Impact of Network Address Translation on Router Performance

Sarabjeet Singh Chugh

Thesis submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Luiz A. Dasilva, Chair
Scott F. Midkiff
Annamalai Annamalai, Jr.

September 18, 2003

Falls Church, Virginia

Keywords: Throughput, Delay, Utilization, CPU, Network.

Impact of Network Address Translation on Router Performance

Sarabjeet Singh Chugh

Abstract

"When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind: it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science."

-- William Thomson, Lord Kelvin

Network Address Translation (NAT) is a method by which Internet Protocol (IP) addresses are translated from one group to another, in a manner transparent to the end users. It translates the source and destination addresses and ports in the Internet Protocol datagram. There are several benefits for using NAT. NAT can be installed without changes to hosts or routers, it allows reuse of globally routable addresses, it facilitates easy migration or addition of new networks and it provides a method to keep private network addresses hidden from the outside world.

NAT, however, is a processor- and memory-intensive activity for any device that implements it. This is because NAT involves reading from and writing to the header and payload information of every IP packet to do the address translation, a performance-intensive activity. It causes an increase in Central Processing Unit (CPU) and memory utilization and may impair throughput and increase the latency experienced by a packet. Thus, understanding the performance impact of NAT on a network device (in particular, a router) becomes an important factor when implementing NAT in any live network.

This thesis aims to *understand* and *quantify* the impact of Network Address Translation on a network router by doing a series of performance tests after specifying the performance parameters to measure and, then, clearly defining the performance testing methodology that is used to study each of the performance parameters. After a discussion of previous research, the measurement system and subsequent measurement results are described.

Table Of Contents

LIST OF FIGURES ..... **IV**

LIST OF TABLES ..... **IV**

CHAPTER 1. INTRODUCTION **1**

1.1 BACKGROUND 1
1.2 PROBLEM STATEMENT 2
1.3 RESEARCH GOALS 3
1.4 DOCUMENT OVERVIEW 4

CHAPTER 2. NETWORK ADDRESS TRANSLATION **5**

2.1 HOW NAT WORKS 5
2.2 MERITS AND DEMERITS OF NAT 11
2.3 CURRENT RESEARCH 13
2.4 SUMMARY 14

CHAPTER 3. METHODOLOGY **15**

3.1 NAT PERFORMANCE CHARACTERIZATION 15
3.2 TEST NETWORK DESIGN METHODOLOGY 16
3.3 NAT CONFIGURATION INFORMATION 18
3.4 NETWORK COMPONENTS 19
3.5 TESTS 22
3.6 SUMMARY 24

CHAPTER 4. OBSERVATIONS AND RESULTS **25**

4.1 TEST 1: DELAY MEASUREMENT 25
4.2 TEST 2: THROUGHPUT MEASUREMENT 26
4.3 TEST 3: MEMORY UTILIZATION 27
4.4 TEST 4: CPU UTILIZATION 28
4.5 SUMMARY 29

CHAPTER 5. ANALYSIS AND CONCLUSIONS **30**

5.1 TEST 1: DELAY MEASUREMENT 30
5.2 TEST 2: THROUGHPUT MEASUREMENT 31
5.3 TEST 3: MEMORY UTILIZATION 32
5.4 TEST 4: CPU UTILIZATION 33
5.5 CONCLUSIONS 34
5.6 SUGGESTED PERFORMANCE ENHANCEMENTS 35
5.7 CONTRIBUTIONS 38
5.8 FURTHER WORK 38

REFERENCES **41**

VITA **45**

List of Figures

Figure 2.1. NAT inside and outside interfaces. 6
Figure 2.2. Inside local and inside global address translation. 8
Figure 2.3. Outside local and outside global address translation. 9
Figure 3.1. NAT performance test topology for delay. 16
Figure 3.2. NAT performance test topology for delay, throughput, CPU, and memory... 17
Figure 5.1. Delay versus number of NAT entries. 30
Figure 5.2. Throughput versus packet size. 31
Figure 5.3. Memory utilization versus number of NAT entries. 32
Figure 5.4. CPU utilization versus packet size. 33
Figure 5.5. CPU utilization versus packet size. 34
Figure 5.6. Hash-based lookup. 36
Figure 5.7. Binary radix-tree based lookup. 37
Figure 5.8. NAT-PT translating between IPv6 and IPv4 clouds. 39
Figure 5.9. NAT-PT stateful NAT failover. 40

List of Tables

Table 3.1. IMIX Traffic Profile Details 21
Table 4.1. Delay Measurement Observations 25
Table 4.2. Throughput Measurement Observations 26
Table 4.3. Memory Measurement Observations 28
Table 4.4. CPU Utilization Measurement Observations 29

Chapter 1. Introduction

1.1 Background

Network Address Translation (NAT) is a method by which IP addresses are translated from one group to another, transparent to the end users, in effect expanding the limited number of globally unique addresses [1]. It was first introduced in RFC 1631 [6]. It replaces the source address with a globally routable address and enables privately addressed hosts to access registered networks, such as the Internet. Network Address Translation involves changing the source and/or destination IP addresses in the IP header and the IP addresses in application data streams.

The advantage of this approach is that it can be adopted without changes to hosts or routers. Globally routable addresses can be reused, migration or addition of new networks becomes easier and private network addresses can remain hidden from the outside world. RFC 2993 discusses the architectural implications of Network Address Translation in further detail [2]. The use of NAT creates a single point where addressing decisions are made, in the device maintaining connection state and dynamic mapping information [2]. One of the key disadvantages of Network Address Translation is that it impacts maintenance and serviceability of networks by creating a single point of failure at the device performing address translations. The failures are usually due to the performance limitations of the edge router doing address translations. Thus, the performance impact of NAT translations on the routers doing NAT becomes an important factor while implementing NAT in a production network.

To ensure that multiple applications can use the same IP address from the limited IP pool, the Port Address Translation (PAT) feature is also available. Once this feature is enabled, NAT requires only unique ports and not unique IP addresses to establish a connection between two hosts. PAT allows many-to-one address mapping, since many inside IP addresses can be mapped to one outside IP address. Many applications like H.323 and FTP have problems negotiating a path across an application proxy such as NAT/PAT. H.323 causes a problem with NAT because it uses two Transmission Control Protocol (TCP) connections and several User Datagram Protocol (UDP) sessions for a single call. The response IP addressing information needed for the H.323 session is placed within these data packet's' payload also causes problems with proper NAT H.323 support mechanisms. H.323, further, uses an encoding in the packet's payload, called

Abstract Syntax Notation (ASN), which is too complex for a standard NAT process to decode. H.323 uses ephemeral ports (i.e. port numbers greater than 1024) in its connection call setup process, which PAT has difficulty supporting. Outside session setup must also be allowed for external network call inquires. This will require static address relation tables or firewall conduits to be constructed, which increases the management overhead of supporting H.323 applications through a NAT system.

The final outcome of these H.323 structural issues is that addresses in the payloads are not decoded correctly by standard NAT, which leads to connection failures or poor performance on H.323 sessions. The only solution is to deploy a special NAT system that is customized to support the intricacies of H.323.

The FTP layer-3 connection addressing is also embedded (hidden) within the payload of an outgoing data packet. NAT process does not decode this FTP packet properly, which leads to a failure in the FTP host connection [26].

1.2 Problem Statement

Network Address Translation translates the IP packet source and destination addresses and ports. It also translates the embedded IP addresses and ports for the protocols of which it is aware. That means before making a switching decision, the router's NAT process needs to be aware of the offset to the IP address and port number that are embedded within the payload of all incoming application packets and manipulate them.

NAT is performance intensive because of the above mentioned reasons. As a result, it does not scale well when there are large number of translations or when the traffic volume is high. The performance of the router in terms of throughput degrades significantly, while memory and CPU utilization is high. The purpose of this thesis is to identify key areas to improve the performance of a router running Network Address Translation.

1.3 Research Goals

The intent of this research is to understand and quantify the impact of Network Address Translation on a network router by conducting performance tests. The thesis defines the performance parameters for NAT features and defines a performance testing methodology to obtain values for these parameters.

There are a number of factors introduced by NAT that impact a router's performance. First, the IP address translation and NAT table lookup utilize extra CPU processing power. Second, the NAT translation table consumes extra system memory. Third, due to the address lookups and translations, the packet stream experiences a certain amount of delay. This can be a cause for concern, especially for applications like voice or video, where end-to-end latency is a crucial issue.

The performance of NAT-enabled routers is a function of traffic load, traffic type (different applications), number of NAT entries and specific NAT configuration. It can be measured in terms of:

1. traffic delay, the extra packet delay introduced by NAT;
2. throughput, the traffic throughput with NAT;
3. memory utilization, the memory consumption by different NAT entries; and
4. CPU utilization, the extra CPU utilization introduced by NAT.

Tests were run to find out the impact of NAT in terms of the parameters listed above. This was followed by an analysis of the data. Finally, some specific inferences were drawn from the test results about the impact of NAT on network routers. These results provide performance and scalability information to evaluate and optimize the NAT feature design and implementation of NAT features.

1.4 Document Overview

This document is organized as follows. Chapter 2 gives a general overview of how NAT works and the factors introduced by NAT that impact the performance of a router. Chapter 3 presents the general methodology of performance testing including the testbed setup, test execution procedure and the method of collecting raw data from the router under test. Different assumptions and limitations are also discussed. Chapter 4 contains the experimental results and discussion of the results. Chapter 5 draws conclusions based on the results of this research and offers recommendations for further work.

Chapter 2. Network Address Translation

2.1 How NAT Works

NAT is a process that runs on a network device connecting two networks. One of these networks, designated as “inside", is addressed with either private or obsolete addresses that are to be converted into "legal", or globally routable, IP addresses before packets are forwarded onto the other network (designated as "outside"). The translation operates in conjunction with routing, so that the feature can simply be enabled on the enterprise side of an Internet access router. In that case, the NAT router will advertise reachability to the outside local addresses into the enterprise and also will not advertise any of the inside local addresses to the Internet Service Provider (ISP) network. It modifies destination addresses en-route to maintain state so that datagrams pertaining to a session are routed to the correct destination node in either realm. In enabling NAT, the aim is that everything works as if the private network had globally unique addresses and as if the NAT device were not present.

Before describing the operation, we first review the terminology that is used to describe various elements of a Network Address Translation scheme.

- Inside network – The set of network nodes whose IP address is subject to translation. These IP addresses are usually not globally unique.

- Outside network – All the nodes on the other side of the inside network, separated by a NAT-enabled device.

- Inside local IP address – Address that was assigned to a host on the inside network. The address was either globally unique but obsolete, allocated from an address space defined by RFC 1918 [25] or just picked at random. The address may or may not be globally routable.

- Inside global IP address – The IP address of an inside host as it appears to the outside world. The address was allocated from globally unique address space, typically provided by the Internet service provider.

- Outside global IP address – The IP address that was assigned to a host on the outside network.

- Outside local IP address – The IP address of an outside host as it appears to the inside network.

- Simple translation entry – A translation entry that maps one IP address to another.

- Extended translation entry – A translation entry that maps one IP address and port pair to another.

An interface on the router can be defined as "inside" or "outside," as illustrated in Figure 2.1. Translation occurs only from inside to outside interfaces or vice versa, but never between the same type of interfaces.

