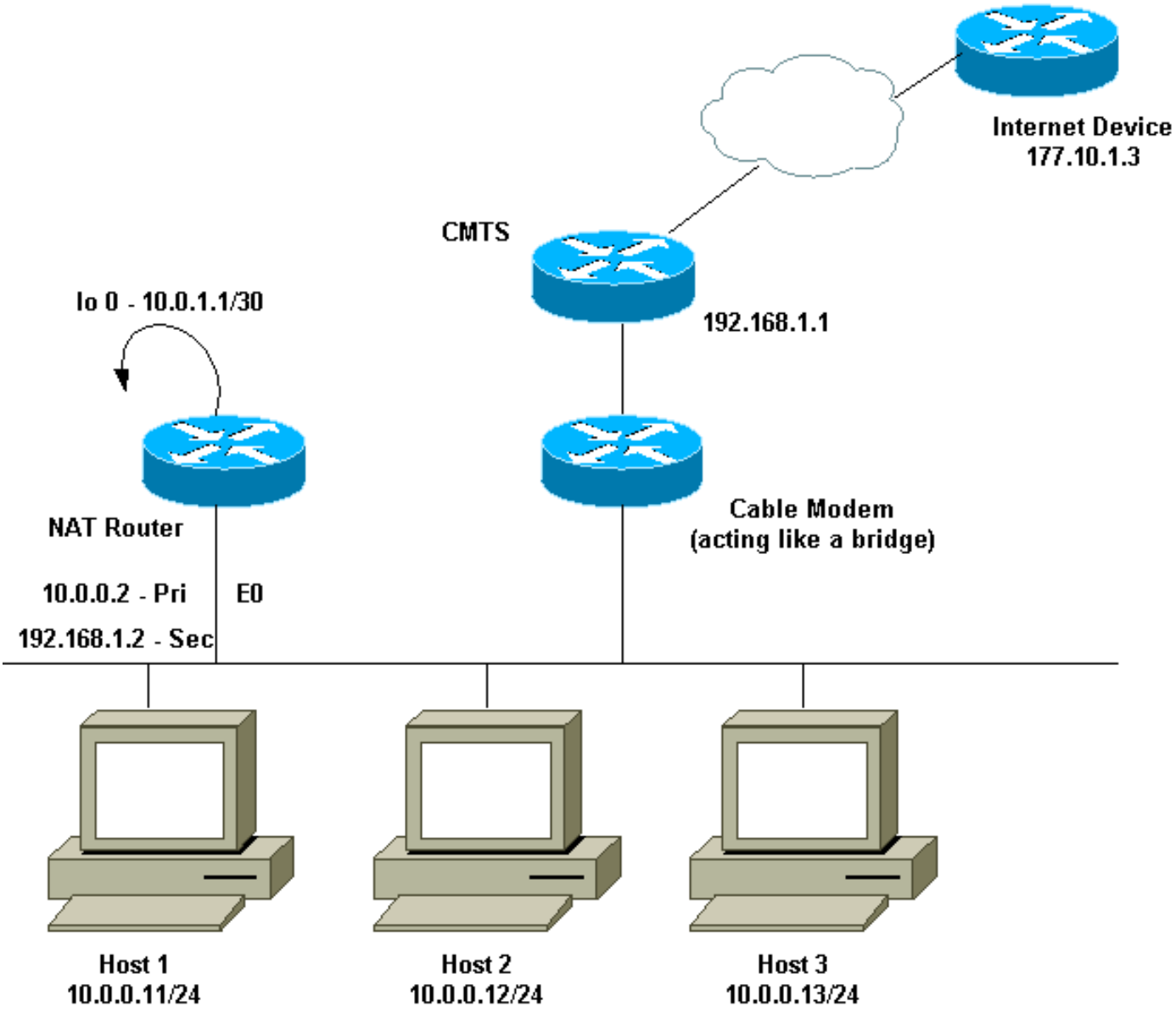
## Achtergrondinformatie

Om NAT te kunnen uitvoeren, moet een pakje zijn overgeschakeld van een door NAT gedefinieerde interface naar een door NAT gedefinieerde "buiten" interface of vice versa. Dit vereiste voor NAT is niet gewijzigd, maar dit document demonstreert hoe u een virtuele interface kunt gebruiken, ook wel bekend als een loopback-interface, en op beleid gebaseerde routing om NAT op een router met één fysieke interface te maken.

