NAT

Network Address Translation

Agenda

- **NAT Basics**
- **NAPT**
- **Complex NAT**
- **DNS Aspects**
- **Load Balancing**
- **RFCs**

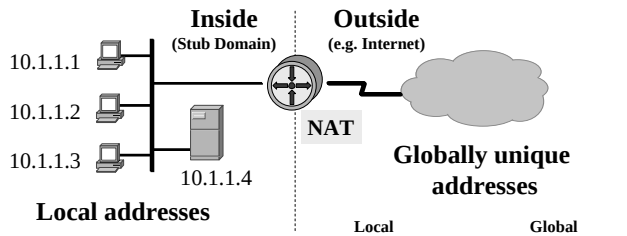
Network Address Translation (NAT)

- **NAT**
  - was originally developed as an interim solution to combat IPv4 address depletion by allowing IP addresses to be reused by several hosts
  - first explained in RFC 1631
    - the address reuse solution is to place Network Address Translators (NAT) at the borders of stub domains
    - each NAT box has a table consisting of pairs of local IP addresses and globally unique addresses performing address translation when passing IP Datagram's between a stub domain and the Internet and vice versa
    - the IP addresses inside the stub domain are not globally unique, they are reused in other domains, thus solving the address depletion problem
    - in most cases private addresses (RFC 1918) are used inside the stub domain (10.0.0.0/8, 172.16.0.0/16, 192.168.0.0/16)

Reasons for NAT

- **Mitigate Internet address depletion**
- **Save global addresses (and money)**
  - if not all inside hosts need to go outside
  - if all inside hosts can be mapped to one unique global address using NAPT (Network Address Port Translation)
- **Conserve internal address plan**
- **Hide internal topology**
  - Security aspect
- **TCP load sharing**
  - Several physical servers are hided behind one IP address and traffic to them is balanced

Terms

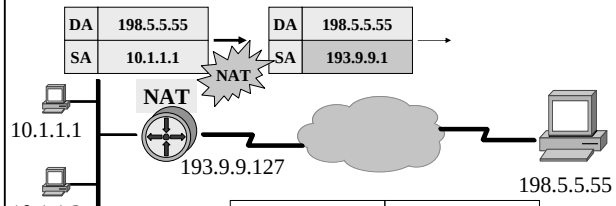


Binding is maintained by static NAT–Table

Local IP address		Global IP address
10.1.1.1	⟷	193.99.99.1
10.1.1.2	⟷	193.99.99.2
10.1.1.3	⟷	193.99.99.3
10.1.1.4	⟷	193.99.99.4

 NAT, v4.3 5

Basic Principle (1a)

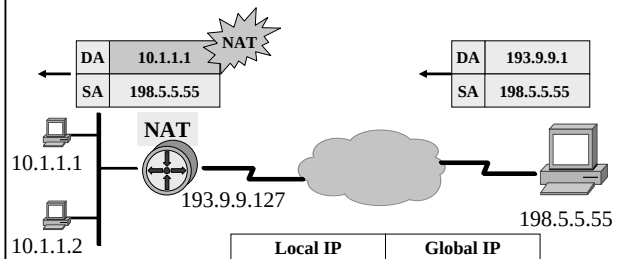


Simple NAT Table

Local IP	Global IP
10.1.1.1	193.9.9.1
10.1.1.2	193.9.9.2
....	....

 NAT, v4.3 6

Basic Principle (1b)



Simple NAT Table

Local IP	Global IP
10.1.1.1	193.9.9.1
10.1.1.2	193.9.9.2
....	....

