Stick의 네트워크 주소 변환

목차

[소개](#)
[사전 요구 사항](#)
[요구 사항](#)
[사용되는 구성 요소](#)
[표기 규칙](#)
[배경 정보](#)
[예 1 네트워크 다이어그램 및 구성](#)
[네트워크 다이어그램](#)
[요구 사항](#)
[NAT 라우터 컨피그레이션](#)
[예 1 show and debug 명령 출력](#)
[테스트 1](#)
[테스트 2](#)
[예 2 네트워크 다이어그램 및 구성](#)
[네트워크 다이어그램](#)
[요구 사항](#)
[NAT 라우터 컨피그레이션](#)
[예 2 show and debug 명령 출력](#)
[테스트 1](#)
[요약](#)
[관련 정보](#)

소개

NAT(Network Address Translation)를 막대기로 사용하면 무엇을 의미합니까? "on a stick"이라는 용어는 일반적으로 작업에 라우터의 단일 물리적 인터페이스를 사용하는 것을 의미합니다. 동일한 물리적 인터페이스의 하위 인터페이스를 사용하여 ISL(Inter-Switch Link) 트렁킹을 수행할 수 있는 것처럼, 라우터에서 단일 물리적 인터페이스를 사용하여 NAT를 수행할 수 있습니다.

참고: 라우터는 루프백 인터페이스로 인해 모든 패킷을 전환해야 합니다. 이렇게 하면 라우터의 성능이 저하됩니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 기능을 사용하려면 NAT를 지원하는 Cisco IOS® Software 버전을 사용해야 합니다. Cisco Feature Navigator II(등록된 고객만 해당)를 사용하여 이 기능에 사용할 수 있는 IOS 버전을 결정합니다.

표기 규칙

문서 표기 규칙에 대한 자세한 내용은 Cisco 기술 팁 표기 규칙을 참조하십시오.

