SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.557

RESEARCH ARTICLE

A scalable NAT-based solution to Internet access denial by higher-tier ISPs

Marwan Abu-Amara[1]*, Abdulaziz Al-Baiz[1], Ashraf S. Mahmoud[1], Mohammed H. Sqalli[1] and Farag Azzedin[2]

[1] Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
[2] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

ABSTRACT

The Internet is an interconnection of autonomous systems (ASes) that are mostly controlled by Internet service providers (ISPs). ASes use Border Gateway Protocol (BGP) to communicate routing information in the form of reachability paths. However, BGP does not guarantee that the advertised reachability paths will be exactly followed. As a result, traffic belonging to a specific network can be intentionally dropped as it is routed by BGP through a malicious ISP; a behavior we define as *Internet access denial*. The impact of Internet access denial, especially when performed by higher-tier ISPs, is significant. In this work, network address translation (NAT) is used as a solution to overcome the Internet access denial problem by hiding the traffic identity. The proposed solution is scalable to fit large networks, by using pools of IP addresses across several NAT routers. Moreover, the proposed solution addresses the server reachability problem that is associated with NAT routers by introducing a novel approach. The performance degradation of introducing NAT is significantly small as shown by our experiments' results. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS

Internet availability; malicious ISP; Internet access denial; resilient Internet; NAT

***Correspondence**

Marwan Abu-Amara, Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.
E-mail: marwan@kfupm.edu.sa

1. INTRODUCTION

The Internet is a result of interconnecting numerous autonomous systems (ASes). An AS is a network that is under a single administrative control. Most ASes are operated by Internet service providers (ISPs). ISPs are loosely classified into 3 *tiers*, on the basis of their size and interconnections. Tier-1 ISPs own large networks that cover one or more than one continent, and they form the Internet core. Tier-2 ISPs are smaller networks that mostly cover one or few countries. Tier-3 ISPs are the smallest, covering a country or a metropolitan area of a country. Tier-3 ISPs provide Internet service to end users and connect to one or few larger ISPs for the delivery of their customers' traffic to destinations outside their networks. Higher-tier ISPs, that is, tier-1 and tier-2 ISPs, carry not only traffic that belongs to their networks, but also traffic that is originated from or destined to one of the networks that they are connected to. Thus, packets that are sent from one end-user to another are carried over multiple different tier ISPs.

Autonomous systems are interconnected using inter-domain routing protocols. Border Gateway Protocol (BGP) is the dominant inter-domain routing protocol in the Internet. Hence, BGP is the inter-AS routing protocol that interconnects ISPs. It provides routing information as a set of IP address subnets (known as *prefixes*) and reachability information related to each prefix. Routes in BGP are described as a sequence of ASes that traffic will traverse to reach its destination.

Border Gateway Protocol suffers from many security weaknesses [1]. Many vulnerabilities in the design of BGP have become increasingly critical as the Internet has grown. One of the issues with BGP is the inability to control how traffic is routed through ASes. The received prefix reachability paths can only be considered as "promises." There is no way to ensure that traffic will actually be routed through these paths. Practically, routers may provide the list of ASes that propagated the BGP update messages, which are not necessarily the same as the list of ASes traversed by data packets [2]. BGP allows the network to control only which neighbor AS will receive the packet but not how that neighbor AS, or any other AS in the remainder of the path, will handle that packet. Moreover, many networks use load-balancing and multihoming

techniques to distribute traffic over multiple links. Thus, the traffic may go through different paths other than the advertised ones and may go through ASes that the traffic originator is not aware of.

This issue does not normally affect the delivery of traffic as packets will eventually reach their destinations regardless of the used path. However, many security concerns are raised because of this behavior. Packets may go through ASes that the traffic originator is unaware of as they do not appear in the AS path. The presence of a malicious ISP in any path to the destination results in the potential risk of routing the packets through that malicious ISP.

A malicious ISP can, for example, monitor, record, or even modify packets that are routed through it, performing man-in-the-middle attacks. It may also *blackhole* the traffic that belongs to a specific network (referred to as the *victim* network), that is, drop all the packets originated from or destined to the victim network. Hence, it denies providing routing services for that particular network, preventing it from accessing many destinations, namely the ones that are reachable through paths that go through the malicious ISP. Accordingly, we define *Internet access denial* by malicious ISPs as the process of filtering transit traffic to drop packets that belong to a specific network. Figure 1 illustrates how Internet access denial by a malicious ISP results in unreachability of the destination servers. As shown, the malicious ISP between the source and the destination routes packets from networks 1 and 2 normally but performs Internet access denial on network 3 traffic by dropping packets that carry network 3 addresses. In this case, network 3 is unable to reach the destination host.

The idea of malicious higher-tier ISPs seems unlikely at first because ISPs that perform Internet access denial are risking their reputation, and eventually their business, as they will lose customers. However, there are several reasons that may force an ISP to become malicious and perform Internet access denial against a specific organization or country. For example, Internet access denial can be driven by political motivations as governments may force ISPs to block Internet access to a specific region or country in an attempt to establish an Internet embargo on that specific region. Recently, many large services and networks have been attacked for political motivations. On December 2009, Gmail, for example, had many attacks targeting e-mail accounts of Chinese human rights activists [3]. Twitter, a popular social network, has also been attacked during 2009 by hackers from Iran [4]. Another prime example of political motivations of a service provider to deny Internet access to an organization are the recent attempts by many governments to pressure service providers to block access to *WikiLeaks* [5]. These types of attacks are driven by political forces. Moreover, ISPs' routers may be hacked by attackers and reconfigured to drop traffic, which causes Internet access denial. Although the latter case might be temporary, it still has an impact on the victim network. Moreover, malicious BGP path advertisements can redirect traffic to malicious ASes, an attack technique known as *BGP hijacking* [1]. Such an attack has actually taken place many times in the past, where an AS, mistakenly or intentionally, advertises BGP routes to prefixes that do not belong to it, and hence redirect all the traffic towards that AS [6].

In this paper, we tackle the problem of Internet access denial by malicious ISPs. Section 2 describes the problem causes and implications in more details. Section 3 provides a summary of related work on Internet access denial and the potential solutions. In Section 4, we describe how network address translation (NAT) can be adapted to provide a transparent solution for Internet access denial. We then evaluate the performance of the proposed NAT-based solution in Section 5. In Section 6, we discuss the limitations that the proposed solution raises and how they can be handled. Finally, we conclude the paper in Section 7.

2. INTERNET ACCESS DENIAL BY HIGHER-TIER ISPs

Most of the ASes that form the Internet core are owned by tier-1 and tier-2 ISPs. Internet traffic, sent from a host on one network to a destination on a different network, is likely to go through multiple ASes, of which one or more is a higher-tier ISP.

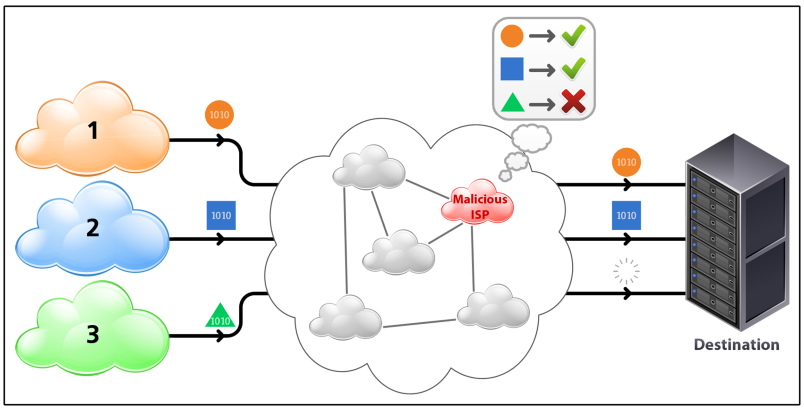

Figure 1. The presence of a malicious ISP in the path of packets results in destination unreachability.

As stated in Section 1, we define the *Internet access denial* by malicious ISPs to be the process of filtering transit traffic for the purpose of dropping packets that belong to a specific victim network. The malicious ISP configures its routers to drop, or blackhole, some or all the traffic that is originated from or destined to one or more IP prefixes of the victim network. We assume that the malicious ISP will use the network-layer information (i.e., source and destination IP addresses), to determine if a packet belongs to the victim network. Malicious ISPs can perform Internet access denial on any IP address blocks, ranging from a single host to an entire country.

The impact of Internet access denial depends on the location, size, and connection topology of the malicious ISP. Lower-tier ISPs can only cause Internet access denial if they exist in the route of the traffic, whereas higher-tier ISPs may have a larger impact.

Because tier-3 ISPs do not act as transit for other networks, they only carry traffic that belongs to their networks. Therefore, a malicious tier-3 ISP can only block access to its own network. Hence, the impact of this type of ISP is limited to only a small set of hosts and services. On the other hand, malicious higher-tier ISPs can have more impact as they can block not only traffic that belongs to their networks, but also all other traffic that passes through them in transit. For example, a malicious tier-2 ISP blocks access to its own network and to all its customer ISPs' networks. Furthermore, Internet access denial by tier-1 ISPs presents a more critical problem. A malicious tier-1 ISP can isolate the victim network and block it from accessing a large portion of the Internet. Because of the major impact that a malicious higher-tier ISP can cause, solutions to the Internet access denial problem should be studied and deployed. Figure 2 shows a simplified network of ISPs of different tiers and how Internet access denial by higher-tier ISPs results in a larger inaccessibility to other parts of the network.