 NAT, v4.3 7

NAT Tasks and Behaviour

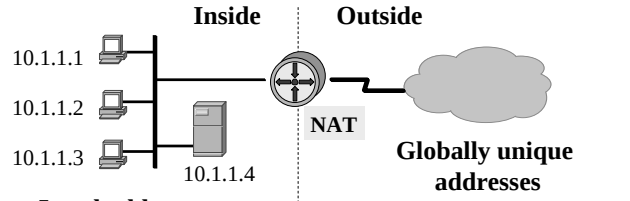
- modify IP addresses according to NAT table
- but also must modify the IP checksum and the TCP checksum
  - note: TCP's checksum also covers a pseudo IP header which contains the source and destination address.
- must also look out for ICMP and modify the places where the IP address appears
- there may be other places, where modifications must be done (FTP, NetBIOS over TCP/IP, SNMP, DNS, Kerberos, X-Windows, SIP, H.323, IPsec, IKE…)
- the packet sender and receiver (should) remain unaware that NAT is taking place
- NAT devices were intended to be unmanaged devices that are transparent to end-to-end protocol interaction
- hence no specific interaction is required between the end systems and the NAT device

 NAT, v4.3 8

NAT Binding Possibilities

- **Static ("Fixed Binding")**
  - in case of one-to-one mapping of local to global addresses
- **Dynamic ("Binding on the fly")**
  - in case of sharing a pool of global addresses
  - connections initiated by private hosts are assigned a global address from the pool
  - as long as the private host has an outgoing connection, it can be reached by incoming packets sent to this global address
  - after the connection is terminated (or a timeout is reached), the binding expires, and the address is returned to the pool for reuse
  - is more complex because state must be maintained, and connections must be rejected when the pool is exhausted
  - unlike static binding, dynamic binding enables address reuse, reducing the demand for globally unique addresses.

Scenario Dynamic Binding



Local IP address		Global IP address
10.1.1.1	⟷	193.99.99.1
10.1.1.2	⟷	193.99.99.2
10.1.1.3	Currently not possible	
10.1.1.4	Currently not possible	

Binding is maintained by dynamic NAT–Table

Note: a connection state or timer must be maintained per mapping

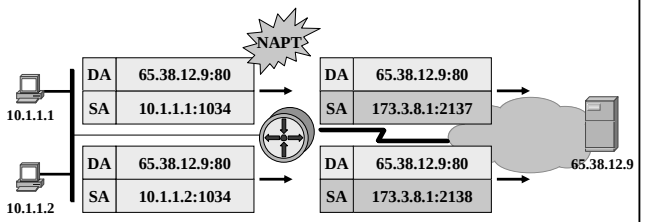
Agenda

- **NAT Basics**
- **NAPT**
- **Complex NAT**
- **DNS Aspects**
- **Load Balancing**
- **RFCs**

Overloading (NAPT)

- Common problem:
  - **Many hosts inside initiating connections to the outside world**
  - **But only one or a few inside-global addresses available**
- Solution:
  - **Many-to-one Translation with NAPT (Network Address Port Translation)**
  - **Usable in context of TCP and UDP sessions**
  - **Aka *"Overloading Global Addresses*"**
  - **Aka "*PAT„ (Port Address Translation)***

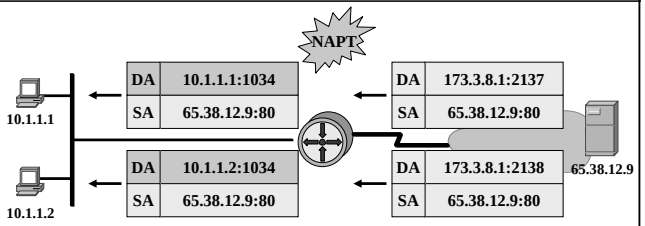
NAPT Example (1)



Prot.	Local	Global
TCP	10.1.1.1:1034	173.3.8.1:2137
TCP	10.1.1.2:1034	173.3.8.1:2138

Extended Translation Table

NAPT Example (2)



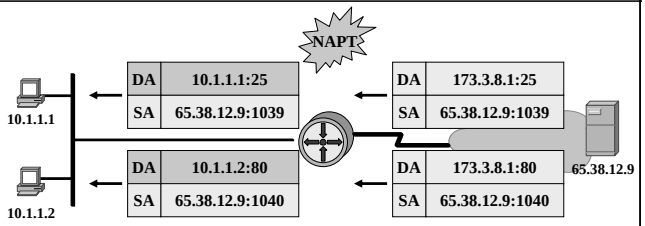
Prot.	Local	Global
TCP	10.1.1.1:1034	173.3.8.1:2137
TCP	10.1.1.2:1034	173.3.8.1:2138