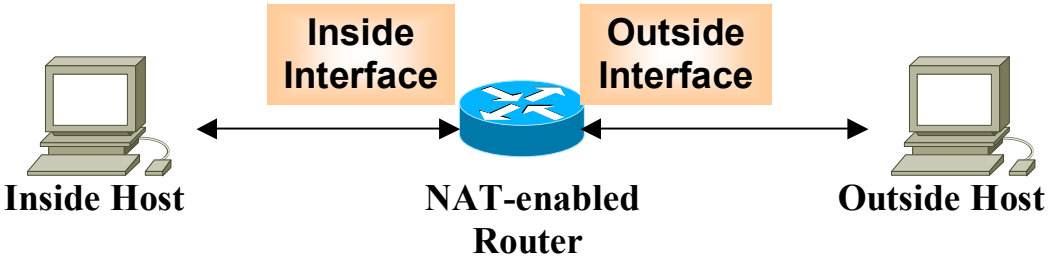

Figure 2.1. NAT inside and outside interfaces.

There are two types of NAT, static and dynamic. Static translations are fixed address translations from the inside network to the outside network. For every IP address in the local network, there exists a fixed unique IP address in the global network. Static NAT need not just be a one to one mapping of IP addresses. Port Address Translation may be used along with Static NAT to enable

one-to-many or many-to-one mapping when there are insufficient number of IP addresses available to translate all of the inside addresses to unique addresses.

Port Address Translation (PAT) extends NAT from "one-to-one" to "many-to-one" by associating a unique source port with each session. In this case, the unique source port is used in addition to the source IP address to identify a session. Static translations are used when one needs to be able to initiate a connection from both the inside and outside interfaces e.g. for Simple Mail Transfer Protocol (SMTP)/ Hypertext Transfer Protocol (HTTP) servers or when one wants a specific host to be associated with a specific IP address.

Dynamic translations translate from a pool of local IP addresses to a pool of global IP addresses. Defining some rules in the router configuration can control the assignment of an IP address from the pool. These rules, called access lists or filters, are used extensively in most network devices like firewalls and network-edge routers to implement security policies of an organization. Such translations are generally used when one wants to initiate a connection only from the inside or only from the outside.

When a packet from an inside network is outbound, the source IP address in the packet has to be translated into a globally routable IP address, that is, a translation of an inside local IP address to an inside global IP address takes place. The inside global IP address may be allocated dynamically from the NAT address pool. Corresponding to each translation that is done, an entry is logged in the address translation table in the router's non-volatile random access memory (NVRAM). An address translation entry in the translation table is used to translate inside global IP addresses to inside local IP addresses in the return packets. This is illustrated in Figure 2.2.

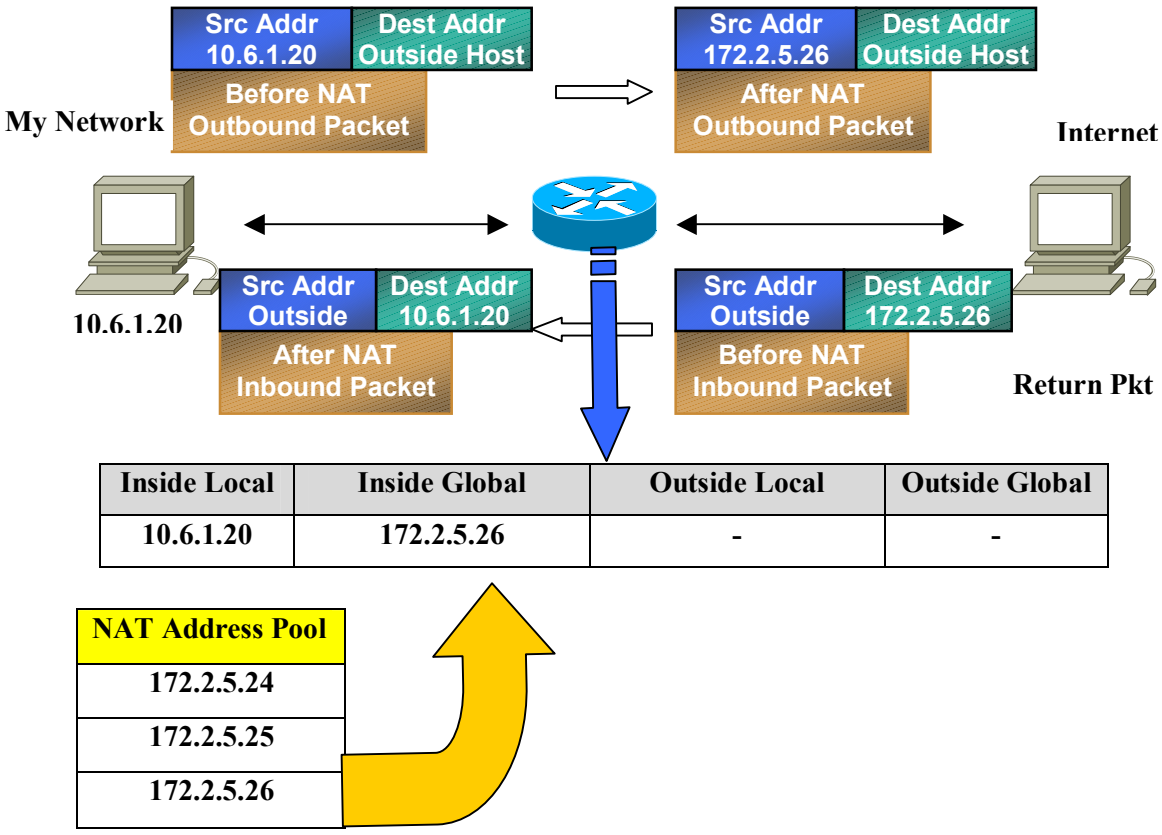

Figure 2.2. Inside local and inside global address translation.

Similarly, when a packet from an outside network is inbound, the source IP address in the packet has to be translated into a locally significant IP address for the private network. That is, a translation of an "outside global" IP address to an "outside local" IP address must take place. The outside local IP address may be allocated dynamically from the NAT address pool. This is needed in cases of overlapping networks.

Overlapping networks result when an IP address is assigned to a device on a network that is already legally owned and assigned to a different device on the Internet or outside network. Overlapping networks also result when two companies, both of whom use RFC 1918 [25] IP addresses in their networks, merge. These two networks need to communicate, preferably without having to readdress all their devices [27]. If source address translation is not done, the NAT router will not be able to route any inbound packets originating from an outside network that have

same set of IP addresses assigned to its hosts because the source address of certain packets would be the same as the destination address in the internal addressing scheme of the inside network.

Corresponding to each translation that is done, an entry is logged in the address translation table in the router's NVRAM. An address translation entry in the translation table is used to translate outside local IP addresses to outside global IP addresses in the return packets. This is illustrated in Figure 2.3.

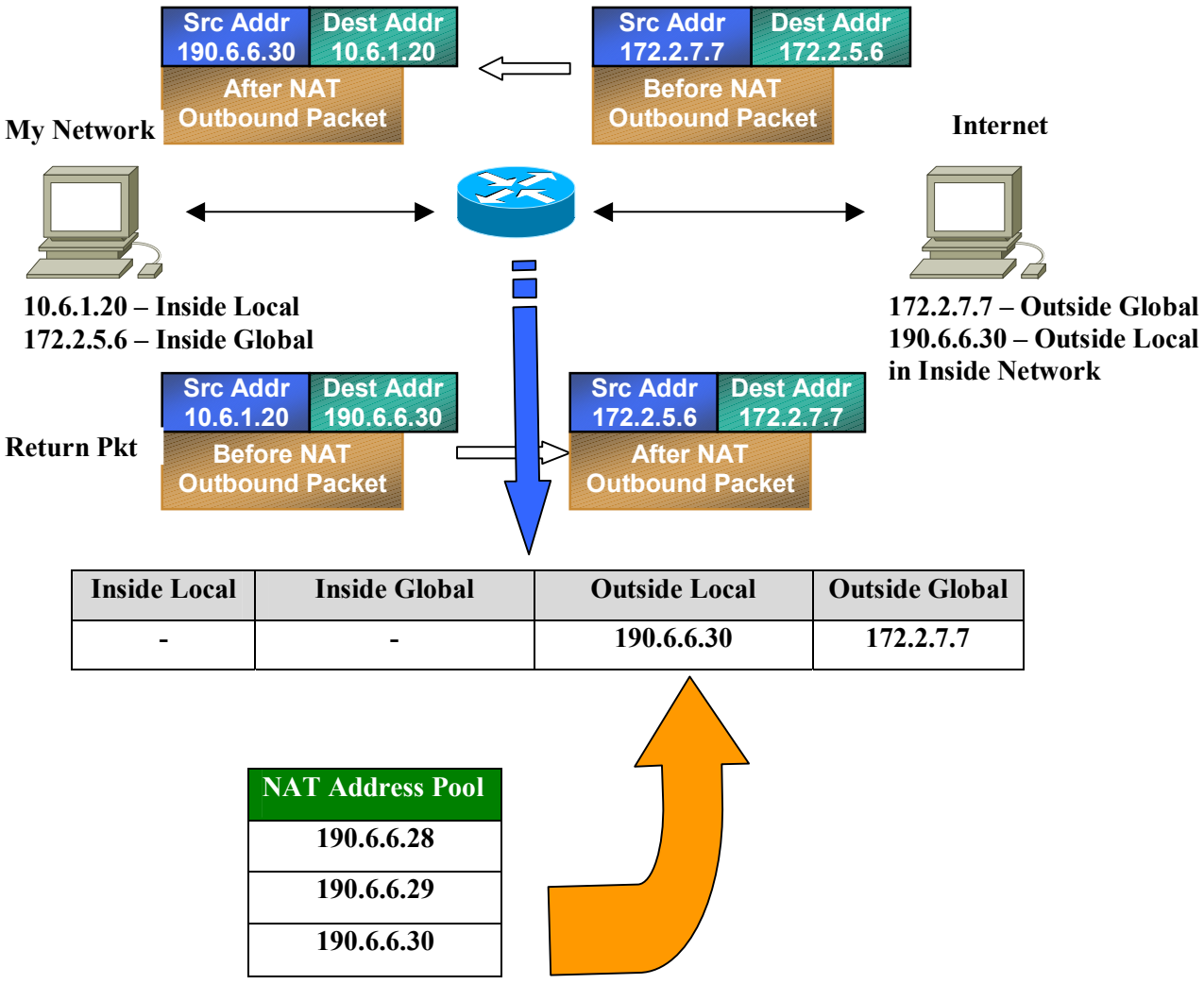

Figure 2.3. Outside local and outside global address translation.

The order of operation of different processes when performing NAT is based on whether the packet is going from the inside network to the outside network or from the outside network to the inside network [3]. The router will always try to find an existing entry from the translation table before it attempts to assign IP address from a pool based on some access-lists or filters. The full order of NAT operation is listed in [3].