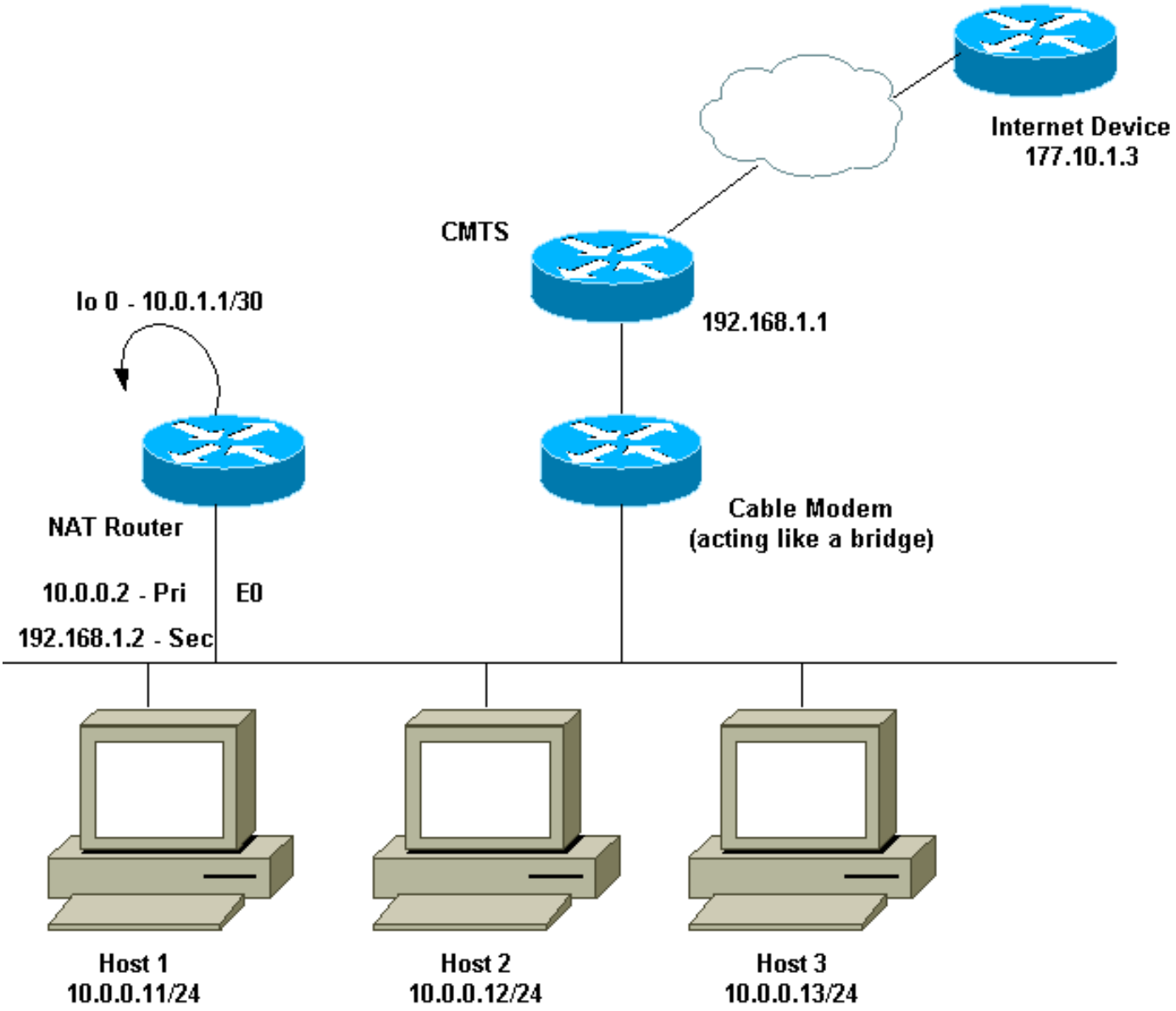
배경 정보

NAT가 발생하려면 NAT "내부" 정의 인터페이스에서 NAT "외부" 정의 인터페이스로 또는 그 반대로 패킷을 전환해야 합니다. NAT에 대한 이 요구 사항은 변경되지 않았지만, 이 문서에서는 가상 인터페이스, 즉 루프백 인터페이스로 알려진 인터페이스 및 정책 기반 라우팅을 사용하여 단일 물리적 인터페이스를 사용하는 라우터에서 NAT를 작동하는 방법을 보여 줍니다.

스틱에 NAT가 필요한 경우는 드물다. 실제로 이 문서의 예시만 이 구성이 필요할 수 있습니다. 사용자가 NAT와 함께 정책 라우팅을 사용하는 다른 경우에도 이러한 인스턴스는 여전히 두 개 이상의 물리적 인터페이스를 사용하므로 이를 스틱에 NAT로 간주하지 않습니다.

예 1 네트워크 다이어그램 및 구성

네트워크 다이어그램



위의 네트워크 다이어그램은 케이블 모뎀 설정에서 매우 일반적입니다. CMTS(Cable Modem Termination System)는 라우터이고 CM(Cable Modem)은 브리지 같은 장치입니다. 문제는 ISP(Internet Service Provider)가 인터넷에 연결해야 하는 호스트 수에 대해 충분한 유효한 주소를 제공하지 않았다는 것입니다. ISP에서 장치에 사용할 192.168.1.2 주소를 제공했습니다. 추가 요청 시 NAT가 10.0.0.0/24 범위의 호스트를 변환하는 3개(192.168.2.1~192.168.2.3)을 추가로 받았습니다.

요구 사항

당사의 요구 사항은 다음과 같습니다.

- 네트워크의 모든 호스트는 인터넷에 연결할 수 있어야 합니다.
- 호스트 2는 IP 주소가 192.168.2.1인 인터넷에서 연결할 수 있어야 합니다.
- 법적 주소보다 더 많은 호스트가 있을 수 있으므로 내부 주소 지정에 10.0.0.0/24 서브넷을 사용합니다.

이 문서에서는 NAT 라우터의 컨피그레이션만 보여줍니다. 그러나 호스트와 관련된 몇 가지 중요한 컨피그레이션 참고 사항은 언급되지 않습니다.

NAT 라우터 컨피그레이션

NAT 라우터 컨피그레이션

```
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#
```

참고: 모든 호스트의 기본 게이트웨이는 NAT 라우터인 10.0.0.2으로 설정됩니다. ISP와 CMTS는 반환 트래픽이 작동하려면 NAT 라우터를 가리키는 192.168.2.0/29로의 경로가 있어야 합니다. 내부 호스트의 트래픽이 이 서브넷에서 도착하는 것으로 표시되기 때문입니다. 이 예에서 CMTS는 192.168.2.0/29에 대한 트래픽을 NAT 라우터에 구성된 보조 IP 주소인 192.168.1.2으로 라우팅합니다.

예 1 show and debug 명령 출력

이 섹션에서는 컨피그레이션이 제대로 작동하는지 확인하는 데 사용할 수 있는 정보를 제공합니다.