NAT is zelden nodig op een stok. In feite zijn de voorbeelden in dit document wellicht de enige situaties waarin deze configuratie nodig is. Hoewel andere gevallen zich voordoen waarin gebruikers beleidsrouting gebruiken in combinatie met NAT, beschouwen we dit niet als NAT op een stok omdat deze gevallen nog steeds meer dan één fysieke interface gebruiken.

## Voorbeeld 1 Netwerkdiagram en -configuratie

### Netwerkdiagram




Het bovenstaande netwerkdiagram komt zeer veel voor in een kabelmodeminstelling. Het Cable Modem Termination System (CMTS) is een router en de Cable Modem (CM) is een apparaat dat werkt als een brug. Het probleem waarmee we worden geconfronteerd, is dat onze Internet Service Provider (ISP) ons niet genoeg geldige adressen heeft gegeven voor het aantal hosts die het internet moeten bereiken. De ISP gaf ons het adres 192.168.1.2, dat voor een apparaat moest worden gebruikt. Op verder verzoek ontvingen we drie meer-192.168.2.1 tot 192.168.2.3—waarin NAT de hosts in het 10.0.0.0/24 bereik vertaalt.

## Vereisten

Onze eisen zijn:

- Alle hosts op het netwerk moeten het internet kunnen bereiken.
- Host 2 moet vanaf het internet kunnen worden bereikt met het IP-adres van 192.168.2.1.
- Omdat we meer hosts dan juridische adressen kunnen hebben, gebruiken we het 10.0.0.0/24-net voor onze interne adressering.

Voor de doeleinden van dit document, tonen we alleen de configuratie van de NAT-router. Maar we noemen wel een paar belangrijke configuratienotities met betrekking tot de hosts.

## NAT-routerconfiguratie

**NAT-routerconfiguratie**

```
interface Loopback0
 ip address 10.0.1.1 255.255.255.252
 ip nat outside
!--- Creates a virtual interface called Loopback 0 and
assigns an !--- IP address of 10.0.1.1 to it. Defines
interface Loopback 0 as !--- NAT outside. ! ! interface
Ethernet0 ip address 192.168.1.2 255.255.255.0 secondary
ip address 10.0.0.2 255.255.255.0 ip Nat inside !---
Assigns a primary IP address of 10.0.0.2 and a secondary
IP !--- address of 192.168.1.2 to Ethernet 0. Defines
interface Ethernet 0 !--- as NAT inside. The 192.168.1.2
address will be used to communicate !--- through the CM
to the CMTS and the Internet. The 10.0.0.2 address !---
will be used to communicate with the local hosts. ip
policy route-map Nat-loop !--- Assigns route-map "Nat-
loop" to Ethernet 0 for policy routing. ! ip Nat pool
external 192.168.2.2 192.168.2.3 prefix-length 29 ip Nat
inside source list 10 pool external overload ip Nat
inside source static 10.0.0.12 192.168.2.1 !--- NAT is
defined: packets that match access-list 10 will be !---
translated to an address from the pool called
"external". !--- A static NAT translation is defined for
10.0.0.12 to be !--- translated to 192.168.2.1 (this is
for host 2 which needs !--- to be accessed from the
Internet).


ip classless
!
!
ip route 0.0.0.0 0.0.0.0 192.168.1.1
ip route 192.168.2.0 255.255.255.0 Ethernet0
 !--- Static default route set as 192.168.1.1, also a
static !--- route for network 192.168.2.0/24 directly
attached to !--- Ethernet 0 ! ! access-list 10 permit
10.0.0.0 0.0.0.255 !--- Access-list 10 defined for use
by NAT statement above.


access-list 102 permit ip any 192.168.2.0 0.0.0.255
access-list 102 permit ip 10.0.0.0 0.0.0.255 any
!--- Access-list 102 defined and used by route-map "Nat-
loop" !--- which is used for policy routing.

!
Access-list 177 permit icmp any any
 !--- Access-list 177 used for debug.

!
route-map Nat-loop permit 10
 match ip address 102
 set ip next-hop 10.0.1.2
!--- Creates route-map "Nat-loop" used for policy
routing. !--- Route map states that any packets that
match access-list 102 will !--- have the next hop set to
10.0.1.2 and be routed "out" the !--- loopback
interface. All other packets will be routed normally. !-
-- We use 10.0.1.2 because this next-hop is seen as
```

```
located !--- on the loopback interface which would
result in policy routing to !--- loopback0.
Alternatively, we could have used "set interface !---
loopback0" which would have done the same thing. ! end
NAT-router#
```