Now let us look at the overall operation of a NAT-enabled device. When a packet is received from an internal host destined for the outside network, the NAT process looks for the matching source address-source port combination in the port-mapping table. A port-mapping table contains a list of inside local port numbers that get translated to particular mapped ports. If the entry is not found for the received source port, a new port translation entry is created and the new port is allocated for the received source port. The source IP address is replaced with a translated global address from the translation table or NAT address pool and the source port is replaced with port from the port-mapping table. Finally, the packet is sent out with the translated source IP address and source port number.

Packets that are received from the outside network and destined for the inside network undergo a similar but reverse translation. The destination port number is looked up in a port mapping table. If an entry is found, the destination port number gets replaced with a port number from the mapping table, and the destination IP address is replaced with an IP address from the translation table. If an entry is not found for the destination IP and destination port, the router concludes that the packet is not for its internal network and rejects it.

Each entry in the translation table has a timeout value associated with it. This timeout value is reset whenever new traffic is received for that entry. When the timeout expires, the entry is removed from the table. This ensures that the table is kept to a reasonable size. Most NAT implementations can also track Transmission Control Protocol (TCP) clients on a per-connection basis and remove them from the table as soon as the connection is closed. This is not possible for User Datagram Protocol (UDP) traffic, since it is not connection based.

Many higher-level TCP/IP protocols embed client-addressing information in the packets. For example, during an "active" File Transfer Protocol (FTP) transfer the client informs the server of its IP address and port number and then waits for the server to open a connection to that address. NAT has to monitor these packets and modify them on the fly to replace the client's IP address

(which is on the inside network) with the NAT address. Since the IP addresses are encoded in American Standard Code for Information Interchange (ASCII), this may result in a change in the size of a packet (for instance 10.18.177.42 is 12 ASCII characters, while 193.45.228.137 is 14 ASCII characters [6]). The TCP sequence/acknowledge numbers must be modified as well. Most protocols can be supported within NAT; some protocols, however, may require that the clients themselves are made aware of the NAT and that they participate in the address translation process.

Because the port-mapping table relates complete connection information - source and destination address and port numbers - it is possible to validate any or all of this information before passing incoming packets back to the client. This checking helps to provide effective firewall protection against Internet-launched attacks on the inside network.

Each IP packet also contains checksums that are calculated by the originator. They are recalculated and compared by the recipient to determine whether the packet has been corrupted in transit. The checksums depend on the contents of the packet. Since NAT must modify the packet addresses and port numbers, it must also recalculate and replace the checksums. Careful design in the NAT software can ensure that this extra processing has a minimal effect on the gateway's throughput [4].

Thus, we see that NAT provides transparent and bi-directional connectivity between networks having arbitrary addressing schemes. The next section discusses the merits and demerits of NAT.

2.2 Merits and Demerits of NAT

IP NAT has become an increasingly common function in the Internet for a variety of reasons. Network address translations are used to interconnect a private network consisting of unregistered IP addresses with a global IP network using a limited number of registered IP addresses. Network address translations are also used to avoid address renumbering in a private network when the topology outside the private network changes for a variety of reasons [5]. NAT enhances network privacy since assigned addresses of the internal hosts are hidden from the outside world. It, thus, defeats port scanning of subnets by malicious hackers.

A number of NAT deployments are currently in use and, naturally, a large number of Internet applications work transparently with Network Address Translations. However, there are applications for which Network Address Translations would fail and customized Application Level Gateways (ALGs) are required to perform translations for those applications.

NAT has the potential to interrupt the end-to-end nature of Internet applications, thereby threatening end-to-end security and other end-to-end functions, such as quality of service guarantees. In addition, NAT imposes topology restrictions and other constraints on the protocols and applications that run across Network Address Translations [5].

NAT can also be used to do load balancing for network servers being accessed by an organization. The NAT-enabled router can implement a load-balancing algorithm that may periodically check server loads. Based on some criteria it can decide which translated destination IP address to assign to the client request packet. The translated IP address will be the address of a real server that can accept the client request at that time, based on user defined load-balancing algorithm. The same principle also allows NAT to be used in high-availability and fail over schemes, where the NAT-enabled router may divert all incoming traffic to a backup server by simply translating the destination address to the IP address of the backup server when it detects that the primary server is down.

However, the NAT device must store significant information about the connection it translates, which "normal" routers do not need to do. This is because only a NAT device knows how to bridge the link between the two or more disparate addressing schemes and makes the domains appear seamless. In addition, the NAT device must maintain the state information for all connections going through it, including, for instance, load balancing or redundancy information of different network servers that are deployed in a cluster. The storage of port as well as IP address translation information in a translation table leads to a finer granularity in distributing loads across multiple servers, but also causes the connection state information to swell.

The problem of maintaining state information is further compounded when the packets undergo fragmentation. It is sometimes desirable to use TCP or UDP port numbers in addition to IP addresses in performing Network Address Translations. If a packet undergoes fragmentation, the state information of each fragment needs to be maintained by the router for all connections. The problem is that as soon as a packet has been fragmented, the NAT device cannot determine the port number except of the first fragment containing the TCP header. That is why the NAT device

must also keep state information about fragments. The header bytes associated with the first fragment must be stored for the duration of the session.

Each translation entry is associated with a timeout value so that hosts that have long been switched off or have stopped transmitting packets are eventually deleted from the list. This facilitates the reuse of the NAT IP address assigned to hosts.

Furthermore, NAT raises more interesting questions when it is implemented on inter-ISP boundaries. It complicates the issues relating to the trust boundaries and the service level agreements between two providers.

2.3 Current Research

NAT first started with only simple address-based translation [6]. Network Address Port Translation and the use of NAT for applications like high-availability and load balancing [7] came later.

Current research on Network Address Translation falls under three broad categories. The first category seeks to attain interoperability between NAT and other security mechanisms in a network, like IP Security (IPSec) [8], Internet Key Exchange (IKE), etc. NAT breaks the Authentication Header (AH) in IPSec because IPSec uses a Hashed Message Authentication Code (HMAC) to detect and prevent any change to IP packets. Also, NAT must translate private source/destination IP addresses into globally routable IP addresses inside the IP-header. Thus, if an IP packet is protected by an AH before it passes through a NAT device, the HMAC integrity check on the modified packet will fail and the receiver will discard the modified packet. This is a compelling interoperability issue that led to the IPSec group of Internet Engineering Task Force (IETF) to investigate NAT traversal through IPSec [9] and NAT-IKE interoperability [10].

The second area of research focuses on having NAT support newer applications and protocols like voice over IP [11], IP version 6 (IPv6) [12] and mobile networks [13]. In a voice over IP call, the underlying H.323 or Session Initiation Protocol (SIP) session is typically composed of multiple simultaneous TCP or UDP connections. This poses an additional burden on a NAT-enabled device since the device has to keep track of the different segments of the same call in addition to using the TCP/UDP layer addresses to do IP address translations in the IP header.

There is also ongoing research in design and development of application level gateways. When a protocol is unable to pass cleanly through a NAT, the use of an application level gateway (ALG) may still permit operation of the protocol [14]. DNS_ALG [15] is an example of an application level gateway that implements Domain Name Service (DNS) extensions to NAT.

The third category of research in Network Address Translation studies the performance impact of the Network Address Translation feature on a network device including newer NAT features and extensions that are being proposed literally everyday. It is in this area that this thesis will enhance the body of knowledge. The performance impact of NAT on a NAT device, in general, depends on the operating system of the device, the traffic load and the application that the device is routing. Hasenstein describes a number of NAT configurations and conducts performance measurements of NAT on Linux in [16]. That research focused on measuring the delay a packet undergoing NAT translations experiences. There is no research to measure the performance impact on NAT in terms of memory utilization and CPU utilization in the network device. In this thesis, we chose routers running Cisco's Internetworking Operating System (IOS) to measure the impact on the CPU utilization and memory utilization in a router running NAT under different traffic loads and packet sizes. IOS is an event-driven, non-preemptive operating system that is implemented on different models of processors including the MIPS family and the Motorola 68000 family. NAT was first introduced in the IOS release 11.2 in Cisco products. A Cisco technical report describes the details of IOS's NAT features and configuration commands [17]. Although this research on the performance impacts of NAT was carried out on a particular platform and operating system, the conclusions drawn from it are reasonably general and should hold true for most devices running NAT.

2.4 Summary

Chapter 2 gave a general overview of how NAT works and the factors introduced by NAT that impact the performance of a router. Chapter 3 describes the performance testing methodology used to measure the performance impact of NAT on a network router in terms of delay, memory utilization, CPU utilization and throughput. Chapter 4 tabulates and discusses the experimental results.

Chapter 3. Methodology

This chapter describes the methodology for measuring the impact of Network Address Translation on the performance of a router. The chapter starts with a description of the various parameters that will be used to characterize the performance impact of NAT on the router and the guidelines used to carry out the performance tests. Next, the performance workloads and the corresponding test cases are defined. Description of the test network used to carry out the performance tests is given next. Finally, the statistics that are collected from the test network and the procedure used to collect them are described.

3.1 NAT Performance Characterization

NAT is a software feature of a router. Like any feature, it also has an impact on the router's performance. To characterize the performance of NAT on a router, clear definitions of the parameters to be collected from the router are needed. As discussed in Chapter 2, a NAT router's performance is a function of traffic load, traffic type (different applications), number of NAT entries and specific NAT configuration. It can be measured in terms of the following metrics:

- Packet Delay: The extra delay introduced by just the NAT feature, not taking into consideration the variable network delay
- Throughput: The rate (in bits per second) at which the router forwards packets with NAT
- Memory Utilization: Memory consumption by different NAT entries
- CPU Utilization: extra CPU utilization introduced by NAT table lookups, especially for packets with embedded IP addresses

There is a set of guiding principles that must be followed throughout the data collection process of the performance experiment. Performance testing uses standard workloads, which are used to determine the capacity limits of a device. These capacity limits can be in terms of certain parameters, like the few that are listed in the previous paragraph. To make realistic assessments of capacity thresholds, extremely high or extremely low input load conditions should be avoided [18]. It is good practice to commence collection of test data after the device has reached steady state. Collection of data should end before the device load is reduced to conclude the test. The testing of several workloads may be necessary to determine an optimal number of readings from

which to extrapolate any meaningful conclusions. These guiding principles were put to good use during performance data collection phase for this thesis.

A workload is the collection of all individual tasks that are processed by a device during a specific time period. Workloads vary due to the user varying certain configuration parameters for the feature under test, which lead to the creation of individual tasks that devices are expected to process [18]. We characterize the workload using four parameters: traffic volume, traffic type, packet size and bandwidth. The number of possible workloads that can be presented to a NAT-enabled router can be large. Therefore, standard workloads to be used in performance characterization are first defined. These standard workloads will be used to carry out the performance tests so as to have a common basis for performance evaluation and analysis.

The next section describes the design methodology for the test network that was used to carry out the performance tests.

3.2 Test Network Design Methodology

The hardware and software specification of each component is described in Section 3.3. Figure 3.1 shows the test topology for measurement of traffic delay.