Extended Translation Table

Virtual Server Table

- Problem:
  - **How to reach an inside server from the outside**
  - **NAPT/NAT let IP datagram's (with UDP or TCP segments as payload) from to outside only in if a binding is found**
  - **But server waits for connections from the outside hence cannot install binding in the NAPT/NAT device**
- Solution:
  - **Virtual Server Table**
  - **Creating manually a static binding in the NAPT/NAT device to forward IP datagram's to the real inside server**

Virtual Server Table Example



Prot.	Local	Global
TCP	10.1.1.1:25	173.3.8.1:25
TCP	10.1.1.2:80	173.3.8.1:80

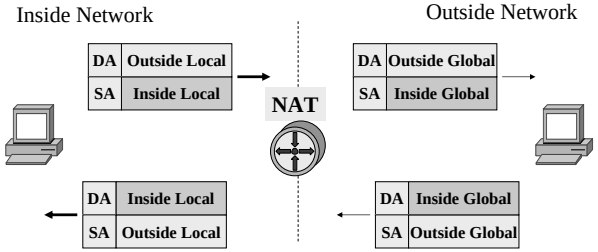
Extended Translation Table

Agenda

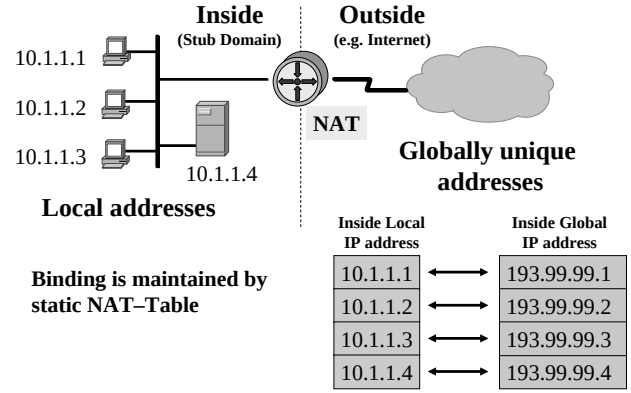
- NAT Basics
- NAPT
- Complex NAT
- DNS Aspects
- Load Balancing
- RFCs

Terms Used in complex NAT Devices

- *Local* versus *global* address
  - Reflects area of usage (inside or outside)
- *Inside* versus *outside* world
  - Reflects the origin

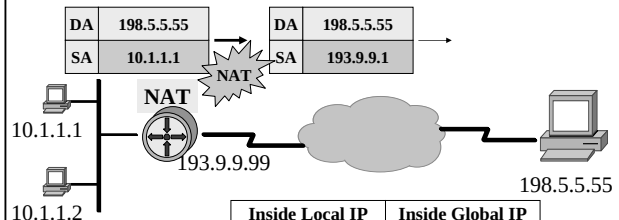


Static NAT Example with New Terms



Binding is maintained by static NAT–Table

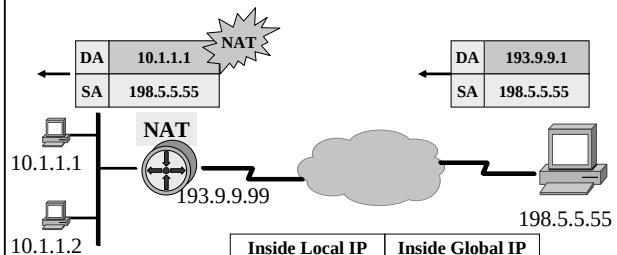
Basic Principle (1a) with New Terms Inside Address Translation



Simple NAT Table

Inside Local IP	Inside Global IP
10.1.1.1	193.9.9.1
10.1.1.2	193.9.9.2
....	....

Basic Principle (1b) with New Terms Inside Address Translation



Simple NAT Table

Inside Local IP	Inside Global IP
10.1.1.1	193.9.9.1
10.1.1.2	193.9.9.2
....	....