**Opmerking:** alle hosts hebben hun standaardgateway ingesteld op 10.0.0.2, wat de NAT-router is. Zowel de ISP als CMTS moeten een route naar 192.168.2.0/29 hebben die naar de NAT router wijst voor het retourverkeer aan het werk, omdat het verkeer van de binnenhosts verschijnt als het aankomen van dit net. In dit voorbeeld zou CMTS het verkeer voor 192.168.2.0/29 naar 192.168.1.2 leiden, wat het secundaire IP adres is dat op de NAT-router is geconfigureerd.

# Voorbeeld 1 tonen en uitvoeren

Deze sectie verschaft informatie die u kunt gebruiken om te bevestigen dat uw configuratie correct werkt.

Om te illustreren dat de bovenstaande configuratie werkt, hebben we een aantal **ping**-tests uitgevoerd terwijl de **debug**-uitvoer op de NAT-router wordt gevolgd. U kunt zien dat de opdrachten **ping** succesvol zijn en de uitvoer **debug** toont precies wat er gebeurt.

**Opmerking:** Voordat u **debug**-opdrachten gebruikt, raadpleegt u Belangrijke informatie over debug-opdrachten.

## Test één

Voor onze eerste test, **pingelen** we van een apparaat in ons lab-bepaald Internet aan Host 2. Onthoud dat één van de vereisten was dat de apparaten in het Internet met Host 2 moeten kunnen communiceren met het IP-adres van 192.168.2.1. Het volgende is de **debug**-uitvoer zoals die op de NAT-router wordt gezien. De **debug** opdrachten die op de NAT-router actief waren, **debug ip-pakket 177 details** die de gedefinieerde **toegangslijst 177** gebruiken, **debug ip Nat, en debug ip-beleid dat ons de door beleid gestuurde pakketten toont.**

Dit is de output van het **show ip Nat translatie** opdracht uitgevoerd op de NAT router:

```
NAT-router# show ip Nat translation
Pro Inside global      Inside local       Outside local      Outside global
--- 192.168.2.1        10.0.0.12          ---                ---
NAT-router#
```

Van een apparaat op het internet, in dit geval een router, **ping**-192.168.2.1. Dit is een succes zoals hier wordt getoond:

```
Internet-device# ping 192.168.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 92/92/92 ms
Internet-device#
```

Om te zien wat in de NAT router gebeurt, raadpleeg deze **debug**-uitvoer en opmerkingen:

```
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1, len 100, policy match
    ICMP type=8, code=0
IP: route map Nat-loop, item 10, permit
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1 (Loopback0), Len 100, policy routed
    ICMP type=8, code=0
```

*!--- The above* **debug** output shows the packet with source 177.10.1.3 destined !--- to 192.168.2.1. The packet matches the statements in the "Nat-loop" !--- policy route map and is permitted and policy-routed. The Internet !--- Control Message Protocol (ICMP) type 8, code 0 indicates that this !--- packet is an ICMP echo request packet.

```
IP: Ethernet0 to Loopback0 10.0.1.2
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1 (Loopback0), g=10.0.1.2, Len 100,
forward
    ICMP type=8, code=0
```