위의 컨피그레이션이 작동하는지 확인하기 위해 NAT 라우터의 **디버그** 출력이 모니터링되는 동안 몇 가지 **ping** 테스트를 실행했습니다. **ping** 명령이 성공하고 **debug** 출력이 정확하게 진행 중인 것을 확인할 수 있습니다.

참고: debug 명령을 사용하기 전에 디버그 명령에 대한 중요 정보를 참조하십시오.

테스트 1

첫 번째 테스트에서는 랩 정의 인터넷의 디바이스에서 호스트 2로 **ping**을 수행합니다. 요구 사항 중 하나는 인터넷에 있는 디바이스가 IP 주소가 192.168.2.1인 호스트 2와 통신할 수 있어야 한다는 것입니다. 다음은 NAT 라우터에 표시된 **디버그** 출력입니다. NAT 라우터에서 실행 중이던 **debug** 명령은 정의된 **액세스 목록 177**, **debug ip Nat** 및 **정책 라우팅 패킷을 보여주는 디버그 ip 정책을 사용하는 debug ip packet 177 detail입니다.**

다음은 NAT 라우터에서 실행된 **show ip** NAT **변환** 명령의 출력입니다.

```
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#
```

인터넷의 디바이스에서 라우터인 경우 다음과 같이 성공적인 192.168.2.1 ping을 수행합니다.

```
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#
```

NAT 라우터에서 어떤 일이 발생하는지 확인하려면 다음 **디버그** 출력 및 주석을 참조하십시오.

```
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
```

테스트 2

또 다른 요구 사항은 호스트가 인터넷과 통신할 수 있도록 허용하는 것입니다. 이 테스트에서는 호스트 1에서 인터넷 장치를 ping합니다. 결과 show 및 debug 명령은 아래와 같습니다.

처음에는 NAT 라우터의 NAT 변환 테이블이 다음과 같습니다.

```
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#
```

호스트 1에서 ping을 실행하면 다음과 같은 결과가 표시됩니다.

```
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#
```

위에서 ping이 성공했음을 알 수 있습니다. 이제 NAT 라우터의 NAT 테이블은 다음과 같습니다.

```
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#
```

이제 위의 NAT 변환 테이블에는 동적 NAT 컨피그레이션의 결과인 추가 변환이 표시됩니다(고정 NAT 컨피그레이션과 반대).