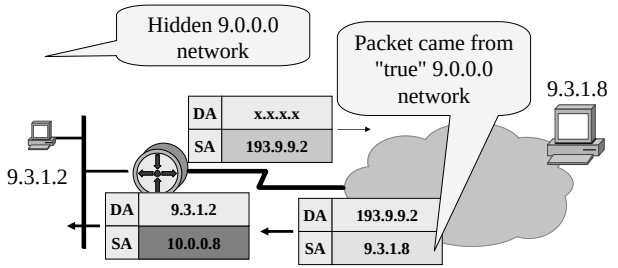
Overlapping Networks

= Same addresses **are used**

locally and ***globally***

What can happen?

Outside Address Translation

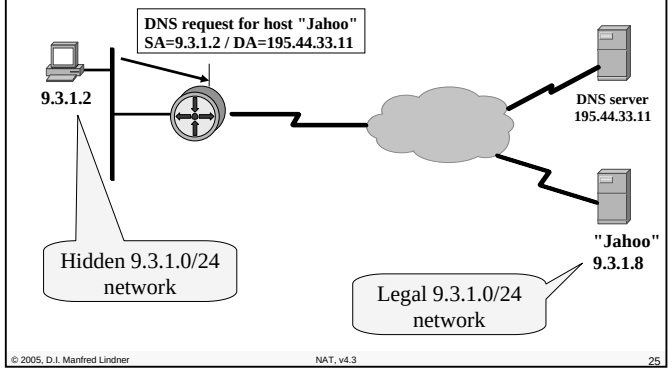


Inside Local	Inside Global	Outside Local	Outside Global
9.3.1.2	193.9.9.2	10.0.0.8	9.3.1.8

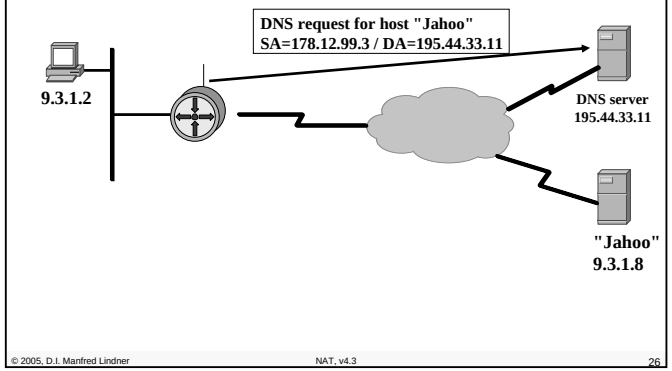
Agenda

- **NAT Basics**
- **NAPT**
- **Complex NAT**
- **DNS Aspects**
- **Load Balancing**
- **RFCs**

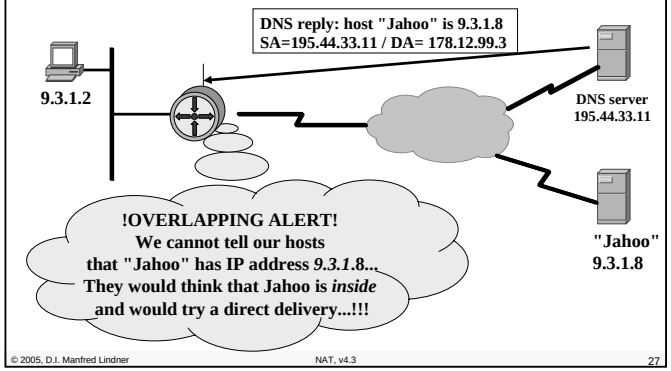
DNS Problem (1)



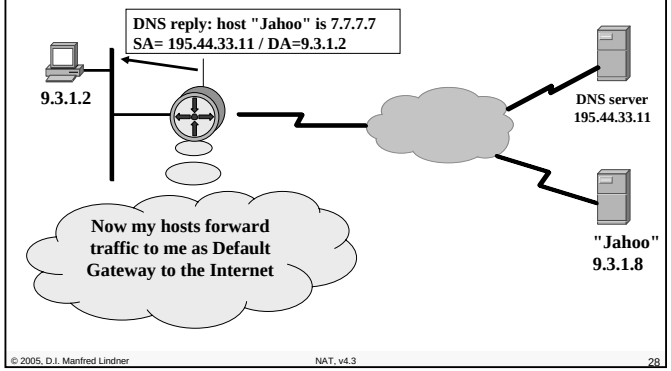
DNS Problem (2)