*!--- The packet now is routed to the new next hop address of 10.0.1.2 !--- as shown above.* IP: NAT enab = 1 trans = 0 flags = 0 NAT: s=177.10.1.3, d=192.168.2.1->10.0.0.12 [52] IP: s=177.10.1.3 (Loopback0), d=10.0.0.12 (Ethernet0), g=10.0.0.12, Len 100, forward ICMP type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 *!--- Now that the routing decision has been made, NAT takes place. We can !--- see above that the address 192.168.2.1 is translated to 10.0.0.12 and !--- this packet is forwarded out Ethernet 0 to the local host. !---* **Note**: When a packet is going from inside to outside, it is routed and !--- then translated (NAT). In the opposite direction (outside to inside), !--- NAT takes place first.

```
IP: s=10.0.0.12 (Ethernet0), d=177.10.1.3, Len 100, policy match
    ICMP type=0, code=0
IP: route map Nat-loop, item 10, permit
IP: s=10.0.0.12 (Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed
    ICMP type=0, code=0
IP: Ethernet0 to Loopback0 10.0.1.2
```

*!--- Host 2 now sends an ICMP echo response, seen as ICMP type 0, code 0. !--- This packet also matches the policy routing statements and is !--- permitted for policy routing.* NAT: s=10.0.0.12->192.168.2.1, d=177.10.1.3 [52] IP: s=192.168.2.1 (Ethernet0), d=177.10.1.3 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=0, code=0 IP: s=192.168.2.1 (Loopback0), d=177.10.1.3 (Ethernet0), g=192.168.1.1, Len 100, forward ICMP type=0, code=0 IP: NAT enab = 1 trans = 0 flags = 0 *!--- The above output shows the Host 2 IP address is translated to !--- 192.168.2.1 and the packet that results packet is sent out loopback 0, !--- because of the policy based routing, and finally forwarded !--- out Ethernet 0 to the Internet device. !--- The remainder of the* **debug** output shown is a repeat of the previous !--- for each of the additional four ICMP packet exchanges (by default, !--- five ICMP packets are sent when pinging from Cisco routers). We have !--- omitted most of the output since it is redundant.

```
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1, Len 100, policy match
    ICMP type=8, code=0
IP: route map Nat-loop, item 10, permit
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1 (Loopback0), Len 100, policy routed
    ICMP type=8, code=0
IP: Ethernet0 to Loopback0 10.0.1.2
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1 (Loopback0), g=10.0.1.2, Len 100,
forward
    ICMP type=8, code=0
IP: NAT enab = 1 trans = 0 flags = 0
NAT: s=177.10.1.3, d=192.168.2.1->10.0.0.12 [53]
IP: s=177.10.1.3 (Loopback0), d=10.0.0.12 (Ethernet0), g=10.0.0.12, Len 100,
forward
    ICMP type=8, code=0
IP: NAT enab = 1 trans = 0 flags = 0
IP: s=10.0.0.12 (Ethernet0), d=177.10.1.3, Len 100, policy match
    ICMP type=0, code=0
IP: route map Nat-loop, item 10, permit
IP: s=10.0.0.12 (Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed
    ICMP type=0, code=0
IP: Ethernet0 to Loopback0 10.0.1.2
```

```
NAT: s=10.0.0.12->192.168.2.1, d=177.10.1.3 [53]
IP: s=192.168.2.1 (Ethernet0), d=177.10.1.3 (Loopback0), g=10.0.1.2, Len 100,
forward
    ICMP type=0, code=0
IP: s=192.168.2.1 (Loopback0), d=177.10.1.3 (Ethernet0), g=192.168.1.1, Len 100,
forward
    ICMP type=0, code=0
IP: NAT enab = 1 trans = 0 flags = 0
```

## Test twee

Een andere van onze vereisten is om de gastheren de mogelijkheid te geven met het internet te communiceren. Voor deze test, **pingelen** we het Internet apparaat van Host 1. De resulterende **show** en **debug** opdrachten zijn hieronder.