3. RELATED WORK AND SOLUTIONS

3.1. Internet unavailability

The growing importance of the Internet has motivated many studies on the Internet resilience against different types of outages, failures, and attacks. Internet unavailability takes place as a result of either accidental or malicious causes. Hardware and/or software failures, misconfiguration, and traffic congestion are non-malicious activities that may cause Internet unavailability. Many solutions have been proposed to address these issues in the physical, routing, and application levels [7–10].

Malicious activities that may cause Internet unavailability include denial-of-service (DoS) attacks, security breaches, terrorist attacks, intentional hardware failures, and deliberate Internet access denial by service providers. Most of the research that has been done in this area targets DoS attacks and security breaches [11–14]. Only few research efforts targeted terrorist attacks and intentional hardware failures [15,16].

Internet access denial takes place when two conditions are met: packets are routed through a malicious ISP, and the malicious ISP drops these packets. Hence, the Internet access denial problem can be resolved by eliminating one or both of these conditions. Therefore, two classes of solutions can be considered: solutions to control the traffic path so that it does not pass through the malicious ISP and solutions to prevent traffic from being dropped by the malicious ISP by concealing the traffic identity.

3.2. Controlling the traffic path

The first class of solutions to the Internet access denial problem depends on preventing the traffic from being sent through the malicious ISP. Although BGP provides reachability information that includes the AS path, it does not allow a network to control the actual routing path of its traffic. A network can only select which neighbor ASes will route its packets but does not know how that neighboring AS is going to handle them.

Controlling the outgoing and incoming traffic requires modifications or adjustments of the routing protocols. *Source routing* [17], which allows the traffic originator to specify the path its traffic will travel through, is a solution to control the outgoing traffic so that it avoids the malicious ISP. However, the existing Internet protocols do not implement this type of routing. Modification of BGP is needed at all routers in the Internet to achieve this type of traffic control.

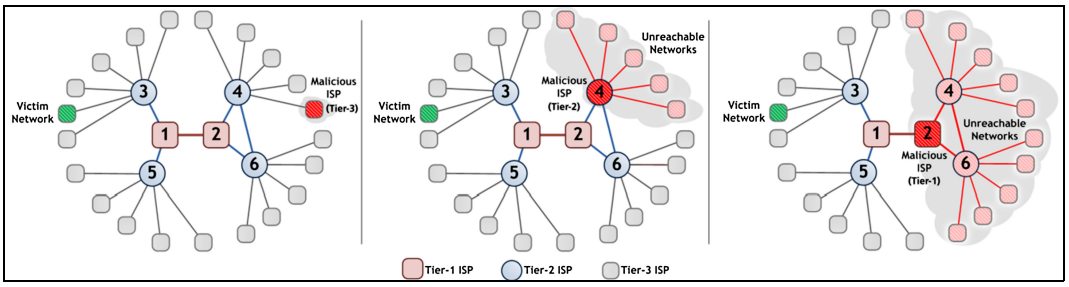

Figure 2. Impact of Internet access denial by different tiers of malicious ISPs.

Quoitin *et al*. [18] proposed *BGP tuning*, which controls incoming traffic by using some techniques to influence the path selection process of remote ASes. Three techniques were presented: *AS-path prepending*, where the length of the advertised AS paths is reduced to present it as a shorter path; *prefix splitting*, where the advertised IP prefix is disaggregated into a set of smaller IP prefixes to lead remote routers into selecting it as the longest prefix match; and the use of *community*, where remote cooperating routers use the community field in the BGP advertisements to identify the preferred paths.

Virtual peering, also proposed by Quoitin [19], is a technique to control incoming traffic by using multi-hop BGP sessions. Remote ASes establish virtual-peering tunnels to control the traffic destined to the local AS. This solution is not scalable as it requires all remote ASes to implement virtual peering and establish tunnels for all communications.

Virtual transit, proposed by Mahmoud *et al*. [20], is a modification of virtual peering. The introduced difference is that remote ASes advertise the virtual-peering tunnel reachability information to their neighbor ASes allowing them to use the same established tunnel to transmit traffic to the local AS. Virtual transit has better scalability than virtual peering as only a portion of Internet ASes need to implement it.

3.3. Hiding traffic identity

The other class of Internet access denial solutions is based on hiding traffic identity from the malicious ISP so that it does not identify the traffic's origin or destination. These techniques use IP addresses that are different from the blocked ones. Therefore, the malicious ISP will be misled into routing the traffic without filtering it.

One solution is to change the IP addresses of the victim network to different ones. The victim network can just register a new IP block and use it instead of its current one. This, however, only provides a temporary solution as the malicious ISP can easily detect the new IP block and will simply block it again. Hence, this solution is not robust.

Network-layer encapsulation and tunnels are other methods of hiding the identity. Traffic is carried through a tunnel created between the two tunnel endpoints. First, packets are routed as usual until they reach the first tunnel endpoint. At the first tunnel endpoint, each packet is encrypted (optionally) and encapsulated, as payload, into another packet, and then sent to the other tunnel endpoint. The intermediate routers will only see the two tunnel ends as the source and destination addresses. Packets then are decapsulated at the other endpoint of the tunnel and sent to their destination.

There are many tunneling protocols, such as *IP-in-IP* [21], *Internet Protocol Security* (IPSec) [22], and *Generic Routing Encapsulation* [23]. Additionally, *anonymous routing* protocols, such as *Onion Routing* [24], *Cashmere* [25], *Crowds* [26], and *Hordes* [27], provide means to hide the content of the packet, as well as the identities of the source and destination, from the routers that carry the traffic.

Implementing tunneling as a solution to bypass Internet access denial requires at least two cooperating networks as the endpoints of the tunnel [28]. One of them should be located before the malicious network in the route path, and the other is located after it, so that the tunnel is established through the malicious ISP. Although this solution is highly reliable once deployed, it does not work if no cooperating networks are found before and after the malicious ISP, such as the case of stub malicious networks. It also does not work when the destination host is within the malicious network. Moreover, the use of anonymous routing protocols as a solution for Internet access denial results in a very high performance degradation [24,29].

Network address translation [30,31] is a technique that allows a large number of hosts to use a small set of IP addresses to communicate with other hosts on the Internet. A NAT router separates the network into two subnetworks, a private network, where the hosts are given private IP addresses, and a public network, where the NAT router is connected to the Internet using its public IP address.

Network address translation can be used as an identity-hiding technique, by using a set of non-blocked IP addresses as the NAT's external IP addresses. All traffic will then use these non-blocked IP addresses when it is sent through the Internet. The solution that we adopt in this paper is based on using NAT as an identity-hiding technique at the gateway level of the victim network.

4. NAT-BASED SOLUTION FOR INTERNET ACCESS DENIAL

Network address translation is a technique that enables a number of hosts to use the same public IP address to connect to the Internet. It was first proposed by Paul Francis [31] as a temporary solution for the IPv4 address exhaustion problem. As stated in Section 3, a typical NAT network consists of a private network and the external, public network, through which the NAT router is connected using a public IP address. The NAT router and the private network behind it appear to the Internet as a single host, with a single public IP address. Together with its main purpose of extending the IP address space, NAT also provides a level of security for the private network by hiding its internal addressing structure and topology.

Network address translation can be used as an identity-hiding technique to bypass Internet access denial. The victim network uses NAT routers as gateway to connect to neighboring networks and use a set of non-blocked IP addresses as the NAT routers' external public IP addresses. These addresses are not part of the IP ranges registered to the victim network; they are obtained from a neighboring network. The outgoing packets, therefore, will not be blocked by the malicious ISP as they will not be recognized as part of the victim network.

4.1. Adapting NAT as a solution to Internet access denial

Implementing the NAT solution requires setting the gateway routers to use NAT to translate all traffic into the non-blocked public IP addresses. Once NAT is enabled and configured properly, clients within the victim network can send requests and receive responses. Even if traffic passes through the malicious ISP, it will not be recognized as traffic that belongs to victim networks, and the malicious ISP will route it normally through its network.

The non-blocked public IP addresses that will be used by the victim network can be obtained through different means. For example, IP addresses from a neighboring network or country could be leased privately. This prevents the malicious ISP from identifying the origin of traffic. It is important that the records of these IP addresses are kept pointing to their original network, rather than the new network, to avoid being exposed to the malicious ISP. We assume that the malicious ISP will only block the IP addresses of the victim network and will not attempt to track or identify the source of the traffic that uses the new IP addresses as long as these addresses are not pointing to the same victim network.

Although entities in the private network behind NAT are recommended to have IP addresses from the reserved private address blocks, they can still work with different IP address blocks if the NAT routers are configured properly. Therefore, for the NAT solution to Internet access denial, entities within the victim network, including hosts and routers, do not need any modifications to adapt with this solution. The only modification needed is at the gateway routers. NAT can be set in the existing gateway routers, or dedicated NAT routers can be used as a layer between the private network and the gateway routers.

