Introduction to Network Address Translation

Campus Network Design & Operations Workshop

UNIVERSITY OF OREGON

Last updated 9th November 2020

NSRC
Network Startup Resource Center

Network Address Translation

- NAT has become a commonly used technique for prolonging the use of IPv4 on today's Internet
  - Originally designed as a means of allowing isolated networks to connect to Internet without renumbering into public IP address space

- Presentation introduces NAT terminology, the typical use case in a Campus Network, and sample Cisco IOS configuration

Network Address Translation

- NAT is translation of one IP address into another IP address
- NAPT (Network Address & Port Translation) translates multiple IP addresses into one other IP address
  - TCP/UDP port distinguishes different packet flows
- NAT-PT (NAT – Protocol Translation) is a particular technology which does protocol translation (v4 to v6) in addition to address translation
  - NAT-PT is has long been made obsolete by the IETF

Carrier Grade NAT (CGN)

- Service Provider version of subscriber NAT
  - Subscriber NAT can handle only hundreds of translations
  - ISP NAT can handle millions of translations
  - Expensive high performance hardware
- Not limited to just translation within one address family, but does address family translation as well
- Sometimes referred to as Large Scale NAT (LSN)

NAT Use Case

- A campus network does not have sufficient public IPv4 address space to address all the devices on their network
- Their service provider lets them use a small range of addresses – e.g. /28
- The campus might divide the address space into two /29s
  - One /29 for services requiring public IP addresses
  - One /29 for translating internal addresses to public addresses

NAT Use Case

- The /29 for public services:
  - Total of 8 addresses in the subnet
    - 1 address reserved for the gateway router
    - 2 addresses reserved for the subnet
    - 5 addresses available for servers & services
- The /29 for address translation:
  - Campus uses NAPT (network address and port translation) allowing mapping of multiple internal addresses to up to 8 external addresses

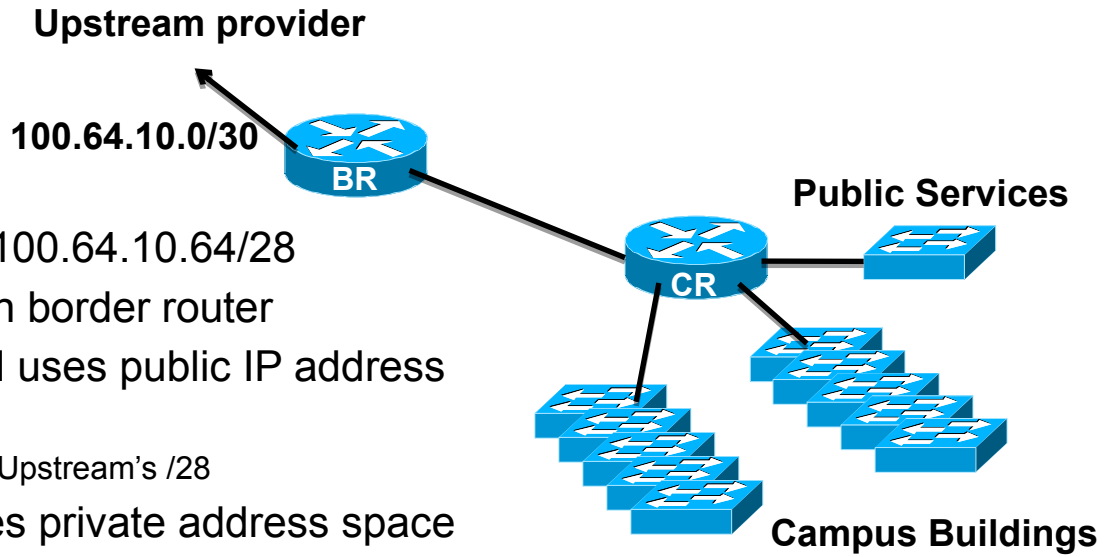
How NAPT works

- NAPT allows mapping of multiple internal addresses to one external address.
  - Each TCP or UDP session is mapped to one TCP or UDP port of an external address
  - There are ~64000 unprivileged TCP and UDP ports
  - Typical end user device consumes ~400 TCP and UDP ports at any one time
  - Which allows around 150 end user devices per public IP address
- One /29 would allow only 1200 simultaneous fully active end user devices