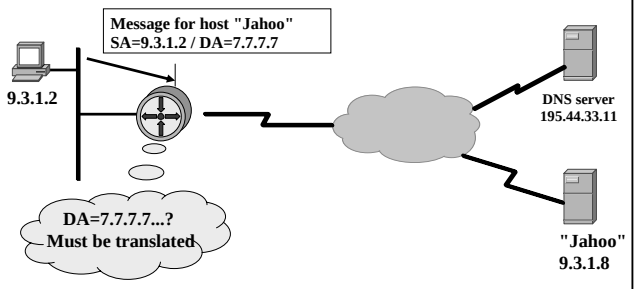
DNS Problem (3)



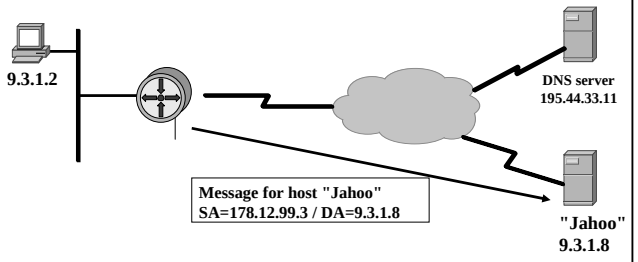
DNS Problem (4)



DNS Problem (5)



DNS Problem (6)



NAT Table

Inside Local	Inside Global	Outside Global	Outside Local
9.3.1.2	178.12.99.3	**9.3.1.8**	**7.7.7.7**

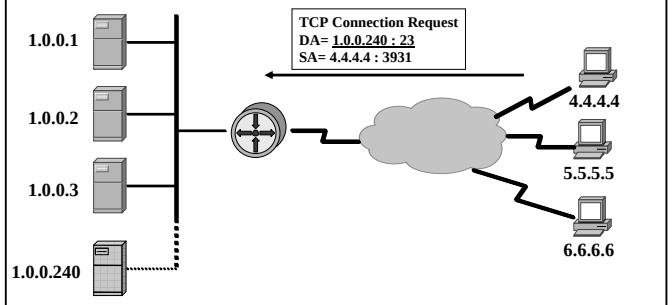
Agenda

- **NAT Basics**
- **NAPT**
- **Complex NAT**
- **DNS Aspects**
- **Load Balancing**
- **RFCs**

TCP Load Sharing (1)

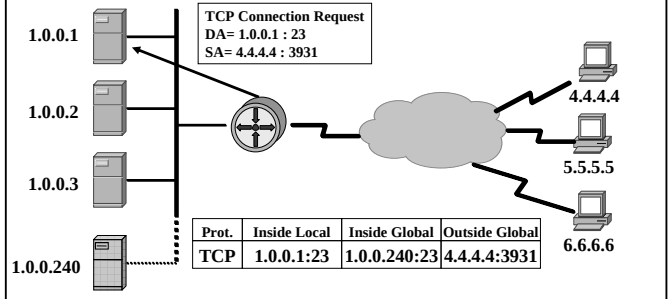
- **Multiple servers represented by a single inside-global IP address**
  - *Virtual host* address
- **New TCP session requests to the Virtual Host are forwarded to one of a group of real hosts**
  - *Rotary group*

TCP Load Sharing (2)



 NAT, v4.3 33

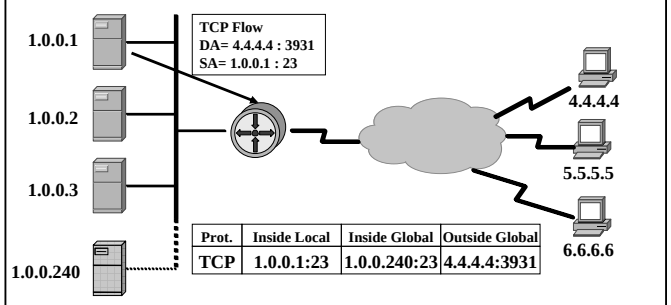
TCP Load Sharing (3)