As stated earlier, hosts and routers in a typical NAT setup are assigned private IP addresses from the reserved private IP blocks. However, in our proposed solution, we keep the existing IP addressing without changes. NAT routers can be set such that they recognize the internal IP address blocks as private addresses, and the translation is carried out between the internal IP blocks and the external public IP addresses.

There are many advantages to keeping the same IP addresses. The NAT solution would be transparent to the clients within the victim network as they do not have to make any changes in their networks. Moreover, local *Domain Name System* (DNS) servers do not have to update their records with private IP addresses because no changes are made internally. In addition, keeping the same addresses would prevent addressing conflicts in case that there are existing NAT networks within the victim network, an issue that many NAT networks suffer from [32].

4.2. Solution scalability

Because the proposed NAT solution is meant to solve the Internet access denial problem, the victim network can range from a small LAN to an entire country or region. Therefore, the deployed solution must be scalable to fit the size and requirements of the victim network.

For a small network, a single NAT router with an external IP address is used. The NAT router is used to connect to the Internet, and all the traffic is translated into its public IP address. As the size of the private network increases, scalability issues start to appear.

The first issue is the limited number of possible port mappings. NAT maps each session to a single external port number. The tuple of source IP:Port and destination IP:Port is used to map subsequent traffic to the same external port number. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) use 16-bit port numbers, providing 65 536 ports. Ports from 1 to 1023 are called the "well-known ports" as they are reserved for specific applications by the *Internet Assigned Numbers Authority* (IANA), and they should not be used as source ports. Hence, a NAT router can map up to 64 512 sessions at the same time with a single external IP address. If there are more connections coming from the private network to the router, it may not be able to serve them as there are no more available ports. This issue can be resolved by using a pool of public IP addresses instead of using a single public IP address. Adding public IP addresses increases the available ports exponentially because every added address provides the complete port space to be used for mapping. Figure 3 shows the extended network where the NAT router uses an IP pool, 3.3.4.0/28 for example, which consists of 16 public IP addresses, from 3.3.4.0 to 3.3.4.15.

Other NAT scalability issues include memory, bandwidth, and processing requirements. For each NAT mapping, an entry is added to the NAT table. Because a NAT router can map up to 64 512 sessions at the same time with a single IP address, that many NAT entries are expected to be in the NAT table.

A NAT table entry requires about 160 B [33]. Therefore, a fully utilized NAT table with 64 512 entries would require a little less than 10 MB of memory, which represents a small portion of the available memory in routers nowadays. Hence, the growth of the NAT table is not an issue when a single public IP address is used. However,

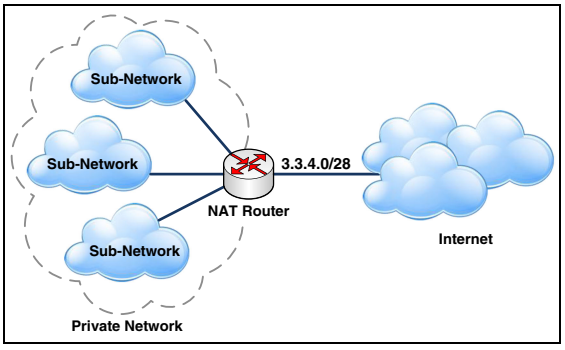

Figure 3. Extended NAT solution design using a pool of public IP addresses.

the use of pools of public IP addresses will significantly increase the required memory. For example, the NAT table resulting from the full mapping of a pool of 16 IP addresses would require 160 MB, which is considerably high. Therefore, router memory may become a limitation on the design. Moreover, the NAT router has a limited processor power such that it may not be able to handle that much traffic. Bandwidth and processor limitations need to be considered as well.

To resolve these issues, load-balancing can be used by adding more NAT routers at the gateway level. Each NAT router handles a portion of the private network and has its own pool of IP addresses, as shown in Figure 4. This method provides large scalability of the solution because more NAT routers can be added as needed.

The partitioning of the internal network can be carried out based on the physical topology. The private network is partitioned into a number of subnetworks, and each subnetwork uses its own NAT router to translate traffic. For example, if NAT solution is to be implemented on a country level, the country's network can be partitioned by ISPs. Each ISP is a subnetwork that is connected to the higher-tier ISP with the use of one or more NAT routers.

5. PERFORMANCE EVALUATION

Enabling NAT in a router introduces a computational overhead that, theoretically, affects performance. NAT performs a number of added operations on packets. For each packet, the NAT router changes the IP address of the source (or destination, for incoming packets) and replaces the source and destination ports. The router also performs NAT table lookup to find a matching entry, and if none is found, it adds a new entry. TCP packets have a packet-checksum in their TCP header, which also needs to be recomputed after the IP address and port translation.

However, many router vendors suggest that the extra delay added by enabling NAT is very small and negligible because routers are designed to minimize the NAT computational overhead [34]. NAT may even have *zero* impact on performance, as some routers, such as Juniper's SSG500 [35], are designed using session-based architecture, where the router keeps track of complete connection sessions and is aware of the packet's transport-layer information. Nevertheless, in the following three subsections, we evaluate the effect of NAT on network performance by first modeling the NAT processing overhead, then describing the simulation setup, and finally presenting the simulation results.

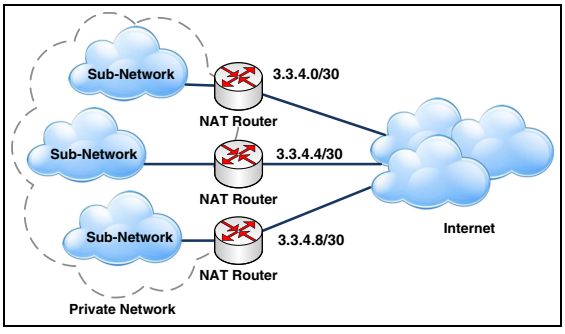

Figure 4. Extended NAT design using load-balancing over a number of NAT routers.

5.1. NAT processing overhead

Most popular network simulators, such as OPNET [36] and ns-2 [37], do not consider NAT processing delay in their simulations [38]. In order to correctly evaluate the performance of NAT, a correct delay model of NAT needs to be implemented in the simulator.

Clark *et al*. [39] studied the overhead of TCP. They measured the computational overhead done at the transport layer, such as TCP checksum computation, and memory read and write accesses. They concluded that the TCP overhead is very small and it is not the source of processing overhead. The overall overhead per packet does not exceed a fraction of a millisecond.

Network processors have significantly been developed over the last 2 decades, and the measured TCP overhead would be even smaller by now. NAT computational overhead is somehow similar to the TCP overhead, as both are in the transport layer, and they have similar computations, such as the checksum calculation. Hence, it is possible to approximate the NAT delay to the measured TCP overhead.

Ramaswamy *et al*. [38] have studied the network processing delay that packets experience. They estimated that on a 1-Gbps network, the processing delay of complex packet modifications, including NAT, firewall, and IPSec encryption, is 1000 μs. They modeled a simplified network processor to measure the end-to-end delay that a single packet experiences. They did not consider the effect on the overall throughput, as routers are designed to improve performance by processing many packets in parallel using multi-core processors, and the processing overhead would have a significant effect only on the end-to-end delay of a single packet.

Although the study performed by Ramaswamy *et al*. [38] shows that processing delay is not very small, we still can consider it negligible for the NAT-based Internet access denial solution. The reason is that the measured delay is much smaller than the Internet delay, which ranges from tens to hundreds of milliseconds. Also, Ramaswamy *et al*. measured delays that included not only NAT, but also more complex operations such as encryption and firewall. Hence, the NAT delay is only a small portion of the measured processing delay. Moreover, routers process traffic with high level of parallelism and pipelining. This hides the processing delay for a flow of packets.

Therefore, the NAT processing delay is expected not to have any significant impact on the performance of the network, as long as the same network resources are available. Nevertheless, in order to evaluate the impact of implementing the proposed NAT solution on the network,

simulations are performed using OPNET Modeler network simulator [36].

The objective of the simulations is to compare the network performance before and after implementing the NAT solution. The used performance metrics are end-to-end delay, traffic throughput, and packet drop rate. Different applications are tested under different traffic loads.

We select to simulate the range of NAT delay values between 10 and 250 μs. In reality, the range for real routers is between 10 and 50 μs. The remaining range, that is, from 50 to 250 μs, does not reflect the real routers' performance. It is simulated only to see the effect of high processing delay on performance.

5.2. Simulation setup

The simulated network is shown in Figure 5. It consists of two networks, local and remote. Each network consists of a local area network (LAN) and a gateway router. NAT is enabled in the local network's gateway router. An IP cloud, representing the Internet, is connecting the two gateway routers.

The local and remote networks are set to 100-Mbps *Fast Ethernet* networks. Each network has 10 connected hosts that will serve as clients and servers for each application. The gateway routers are based on the generic router model in OPNET. It supports many protocols, including BGP and NAT. Both routers are connected to the central Internet cloud using DS-1 links, providing a data rate of 1.544 Mbps.