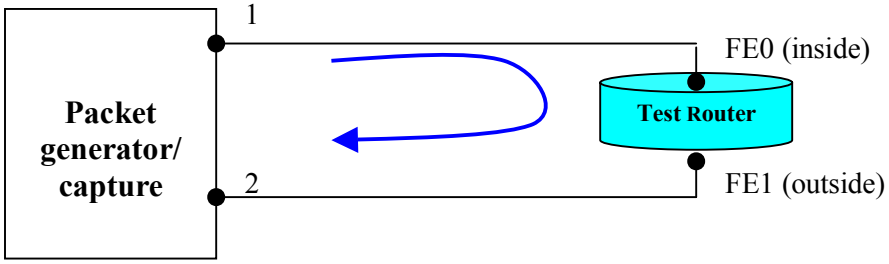

Figure 3.1. NAT performance test topology for delay.

The test router has two Fast Ethernet interfaces (FE0 and FE1). FE0 is configured to be the inside network (protected) interface. It is, by definition, the interface through which all internal hosts can reach the outside networks. FE1 is configured to be the interface to the internal network from the outside world. It is, by definition, the interface through which all external hosts can talk with hosts on the inside network. To eliminate network-induced latency, a simple two-node test topology has been adopted. This eliminates the effect of intermediate nodes on the end-to-end

packet latency. The time-stamped traffic originating from the packet generator is captured by the packet capture at the egress interface (FE1) of the router under test. In this way, only the increment in delay because of turning on NAT is captured. The “inside” and “outside” interfaces shown as Fast Ethernet interfaces (FE0 and FE1) serve only as examples and are not the norm. These interfaces can also be optical OC3/OC12 or any other high data-rate IP interfaces. The exact interfaces used depend on the transmission media used by a test router and the type of interfaces it can support on its chassis.

A two-node topology is also used to simulate a NAT gateway scenario for all other performance tests. It is represented as a three-node topology in Figure 3.2 only to explicitly show the “inside” and “outside” sections of the network. In the test actual setup, a single router with two Fast Ethernet interfaces acts as a packet generator as well as a packet capture tool as shown in Figure 3.1.

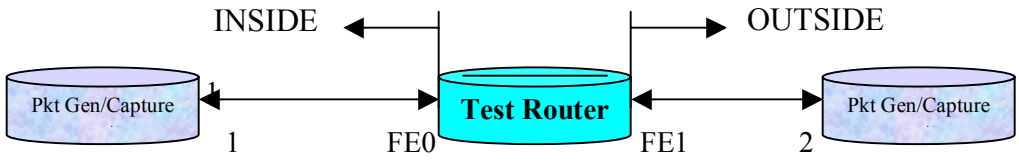

Figure 3.2. NAT performance test topology for delay, throughput, CPU, and memory.

Traffic generator is used to generate traffic on the end. Packet generator/capture router will receive and transmit packets simultaneously for bi-directional traffic. The network diagram consists of an outside network (packet generator/capture), a NAT gateway (test router), and an inside network (packet generator/capture). The inside refers to the protected network behind the test router and the outside network refers to the network on the other side of the test router. We designate one Fast Ethernet interface of the test router as "inside" and another Fast Ethernet interface of the same router as "outside" by way of router configuration commands. The address translation will occur from inside to outside interfaces or vice versa.

In each of the test scenarios, after the traffic flow is established, the Test Router is polled for various parameters, such as the test router’s CPU utilization and the number of packet drops.

The performance test setup includes two routers: one acting as an inside/outside network packet generator/capture and the other as the NAT gateway. Tests are performed with bi-directional

traffic streams. Fast Ethernet connections are used between the test router and packet generator/capture, with the test router having a Fast Ethernet Port-Adaptor module.

3.3 NAT Configuration Information

This section explains the configuration commands for a test router to enable the Network Address Translation feature.

In Cisco routers, there are different levels or modes in which a router may be configured. There is the concept of a global configuration mode, which configures features that apply to the overall operation of the router. Then there are sub-configuration modes such as the interface configuration mode, T1/E1 controller configuration mode, etc., wherein you specify the details of that particular interface or subsystem. Cisco documentation provides further details about router configuration modes [19].

To configure a basic NAT scenario on a router, the basic steps are as follows. One should first designate one Ethernet or Fast Ethernet interface as being "inside" and another as being "outside" on the router. Appropriate IP addresses must then be assigned to these interfaces. Next, in the global configuration mode, one must associate the IP address to the MAC address and specify the Ethernet framing type. This is further explained by the example configuration below, along with comments for each section.

```
interface FastEthernet 0
 ip address 172.16.10.1 255.255.255.0
 ip nat outside
 mtu 100
!-- Defines FastEthernet 0 interface with an IP address and designates it as a NAT outside
!-- interface.

interface FastEthernet 1
 ip address 172.16.50.1 255.255.255.0
 ip nat inside
 mtu 100
!-- Defines FastEthernet 1 interface with an IP address and designates it as a NAT inside
```

!-- interface.

ip nat inside source static 172.16.50.8 172.16.10.8

!-- States that any packet received on the inside interface with a

!-- source address of 172.16.50.8 will be translated to 172.16.10.8.

! -- This is an example of static NAT entry

The packet generator will simulate multiple "inside" and "outside" hosts, each with its own IP address. The router configuration will have many such static NAT entries for each IP address translation.

Now let us describe the individual components of the test network, i.e. the test router to be used and the packet generator and capture tool that will be used to perform the tests.

3.4 Network Components

The test network topology for both delay measurements and CPU, memory, and throughput measurements use the same test router and packet generator/capture tool. They only differ in the way in which they are used in these scenarios. These components are described in this section.

3.4.1 Test router

The test router used to conduct all performance tests is a Cisco 7206VXR router. The Cisco 7206VXR supports multiprotocol, multimedia routing and bridging with a wide variety of protocols and port adapter combinations available for Cisco 7200 series routers. In addition, the Cisco 7206VXR midplane provides increased support for multiple high-bandwidth port adapters. The Cisco 7206VXR has six slots (slot 1 through slot 6) for port adapters, one slot for an Input/Output (I/O) controller, and one slot for a network processing engine or network services engine. We can place the port adapters in any of the six available slots [20]. The particular test router used to carry out the tests had the Network Processing Engine – 300 (NPE-300). Two of these port adaptors are Fast Ethernet port adaptors, where one is designated as an "inside" interface and the other is an "outside" interface. The network-processing engine maintains and executes the system management functions for the routers. It also shares the system memory and environmental monitoring functions with the I/O controller [21]. The test router was also equipped with 254 MB Dynamic Random Access Memory (DRAM), 40 MB Synchronous

Dynamic Random Access Memory (SDRAM), 256 KB Layer-2 cache and 2048 KB Layer-3 cache. The operating system version that was used was Internetworking Operating System version 12.1 (15) Major release. The particular OS image was c7200-js-mz.121-15.bin It is an Enterprise Plus edition (denoted by the "js-mz" portion of the image name), which is a feature-rich edition. For details about Cisco's IOS image version and naming convention the reader should refer to [22] and [23].

The reason for choosing a major release is that it provides a relatively stable code base and, hence, the chances of an operating system malfunction are much lower. This particular OS image also supports NAT among other features.

3.4.2 Packet Generator/ Capture

To generate traffic streams, a Cisco proprietary IOS-based packet generator called Pagent was used. It is a special IOS image that runs on a number of supported Cisco platforms. Pagent has a packet-generation mode of operation called Traffic Generator mode (or TGN mode), which allows it to generate different types of packet streams. For the purpose of this performance testing study, a Cisco 7206VXR router was used to run the Pagent image. Similarly, Pagent also has a packet-capture mode called PKTS. It captures and displays packets from any combination of interfaces on a router.

For the test setup to measure latency shown in Figure 3.1, a timestamp configurable field is included in TGN and PKTS to help measure latency through the test router.

When the timestamp field is configured on a TGN traffic stream, TGN puts a timestamp in the packet just before sending it. This occurs before any transport checksums are calculated, so that the timestamp can be added into a valid TCP or UDP packet. When the timestamp field is configured in a PKTS filter, PKTS can extract the timestamp from the packet and compare it to the time it received the packet.

It is important that the same router is used to send and receive, so that both the send and receive timestamps come from the same clock source.

For the test setup in Figure 3.1, a Cisco 7206VXR router is used to generate as well as and capture packets. In this setup, the Cisco 7206VXR router operates in dual mode (i.e. TGN and PKTS) to capture as well as send packets.

In the TGN mode, streams with parameters like packet rate, packet size and destination Media Access Control (MAC) addresses and IP addresses can be defined. The PKTS mode allows the IP interface that connects to the test router to be in the promiscuous mode, thus allowing it to capture all incoming packets and store packet statistics such arrival timestamp, incoming packet rate, packet count, etc.

In addition to creating streams of fixed packet size, Pagent can also create a special type of traffic profile, an Internet Mix (IMIX) traffic profile. The IMIX traffic profile emulates the distribution of packet sizes typically found in the Internet. The IMIX profile consists of the packet mix described in Table 3.1.

Table 3.1. IMIX Traffic Profile Details

IP Datagram Size (bytes)	Proportion of Total
64	55.0%
570	15.0%
1024	20.0%
1500	10.0%

In the performance tests for this thesis, IMIX traffic consisted of seven 64 byte packets, two 570 byte packets, one 1500 byte packet, and two 1024 byte packets. The average packet length for the IMIX packets is 428 bytes.
UDP is used as the transport-layer protocol in traffic for all tests.

Due to the proprietary nature of this tool, the author cannot publish any further configuration details of the tool.

3.5 Tests

This section outlines test cases that were run to collect the performance data.

3.5.1 Test 1: Delay Measurement

As described in Figure 3.1, traffic delay is measured by comparing the timestamp difference between the transmitted and received packet. The total value of delay consists of various delays caused by test program and test system. In this test, relative traffic delay is calculated by comparing delays measured without NAT and delays measured with static NAT entries. This will eliminate delay factors other than NAT.

Test results are determined with 0, 1, 200, 2000, and 20,000 NAT entries. NAT lookup-table is gradually populated with NAT entries when the traffic generator tool begins to generate packets. The test router is initially configured with a set of translation rules by the administrator. When the first packet corresponding to each translation rule enters the router's interface, the NAT entry-creation process creates a NAT entry for all future packets of the same type. The NAT entries are stored in a lookup-table inside router's DRAM.

Since we only want to capture the effect of NAT on router delay, NAT is not affected if the packet size is changed. It might cause switching delays due to the larger size packets, but there is no increment in delay due to NAT. We performed tests with 64-byte and 128-byte packets to confirm this hypothesis.

3.5.2 Test 2: Throughput Measurement

The general methodology to measure throughput is to send packets of a particular size at a certain packet-rate from the packet generator tool. The same packets are captured at the other end of the test router after undergoing NAT translation. The total number of packets received is compared internally by the tool with the number of packets sent. If the packet-loss reaches a threshold of 0.1% that is the packet rate that will be tabulated. Packet-loss threshold is set to 0.1% as it is considered to be a statistically tolerable level of packet loss in data networks.

The impact on throughput is measured for 64, 128, 512, 1518 byte packet stream, and the IMIX packet stream for 1,000 NAT translation entries in the lookup-table. The packet length includes both header and payload.

3.5.3 Test 3: Memory Utilization

Each NAT entry will take about 120 bytes of system memory. So for a given platform, the total NAT entries that can be supported are limited by the system memory configuration. The aim of this test case is to measure the memory utilization on the router without NAT and by progressively increasing the number of NAT entries. Memory utilization on a Cisco router can be obtained from the "show memory" command. This is described as follows.

