NAT Router Performance Evaluation

HAYASHI yu-ichi 1070173

Supervised by Atsushi Kara

Abstract

This thesis describes a quantitative analysis of NAT routers forwarding performance. The forwarding performance of NAT routers has been outside the scope of RFC documents. NAT performance is only discussed regarding individual NAT implementations. As a result there have been few resources available for effective operation of NAT routers. This research identifies various bottlenecks on NAT routers and proposes a set of network-management methods that enable secure and stable network operation.

The experiments in this thesis show the nature of data loss and transmission delay when a client computer communicates with a server through an NAT router as follows: (1) The kind of CPU in an NAT router becomes a major performance factor when the size of data exchanged via the NAT router is large and the number of packet filters configured on it increases. (2) Most NAT routers discard packets when a client establishes UDP sessions using more than the preconfigured number of dynamically allocated ports. NAT routers use these dynamic ports to identify a unique connection between a local host and a global host. (3) Some NAT routers implement a state transition diagram that effectively utilizes the limited number of dynamic UDP ports available on the NAT router.

The above results give NAT routers with administration rules as follows: (1) configuring the NAT router with the minimum UDP session timer paying attention to its state transition, (2) enforcing a limited packet size for the traffic traversing the NAT router and keeping the number of packet filters at a minimum, and (3) limiting the number of sessions across the NAT router less than the preconfigured number of dynamically allocated ports on the NAT router.

1. Introduction

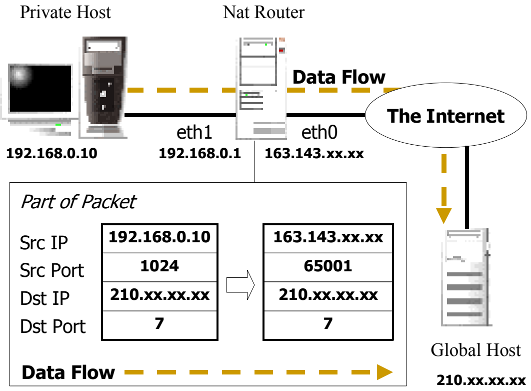
Many networks are connected to the Internet safely with a router, which protects computers from malicious data. In many cases, the router employs NAT [1, 2] and Firewall functions. Under IP address depletion, it is critical to evaluate the performance of NAT routers. Traditional experiments have executed a complete network evaluation with the Network Protocol Independent Performance Evaluator (NetPIPE) [3], Test TCP (TTCP) [4] and File Transfer Protocol (FTP). However, they only reported that NAT router throughput performance, while the performance of various outgoing packets remains unknown. This thesis measures NAT router load averages by (1) sending many outgoing packets to a global host, (2) sending packet sizes to a global host, and (3) sending packets with various MTU sizes from a client to a server.

The traditional session timer style has focused on keeping the session as long as possible on NAT routers, but this experiment releases sessions earlier in order to maximize security.

2. NAT Router

This section describes the principle of NAT routers, the principle of the session timer on NAT routers, and the kinds of session timers.

2.1 Principles of Operation



Src IP	192.168.0.10	163.143.xx.xx
Src Port	1024	65001
Dst IP	210.xx.xx.xx	210.xx.xx.xx
Dst Port	7	7

Fig. 1 Principles operation

Generically, NAT routers are assigned a private IP address to one network interface and a global IP address to another network interface. An NAT router must translate the pair of an address and a port in a local area network packet to a single global address on every outgoing packet. NAT routers have combinations of a service port and an IP address so that the mapping is unique between TCP and UDP sessions, in order to keep track of each client connection. The NAT router has an address/port-mapping table to remember the ports for each outgoing session. The port-mapping table associates the pair of the source IP address/port of a private host to the pair of the source IP address/port of an NAT router (Fig. 1). These functions are called IP masquerade, NAPT or simply NAT.

2.2 Session Management and Session Timer

NAT routers set the session timer to this address/port mapping when a private address client sends the first packet to a wide area network. This mapping table saves sessions in an NAT router. If private hosts send outgoing packets before the preconfigured session timeout, the session timer is reset to the initial condition. If a client does not send packets before the session timeout, this mapping is released (Fig. 2). The session timer is called an NAT timer.