Two applications are simulated: File Transfer Protocol, which runs over TCP, and video conferencing, which runs over UDP. Each application is simulated under three traffic scenarios: low, medium, and high traffic. The *low-traffic* scenario uses 25% of the available link's bandwidth, which is about 380 kbps. The *medium-traffic* scenario uses 50% of the bandwidth (about 770 kbps). The *high-traffic* scenario utilizes 75% of the bandwidth (about 1200 kbps). These scenarios are selected to evaluate the performance of NAT under different traffic loads. Each simulation is run five times, and the average of the five results is taken. Performance is evaluated for the following metrics: end-to-end delay, traffic throughput, and the packet drop rate.

5.3. Simulation results

Each simulation measures the end-to-end delay. End-to-end delay refers to the amount of time that a packet takes to travel from the client to the server and includes the transmission times, the queuing delays, and the added NAT delay.

The effect of NAT delay on the total end-to-end delay for UDP and TCP traffic can be seen in Figure 6. The figure shows the end-to-end delay for low, medium, and high traffic, with and without NAT, versus the simulated NAT delay. When NAT is not enabled, the NAT delay is not taken into consideration, and the end-to-end delay is constant. However, when NAT is enabled, the delay that packets suffer to reach the destination increases linearly.

The added NAT delay is suffered by every packet that passes through the router. Because OPNET does not reflect the parallelism and pipelining that actual routers have, the simulated router is modeled as a G/D/1 queuing system. Hence, the delay suffered by an arriving packet is $N \times \tau$, where N is the number of packets in the system, and τ is the processing time. An added NAT delay of $\Delta\tau$ will result in increasing the processing time to $N \times (\tau + \Delta\tau) = N\tau + N\Delta\tau$. Hence, the increase of $\Delta\tau$ causes a linear increase of the processing time by $N\Delta\tau$.

The relative increase of the end-to-end delay for UDP and TCP traffic is shown in Figure 7. The relative increase is computed as $(Delay_{NAT} - Delay_{NoNAT}) / Delay_{NoNAT}$. It can be noted that for small NAT delays, specifically below 100 μs, the effect of NAT does not exceed 0.1% of the total end-to-end delay. Larger values of the NAT delay cause a relatively higher increase in the end-to-end delay. However, the maximum delay in the highest NAT delay still does not exceed 0.45% of the total delay. It can also be noted that the relative effect of NAT delay is lower when the traffic is high. This is because higher traffic results in higher queuing delay, which eventually becomes more significant than the NAT delay. Hence, the relative effect of NAT delay is lower.

It can be concluded that NAT does not have any significant impact on the end-to-end delay. The maximum increase of the end-to-end delay does not exceed 0.5% in the worst case when the NAT delay is higher than 200 μs, which is an extremely unrealistic scenario. However, for the reasonable range of NAT delay, which is between 10 and 50 μs, NAT adds very small and negligible effect on the end-to-end delay.

Throughput is another performance measure that is evaluated in order to study the impact of NAT on the amount of transmitted and received traffic. Throughput is measured as the amount of application traffic sent and received by the hosts per second. The simulation is set to measure the throughput at the client side. The same

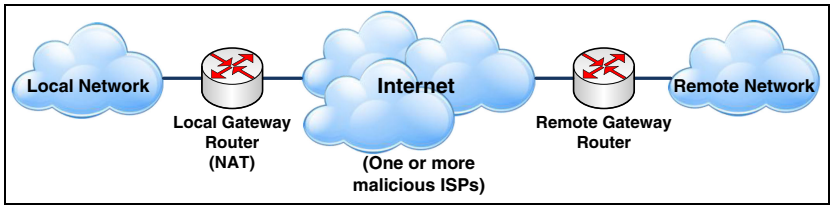

Figure 5. Simulated scenario to measure the effect of NAT delay on network performance.

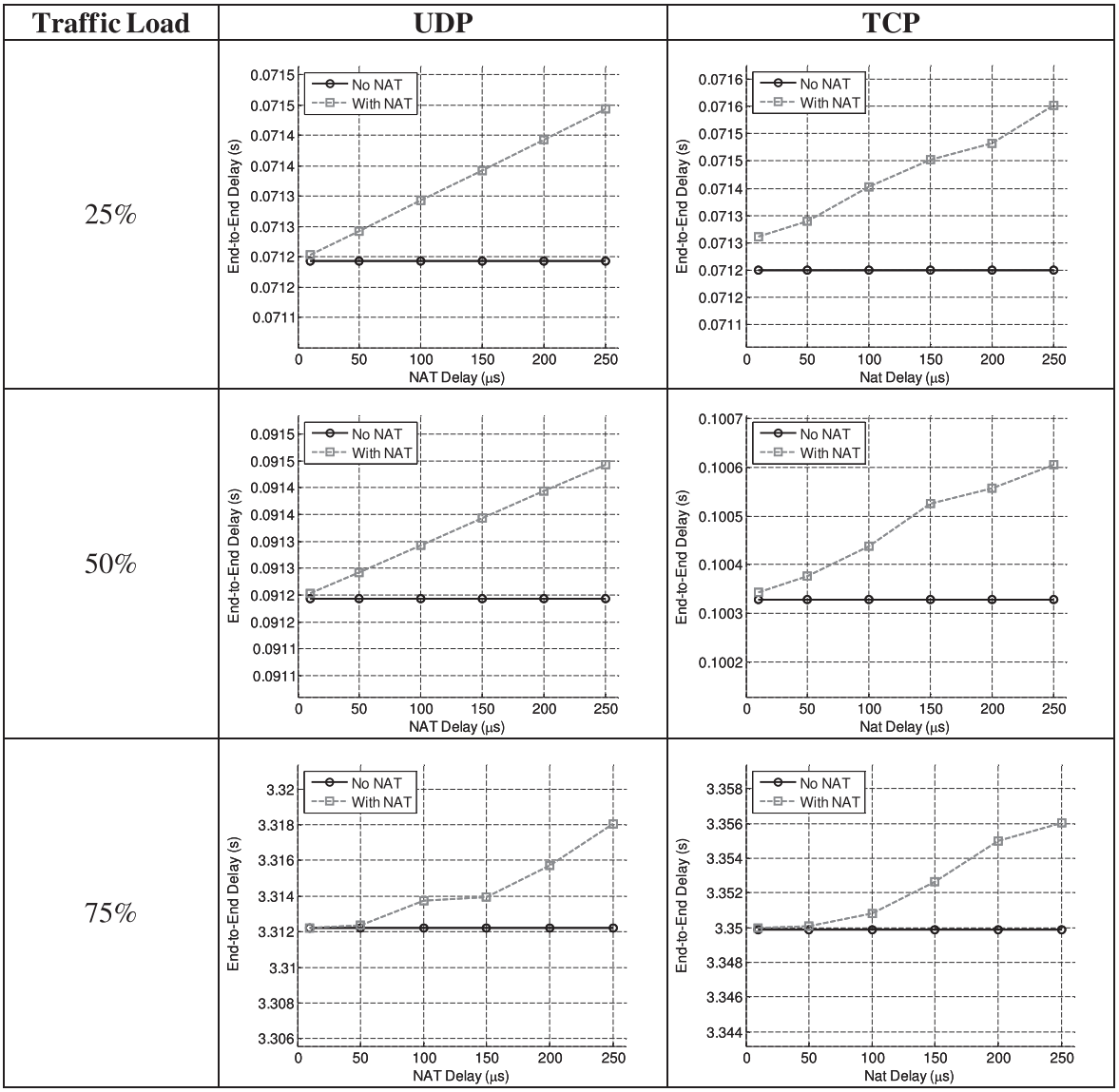

Figure 6. End-to-end delay for UDP and TCP traffic.

simulation setup is used, where the three scenarios of 25%, 50%, and 75% traffic are simulated, and the NAT delay is varied between 10 and 250 μs.

For the cases of low and medium traffic, NAT does not have any effect on the throughput; both scenarios, with and without NAT, have exactly the same measured throughput. In the case of high traffic, NAT only starts to affect the throughput when the NAT delay is very high, that is, more than 150 μs. Figure 8 shows the throughput for UDP and TCP traffic for the high load (75%) scenario. The degradation of throughput is due to the high NAT delay, which slows down the processing of packets and causes the router queue to be filled with waiting packets.

The relative decrease of throughput, which is computed as $(Throughput_{\mathrm{NoNAT}} - Throughput_{\mathrm{NAT}}) / Throughput_{\mathrm{NoNAT}}$, is shown in Figure 9. It can be noted that the degradation of throughput starts earlier in TCP traffic as a NAT delay of 150 μs causes a small decrease in the throughput. The maximum relative decrease is less than 0.3% of the total throughput, which is insignificant. Nevertheless, in the realistic NAT delay range, the throughput is not affected at all. We can conclude that NAT does not affect the network throughput except at the extreme cases of high NAT delay, and even in such cases, the performance degradation is negligibly small.

As for the drop rate, which measures the average number of packets that are discarded per second as a result of network congestion, the simulation results show no increase in the drop rate and, accordingly, were not included in the paper.