Syntax

show memory [*memory-type*] [**free**] [**summary**]

Syntax Description

*memory-type (*Optional) Memory type to display (**processor**, **multibus**, **io**, **sram**). If *type* is not specified, statistics for all memory types present are displayed.

free (Optional) Displays free memory statistics.

summary (Optional) Displays a summary of memory usage including the size and number of blocks allocated for each address of the system call that allocated the block.

For a given number of NAT entries, the quantity "Delta" will be computed. Delta represents the absolute difference in bytes between the measured value of baseline memory utilization (i.e. memory utilization when there are no NAT entries) and the measured values of memory utilization. Values will be provided for 1, 200, 500, and 1,000 NAT entries.

3.5.4 Test 4: CPU Utilization

The packet forwarding rate is governed by how fast the CPU can switch the packets. NAT entries affect the CPU utilization and, hence, the forwarding rate of the router. The aim of this test is to measure and quantify the effect of varying the packet size on CPU utilization of a router containing a fixed number NAT entries. For each packet size the test is repeated with four different numbers of NAT translation entries since the CPU utilization depends on the number of NAT entries as well. The interface media is Fast Ethernet. The results are tabulated for 64, 512,

128, 256, 512, and 1024 byte packet sizes each for 1,000, 3,000, 5,000, and 10,000 NAT translation entries.

3.6 Summary

In summary, we have presented the general performance testing methodology that is used to measure the performance impact of NAT on a router. The performance metrics were identified and the detailed performance testing approach was described along with a detailed description of the test network components and their configurations. Chapter 4 contains the test results obtained by running the performance tests in a laboratory.

Chapter 4. Observations and Results

The performance data corresponding to each test are tabulated and discussed below.

4.1 Test 1: Delay Measurement

Traffic delay is measured by comparing the timestamp difference between the transmitted and received packet. This makes use of the PKTS timestamp function of the Pagent tool described in Section 3.4.2. The absolute value of delay consists of various delays caused by the test program and test system. In this test, the relative traffic delay was measured by comparing delays measured without NAT and delays measured with static NAT entries. The details of the IMIX traffic pattern are also described in Section 3.4.2. Independent measurements were performed five times to get a reasonable amount of confidence in the measured delay values. The measured delay for each of the five experiments varied by a maximum of ~0.00002 ms from one reading to another, which is quite stable to use as representative measurements. Also, delay measurements are fairly accurate because the same device that generates the IP stream also captures the packets to measure latency. Thus, there is no dependency on Network Time Protocol (NTP) to synchronize the sending and receiving test devices, as they are parts of the same device. Test results are shown in Table 4.1.

Table 4.1. Delay Measurement Observations

# of NAT Entries	Average Delay (ms) 64 byte pkt	Average Delay (ms) 128 byte pkt	Average Delay (ms) IMIX pkts
None	0.00183	0.00188	0.00207
1	0.00187	0.00201	0.00209
200	0.00211	0.00236	0.00243
2,000	0.00282	0.00296	0.00302
20,000	0.00287	0.00301	0.00309

The delay measurements in Table 4.1 indicate that larger packets experience relatively greater delays due to NAT as compared to smaller packets. The delay increases with the increase in the number of NAT entries for a given packet length. The delay experienced by IMIX traffic is greater than the delay experienced by 128-byte packets because The average packet length for the IMIX traffic used in the tests was 428 bytes as mentioned in Section 3.4.2. Each packet in the IMIX stream experiences different amount of delay based on its length, which, over time, averages out to a greater delay than the delay measured for the stream of 128-byte packets.

4.2 Test 2: Throughput Measurement

The throughput measurement test was carried out five times for a reasonable degree of confidence in the stability of the measured throughput for different packet sizes and a 0.1% packet-loss threshold. It was observed that the throughput did not vary by more than 50 bytes/s from one reading to another. The mean of the five readings for a given packet size is tabulated in Table 4.2. The IMIX traffic profile, as described in Section 3.4.2, was also used to measure throughput to get a reasonable expectation of throughput for Internet-type traffic.

The impact on throughput is shown in Table 4.2. The total number of NAT entries was fixed at 1,000 to perform these tests. The baseline throughput with no NAT entries was determined to be 36,625 bytes/s.

Table 4.2. Throughput Measurement Observations

Packet Size (bytes)	Throughput (bytes/s)
64	30250
128	29750
512	29000
1518	28800
IMIX	28900

From the measured data in Table 4.2, it is observed that the throughput decreases with the increase in packet size of the incoming packets. IMIX traffic experiences a relatively higher throughput as it consists of packets of different lengths that experience different amounts of throughput loss to give a net throughput that is higher than the observed throughput for the longest packet length.

4.3 Test 3: Memory Utilization

Memory utilization on a Cisco router can be obtained from the "show process memory" command. Here is a typical output of the show command looks like at a router's console.

```
Total: 100798528, Used: 22454000, Free: 78344528
 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0      90456       1848   14057140          0          0 *Init*
   0   0       1068   26821648       1068          0          0 *Sched*
   0   0   71866860   48221956      28192    8396892     453096 *Dead*
   1   0        284        284       3828          0          0 Load Meter
   2   0      30988     768388      11372      20304          0 OSPF Hello
   3   0          0        876       6828          0          0 Check heaps
   4   0      20292          0      27120          0          0 Chunk Manager
   5   0    2096064    2022816     706900     949476    4985832 Pool Manager
   6   0        284        284       6828          0          0 Timers
   7   0          0          0       6828          0          0 ALARM_TRIGGER_SC
   8   0        284        284       6828          0          0 Serial Background
   9   0         96          0       6924          0          0 RM PROCESS
  10   0         96          0       6924          0          0 RM PROCESS
.
.
.
155   0    1271396      23812      22476     314712          0 OSPF Router
                                 22351160 Total
```

This command is described in detail in Chapter 3, and was used repeatedly to get the data points in Table 4.3. The packet size was fixed at 64 bytes. The author decided to measure the amount of memory used at around 80% into the duration of the test, as that would be when the memory utilization counter values are more stable. This also gives us a reasonable degree of confidence in the observed values of memory utilization for a given number of NAT entries. The test was repeated five times and the observed variation in the memory utilization values from one reading to the other, for a given number of NAT entries, was approximately 50 bytes. The values

recorded are those of the "Used" column in the output of the "show process memory" command as described in Section 3.5.3.

Table 4.3. Memory Measurement Observations

# of NAT Entries	Memory Used (bytes)	Delta (bytes)
None (baseline)	22770050	0
1	22770174	124
200	22795457	25407
500	22833650	63600
1000	22897480	127430

Table 4.3 consists of a column called "Memory Used (bytes)" whose values are derived directly from the output of "show process memory" command. For each row, the quantity "Delta" represents the absolute difference (in bytes) between the measured value of baseline memory utilization (i.e. memory utilization when there are no NAT entries) and the measured values of memory utilization for 1, 200, 500, and 1,000 NAT entries respectively. This is used to observe the amount of memory consumed for each additional NAT entry. From the measured data in Table 4.3 it is observed that, with the increase in the number of NAT entries, the memory utilization increases.

4.4 Test 4: CPU Utilization

The packet forwarding rate is governed by how fast the CPU can switch the packets. The purpose of this test case is to measure and quantify the effect of varying the packet size on CPU utilization of a router containing NAT entries.

The test was repeated five times for each combination of values of number of NAT entries and packet size to attain a higher degree of confidence in the measured values. CPU utilization measurements are the 5-second CPU utilization values that were observed by issuing "show process cpu" command on the router console. Internally, the router calculates this value by

polling CPU for the amount of time it is busy and the amount of time it is free in a 5 seconds interval. It appears as a distinct value when the "show process cpu" command is issued on the router console.

The observed values of CPU utilization varied approximately 1% between samples, which makes them reasonably stable. Table 4.4 contains the CPU percentage utilization over a 5-second interval. The mean of five observed CPU utilization values was taken for each combination of values of number of NAT entries and packet size. Results are in Table 4.4.

Table 4.4. CPU Utilization Measurement Observations

Packet Size (bytes)	CPU Utilization (%)			
	1000	3000	5000	10000
	Total Number of NAT Entries			
64	89	92	94	99
128	89	92	94	98
256	80	83	85	89
512	45	50	55	58
1024	28	32	35	39

From the measured data in Table 4.4 it is observed that with increase in packet sizes from 64 bytes to 1024 bytes, the CPU utilization decreases for a given number of NAT entries. It is also observed that the CPU utilization increases when the number of NAT entries increases from 1,000 to 10,000 for a given packet size.

4.5 Summary

This chapter summarized and tabulated the results for each test that was performed to study the effect of Network Address Translation on router performance. The next chapter analyzes the performance data collected and draws conclusions.

Chapter 5. Analysis and Conclusions

Every feature impacts the performance of a router, and Network Address Translation is no different. It is the aim of this thesis to understand this impact in quantitative terms. The purpose of this chapter is to analyze the data collected from the different tests and draw meaningful conclusions about the impact of NAT on a router performance in terms of the four parameters that we set out to test.

5.1 Test 1: Delay Measurement

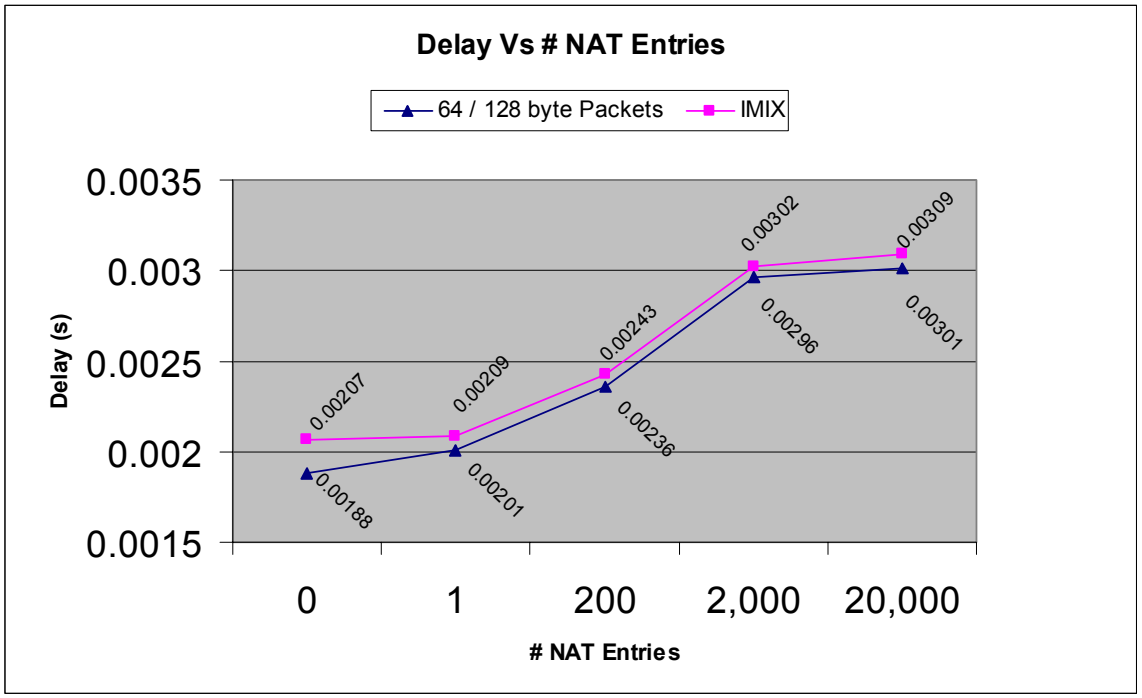

Figure 5.1. Delay versus number of NAT entries.