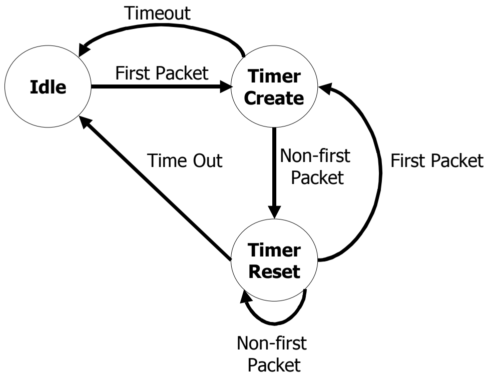


Fig. 2 Session timer transition state

2.3 NAT Timer

Session timers are provided in the value of a typical protocol.

- TCP session timers

All TCP sessions set a session timeout after sending the first TCP packet and receiving a FIN bit in TCP flags packet. The following values are TCP session timer values on IPCHAINS [5] in Linux, which is an IP masquerade implementation in Linux

  - ➢ TCP timer: 60 min. (Default)
  - ➢ FIN timer: 2 min. (Default)

- UDP session timers

IP address/port mappings for a UDP session are released by the UDP session time-out. Outgoing packets from a private host to a global host, and incoming packets from a global host to a private host reset a UDP session timer within a set UDP session timer on NAT routers.

The following values are UDP session timer values on IPCHAINS, which is an IP masquerade implementation in Linux

  - ➢ UDP session timer: 30 sec. (Default)

- ICMP timers

ICMP packets cannot reach wide area networks because ICMP does not use the port number. However, ICMP sessions that correlate queries and responses using a query identifier are uniquely identified by the tuple of (source IP address, ICMP Query Identifier, target IP address). Therefore, a ping command can be used.

  - ➢ ICMP timer: 1 min. (Default)

3. The Problem

NAT [1], [2] routers performance has been outside the scope of RFC documents. NAT performance is only discussed regarding individual NAT implementations. This thesis researches NAT router performance, which is evaluated by various clients' outgoing packets to wide area networks. The problem of packet switching between a private address and a global address involves the load of (1) many simultaneous outgoing sessions, (2) outgoing sessions that include large payloads, (3) outgoing fragmented packets, (4) many filter rules, and (5) many outgoing sessions more than the preconfigured number of dynamically allocated ports via NAT routers. This research analyzes data loss and transmission delay using packet switching, and identifies various bottlenecks of NAT routers and proposes network-management methods that enable secure and stable network operation.

4. Performance Experiments

This section describes the load of various packets sent to NAT routers in the experiment composition below (Fig. 3).

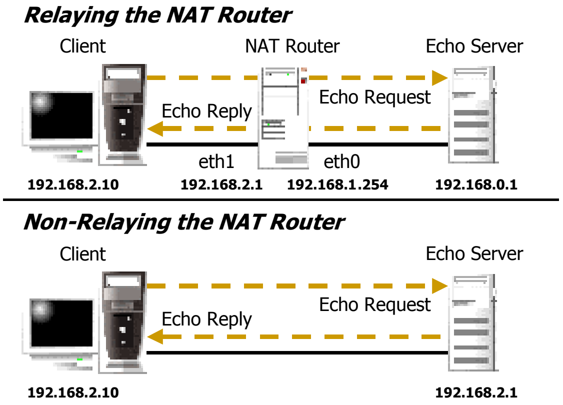


Fig. 3 Experiment composition

Server: CPU: Pentium III 450MHz, Memory: 256MB, Network Interface: 100BASE-TX (Intel)
Client: CPU: Pentium III 800MHz, Memory: 256MB, Network Interface: 100BASE-TX(3com)

4.1 Experiment Procedures

NAT routers forward an Echo Request/Reply between a local host and a global host using Echo-Service in this experiment. This section reports data loss and transmission delay in this experimental result, which compares one Echo Request/Reply packet via NAT routers to another Echo Request/Reply packet via non-NAT routers. IP masquerade preinstalled machines are UltraSPARC [6] (Table 1) with ipchains installed and OpenBlocks [7] (Table 2) with iptables installed.

Device	Value
CPU	UltraSPARC III
Memory	128MB
Network Interface	100BASE-TX(half-duplex)

Table 1 UltraSPARC spec

Device	Value
CPU	PowerPC 405GP 200MHz
Memory	64MB
Network Interface	10/100BASE

Table 2 OpenBlocks spec

Environment variables of SPARC (Table 3) and OpenBlocks (Table 4) are given. The Echo server sent the number of Echo Reply packets as well as receives the number of Echo Request packets.