Squeezing more out of NAPT

- Network operators squeeze more internal users through NAPT devices by:
  - Reducing translation session timeouts
    - Cisco default for TCP is 24 hours!!
  - Reducing the number of TCP & UDP sessions any one internal user can have
    - Shows up as broken mapping applications
    - Shows up as “stuck internet”
    - Shows up as “sites unreachable”
  - Deploying IPv6 (!) which reduces the NAPT burden
    - Most large/popular content providers now support IPv6

Campus Use Case: Simple



- Upstream provides 100.64.10.64/28
- NAT implemented on border router
- Public Services LAN uses public IP address block
  - 100.64.10.72/29 from Upstream's /28
- Rest of Campus uses private address space
  - 192.168.0.0/16
  - NAT'ed to 100.64.10.64/29

Typical Cisco configuration (1)

- NAT Configuration set up on Border Router
- Define the address range we want to NAT

```
ip access-list extended NATplus
 deny   ip 100.64.10.0 0.0.0.255 any
 deny   ip 192.168.0.0 0.0.255.255 192.168.0.0 0.0.255.255
 permit ip 192.168.0.0 0.0.255.255 any
 deny   ip any any log
```

  - This says:
    - Don't NAT any of 100.64.10.0/24
    - Don't NAT when source and destination addresses are both internal
    - NAT internal source to any external destination
    - Anything that doesn't match is logged to catch "errors"

Typical Cisco configuration (2)

- Define the external interface we want to NAT to:

```
interface GigabitEthernet 0/1
 description Link to ISP
 ip address 100.64.10.2 255.255.255.252
 ip nat outside
!
```

- Define the internal interface we want to NAT from:

```
interface GigabitEthernet 0/2
 description Link to Campus Core Switch
 ip address 192.168.255.1 255.255.255.252
 ip nat inside
!
```

Typical Cisco configuration (3)

- Modifying the translation timeouts:

```
ip nat translation dns-timeout 60
ip nat translation icmp-timeout 180
ip nat translation udp-timeout 300
ip nat translation finrst-timeout 60
ip nat translation tcp-timeout 3600
```

- This will
  - Set the translation timeouts for DNS to 60 seconds, ICMP to 180 seconds, UDP to be 300 seconds, FIN/RST to be 60 seconds (all Cisco defaults), and TCP to 3600 seconds (from 86400 seconds default)
    - Timeout is when there is no more traffic using that mapping
  - Other translation timeout options are available in Cisco IOS too but the above are the most commonly used

Typical Cisco configuration (4a)

- Activating the NAT on ONE IPv4 address

```
ip nat inside source list NATplus interface Gigabit 0/1 overload
```

- This will
  - match the NATplus list for traffic going from Gigabit 0/2 to Gigabit 0/1
  - Overload means use NAPT (one to many mapping using TCP/UDP ports)
    - NAPT will use the IP address of the Gigabit 0/1 port to map all the internal addresses to
- Campus traffic will appear as though it is all originated from the 100.64.10.2 address

Typical Cisco configuration (4b)

- Activating the NAT on an IPv4 address pool
- First create the public address pool:

```
ip nat pool CAMPUS 100.64.10.64 100.64.10.67 prefix-length 29
```

  - Which defines the pool CAMPUS having 4 IP public IP addresses out of the 100.64.10.64/28 address block given to the campus
- Now enable NAT

```
ip nat inside source list NATplus pool CAMPUS overload
```

  - Which will match all traffic in the NATplus list translating it into the address pool CAMPUS

Diagnosis on a Cisco Router

- To find out what is being translated:

```
Router# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
...
udp 100.64.10.2:20480 192.168.0.65:20480 193.0.0.228:33436 193.0.0.228:33436
udp 100.64.10.2:20482 192.168.0.65:20482 192.5.5.241:33436 192.5.5.241:33436
udp 100.64.10.2:20483 192.168.0.65:20483 192.36.148.17:33436 192.36.148.17:33436
udp 100.64.10.2:20484 192.168.0.65:20484 202.12.27.33:33436 202.12.27.33:33436
udp 100.64.10.2:20485 192.168.0.65:20485 199.7.83.42:33436 199.7.83.42:33436
udp 100.64.10.2:20486 192.168.0.65:20486 198.41.0.4:33436 198.41.0.4:33436
udp 100.64.10.2:20487 192.168.0.65:20487 192.228.79.201:33436 192.228.79.201:33436
...
```