Figure 5.1 shows delay incurred by NAT as a function of the number of NAT entries. As expected, as the number of NAT entries increase, the delay increases. The delay caused by IMIX packets is higher as compared to the delay caused by packets of length 64 or 128 bytes. This means that with Internet type of traffic flowing through the system, the delay would likely to be higher than that from fixed size packets for any given number of NAT entries. IMIX traffic used in the tests consisted of seven 64-byte packets, two 570-byte packets, one 1500-byte packet, and two 1024-byte packets to conform to the proportions mentioned in Table 3.1. The average packet

length for IMIX traffic used in the tests is 428 bytes. It experiences a delay that is greater than the delay experienced by 64 bytes or 128 byte packet stream. The analysis also shows that packet size, in general, does not have appreciable impact on the delay variation as is observed by the overlapping delay trend line for 64 byte packet stream and the trend line for 128 byte packet stream.

5.2 Test 2: Throughput Measurement

Throughput, as a function of packet size, is shown in Figure 5.2.

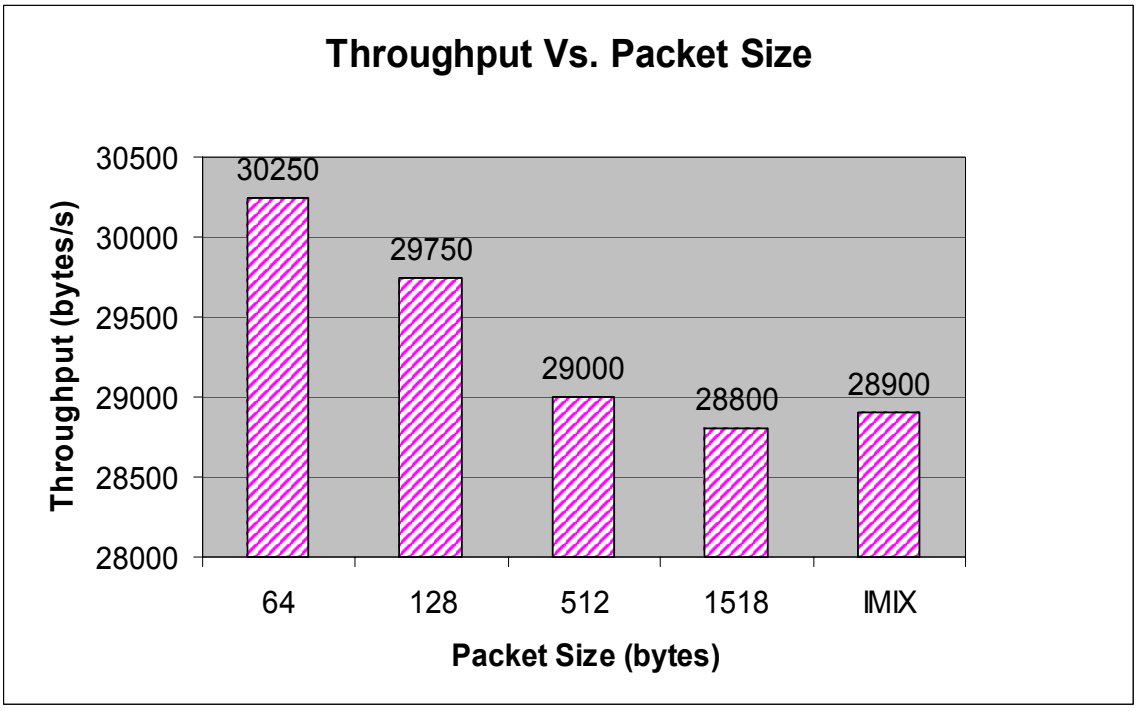

Figure 5.2. Throughput versus packet size.

The throughput of the system with NAT turned on decreases significantly with the increase in packet size because the network congestion increases for larger packets as the packets remain in the router's switching fabric for a longer duration than smaller packets that are switched faster. The total number of larger sized packets that are switched per unit time is less that the total number of smaller sized packets that are switched in the same time. This trend is exacerbated with the additional burden of performing NAT on incoming or outgoing larger-sized IP packets since the large-sized packets may have information that NAT requires embedded deep within in its payload, and the amount of processing required would be more than for smaller packets. There is an additional consideration for traffic stream with packet length 1518 bytes. The standard MTU

of the fast ethernet interface is 1500 bytes. All incoming packets of size greater than 1500 bytes will get fragmented. Thus, an additional amount of delay is incurred due to the fragmentation and reassembly of every 1518-byte packets as compared to the delay experienced for other packet sizes.

From the graph, it is calculated that the throughput decreases at the rate of approximately 8 bytes/s for every byte increase in packet size. The last data point is the data point for IMIX traffic (described in Section 3.4.2), where it is observed that the throughput is slightly higher than that observed for a packet size of 1518 bytes.

5.3 Test 3: Memory Utilization

Memory utilization, as a function of the number of NAT entries, is plotted in Figure 5.3.

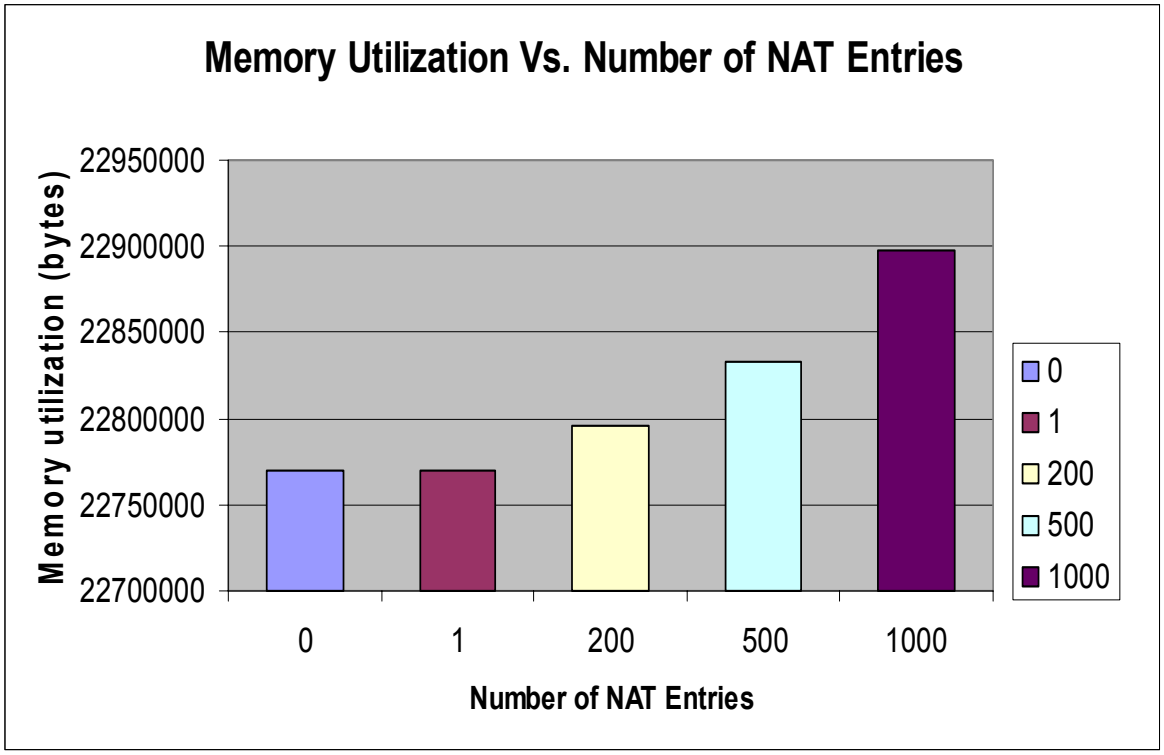

Figure 5.3. Memory utilization versus number of NAT entries.

From the graph, it is evident that the memory utilization and the number of NAT entries are almost linearly dependent. With every NAT entry that is added, there is always a certain fixed amount of increase in memory utilization of the router. An analysis of the graph's data shows that every single NAT entry utilizes approximately 124 bytes of the system's dynamic memory.

5.4 Test 4: CPU Utilization

CPU utilization as a function of packet size is shown in Figure 5.4. The trend shows that for any number of NAT entries, the CPU utilization increases with the decrease in the size of incoming packets. If the arrival rate (in bits/sec) is fixed, smaller packets require more NAT lookups, which causes a relatively higher CPU utilization of a packet stream containing smaller packets as compared to one containing larger packets.

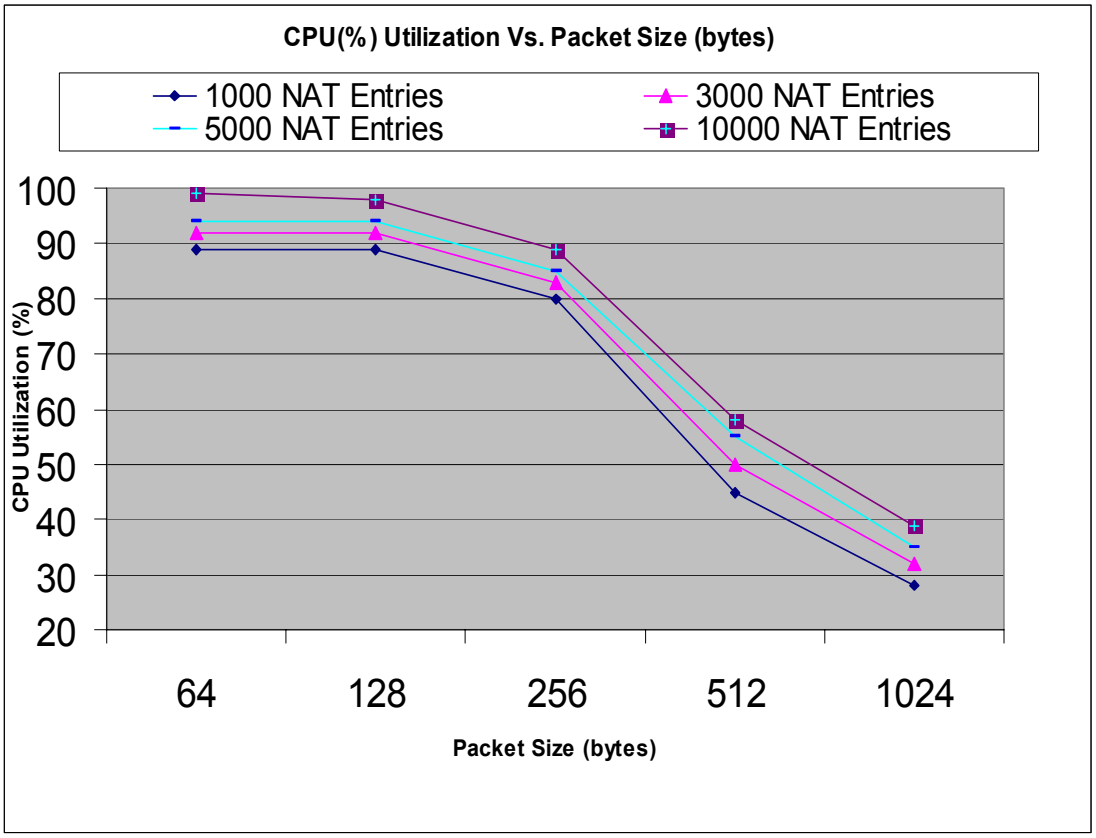

Figure 5.4. CPU utilization versus packet size.