Environment variables	Value
Service Ports	61000 – 65095 (4095)
TCP Session Timeout	60 minutes (prefix)
FIN Timeout	2 minutes (prefix)
UDP Session Timeout	30 seconds (prefix)

Table 3 SPARC' state transition

Environment variables	Value
Service Ports	1026 – 5000 (3974)
TCP Session Timeout	Long timer value (prefix)
FIN Timeout	Long timer value (prefix)
UDP Session Timeout	State transition

Table 4 OpenBlocks' state transition

4.2 Many Simultaneous Outgoing Sessions

When the client sends simultaneous packets from a private client to the global Echo server, these sessions occupy the number of dynamically allocated ports on NAT routers (Fig.4, Fig. 5). The packet payload size regarding these sessions is 100bytes. Experimental parameters are given in Table 5 (SPARC) and Table 6 (OpenBlocks).

Experimental parameters	Value
Occupied the Service Port for IP Masquerade	100%
The packet size of Echo Request/Reply	100byte

Table 5 The case of many sessions

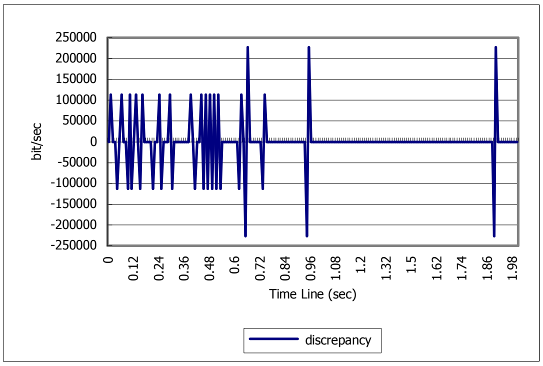


Fig. 4 Difference between echo request and reply packets (SPARC)

The result does not show data loss and transmission delay. The load of address/port mappings and the load of the checking session timer's table do not cause the CPU load of an NAT router when a client sends many outgoing packets.

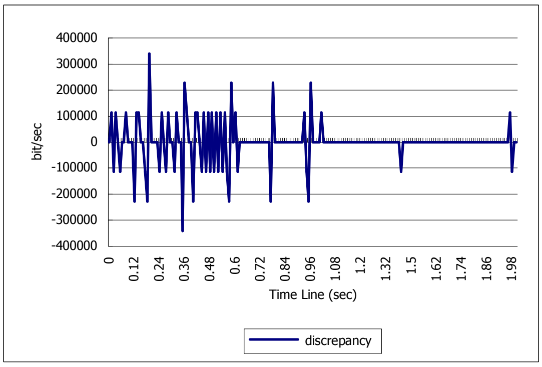


Fig. 5 Difference between echo request and reply packets (OpenBlocks)

4.3 Large Size Outgoing Packets

The Echo client sends outgoing packets of large size to the global Echo server in this section. The outgoing packet payload size is 1000bytes. The Echo client has simultaneously transmitted these 2000 packets to the Echo server. This experimental parameter on NAT router is given in Table 8.

Experimental parameters	Value
The number of sessions	2000
The packet size of Echo Request/Reply	1000byte

Table 6 The case of large packet size

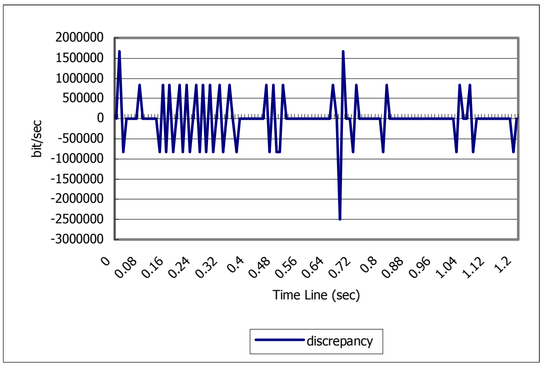


Fig. 6 Difference between echo request and reply packets (SPARC)

The data relaying SPARC does not show both data loss and transmission delay when the Echo client sends outgoing packets of large size to the global echo server (Fig. 6). Also, the data of non-relaying NAT routers do not show either. But the data of relaying OpenBlocks cause the data loss (Fig. 7) because the

CPU loads of OpenBlocks are high (Table 7).