Eerst is de NAT-vertaaltabel in de NAT-router als volgt:

```
NAT-router# show ip Nat translation
Pro Inside global      Inside local       Outside local      Outside global
--- 192.168.2.1        10.0.0.12          ---                ---
NAT-router#
```

Zodra we het **ping** uit Host 1 uitgeven, zien we:

```
Host-1# ping 177.10.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 177.10.1.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 92/92/96 ms
Host-1#
```

We zien hierboven dat de **ping** succesvol was. De NAT-tabel in de NAT-router ziet er nu uit:

```
NAT-router# show ip Nat translation
Pro Inside global      Inside local       Outside local      Outside global
icmp 192.168.2.2:434   10.0.0.11:434      177.10.1.3:434     177.10.1.3:434
icmp 192.168.2.2:435   10.0.0.11:435      177.10.1.3:435     177.10.1.3:435
icmp 192.168.2.2:436   10.0.0.11:436      177.10.1.3:436     177.10.1.3:436
icmp 192.168.2.2:437   10.0.0.11:437      177.10.1.3:437     177.10.1.3:437
icmp 192.168.2.2:438   10.0.0.11:438      177.10.1.3:438     177.10.1.3:438
--- 192.168.2.1        10.0.0.12          ---                ---
NAT-router#
```

In de bovenstaande vertaaltabel van NAT worden nu extra vertalingen weergegeven die het resultaat zijn van de dynamische NAT-configuratie (in tegenstelling tot de statische NAT-configuratie).