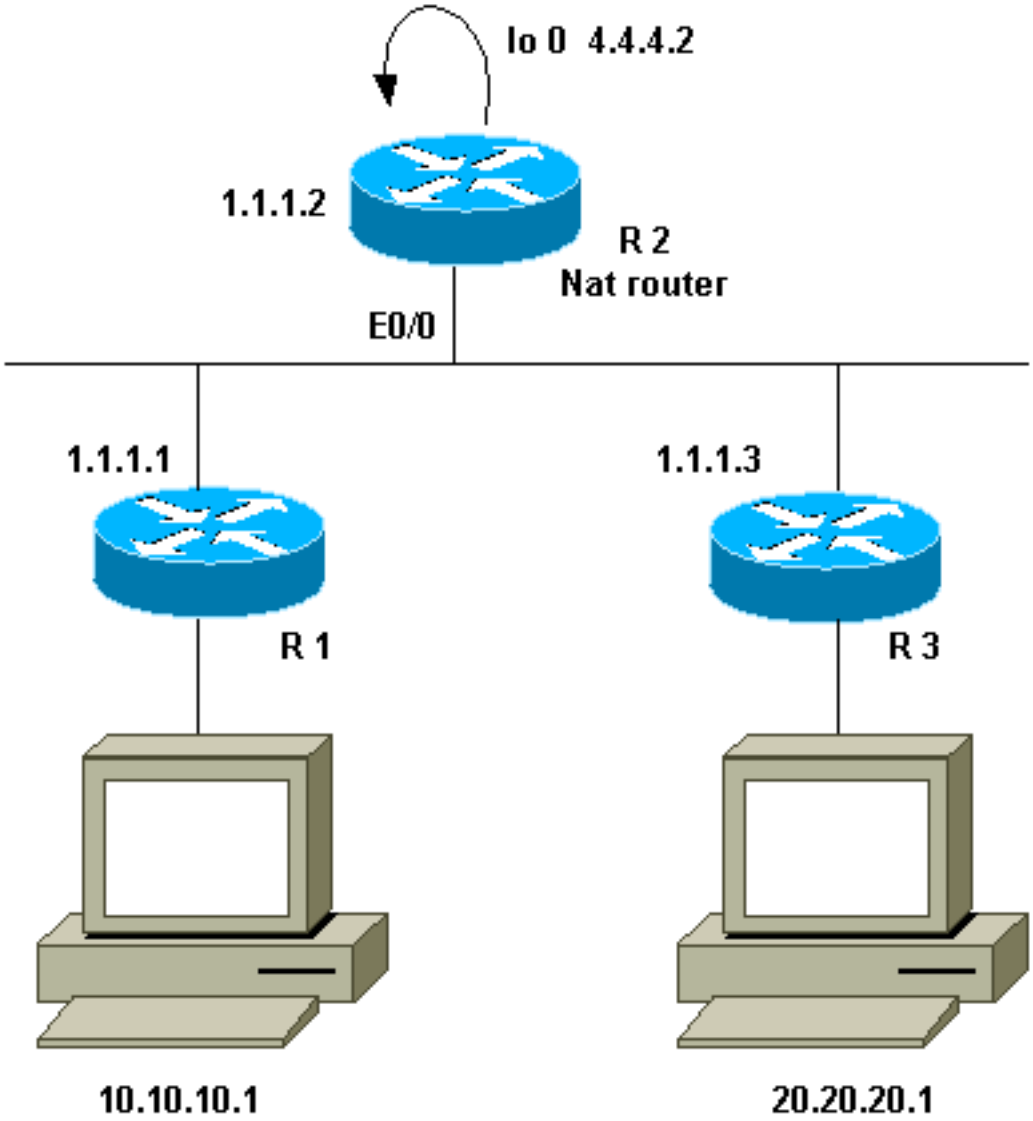
아래의 디버그 출력은 NAT 라우터에서 발생하는 사항을 보여줍니다.

```
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

예 2 네트워크 다이어그램 및 구성

네트워크 다이어그램



요구 사항

두 사이트(R1 및 R3) 뒤에 있는 특정 디바이스가 통신하기를 원합니다. 두 사이트는 등록되지 않은 IP 주소를 사용하므로 서로 통신할 때 주소를 변환해야 합니다. 이 경우 호스트 10.10.10.1은 200.200.200.1으로 변환되고 호스트 20.20.20.1은 100.100.100.1으로 변환됩니다. 따라서 두 방향으로 변환해야 합니다. 이 두 사이트 간의 트래픽은 R2를 통과해야 합니다. 요약하자면, 당사의 요구 사항은 다음과 같습니다.

- R1 뒤에 있는 호스트 10.10.10.1은 전역 주소를 사용하여 R3 뒤에 있는 호스트 20.20.20.1과 통신해야 합니다.
- 이러한 호스트 간 트래픽은 R2를 통해 전송해야 합니다.
- 이 경우 아래 컨피그레이션에 표시된 대로 고정 NAT 변환이 필요합니다.

NAT 라우터 컨피그레이션

NAT 라우터 컨피그레이션

`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
```

예 2 show and debug 명령 출력

참고: 특정 show 명령은 show 명령 출력의 분석을 볼 수 있는 출력 인터프리터 도구에서 지원됩니다. debug 명령을 사용하기 전에 디버그 명령에 대한 중요 정보를 참조하십시오.

테스트 1

위의 컨피그레이션에 표시된 것처럼 R2에서 볼 수 있는 두 개의 고정 NAT 변환(**show ip Nat translation** 명령 포함)이 있습니다.

다음은 NAT 라우터에서 실행된 **show ip** NAT **변환** 명령의 출력입니다.

```
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#
```

이 테스트에서는 R3 뒤에 있는 디바이스의 전역 주소(100.100.100.1)으로 향하는 R1 뒤에 있는 디바이스(10.10.10.1)에서 ping을 소싱했습니다. **debug ip Nat** 및 **debug ip packet**을 R2에서 실행하면 다음 출력이 생성됩니다.

```
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.
```

라우터 3 뒤에 있는 디바이스에서 소싱된 응답 패킷이 라우터 1 뒤에 있는 디바이스로 향하는 결과로 출력됩니다.

```
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.
```

요약

이 문서에서는 NAT 및 정책 기반 라우팅을 사용하여 "NAT on a stick" 시나리오를 생성하는 방법을 시연했습니다. 패킷이 라우터를 통해 프로세스 스위칭될 수 있으므로 이 컨피그레이션은 NAT를 실행하는 라우터의 성능을 줄일 수 있다는 점에 유의해야 합니다.

관련 정보

- NAT 지원 페이지
- Technical Support - Cisco Systems