It is clear that NAT does not have any significant impact on the performance of the network. The performance degradation that was measured using simulations happens

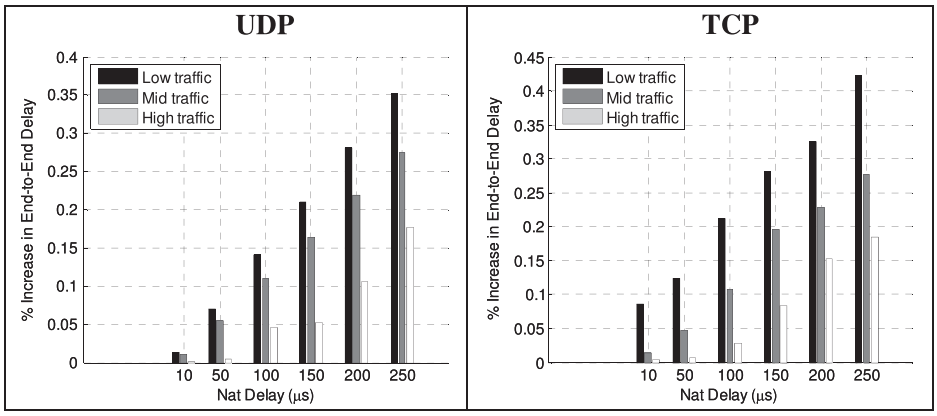

Figure 7. Relative increase of end-to-end delay for UDP and TCP traffic.

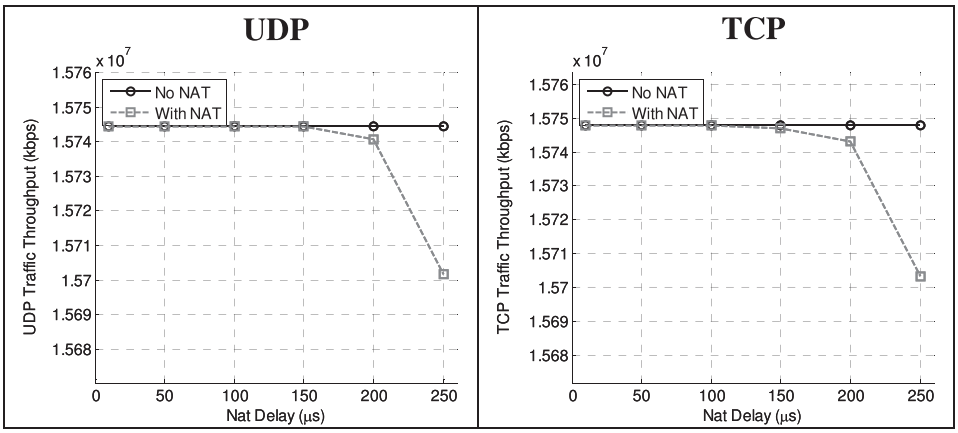

Figure 8. Throughput of high UDP and TCP traffic.

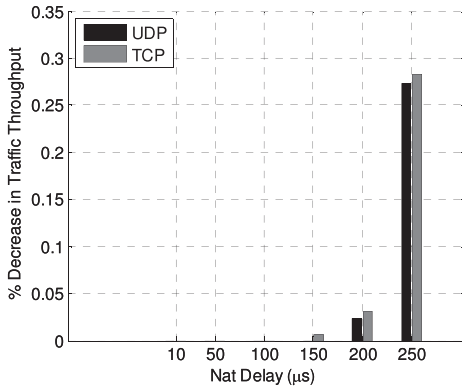

Figure 9. Relative decrease of throughput for TCP and UDP traffic.

only when the NAT delay is set to a very large or extreme value. It is also noticeable that the relative performance degradation decreases as the traffic increases. This is because packets will suffer more from the queuing delay than the processing delay as more packets are transmitted. We can extrapolate this observation to larger networks with higher levels of traffic as the extra delay introduced by the NAT solution is going to be negligibly small compared with other delays, such as queuing and end-to-end delay, that packets may experience.

Therefore, it is concluded that deploying NAT as a solution for Internet access denial will operate transparently without any performance drawbacks. As shown earlier, the solution can easily be scaled for larger networks as well.

6. NAT-BASED SOLUTION LIMITATIONS AND PROPOSED SOLUTIONS

The proposed NAT-based solution does not have any significant impact on the network performance. Nevertheless, there are well-known connectivity limitations that NAT causes. Because the private network behind a NAT router appears as a single entity to other public hosts, incoming

connections will use the public IP address of the NAT router as their destination address. However, the NAT router will not be able to route the incoming connections because there is no information on which a private host should receive this connection. This issue causes two types of limitations: *peer-to-peer* (P2P) application connectivity and *server* reachability behind NAT. The two types of limitations and proposed solutions are summarized in Sections 6.1 and 6.2.

6.1. Peer-to-peer application connectivity

One of the drawbacks of NAT is the limitation of end-to-end connectivity between hosts. This limitation prevents P2P applications from working properly. The reason is that a peer in a P2P network acts as both a client and a server. NAT only allows connections to be initiated from within the private network and destined to a host in the public Internet. Incoming connections are not received by the peer because there is no way of addressing the peer, as the private network behind NAT is seen by outsiders as a single host. This could work for some applications where the connections are initiated by the peers behind NAT. However, if the remote peer is also behind NAT, the problem gets more complicated.

Several research efforts have taken place to resolve the NAT drawback of limiting connectivity. Different *NAT traversal* techniques and protocols were proposed as solutions to this problem. Some of these techniques involve utilizing the NAT router to map ports or tunnel traffic, whereas other techniques are more transparent and exploit the behavior of NAT in order to deliver traffic.

Control-based NAT traversal techniques, such as *Application-Level Gateway* [40], *Internet Gateway Device Protocol* [41], and *Middlebox Communication* [42] provide means for the client to create an external address mapping on the NAT router. This allows the client to receive incoming connections that are destined for that address mapping. Behavior-based techniques, on the other hand, do not use the NAT router as means of receiving connections but rather accomplish connectivity by coordinating with the other peer. Examples of these techniques are *hole-punching* and *relaying*. These are used in many NAT traversal protocols, such as *Traversal Using Relays around NAT* [43] and *Interactive Connectivity Establishment* [44].

Deploying NAT as an Internet access denial solution may require P2P applications to use one or more NAT traversal techniques to be able to function correctly. The modification of these applications may take place on the local clients, the NAT routers, and/or the remote clients on the Internet, depending on the NAT traversal protocol that is used.

6.2. Private server reachability

Servers on the Internet are addressable using a tuple of their IP address and port. Hence, any client can reach a server using this tuple. Normally, servers have public IP addresses, and thus, they are directly reachable. However, introducing NAT changes the IP address of the server to a private IP address, and the server is only seen using the NAT's public IP address. Moreover, running multiple servers for the same service, such as HTTP servers, behind a single NAT router means that these servers are sharing the public IP address used by the NAT router. Therefore, they are all addressable using the same tuple: NAT public IP address and the service port.

Running multiple servers with a single public IP address has been used in many Web-server scalability designs. Web clusters and distributed Web servers are the most common examples of such designs. Some approaches are used to run a single website on multiple servers with a single IP address to achieve load-balancing, whereas other approaches are used to run multiple websites on a single server with one IP address to achieve higher utilization of hardware.

A very common technique to run multiple websites over a single server with a single IP address is *virtual hosting*, which is implemented in most HTTP server applications [45,46]. This technique uses layer-7 information, specifically the *Host* part of the HTTP request headers, to specify which site is the correct destination for that request [47]. However, the virtual hosting technique does not provide means for accessing multiple servers, each with a different private IP address, when the servers are placed behind a NAT router with a single public IP address.

Web clustering is another technique that is used to allow a single website to run on multiple servers (or cluster nodes) with a single public IP address for the purpose of scalability [48,49]. The distribution of requests over multiple servers is carried out transparently from the client by using an intermediate router. There are two types of routing mechanism for Web clusters: layer-4 routing and layer-7 routing.

In layer-4 routing, the router is content-blind; that is, it is not aware of the application-layer information such as the requested page. Therefore, every Web cluster node has the complete content of the website. The router selects a node to serve each incoming connection and binds the selected node with the client address so that subsequent information are sent to and received from the same node.

On the other hand, layer-7 routing is content-aware. Hence, it is possible to distribute the content over different server nodes, where each node can serve a specific type of content. Requests in layer-7 routing are first accepted by the router, which reads the application-layer information. Such a router is also called a *Web switch*. The Web switch accepts the TCP connection, receives the HTTP request, and then decides which server node should handle this request based on some dispatching policy. The request then is handed over to the selected node.

6.2.1. Proposed solution for multiple Web servers behind NAT

As stated earlier, when there are many Web servers behind the NAT router, then all of them share the same

address, that is, the NAT public IP address, and they also share the same TCP port, the standard HTTP port, 80. Thus, when the NAT router receives a request for a Web server behind it, the NAT router is unable to identify the correct destination for that request as there is no network or transport-layer information that tells the NAT router which server is the destination.

The problem, therefore, is that neither network-layer information nor transport-layer information help in identifying the correct destination of the HTTP request. However, application-layer information, namely the HTTP host header, can be used to map a request to the proper destination website. Hence, the solution that we propose is based on using a similar approach to the Web clustering techniques that use layer-7 routing, but on a server-level mapping, rather than a website-level mapping.