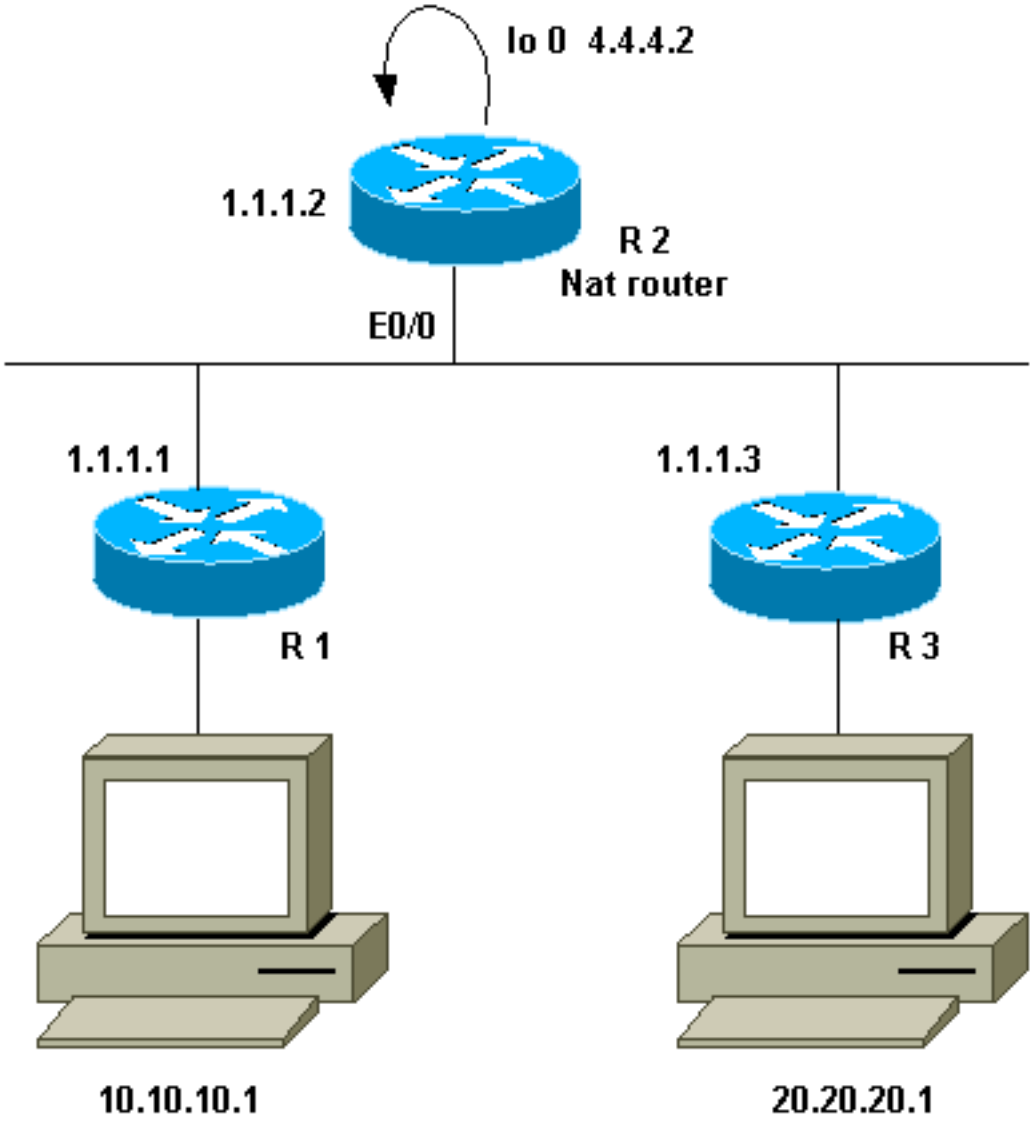
De onderstaande **debug**-uitvoer toont wat op de NAT-router voorkomt.

```
IP: NAT enab = 1 trans = 0 flags = 0
IP: s=10.0.0.11 (Ethernet0), d=177.10.1.3, Len 100, policy match
```

```
    ICMP type=8, code=0
IP: route map Nat-loop, item 10, permit
IP: s=10.0.0.11 (Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed
    ICMP type=8, code=0
IP: Ethernet0 to Loopback0 10.0.1.2
!--- The above output shows the ICMP echo request packet originated by !--- Host 1 which is
policy-routed out the loopback interface. NAT: s=10.0.0.11->192.168.2.2, d=177.10.1.3 [8] IP:
s=192.168.2.2 (Ethernet0), d=177.10.1.3 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=8,
code=0 IP: s=192.168.2.2 (Loopback0), d=177.10.1.3 (Ethernet0), g=192.168.1.1, Len 100, forward
ICMP type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 !--- After the routing decision has
been made by the policy routing, !--- translation takes place, which translates the Host 1 IP
address of 10.0.0.11 !--- to an address from the "external" pool 192.168.2.2 as shown above. !--
- The packet is then forwarded out loopback 0 and finally out Ethernet 0 !--- to the Internet
device. IP: s=177.10.1.3 (Ethernet0), d=192.168.2.2, Len 100, policy match ICMP type=0, code=0
IP: route map Nat-loop, item 10, permit IP: s=177.10.1.3 (Ethernet0), d=192.168.2.2 (Loopback0),
Len 100, policy routed ICMP type=0, code=0 IP: Ethernet0 to Loopback0 10.0.1.2 IP: s=177.10.1.3
(Ethernet0), d=192.168.2.2 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=0, code=0 !---
The Internet device sends an ICMP echo response which matches our !--- policy, is policy-routed,
and forward out the Loopback 0 interface. IP: NAT enab = 1 trans = 0 flags = 0 NAT:
s=177.10.1.3, d=192.168.2.2->10.0.0.11 [8] IP: s=177.10.1.3 (Loopback0), d=10.0.0.11
(Ethernet0), g=10.0.0.11, Len 100, forward ICMP type=0, code=0 !--- The packet is looped back
into the loopback interface at which point !--- the destination portion of the address is
translated from 192.168.2.2 !--- to 10.0.0.11 and forwarded out the Ethernet 0 interface to the
local host. !--- The ICMP exchange is repeated for the rest of the ICMP packets, some of !---
which are shown below. IP: NAT enab = 1 trans = 0 flags = 0 IP: s=10.0.0.11 (Ethernet0),
d=177.10.1.3, Len 100, policy match ICMP type=8, code=0 IP: route map Nat-loop, item 10, permit
IP: s=10.0.0.11 (Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed ICMP type=8,
code=0 IP: Ethernet0 to Loopback0 10.0.1.2 NAT: s=10.0.0.11->192.168.2.2, d=177.10.1.3 [9] IP:
s=192.168.2.2 (Ethernet0), d=177.10.1.3 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=8,
code=0 IP: s=192.168.2.2 (Loopback0), d=177.10.1.3 (Ethernet0), g=192.168.1.1, Len 100, forward
ICMP type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 IP: s=177.10.1.3 (Ethernet0),
d=192.168.2.2, Len 100, policy match ICMP type=0, code=0 IP: route map Nat-loop, item 10, permit
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.2 (Loopback0), Len 100, policy routed ICMP type=0,
code=0 IP: Ethernet0 to Loopback0 10.0.1.2 IP: s=177.10.1.3 (Ethernet0), d=192.168.2.2
(Loopback0), g=10.0.1.2, Len 100, forward ICMP type=0, code=0 IP: NAT enab = 1 trans = 0 flags =
0 NAT: s=177.10.1.3, d=192.168.2.2->10.0.0.11 [9] IP: s=177.10.1.3 (Loopback0), d=10.0.0.11
(Ethernet0), g=10.0.0.11, Len 100, forward ICMP type=0, code=0
```