Prot.	Inside Local	Inside Global	Outside Global
TCP	1.0.0.1:23	1.0.0.240:23	4.4.4.4:3931

 NAT, v4.3 34

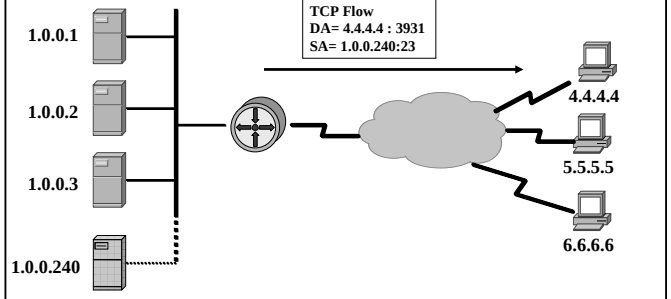
TCP Load Sharing (4)



Prot.	Inside Local	Inside Global	Outside Global
TCP	1.0.0.1:23	1.0.0.240:23	4.4.4.4:3931

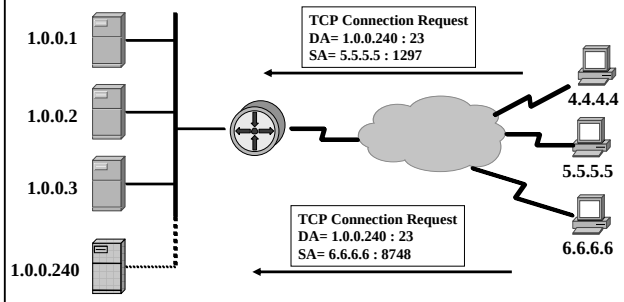
 NAT, v4.3 35

TCP Load Sharing (5)

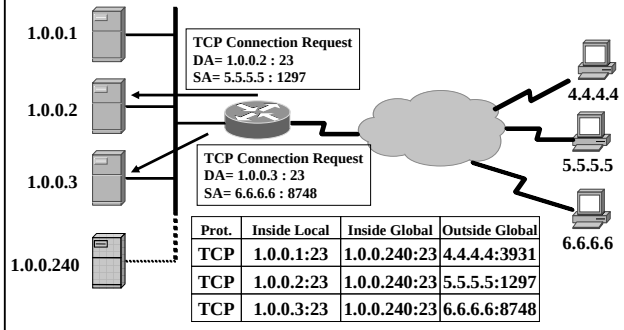


 NAT, v4.3 36

TCP Load Sharing (6)



TCP Load Sharing (7)



Prot.	Inside Local	Inside Global	Outside Global
TCP	1.0.0.1:23	1.0.0.240:23	4.4.4.4:3931
TCP	1.0.0.2:23	1.0.0.240:23	5.5.5.5:1297
TCP	1.0.0.3:23	1.0.0.240:23	6.6.6.6:8748

Agenda

- NAT Basics
- NAPT
- Complex NAT
- DNS Aspects
- Load Balancing
- RFCs

Further Information

- RFC 1631 - NAT
- RFC 2391 - Load Sharing Using IP Network Address Translation (LSNAT)
- RFC 2666 - IP Network Address Translator (NAT) Terminology and Considerations
- RFC 2694 - DNS ALG
- RFC 2776 - Network Address Translation Protocol Translation (NAT-PT)
- RFC 2993 - Architectural Implications of NAT
- RFC 3022 - Traditional IP Network Address Translator (Traditional NAT)

Further Information

- RFC 3027 - Protocol Complications with the IP Network Address Translator,
- RFC 3235 - Network Address Translator (NAT)-Friendly Application Design Guidelines
- RFC3303 - Middlebox Communication Architecture and Framework
- RFC 3424 - IAB Considerations for Unilateral Self Address Fixing (UNSAF) Across Network Address Translation

Further Information

- RFC 3489 - STUN—Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)
- RFC 3715 - IPsec—Network Address Translation (NAT) Compatibility Requirements
- Internet Protocol Journal
  - www.cisco.com/ipj
    - Issue Volume 3, Number 4 (December 2000)
    - „The Trouble with NAT“
    - Issue Volume 7, Number 3 (September 2004)
    - „Anatomy (of NAT)“