Figure 10 shows the setup for the proposed solution. It includes the NAT router connected to the Internet with a public IP address, a number of Web servers in the NAT's private network, and a client that is connected to the Internet and is attempting to access one of the Web servers behind the NAT router. There are also both public and internal DNS servers. A Web switch is used to accept the HTTP request. It can either replace the NAT router, acting as both a NAT router and a Web switch, or the NAT router can just forward all traffic destined to port 80 to the Web switch that is placed in the private network. We assume that the DNS records of the Web servers are stored in the public DNS server and that they all point to the NAT's public IP address. The clients will use the public DNS servers to resolve the domain names of the HTTP servers to their IP addresses (i.e., NAT's public IP address).

After resolving the server's name and getting the IP address, the client initiates a TCP connection to the HTTP port, 80, of that IP address. The Web switch accepts the connection and receives the HTTP request. Then, the Web switch reads the Host header from the received request and uses the internal DNS servers to resolve the host name into an IP address. This IP address, corresponding to the private address of the correct destination server of the request, is directly accessible by the Web switch because the Web switch is part of the private network.

After the Web switch has identified the correct Web server for that HTTP request, it forwards the HTTP request to that server. Forwarding is carried out using transport-layer forwarding of traffic, which utilizes a table for mapping that is similar to the NAT table. We call it a *Web-switch table*.

Once the Web switch receives the HTTP request and identifies the intended Web server, an entry is added to the Web-switch table that maps the client's address tuple (client's IP address and source port) to the server's address tuple (server's private IP address and destination port, 80 in this case). This table entry is used to forward subsequent traffic between the client and the Web server until the end of the HTTP session. Similar to the usual NAT tables, the entries are deleted after some timeout period, or at the termination of the TCP connection using a packet with the FIN flag. Packets are address-translated into local IP addresses, similar to the way NAT translates them, then the translated packets are sent to the proper server. It should be emphasized that only the first packet of each HTTP session is examined at the application layer. All subsequent packets are address-translated and forwarded at the transport layer, in a very similar way to NAT forwarding.

The advantage of using this solution is that no modifications are made to the servers within the private network placed behind the NAT router. Another advantage of the proposed technique is that it is transparent to the clients.

6.2.2. Solution scalability

The proposed solution can be applied to a NAT network of any size. There is no limit on the number of Web servers behind NAT as long as other resources, such as DNS servers and bandwidth, are available. The Web switch is considered the performance bottleneck of the proposed solution. Hence, solution scalability is based on how the Web-switch scales.

One approach is to use load-balancing. Incoming requests are distributed equally over a number of Web switches that are interconnected with the gateway NAT router. This design is shown in Figure 11. When a request is received by the NAT router, it first checks its NAT table to see if this request has already been mapped to one of the

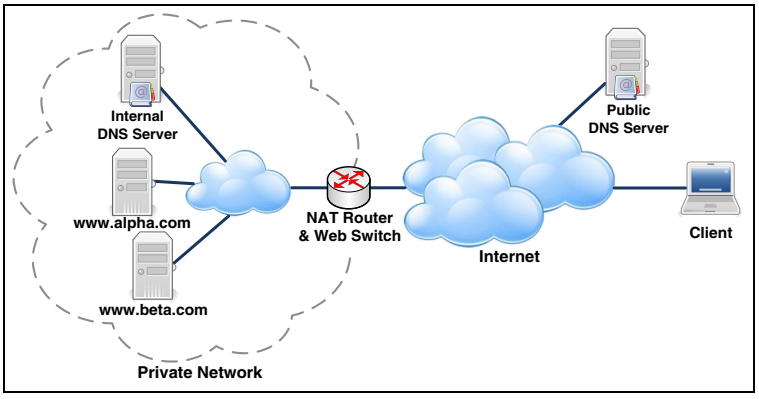

Figure 10. Setup for the solution of multiple Web servers behind NAT using a Web switch.

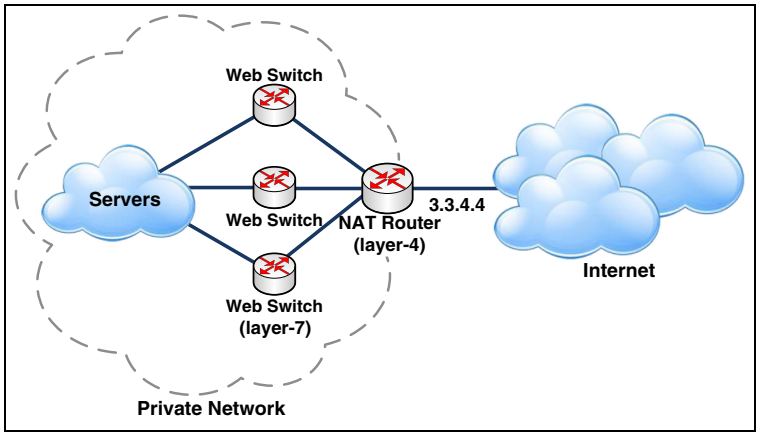

Figure 11. Load-balancing incoming requests over a number of Web switches.

Web switches. If no mapping is found, the NAT router selects one of the Web switches to handle this request, adds an entry to its NAT table to map the connection with that Web switch, and then forwards the packets to the selected Web switch. Note that the processing carried out at this level is only layer-4 processing; no application-layer data are processed yet. The selected Web switch accepts the client's request and finds the correct server using layer-7 information. It then forwards the request to the intended server after adding an entry to its Web-switch table.

The objective of the NAT table at the NAT router is to map Web switches to incoming packets. Subsequent packets from the same client destined to the same server should all be processed by the same Web switch. Hence, the NAT table is used to keep track of this mapping.

The Web-switch table on a Web switch is used to map packets of the same session together. Once the server is located using layer-7 information, all subsequent packets are forwarded directly to the Web server, and all responses are forwarded directly to the client.

The other scalability approach of load-balancing the incoming traffic is to utilize round-robin DNS [50]. Entries in the DNS can have more than one IP address. Thus, multiple public IP addresses can be associated with the public DNS entries corresponding to the private network servers. Hence, after accessing the public DNS to resolve the server name, the clients will use different IP addresses to connect to the same server. By placing a number of Web switches at the gateway level, each with a different public IP address, incoming traffic will be balanced over the different Web switches. Figure 12 shows the topology used to implement this approach.

It is possible to combine both approaches to maximize scalability. A number of NAT routers can be used at the gateway level, each with a different public IP address. These IP addresses are all used in the public DNS for load-balancing. Each NAT router is connected to a number of Web switches that will process layer-7 information and forward the requests to the intended Web server. This way, load-balancing is performed at both gateway level and Web-switch level.

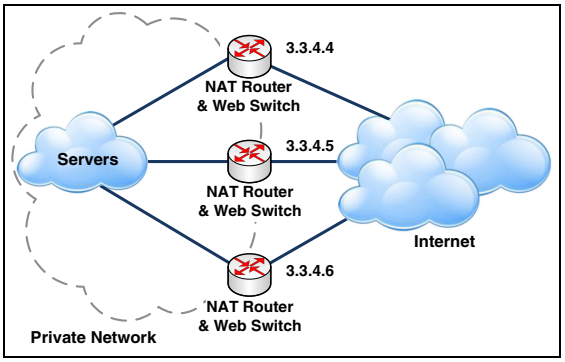

Figure 12. DNS is used to distribute the load over Web switches.

6.2.3. Performance evaluation

The proposed solution adds extra overhead to the network and, therefore, has some impact on the performance. Simulations using OPNET network simulator [36] are used to evaluate the performance of the proposed solution for the HTTP servers behind NAT. The objective of simulating the HTTP solution is to measure the impact of implementing the Web server solution on the network performance. Two metrics are used for measurements: end-to-end delay between the clients and servers and the throughput of the sent and received traffic.

6.2.3.1. Modeling of Web-switch delay. The proposed solution requires layer-7 processing of only the first packet, and subsequent packets are processed at layer-4. Hence, it can be assumed that the performance evaluation of NAT is a good approximation of the performance of a Web switch, except for the layer-7 processing of only the first packet.

As stated in Section 5, the NAT delay is considered insignificant. Therefore, the performance impact of the proposed solution may be affected by layer-7 processing of the first packet.

In order to measure the effect of layer-7 processing on the performance, the Web-switch delay is implemented in

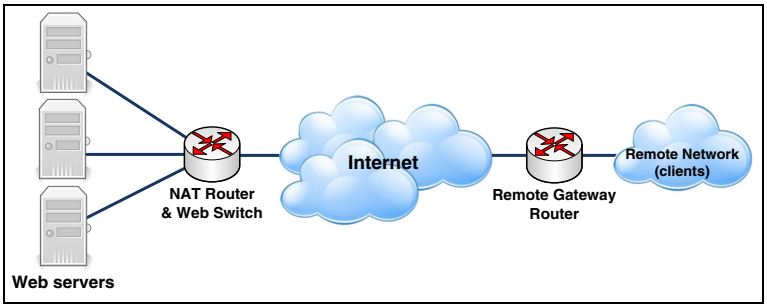

Figure 13. Simulated scenario for the HTTP server reachability solution.