# Voorbeeld 2 Netwerkdiagram en -configuratie

## Netwerkdiagram




## Vereisten

We willen dat bepaalde apparaten achter de twee sites (R1 en R3) communiceren. De twee sites gebruiken niet-geregistreerde IP-adressen, dus we moeten de adressen vertalen wanneer ze met elkaar communiceren. In ons geval wordt gastvrouw 10.10.10.1 vertaald naar 200.200.1 en gastgastheer 20.20.20.1 wordt vertaald naar 100.100.1. Daarom moeten we in beide richtingen vertalen. Voor boekhoudkundige doeleinden moet het verkeer tussen deze twee locaties door R2 lopen. Samengevat zijn onze eisen:

- Host 10.10.10.1, achter R1, moet met Host 20.20.20.1 achter R3 communiceren met het gebruik van hun globale adressen.
- Verkeer tussen deze hosts moet door R2 worden verstuurd.
- Voor ons geval hebben we statische NAT-vertalingen nodig, zoals in de onderstaande configuratie wordt getoond.

## NAT-routerconfiguratie

| NAT-routerconfiguratie |
| --- |
| |

```
interface Loopback0
 ip address 4.4.4.2 255.255.255.0
 ip Nat inside
!--- Creates a virtual interface called "loopback 0" and
assigns IP address !--- 4.4.4.2 to it. Also defines for
it a NAT inside interface. ! Interface Ethernet0/0 ip
address 1.1.1.2 255.255.255.0 no ip redirects ip Nat
outside ip policy route-map Nat !--- Assigns IP address
1.1.1.1/24 to e0/0. Disables redirects so that packets
!--- which arrive from R1 destined toward R3 are not
redirected to R3 and !--- visa-versa. Defines the
interface as NAT outside interface. Assigns !--- route-
map "Nat" used for policy-based routing. ! ip Nat inside
source static 10.10.10.1 200.200.200.1 !--- Creates a
static translation so packets received on the inside
interface !--- with a source address of 10.10.10.1 will
have their source address !--- translated to
200.200.200.1. Note: This implies that the packets
received !--- on the outside interface with a
destination address of 200.200.200.1 !--- will have the
destination translated to 10.10.10.1.


ip Nat outside source static 20.20.20.1 100.100.100.1
!--- Creates a static translation so packets received on
the outside interface !--- with a source address of
20.20.20.1 will have their source address !---
translated to 100.100.100.1. Note: This implies that
packets received on !--- the inside interface with a
destination address of 100.100.100.1 will !--- have the
destination translated to 20.20.20.1.


ip route 10.10.10.0 255.255.255.0 1.1.1.1
ip route 20.20.20.0 255.255.255.0 1.1.1.3
ip route 100.100.100.0 255.255.255.0 1.1.1.3
!
access-list 101 permit ip host 10.10.10.1 host
100.100.100.1
route-map Nat permit 10
 match ip address 101
 set ip next-hop 4.4.4.2
```

# Voorbeeld 2 tonen en tegenhouden

**Opmerking:** Bepaalde knoppen voor tonen worden ondersteund door het gereedschap Uitvoer-tolk, waarmee u een analyse van de uitvoer van show-opdrachten kunt bekijken. Voordat u **debug**-opdrachten gebruikt, raadpleegt u Belangrijke informatie over Debug Commands.

## Test één

Zoals in de bovenstaande configuratie wordt getoond, hebben we twee statische NAT-vertalingen die op R2 kunnen worden gezien met de opdracht **ip Nat-vertaling tonen**.