- This shows
  - The local public IP address: UDP port
  - The local internal address and UDP port it maps to
  - And then the global destination addresses & ports

Juniper example with a NAT pool

- First we create a service set definition

```
[edit services]
service-set s1 {
    nat-rules rule-napt-44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
```

- Juniper routers require a multiservices PIC to do NAPT (MS-MIC or MS-MPC)
- As of 2020 the TRIO chipset can only do 1:1 NAT inline

Juniper example with a NAT pool(2)

- Next we create the NAT pool

```
[edit services]
nat {
    pool napt-pool {
        address-range low 100.64.10.64 high 100.64.10.67;
        port {
            automatic auto;
        }
    }
}
```

Juniper example with a NAT pool(3)

- Next we create the NAT rules

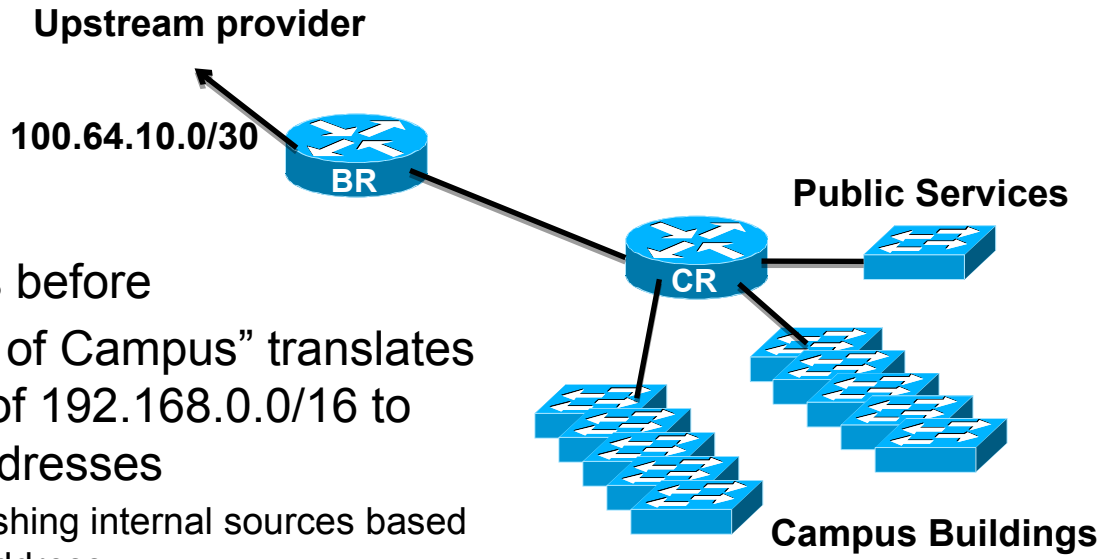
```
[edit services]
nat {
 rule rule-napt-44 {
        match-direction input;
        term t1 {
        from {
         source-address 192.168.0.0/16;
        }
             then {
                 translated {
                     source-pool napt-pool;
                     translation-type {
                         napt-44;
 }}}}}}
```

Juniper example with a NAT pool(4)

- Lastly, some logging

```
[edit services]
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Campus Use Case: Per Subnet NAT



- Same scenario as before
- But NAT for “Rest of Campus” translates different subnets of 192.168.0.0/16 to different public addresses
  - Useful for distinguishing internal sources based on their public IP address

Campus Use Case: Per Subnet NAT

- Campus 192.168.0.0/16 needs to be NAT'ed to different public IP addresses
- Assume that 100.64.10.64 to 100.64.10.71 are used for the NAT pool – 8 addresses out of the 100.64.10.64/28 address block available

Function	Internal Subnet	External Address
Device Management & Internal Services	192.168.0.0/22	100.64.10.64
Science Faculty	192.168.16.0/20	100.64.10.65
Arts Faculty	192.168.32.0/20	100.64.10.66
Engineering Faculty	192.168.48.0/20	100.64.10.67
Library	192.168.64.0/20	100.64.10.68
Administration	192.168.96.0/20	100.64.10.69
Campus Wireless	192.168.128.0/17	100.64.10.70 & 100.64.10.71

Typical Cisco configuration (1)

- Define the address ranges we want to NAT