OPNET network simulator such that it adds an extra processing delay for the first packet of a request. Subsequent packets only suffer from the layer-4 NAT delay, which is smaller than the Web-switch delay. The implementation is carried out such that when a NAT table lookup is added, the packet processing delay is increased by the Web-switch delay. This is because NAT table entries are only added for the first packet of every session, which is the same packet that will have layer-7 processing. The processing time of all subsequent packets and responses only suffers from a small extra NAT delay.

6.2.3.2. Simulated scenario. The simulated scenario is shown in Figure 13. It consists of two networks, private and remote. Each network is connected to the Internet through a gateway router. The private network consists of three Web servers, and the private network gateway is a Web switch that has the implementation of NAT delay and Web-switch delay. The remote network consists of a 100-Mbps Fast Ethernet LAN, with 10 hosts that act as Web clients. The intermediate links, connecting gateway routers to the Internet, are DS-1 links with 1.544-Mbps data rate.

The measurements are selected to compare the performance of using a normal router, where servers have public IP addresses, with the use of a Web switch, where packets suffer an added network address translation and Web-switch delays. Because all packets that pass through the Web switch are translated, NAT delay is added to the processing time. Based on the discussion in Section 5, the selected simulation NAT delay is 50 μs.

It was shown earlier that the degradation of performance caused by NAT is very similar for low and medium traffic. Therefore, we will only simulate two traffic load: low, where 25% of the DS-1 bandwidth is utilized (about 380 kbps), and high, where 75% of the DS-1 bandwidth is used (about 1200 kbps).

Because the Web-switch delay is caused by the processing of layer-7 information, this delay is expected to be higher than the one caused by NAT for layer-4 processing. In the simulations, we simulate different scenarios with varying values of Web-switch delay. Based on the fact that more complex operations such as firewall and IPSec encryption have a processing delay that is less than 700 μs [38], a range between 100 and 400 μs is used as a Web-switch delay. This delay is only added to the first incoming packet of the session.

6.2.3.3. Results and analysis. The impact of the Web switch on end-to-end delay is simulated. Figure 14 (a) shows the simulated end-to-end delay for the low-traffic load scenario. Note that the solid line in Figure 14(a) refers to the end-to-end delay when both the NAT router and the Web switch are not installed, whereas the dashed line refers to the end-to-end delay when both the NAT router and the Web switch are installed. It is noticed that a small

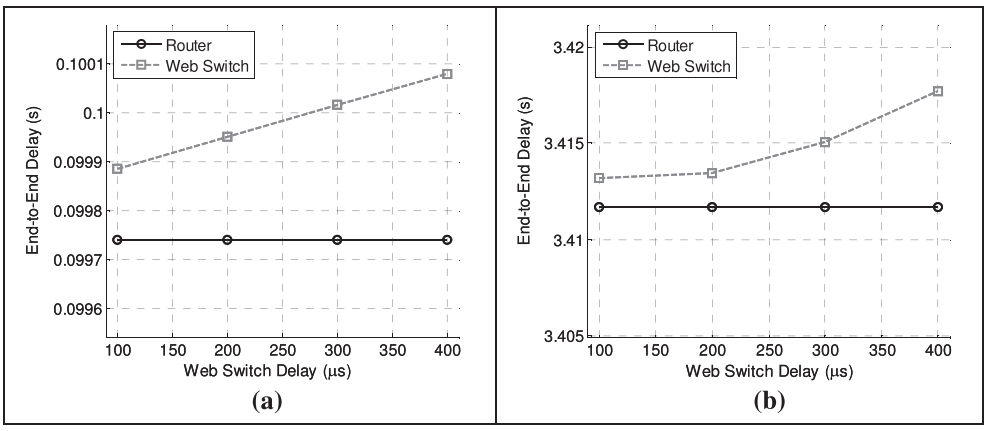

Figure 14. End-to-end delay for (a) low Web traffic, and (b) high Web traffic.

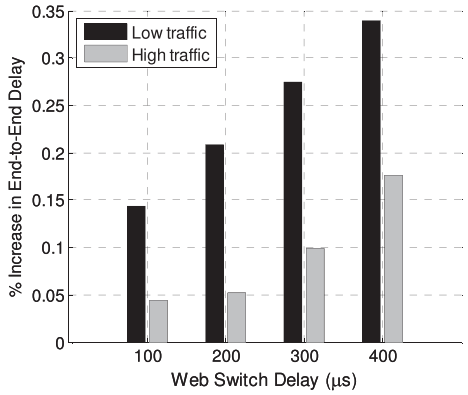

Figure 15. Relative Increase in end-to-end delay for Web traffic.

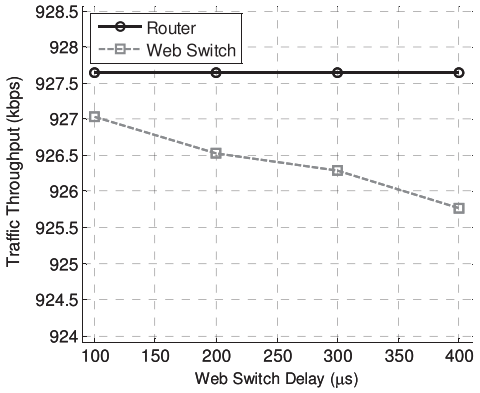

Figure 16. Throughput for high Web traffic.

increase is caused by the Web switch. The effect increases as the Web-switch delay is increased. About 150 μs of additional end-to-end delay (i.e., a total of 99.8 ms of end-to-end delay) is measured when the Web-switch delay is 100 μs, but the additional end-to-end delay increases to 350 μs (i.e., a total of 100 ms of end-to-end delay) for a Web-switch delay of 400 μs. There are two factors causing this increase of the end-to-end delay. First, all packets require extra processing time because of the added NAT delay. Second, the first packet of every HTTP session suffers an extra Web-switch processing delay.

Figure 14(b) shows the measured end-to-end delay from the client to the server for the high-traffic load scenario. Similar to the previous scenario, the end-to-end delay increases when a Web switch is used because of the added Web switch and NAT delays, which cause all packets to require extra processing time.

The relative increase in the end-to-end delay, computed as $(Delay_{\text{WebSwitch}} - Delay_{\text{Router}}) / Delay_{\text{Router}}$ is shown in Figure 15. Although the amount of increase in the end-to-end delay in high-traffic load scenario is higher than the amount of increase in the low-traffic load scenario, the relative increase in the end-to-end delay in high-traffic load scenario is smaller than the low-traffic load scenario. The reason is that the processing delay for high traffic becomes less significant than the queuing delay. Hence, the relative increase in the end-to-end delay, caused by NAT and Web-switch delay, is smaller.

We also notice that the maximum increase in the end-to-end delay in the worst case does not exceed 0.35% of the total end-to-end delay. This increase does not have significant effect on the performance. Hence, we can conclude that the proposed HTTP server solution has a small, insignificant impact on the end-to-end delay.

The other measure of performance considered is throughput, which is the amount of traffic received on the server side. The simulations show no impact on the traffic throughput in the low-traffic load scenario. However, the impact starts to appear in the high-traffic load scenario, as shown in Figure 16. We notice that the throughput starts to decrease when the simulated Web-switch delay increases. This is because the added processing delay causes more packets to be queued in the Web-switch's queue. Hence, the amount of transmitted traffic is lower.

The relative amount of throughput decrease caused by the introduction of a Web switch is shown in Figure 17. The decrease is computed as $(Throughput_{\text{Router}} - Throughput_{\text{WebSwitch}}) / Throughput_{\text{Router}}$. We notice that the impact on the throughput is relatively low; a maximum decrease of 0.2% of the total throughput is experienced when the Web-switch delay is as high as 400 μs. This decrease is very low and, hence, can be considered negligible.

The simulation results show that the proposed solution for HTTP servers behind NAT does not cause any significant impact on the end-to-end delay nor on the throughput. The simulations were performed with the worst-case scenario parameters, that is, high NAT delay and high Web-switch delay. Therefore, the realistic implementation of this solution on hardware would cause even less impact on the performance. Hence, we can conclude that the proposed solution has negligibly small impact on the performance of the network.

We note further that the proposed *Web-switch* technique can be modified to work with other protocols that have host information in layer-7, such as Simple Mail Transfer Protocol. However, other types of protocols will still suffer the limitation of NAT and will not be directly reachable from the public Internet.

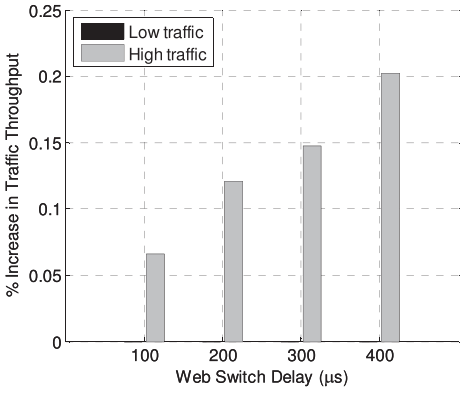

Figure 17. Relative decrease of throughput for Web traffic.

7. CONCLUSION AND FUTURE WORK

This paper introduces the Internet access denial problem by malicious ISPs and proposes a NAT-based solution that provides a method of bypassing the Internet denial by hiding the victim network behind a non-blocked IP address. The solution is shown to be scalable and has minimal performance impact on end-to-end delay, throughput, and drop rate. Although NAT introduces some connectivity limitations, they can be overcome by using application-layer routing for server reachability behind NAT, and NAT traversal techniques for P2P applications.