Dit is de output van het **show ip Nat translatie** opdracht uitgevoerd op de NAT router:

```
NAT-router# show ip Nat translation
Pro Inside global      Inside local       Outside local      Outside global
--- ---                ---                100.100.100.1      20.20.20.1
--- 200.200.200.1      10.10.10.1         ---                ---
R2#
```

Voor deze test brachten we een **ping** van een apparaat (10.10.10.1) achter R1 dat bestemd is voor het algemene adres van een apparaat (100.100.100.1) achter R3. Het uitvoeren van **debug ip Nat** en het **debug IP pakketje** op R2 resulteerde in deze uitvoer:

```
IP: NAT enab = 1 trans = 0 flags = 0
IP: s=10.10.10.1 (Ethernet0/0), d=100.100.100.1, Len 100, policy match
    ICMP type=8, code=0
IP: route map Nat, item 10, permit
IP: s=10.10.10.1 (Ethernet0/0), d=100.100.100.1 (Loopback0), Len 100, policy
routed
    ICMP type=8, code=0
IP: Ethernet0/0 to Loopback0 4.4.4.2
!--- The above output shows the packet source from 10.10.10.1 destined !--- for 100.100.100.1
arrives on E0/0, which is defined as a NAT !--- outside interface. There is not any NAT that
needs to take place at !--- this point, however the router also has policy routing enabled for
!--- E0/0. The output shows that the packet matches the policy that is !--- defined in the
policy routing statements. IP: s=10.10.10.1 (Ethernet0/0), d=100.100.100.1 (Loopback0),
g=4.4.4.2, Len 100, forward ICMP type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 !--- The
above now shows the packet is policy-routed out the loopback0 !--- interface. Remember the
loopback is defined as a NAT inside interface. NAT: s=10.10.10.1->200.200.200.1, d=100.100.100.1
[26] NAT: s=200.200.200.1, d=100.100.100.1->20.20.20.1 [26] !--- For the above output, the
packet is now arriving on the loopback0 !--- interface. Since this is a NAT inside interface, it
is important to !--- note that before the translation shown above takes place, the router !---
will look for a route in the routing table to the destination, which !--- before the translation
is still 100.100.100.1. Once this route look up !--- is complete, the router will continue with
translation, as shown above. !--- The route lookup is not shown in the debug output.


IP: s=200.200.200.1 (Loopback0), d=20.20.20.1 (Ethernet0/0), g=1.1.1.3, Len 100,
forward
    ICMP type=8, code=0
IP: NAT enab = 1 trans = 0 flags = 0
!--- The above output shows the resulting translated packet that results is !--- forwarded out
E0/0.
```

Dit is de output als resultaat van het responspakket dat van het apparaat achter router 3 is afkomstig en bestemd is voor het apparaat achter router 1:

```
NAT: s=20.20.20.1->100.100.100.1, d=200.200.200.1 [26]
NAT: s=100.100.100.1, d=200.200.200.1->10.10.10.1 [26]
 !--- The return packet arrives into the e0/0 interface which is a NAT !--- outside interface.
In this direction (outside to inside), translation !--- occurs before routing. The above output
shows the translation takes place. IP: s=100.100.100.1 (Ethernet0/0), d=10.10.10.1
(Ethernet0/0), Len 100, policy rejected -- normal forwarding ICMP type=0, code=0 IP:
s=100.100.100.1 (Ethernet0/0), d=10.10.10.1 (Ethernet0/0), g=1.1.1.1, Len 100, forward ICMP
type=0, code=0 !--- The E0/0 interface still has policy routing enabled, so the packet is !---
check against the policy, as shown above. The packet does not match the !--- policy and is
forwarded normally.
```

## Samenvatting

Dit document heeft aangetoond hoe het gebruik van NAT en op beleid gebaseerde routing kan worden gebruikt om een "NAT op een stok"-scenario te creëren. Het is belangrijk om in gedachten te houden dat deze configuratie de prestaties op de router die NAT loopt kan verminderen omdat

de pakketten door de router kunnen worden aangepast.

## Gerelateerde informatie

- NAT-ondersteuningspagina
- Technische ondersteuning - Cisco-systemen