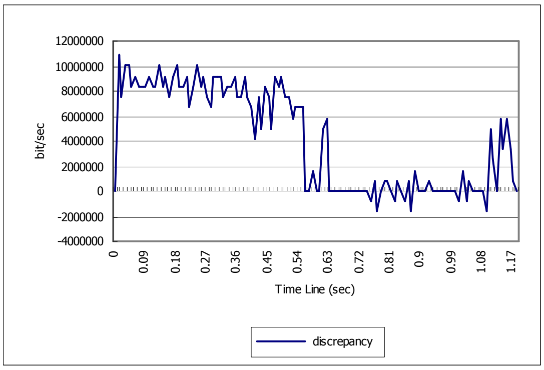


Fig. 7 Difference between echo request and reply packets (OpenBlocks)

A kind of packets	Packets loss
Relaying SPARC	0.01%
Relaying OpenBlocks	29.28%
Non-Relaying NAT routers	0.00%

Table 7 Packet Losses

In spite of decreasing the number of address/port mappings, the result of relaying OpenBlocks shows that packet loss is caused by the check sum and the calculation of the packet size in OpenBlocks.

4.4 Outgoing Fragmented Packets

The Echo client sends packets of setting fragment bit in IP flags to the global Echo server in this section. This outgoing packet payload size is 1443bytes (Payload = Ether Header (14bytes) + IP Header (20bytes) + UDP Header (8bytes) + Data Payload (1443) = 1501bytes). The Echo client has simultaneously transmitted these 2000 packets to the Echo server. The experimental parameters on an NAT router are given in Table 8.

Experimental parameters	Value
The number of sessions	2000
The packet size of Echo Request/Reply	1443bytes

Table 8 The case of fragment bit in IP flags

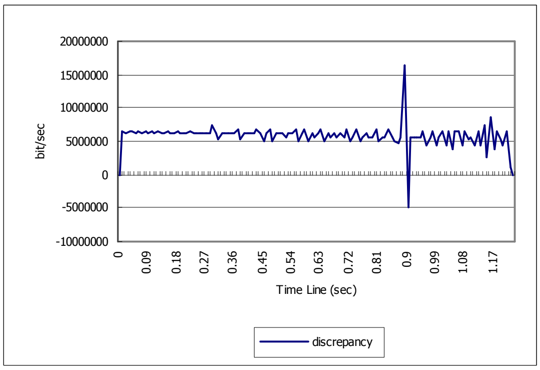


Fig. 8 Difference between echo request and reply packets (SPARC)

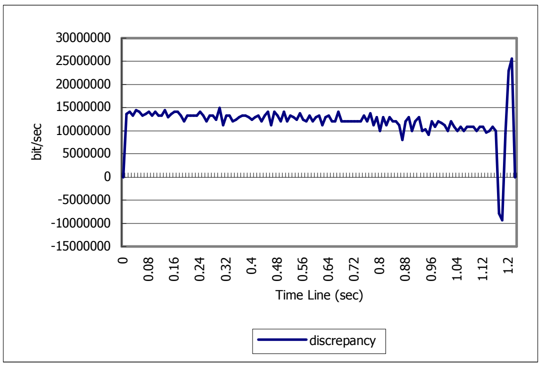


Fig. 9 Difference between echo request and reply packets (OpenBlocks)

The data of relaying SPARC (Fig. 8) shows 32.32% data loss (Table 9), but the data of non-relaying NAT routers show 32.28% data loss too (Table 9).

These data show that many data loss does not cause with NAT router. Also, The data relying OpenBlocks shows 57.83% data loss. This is more than relaying SPARC and non-NAT routers (Fig. 9). These experimental results show that packets of fragment bit in IP flags have data loss on relaying NAT routers as well as non-relaying NAT routers. Also, these results show that the performance of the packet processing differs according to the performance of the CPU too.

Kind of clients	Packet loss rate
Relaying packets with SPARC	32.32%
Relaying packets with OpenBlocks	57.83%
Non-relaying NAT routers	32.28%

Table 9 Packet loss rate on NAT routers

5. Exhaustion of TCP/UDP Ports

This section describes the status of NAT routers, which accept many outgoing sessions over the number of dynamically allocated ports.

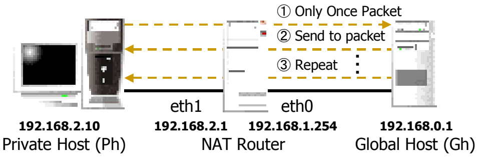
SPARC returns messages, which are full for address/port mappings when clients request more outbound sessions than the number of dynamically allocated ports. This result shows that SPARC does not accept new outgoing sessions until other sessions terminate. Also, OpenBlocks does not return these messages.