Future work would include the process of detecting the existence of a malicious ISP and the use of different techniques, other than NAT, to bypass the Internet access denial problem.

ACKNOWLEDGEMENTS

The authors acknowledge the support provided by King Fahd University of Petroleum and Minerals (KFUPM). This project is funded by King Abdulaziz City for Science and Technology (KACST) under the National Science, Technology, and Innovation Plan (project number 08-INF97-4).

REFERENCES

1. Butler K, Farley T, McDaniel P, Rexford J. A survey of BGP security issues and solutions. *IEEE/ACM Transactions on Networking*January 2010 ; **98**:100–122.
2. Mao Z, Rexford J, Wang J, Katz R. Towards an accurate AS-level traceroute tool. *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, Germany, pp. 365–378, August 2003.
3. Drummond D. A new approach to china. *The Official Google Blog*, http://googleblog.blogspot.com/2010/01/new-approach-to-china.html, January 2010.
4. Finkle J, Bartz D. Twitter hacked, attacker claims Iran link. *Reuters*, http://www.reuters.com/article/idUSTRE5BH2A620091218, December, 2009.
5. WikiLeaks. *Wikipedia*, http://en.wikipedia.org/wiki/WikiLeaks#cite_note-197, 2011.
6. Chinese ISP hijacks the Internet. *BGP Mon*, http://bgpmon.net/blog/?p=282, April 2010.
7. Labovitz C, Malan R, Jahanian F. Origins of Internet routing instability. *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1999)*, New York, USA, pp. 218–226, March 1999.
8. Lee S, Yu Y, Nelakuditi S, Zhang Z-L, Chuah C-N. Proactive vs. reactive approaches to failure resilient routing. *Proceedings of the IEEE INFOCOM 2004*, Hong Kong, pp. 176–186, March 2004
9. Poolsappasit N, Ray I. Enhancing Internet domain name system availability by building rings of cooperation among cache resolvers. *IEEE SMC Information Assurance and Security Workshop*, West Point, New York, pp. 317–324, June 2007
10. Nucci A, Bhattacharyya S, Taft N, Diot C. IGP link weight assignment for operational tier-1 backbones. *IEEE/ACM Transactions on Networking*August 2007; **15**(4):789–802.
11. Ford R, Bush M, Boulatov A. Internet instability and disturbance: goal or menace? *Proceedings of the 2005 Workshop on New Security Paradigms*, Lake Arrowhead, USA, pp. 3–8, September 2005.
12. Zheng J, Hu M, Zhao L. Enhancing Internet robustness against malicious flows using active queue management. *Proceedings of the Second International Conference on Embedded Software and Systems*, Xi'an, China, pp. 501–506, December 2005.
13. Guo F, Chen J, Chiueh T-C. Spoof detection for preventing DoS attacks against DNS servers. *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, Washington, DC, USA, p. 37, IEEE Computer Society, 2006.
14. Lad M, Oliveira R, Zhang B, Zhang L. Understanding resiliency of Internet topology against prefix hijack attacks. *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Edinburgh, UK, pp. 368–377, June 2007.
15. Haungs M, Pandey R, Barr E. Handling catastrophic failures in scalable Internet applications. *Proceedings of 2004 International Symposium on Applications and the Internet*, Tokyo, Japan, pp. 188–194, January 2004.
16. Dolev D, Jamin S, Mokryn O, Shavitt Y. Internet resiliency to attacks and failures under BGP policy routing. *Computer Networks* 2006; **50**(16):3183–3196.
17. Postel J. "Internet protocol," RFC 791, Internet Engineering Task Force, September 1981.
18. Quoitin B, Pelsser C, Swinnen L, Bonaventure O, Uhlig S. Interdomain traffic engineering with BGP. *IEEE Communications Magazine* 2003; **41**(5):122–128.
19. Quoitin B, Bonaventure O. A cooperative approach to interdomain traffic engineering. *Proceedings of Next Generation Internet Networks*, Rome, Italy, pp. 450–457, 18–20 April 2005.
20. Mahmoud A, Alrefai A, Abu-Amara M, Sqalli M, Azzedin F. Qualitative analysis of methods for circumventing malicious ISP blocking. *Arabian Journal for Science and Engineering* 2012; in press.
21. Perkins C. "IP encapsulation within IP," RFC 2003, Internet Engineering Task Force, October 1996.

22. Atkinson R. "Security architecture for the Internet protocol," RFC 1825, Internet Engineering Task Force, August 1995.
23. Farinacci D, Li T, Hanks S, Meyer D, Traina P. "Generic routing encapsulation (GRE)," RFC 2784, Internet Engineering Task Force, March 2000.
24. Syverson PF, Reed MG, Goldschlag DM. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications* 1998; **16**:482–494.
25. Zhuang L, Zhou F, Zhao BY, Rowstron A. Cashmere: resilient anonymous routing. *Proceedings of the 2nd Conference on Symposium on Networked Systems Design and Implementation—Volume 2*, Boston, USA, pp. 301–314, May 2005.
26. Reiter MK, Rubin AD. Anonymous Web transactions with crowds. *Communications of the ACM*February 1999 ; **42**(2):32–38.
27. Shields C, Levine BN. A protocol for anonymous communication over the Internet. *Proceedings of the 7th ACM Conference on Computer and Communications Security*, Athens, Greece, pp. 33–42, November 2000.
28. Abu-Amara M, Asif M, Sqalli M, Mahmoud A, Azzedin F. Resilient Internet access using tunnel-based solution for malicious ISP blocking. *Proceedings of the 3rd IEEE International Conference on Communication Software and Networks*, Xi'an, China, May 27–29, 2011.
29. Liu J, Kong J, Hong X, Gerla M. Performance evaluation of anonymous routing protocols in MANETs. *Proceedings of the 2006 IEEE Wireless Communications and Networking Conference*, Las Vegas, USA, pp. 646–651, April 2006.
30. Rekhter Y, Moskowitz B, Karrenberg D, Groot G, Lear E. "Address allocation for private Internets," RFC 1918, Internet Engineering Task Force, February 1996.
31. Egevang K, Francis P. "The IP network address translator (NAT)," RFC 1631, Internet Engineering Task Force, May 1994.
32. Srisuresh P, Ford B. "Unintended consequences of NAT deployments with overlapping address space," RFC 5684, Internet Engineering Task Force, February 2010.
33. Doyle J, Carroll J. *Routing TCP/IP, Volume II*. Cisco Press: Indianapolis, 2005.
34. CISCO IOS Network Address Translation (NAT) Q&A, http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6640/prod_qas0900aecd801ba55a.html.
35. System architecture overview for the Juniper networks SSG500 line. Juniper Networks, http://www.juniper.net/us/en/local/pdf/whitepapers/2000177-en.pdf, February 2009.
36. OPNET Modeler. http://www.opnet.com/.
37. The Network Simulator—ns-2. http://www.isi.edu/nsnam/ns/.
38. Ramaswamy R, Weng N, Wolf T. Characterizing network processing delay. *Proceedings of IEEE GLOBECOM*Texas, USA, November 2004 ; **3**:1629–1634.
39. Clark D, Jacobson V, Romkey J, Salwen H. An analysis of TCP processing overhead. *IEEE Communications Magazine* 1989; **27**(6):23–29.
40. Srisuresh P, Holdrege M. "IP network address translator (NAT) terminology and considerations," RFC 2663, Internet Engineering Task Force, August 1999.
41. U. Forum. Internet gateway device (IGD) standardized device control protocol. http://www.upnp.org/standardizeddcps/igd.asp, November 2001.
42. Srisuresh P, Kuthan J, Rosenberg J, Molitor A, Rayhan A. "Middlebox communication architecture and framework," RFC 3303, Internet Engineering Task Force, August 2002.
43. Mahy R, Matthews P, Rosenberg J. "Traversal using relays around NAT (TURN): Relay extensions to session traversal utilities for NAT (STUN)," RFC 5766, Internet Engineering Task Force, April 2010.
44. Rosenberg J. "Interactive connectivity establishment (ICE): a protocol for network address translator (NAT) traversal for offer/answer protocols," RFC 5245, Internet Engineering Task Force, April 2010.
45. Apache virtual host documentation. http://httpd.apache.org/docs/2.2/vhosts/.
46. Use host header names to configure multiple web sites in IIS 6.0, http://go.microsoft.com/fwlink/?LinkId=36045, December 2007.
47. Fielding R, Gettys J, Mogul J, et al. "Hypertext Transfer Protocol—HTTP/1.1," RFC 2616, Internet Engineering Task Force, June 1999.
48. Cardellini V, Yu PS, Cardellini V, *et al*. The state of the art in locally distributed web-server systems. *ACM Computing Surveys* 2001; **34**(34):263–311.
49. Schroeder T, Goddard S, Ramamurthy B. Scalable web server clustering technologies. *IEEE Network* 2000; **14**(3):38–45.
50. Mourad A, Liu H. Scalable web server architectures. *Proceedings of the IEEE Symposium on Computers and Communications*, p. 12, 1997.