```
ip access-list extended Services-NAT
 deny   ip 100.64.10.0 0.0.0.255 any
 permit ip 192.168.0.0 0.0.3.255 any
 deny   ip any any
ip access-list extended Science-NAT
 deny   ip 100.64.10.0 0.0.0.255 any
 permit ip 192.168.16.0 0.0.15.255 any
 deny   ip any any
ip access-list extended Arts-NAT
 deny   ip 100.64.10.0 0.0.0.255 any
 permit ip 192.168.32.0 0.0.15.255 any
 deny   ip any any
ip access-list extended Engineering-NAT
 deny   ip 100.64.10.0 0.0.0.255 any
 permit ip 192.168.48.0 0.0.15.255 any
 deny   ip any any
```

Typical Cisco configuration (1)

- Continued:

```
ip access-list extended Library-NAT
 deny   ip 100.64.10.0 0.0.0.255 any
 permit ip 192.168.64.0 0.0.15.255 any
 deny   ip any any
ip access-list extended Admin-NAT
 deny   ip 100.64.10.0 0.0.0.255 any
 permit ip 192.168.96.0 0.0.15.255 any
 deny   ip any any
ip access-list extended Wireless-NAT
 deny   ip 100.64.10.0 0.0.0.255 any
 permit ip 192.168.128.0 0.0.127.255 any
 deny   ip any any
```

  - Define one access-list per internally assigned address block

Typical Cisco configuration (2)

- Internal and External interface NAT definitions are as in the previous example
- NAT translation timeouts also are as in the previous example

Typical Cisco configuration (3)

- Now define the address pools:

```
ip nat pool Services    100.64.10.64 100.64.10.64 prefix-length 29
ip nat pool Science     100.64.10.65 100.64.10.65 prefix-length 29
ip nat pool Arts        100.64.10.66 100.64.10.66 prefix-length 29
ip nat pool Engineering 100.64.10.67 100.64.10.67 prefix-length 29
ip nat pool Library     100.64.10.68 100.64.10.68 prefix-length 29
ip nat pool Admin       100.64.10.69.100.64.10.69 prefix-length 29
ip nat pool Wireless    100.64.10.70 100.64.10.71 prefix-length 29
```

- Note that the public subnet we are NAT'ing into is 100.64.10.64/29
  - We can use all IP addresses in the /29
  - (The University's public servers use the other /29)

Typical Cisco configuration (4)

- Now define the NAT function:

```
ip nat inside source list Services-NAT    pool Services overload
ip nat inside source list Science-NAT     pool Science overload
ip nat inside source list Arts-NAT        pool Arts overload
ip nat inside source list Engineering-NAT pool Engineering overload
ip nat inside source list Library-NAT     pool Library overload
ip nat inside source list Admin-NAT       pool Admin overload
ip nat inside source list Wireless-NAT    pool Wireless overload
```

  - This will match the internal address block with the correct external address
- The example shows how a more sophisticated NAT strategy could be developed for the campus

Summary

- NAPT is useful technique for connecting large numbers of campus network devices to the public IPv4 Internet when the campus has limited or no public IPv4 address space
  - Private address space used for campus networks:
    - 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
- Border router is the most common location of the NAT device
  - Be aware of CPU loading though
- Be aware of NAT scaling limitations

Questions?

Aside: NAT Issues (1)

- How to scale NAT performance for large networks?
  - Limiting tcp/udp ports per user harms user experience
  - Redesigning network
- Breaks the end-to-end model of IP
- Breaks end-to-end network security
- Breaks non-NAT friendly applications
  - Or NAT has to be upgraded (if possible)
- Content cannot be hosted behind a NAT

Aside: NAT Issues (2)

- Makes fast rerouting and multihoming more difficult
  - Moving IPv4 address pools between CGNs for external traffic engineering
- Address sharing has reputation, reliability and security issues for end-users
- NAT device keeps the state of the connections
- Makes the NAT device a target for miscreants due to possible impact on large numbers of users

Aside: NAT Issues (3)

- Consumer NAT device:
  - 5000 sessions means only 12 connected devices!
  - “NAT table FULL” error messages
  - “Broken Googlemaps”
  - “Stuck Internet”
- Carrier Grade NAT device:
  - 20 million sessions (Cisco ASR9001 ISM)
  - Which realistically is 50k users (400 sessions per user)
  - RIR 2x final IPv4 /22s only allows 640k users