Figure 5.5 is an alternative way of representing the same data.

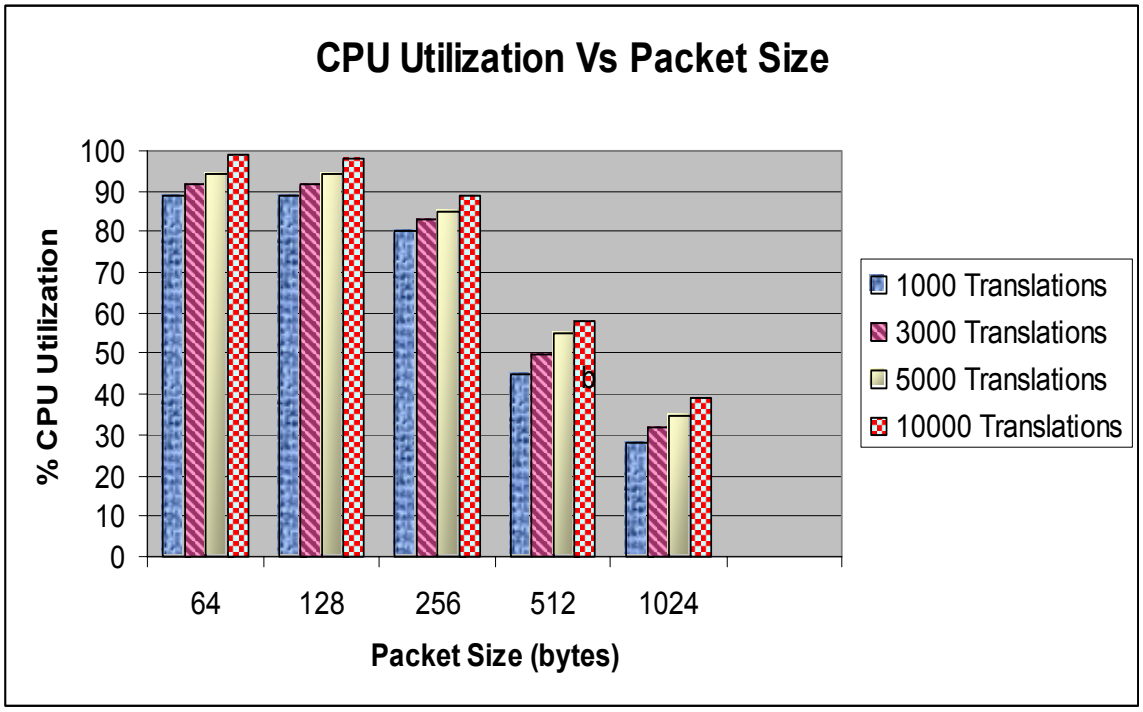

Figure 5.5. CPU utilization versus packet size.

5.5 Conclusions

With NAT enabled, there is a throughput performance degradation of approximately 17% with 1,000 NAT entries for an input stream containing 64 byte packets. The baseline throughput was determined to be 36,625 bytes/s when NAT was not configured on the system. With 1,000 NAT entries the throughput degraded to 30,250 bytes/s, which represents a performance degradation of approximately 17%. As the number of translation entries increases, the number of packets per second the router can forward decreases.

The rate at which the router can create new translation entries decreases as the number of NAT translation entries in the translation table increases. A corollary to this characteristic is, with a fixed packet rate (throughput), as the number of translation entries increases the percentage of CPU required to produce new entries increases. Four sets of tests were used to test this characteristic. One aspect of these tests determined how fast a router could create a specified number of new NAT entries without dropping a packet. The second aspect of these set of tests was to determine the impact on CPU utilization with an increase in the number of NAT entries.

CPU utilization was monitored while the number of NAT entries increased. The CPU utilization increased with the number of entries as in Table 4.4. The rate at which 100% CPU utilization is reached depends on the rate of creation of new translation entries and how many translation entries are already in the NAT translation table. For example, the packet rate was limited to 150 packets per second (PPS) and the CPU utilization climbed as the number of entries increased in the NAT translation table. At 100% utilization, only 40 new translations-per-second could be created when the NAT table contained 10,000 entries.

As the number of translation entries in the translation table increases, the amount of memory used increases. The maximum number of NAT entries depends only on the amount of memory in the router. As the number of embedded applications supported by NAT increases in more recent versions of router operating system images, the number of bytes per entry will increase due to flags and other required data. In the router's Operating System (OS) image that was used in the tests for this thesis, memory utilization was determined to be 124 bytes per NAT entry.

IOS NAT is a performance-intensive feature that performs read and write operations on packets. As a result of this, moderate to heavy performance degradation is observed in networks. Since the NAT is a single point of entry and exit into a network, the NAT box happens to be a performance bottleneck. The next section addresses the enhancements that can be considered to the NAT software to optimize system performance.

5.6 Suggested Performance Enhancements

Based on the tests carried out and their analysis, there are several performance enhancements that can be suggested. Some of the design considerations that we need to address with NAT performance enhancements are as follows.

- NAT should scale to thousands of entries without considerable throughput degradation.
- NAT should optimize the table lookups and minimize the total number of table-lookups to get to the right entry.
- New NAT entries should consume the least amount of memory.

The first two design consideration are valid only for large (high throughput) routers such as the Cisco 7206VXR router used in this thesis, but not for small (residential and small office/home office) routers.

Currently, the router implements a 32-bit hash-based translation lookup mechanism. In this mechanism, an incoming IP packet first makes a pseudo hash-key from an incoming packet by concatenating all the fields of the IP header that need translation (source address, destination address or payload application-specific information). An AND operation is performed on the pseudo hash-key to convert it a 32-bit hash-key. The 32-bit hash-key matches a hash-bucket that points to the exact translated entry associated with the packet. This is illustrated in Figure 5.6.

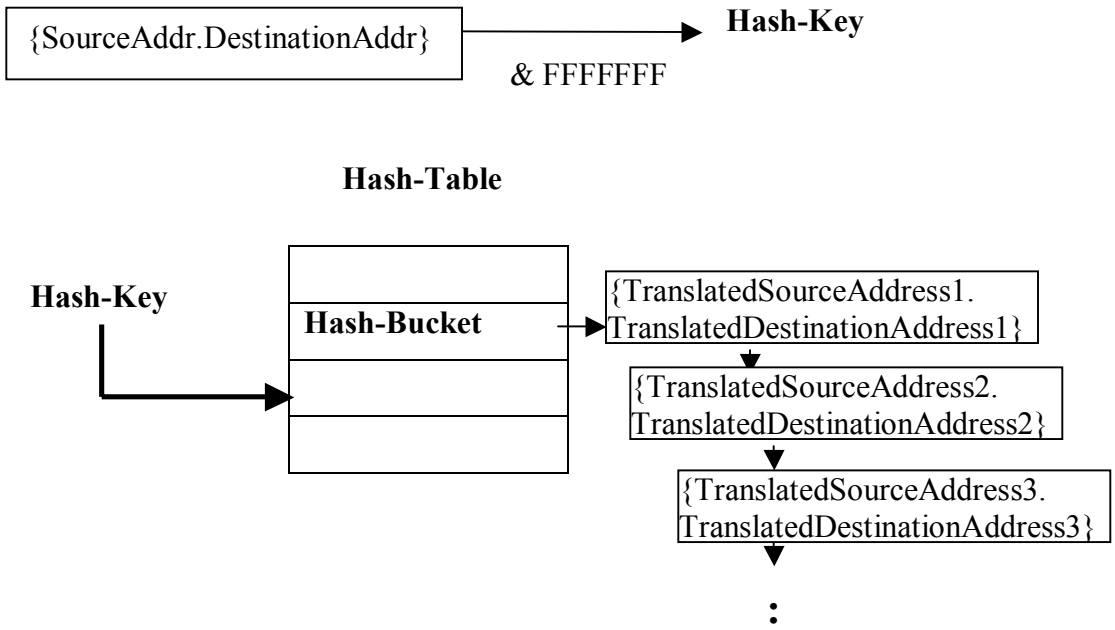

Figure 5.6. Hash-based lookup.

In this mechanism, there can be collisions as several different NAT entries can match the same hash-bucket when an AND operation is performed on the pseudo-key. In that case, the hash-algorithm stores the translated entries in a linked list. So, to determine translated addresses, the NAT routine must traverse the linked list as well to determine the exact NAT translation. Thus, we see that there are two levels of lookups in this scheme that lend to its inherent inefficiency.

A better table lookup algorithm could be used. For example, a radix-tree based algorithm can be used instead of a hash-based algorithm so as to reduce the total number of lookups and optimize the overall lookup process. In the radix-tree based algorithm, the entire pseudo-key is used to do the lookup using a bit-by-bit comparison while traversing down the binary radix-tree as shown in Figure 5.7.

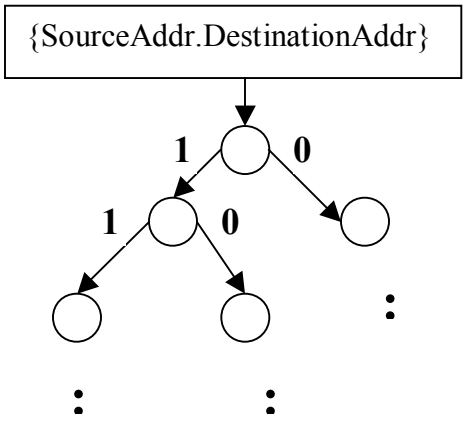

Figure 5.7. Binary radix-tree based lookup.

One local tree for all local destination entries and a global tree for non-local destinations is needed. When a packet flows from outside to inside, the lookup will be made in the global tree and when a packet flows from inside to outside the lookup will be made in the local tree.

Allocating memory for new NAT entries is an issue that needs to be addressed. A background process should keep track of the number of available memory chunks for creating new entries.

Drilling down into each IP packet to read application-specific information is a performance-intensive activity and can cause severe performance degradation. The use of NAT for Application-layer protocols such as DNS, FTP, SIP, etc., should be discouraged. The network administrators can selectively disable NAT functionality for specific applications packets by using router configuration commands.

NAT operation can be broadly broken down into two sub-operations: the entry creation operation matches an incoming pattern to a translation rule defined through access-lists or route filter commands; and the translation part matches the packet to an existing translation entry in the

lookup table. Currently, both operations are performed in the process path and not the interrupt path. In the process path, the packets are queued, the scheduler picks up the packets in a predefined manner and hands them over to the NAT application to process and send out. This is rather inefficient. An alternative approach is to do the entry creation function in the interrupt path. The interrupt path is faster and has all the intelligence needed to create a NAT entry. The only disadvantage of this approach is that we cannot do memory allocation (malloc) in the interrupt path, so we have to allocate memory in chunks at the initialization phase instead. Thus, there is a trade-off between the memory utilization efficiency and the speed of operation. Since memory is cheap and compact with current technology, the increase in memory utilization is not of much concern as compared to the efficient packet processing capabilities it provides in return.