Messages returned by SPARC and OpenBlocks returned are different when clients request outbound sessions over the number of dynamically allocated ports (see Section 5.1). This experiment illustrates releasing sessions in OpenBlocks. The iptables implementation releases a session and deletes the address/port mapping from the mapping, and iptables implementation adds the new session to the mapping table.

6. NAT State Transition

The OpenBlocks' session timer is different from the SPARC session timer. The OpenBlocks' session timer has the state transition.

This experiment shows that the NAT router has state transition (Fig. 10). This result illustrated the NAT router state transition (Fig. 11). Also, another Linux (Red Hat Linux) NAT router installed iptables has the same state transition. Therefore, this NAT router transition is the iptables implementation.



(1) The private host (Ph) sends UDP packets to global host (Gh) via OpenBlocks. Gh sends response UDP packets to Ph after waiting for fixed time (T1).
(2) Ph does not send UDP packets after receiving response packets from Gh, and Gh sends next UDP packets to Ph after waiting at a fix time.
(3) Phase (2) is repeated until the NAT router closes session by preconfigured session timeout.

Fig.10 Experiment procedure

A new UDP session generates one or two extra packets to acquire a DNS for a DNS record. These sessions reduce the number of dynamically allocated ports to half or one-third the numbers. A NAT router releases a session to DNS more quickly than an NAT router releases a session to non-DNS using the state. The NAT state transition explicitly distinguishes UDP sessions between short UDP sessions and long-term UDP sessions, and an NAT router sets the appropriate timeout for each UDP session. The NAT state transition applies a session to DNS, as well as a session to NTP (Network Time Protocol).

6.1 NAT Router Security

UDP T1 and T2 session timeout values are both 30 seconds (Fig. 10). An NAT router session timeout should be preconfigured to approximately one second because the timer configuration utilizes dynamically allocated ports. When an NAT router is configured the one-second timeout, a session to DNS and NTP receive query information from a DNS server and an NTP server respectively within one second. Also, Widows XP has the NTP client function as a standard implementation. If the Windows XP client accesses a preconfigured default NTP server, it can always receive response information within 5 seconds. Setting short session timeout for NAT routers, can limit idle session and protect malicious access from wide area networks to the local area network.

6.2 Session Timer Dynamic Configuration

Many NAT router implementations have fixed session timeout values. System administrators cannot change session timeout values. It is important that an NAT router implementation calculates automatically the minimum session timeout values using an NAT router situation and changes the number of dynamically allocated ports. These automatic configuration systems prevent restricting and dropping many simultaneous UDP sessions, and these functions protect malicious access by early release of idle sessions. Also, system administrators cannot set the UDP session timeout value at less than one second in many NAT router implementations. For many hosts in networks exchange data transmissions on the microsecond time scale, the NAT router session timer should be set at the timeout value on the microsecond time scale. But an NAT router's CPU load is high because an NAT router's CPU checks the address/port-mapping tables every microsecond time. This phenomenon is a problem yet to be solved.

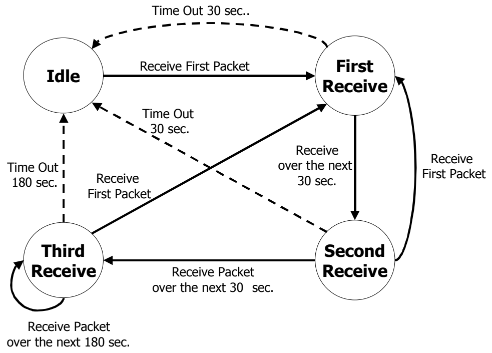


Fig. 11 The Ipchains state transition

7. Packet Filters

Applying many packet filter rules increases an NAT router's CPU load. This section describes data loss when applying many packet filter rules.

This experiment increases the number of packet filter rules for NAT routers, and measures data loss as a function of the number of filter rules experimental parameters are given in Table 10.

Experimental parameters	Value
Occupied the Service Port for IP Masquerade	100%
The packet size of Echo Request/Reply	100byte

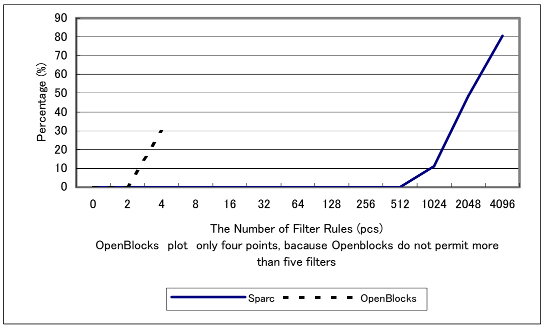


Fig. 12 Data loss by filter rules

Table 10 The case of applying packet filter rules

OpenBlocks permits only four packet filter rules. Data loss dramatically increases every applying packet filter rule. SPARC causes data loss over 1024 packet filter rules (Fig. 12).
This result in SPARC shows that high speed CPU processes data transition under the state of applying many packet filter rules. Also, this result in OpenBlocks shows that a low speed CPU causes data loss under the state of applying few packet filter rules.

8. Load Sharing using Dynamic Routing Protocols

A CPU load is high in the situation where many users simultaneously request sessions via an NAT router (Section 4). This section suggests a load sharing architecture using dynamic routing protocols.

8.1 Routing in a Local Area Network

This experiment sets up the Router in a local area network, and private clients set a default gateway to the Router (Fig. 13).
Routing information exchanged between the Router and NAT routers decides the data transmission path from private hosts to global hosts via the Router and NAT routers. For example, OSPF operates load sharing for a single NAT router.

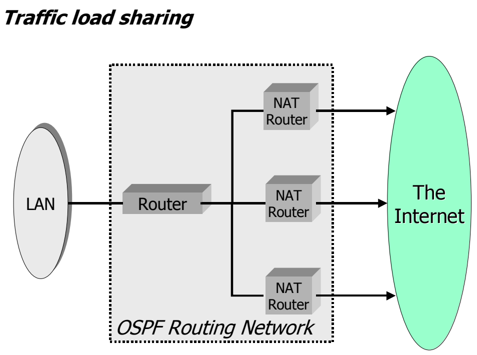


Fig. 13 The load sharing OSPF

9. Conclusions

This research shows the many cases of the NAT router's CPU load. The results show characteristic NAT routers and this section describes stable and safe operations methods of NAT routers with administration.

9.1 Efficient NAT Router Operation Rules

If the NAT router includes transition states in a NAT router session timeout, system administrators disclose the transition states and assign short intervals for session timeout value to the NAT router. This assignment serves not only to utilize the number of dynamically allocated ports but also protects malicious data from global networks to a private network with idle session ports. System administrators should limit as follow; (1) the number of outgoing sessions, (2) the size of outgoing packets, and (3) the number of filter rules according to differences in the packet processing performance in the CPU.

9.2 Stable Operations of Large-scale Private networks

The numbers of ports on NAT routers are 65000 ports because a service port number gives a hexadecimal digit. Most NAT router implementations limit service ports to a number of 5000 ports. If private clients request simultaneous outgoing sessions, an NAT router drops new sessions and overwrites existing sessions with new sessions. System administrators should detect these NAT router operations by limiting user sessions. In the case of large-scale networks in corporations, system administrators set multiple NAT routers in a boundary network between a private network and a wide area network and operate load sharing using dynamic routing protocols.

Reference

[1] P. Srisuresh and D. Gan, "RFC 2391, *Load Sharing using IP Network Address Translation (LSNAT)*, " August 1998.

[2] P. Srisuresh and M. Holdrege. "RFC 2663, *IP Network Address Translator (NAT) Terminology and Considerations*," August 1999.

[3] Scalable Computing Laboratory, "A Network Protocol Independent Performance Evaluator," *NetPIPE*, http://www.scl.ameslab.gov/netpipe (current December 2000).

[4] Printing Communications Assoc., Inc., "Measuring TCP and UDP Performance," *TTCP*, http://www.pcausa.com/Utilities/pcattcp.htm (current December 2000).

[5] Rusty Russell, "Linux IP Firewalling Chains," *ipchains*, http://www.netfilter.org/ipchains/ (current December 2000).

[6] Sun Microsystems, "UltraSPARC Processors," *Ultra SPARC*, http://www.sun.com/processors/ (current December 2000).
Linux, "The Netfilter Project," *iptables*, http://www.netfilter.org/ (current December 2000).

[7] Plat'Home CO., LTD., "Plat'Home Open Laboratory," *OpenBlocks*, http://openlab.plathome.co.jp/
J. Moy, "RFC 2328," *OSPF Version 2*, April 1998,