5.7 Contributions

NAT has some inherent performance issues that need to be identified and addressed. This thesis contributes to improved understanding of various performance issues. The thesis also uses that information to address performance issues by suggesting software improvements. It identifies the most relevant performance parameters and devises a performance testing methodology to measure the impact of NAT on these performance parameters. It uses quantitative measurements to gauge the exact impact on the performance of a network device and identifies the areas that need improvement. This study can serve as a guideline to identify and mitigate NAT performance issues.

5.8 Further Work

There has been a considerable impetus in improving the performance of NAT to make it more scalable and less processor-intensive. The need for performance improvement in NAT is exacerbated by the emergence of newer technologies like IPv6. IPv6 demands high-performance NAT for it to keep track of IPv4 address translations and, also, protocol translations between IPv6 and IPv4 and vice versa. There is still a lot more work to be done in this area, but until IPv6 deployment gains some critical mass around the world, the pace of development in this area will probably be slow.

There is a common myth that IPv6 deployment will obviate the need for address translations because it provides a large and virtually inexhaustible address space. Theoretically, it may seem

true, but the reality of the situation is that there is a huge IPv4 installed base throughout the world. IPv6 cannot afford to ignore the presence of IPv4 completely and exist in isolation as a parallel realm to IPv4. So a hot new area of research and development in NAT is the ability to do protocol translation at the edge devices so that IPv6 clouds are able to interact with IPv4 clouds. The emerging standard called Network Address Translation-Protocol Translation (NAT-PT), which was first introduced in RFC 2766 [24], is quickly gaining ground in markets like Japan where IPv6 deployment is a reality. The details of the mechanism are given in RFC 2766 [24]. In summary, NAT-PT binds addresses in an IPv6 network with addresses in an IPv4 network and vice versa to provide transparent routing for the datagrams traversing between address realms. It uses a pool of globally-unique IPv4 addresses for assignment to IPv6 nodes on a dynamic basis as sessions are initiated across IPv6-IPv4 boundaries [24], as illustrated in Figure 5.8.

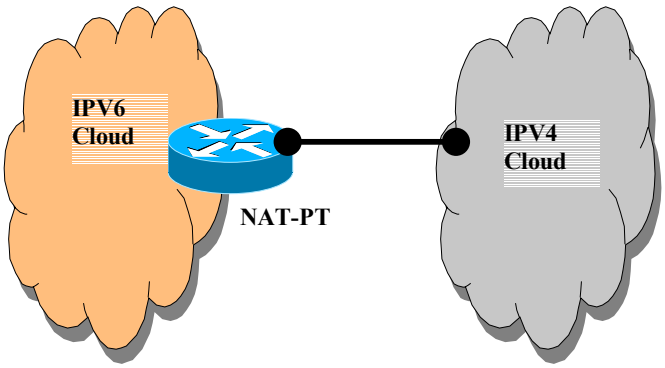

Figure 5.8. NAT-PT translating between IPv6 and IPv4 clouds.

Another area where NAT performance issue is generating opportunities for new development is in the areas of high-availability and failover mechanisms. NAT is usually deployed at the edge of networks and is a single point of failure in case the network device is forced out of service (for example, due to operating-system crash, hacker attack or an exception). This could potentially cause all nodes on the inside network to not be able to communicate with the outside world and vice versa. This scheme is illustrated in Figure 5.9.

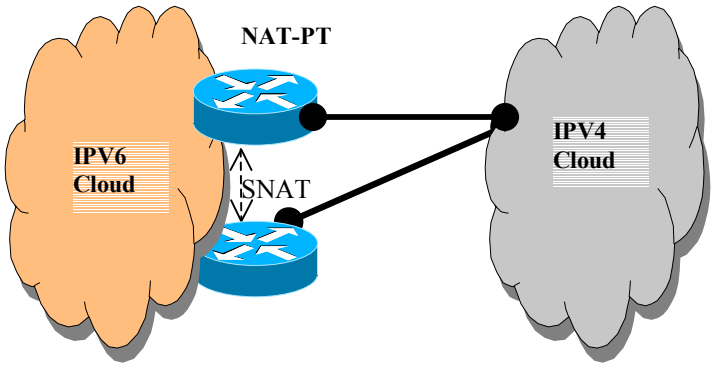

Figure 5.9. NAT-PT stateful NAT failover.

Stateful NAT (SNAT) failover and recovery is desired in which the secondary NAT device assumes the role of the primary NAT device in case the primary NAT device goes down.

References

[1] P. Srisuresh and K. Egevang, “Traditional IP Network Address Translator (Traditional NAT),” Internet Engineering Task Force, RFC 3022, January 2001. Available at http://www.faqs.org/rfcs/rfc3022.html.

[2] T. Hain, “Architectural Implications of NAT,” Internet Engineering Task Force, RFC 2993, November 2000. Available at http://www.faqs.org/rfcs/rfc2993.html.

[3] Cisco Systems, Inc., “NAT Order of Operation,” Technical Report, January 2003. Available at http://www.cisco.com/warp/public/556/5.pdf.

[4] Vicomsoft, Inc., “NAT Operation,” white paper, January 2003. Available at http://www.vicomsoft.com/knowledge/reference/nat.html#5.

[5] Internet Engineering Task Force, “Description of Working Group,” NAT Charter, November 14, 2001. Available at http://www.ietf.org/html.charters/nat-charter.html.

[6] K. Egevang and P. Francis, “The IP Network Address Translator (NAT),” Internet Engineering Task Force, RFC 1631, May 1994. Available at http://www.faqs.org/rfcs/rfc1631.html.

[7] P. Srisuresh and D. Gan, “Load Sharing using IP Network Address Translation (LSNAT),” Internet Engineering Task Force, RFC 2391, August 1998. Available at http://www.faqs.org/rfcs/rfc2391.html.

[8] B. Aboba, "IPSec-NAT Compatibility Requirements," Internet Engineering Task Force, Internet draft, draft-ietf-ipsec-nat-reqts-01.txt, March 2002. Available at http://www.ietf.org/proceedings/02jul/I-D/draft-ietf-ipsec-nat-reqts-01.txt.

[9] A. Huttunen, W. Dixon, B. Swander, T. Kivinen, M. Stenberg, V. Volpe and L. DiBurro, "UDP Encapsulation of IPsec Packets," Internet Engineering Task Force, Internet-draft, draft-ietf-ipsec-udp-encaps-01.txt, October 02, 2001. Available at http://www.ipsec.co.jp/products/security/sentinel/draft/draft-ietf-ipsec-udp-encaps-01.txt.html.

[10] A. Huttunen, W. Dixon, B. Swander, T. Kivinen, M. Stenberg, V. Volpe and L. DiBurro, "Negotiation of NAT-Traversal in the IKE," Internet Engineering Task Force, Internet draft, draft-ietf-ipsec-nat-t-ike-01.txt, October 23, 2001. Available at http://www.ipsec.co.jp/products/security/sentinel/draft/draft-ietf-ipsec-nat-t-ike-01.txt.html.

[11] Intel Corporation, "H.323 and Firewalls: The problems and pitfalls of getting H.323 safely through firewalls," white paper, April 01, 1997. Available at http://www.chebucto.ns.ca/~rakerman/articles/ig-h323_firewalls.html.

[12] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 clouds", Internet Engineering Task Force, RFC3056, February 2001. Available at http://www.faqs.org/rfcs/rfc3056.html.

[13] G. Knauer, "Pioneering mobile networks: NASA tests Mobile IP and Cisco Mobile Networks to forge new era of wireless and satellite communications," Packet Magazine, May 2002. Available at http://www.cisco.com/warp/public/784/packet/apr02/p53-cover.html.

[14] D. Senie, "Network Address Translator (NAT)-Friendly Application Design Guidelines," Internet Engineering Task Force, RFC 3235, September 1999. Available at http://www.faqs.org/rfcs/rfc3235.html.

[15] P. Srisuresh, G. Tsirtsis, P. Akkiraju and A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)," Internet Engineering Task Force, RFC 2694, September 1999. Available at http://www.faqs.org/rfcs/rfc2694.html.

[16] Michael Hasenstein, "IP Network Address Translation," Diplomarbeit, Technische Universitat Chemnitz, Chemnitz, Germany, 1997. Available at http://www.suse.de/~mha/linux-ip-nat/diplom/nat.html.

[17] Cisco Systems, Inc., "Cisco IOS Network Address Translation (NAT)," Technical Report, September 1998. Available at http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ionetn/prodlit/1195_pp.htm.

[18] Bellcore, LSSGR, "Traffic Capacity and Environment," Technical Report TR-TSY-000517, May 1995.

[19] Cisco Connection Online Documentation, "Introduction to Cisco Router Configuration," June 21, 2001. Available at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/1700/1710/1710scg/swskills.htm.

[20] Cisco Connection Online Documentation, "Cisco 7200 VXR Installation and Configuration Guide," March 22, 2003. Available at http://www.cisco.com/univercd/cc/td/doc/product/core/7200vx/72vxicg/.

[21] Cisco Connection Online Documentation, "NPE-300 and NPE-400 Overview," March 15, 2003. Available at http://www.cisco.com/univercd/cc/td/doc/product/core/7206/fru/npense/4448o3.pdf.

[22] Cisco Connection Online Documentation, "The Cisco IOS Image Naming Convention," February 10, 1994. Available at http://www.cisco.com/warp/public/732/abc/additional_info/naming.shtml.

[23] Cisco Connection Online Documentation, "ABC of Cisco IOS Software," December 22, 1996. Available at http://www.cisco.com/warp/public/732/abc/.

[24] P. Srisuresh and G. Tsirtsis, "Network Address Translation – Protocol Translation (NAT-PT)," Internet Engineering Task Force, RFC 2766, February 2000. Available at http://www.faqs.org/rfcs/rfc2766.html.

[25] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot and E. Lear, "Address Allocation for Private Internets," Internet Engineering Task Force, RFC 1918, February 1996. Available at http://www.faqs.org/rfcs/rfc1918.html.

[26] Washington School Information Processing Cooperative (WSIPC), State of Washington, "Network Address Translation (NAT) Issues," white paper, September 19, 2003. Available at http://www.wsipc.org/Information/Network/NAT.htm

[27] Cisco Connection Online Documentation, "Using NAT in Overlapping Networks," March 03, 2003. Available at http://www.cisco.com/warp/public/556/3.html.

VITA

Sarabjeet Singh Chugh, is originally from New Delhi, India. He graduated from Netaji Subhas Institute of Technology (formerly Delhi Institute of Technology) with a Bachelor of Engineering degree in Electronics and Communications Engineering in 1998. He joined Virginia Tech in fall 1999 for M.S. program in Electrical Engineering. He completed all required coursework towards the degree by fall 2000 and moved to San Jose, California to work for Cisco Systems, Inc.