Journal Article

Principles of Eliminating Access Control Lists within a Domain

Davies, J.N., Comerford, P. and Grout, V.

Future Internet **2012**, *4*, 413-429; doi:10.3390/fi4020413

OPEN ACCESS

future internet

ISSN 1999-5903

www.mdpi.com/journal/futureinternet

Article

Principles of Eliminating Access Control Lists within a Domain

John N. Davies *, Paul Comerford and Vic Grout

Centre for Applied Internet Research (CAIR), Glyndŵr University, Wrexham LL11 2AW, UK;
E-Mails: p.comerford@glyndwr.ac.uk (P.C.); v.grout@glyndwr.ac.uk (V.G.)

* Author to whom correspondence should be addressed; E-Mail: j.n.davies@glyndwr.ac.uk;
Tel.: +44-1978-290666; Fax: +44-1978-290008.

Received: 27 December 2011; in revised form: 20 March 2012 / Accepted: 12 April 2012 / Published: 19 April 2012

Abstract: The infrastructure of large networks is broken down into areas that have a common security policy called a domain. Security within a domain is commonly implemented at all nodes. However this can have a negative effect on performance since it introduces a delay associated with packet filtering. When Access Control Lists (ACLs) are used within a router for this purpose then a significant overhead is introduced associated with this process. It is likely that identical checks are made at multiple points within a domain prior to a packet reaching its destination. Therefore by eliminating ACLs within a domain by modifying the ingress/egress points with equivalent functionality an improvement in the overall performance can be obtained. This paper considers the effect of the delays when using router operating systems offering different levels of functionality. It considers factors which contribute to the delay particularly due to ACLs and by using theoretical principles modified by practical calculation a model is created. Additionally this paper provides an example of an optimized solution which reduces the delay through network routers by distributing the security rules to the ingress/egress points of the domain without affecting the security policy.

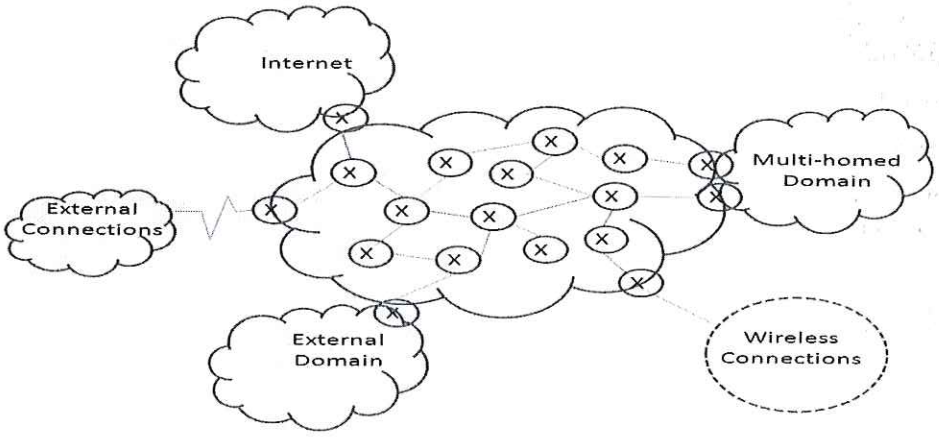
Keywords: routing domain, performance; delay through routers; access control list; ACL optimization; off-line verification of ACLs; firewalls; inter-firewall optimization; IP packet filtering

1. Introduction

Modern computer networks are expected to provide reliable high performance and end to end connectivity at any point in the world. They must also provide the ability to filter packets so that access to services is limited to trusted traffic defined in the security policy for the network. This must be achieved with a minimal delay without compromising the security policy. It can be a challenge for a network manager to meet these two conflicting requirements.

Most networks contain one or multiple connections to external networks e.g., the Internet which is considered a great security risk. To mitigate this, trusted networks are created which perform stringent security checks on packets travelling across the network boundary in either direction. Such networks operate under a common security policy managed by a single authority and are known as domains. If network traffic is filtered at all ingress and egress points in the network then it should only contain traffic which is defined as trusted under the security policy (Figure 1).

Figure 1. Typical Domain allocation.



Infrastructure security within a domain is normally implemented in either firewalls or routers containing Access Control Lists (ACLs). ACLs have a common implementation across all platforms e.g., Cisco, Juniper and Linux [1]. Since every packet has to be tested significant delays can result from the introduction of such techniques due to the filtering requirement [2]. Attempts have been made to use various techniques to optimize the delay through routers caused by ACLs [3]. Optimization of packet filtering performance has been the subject of intense research for the past decade [4]. A number of studies have identified rule relations within an ACL which may result in redundant or conflicting rules.

This paper investigates the significance of the delays encountered through the use of various ACL techniques. Factors which contribute to the delay incurred by packets passing through a router are identified and subsequently, a number of experiments were conducted to quantify these. Delays were investigated from a theoretical perspective which formed the basis of an equation which can be used to calculate the delay for a packet passing through a router running a particular router Operating System (OS). The equation was updated to reflect the packet delay experienced in a path across a domain. Recommendations were made to give guidance to network engineers that can be used during the network design phase. An argument for the use of an OS with appropriate functionality for a given task

is put forward based on experimental results. The authors recommend an optimal configuration using a worked example based on previous findings. Finally, a mechanism for the consolidation of distributed ACLs to a single ACL providing equivalent functionality is presented. Only delays through network equipment were considered in this paper. No account of link speeds was taken since these delays are easily quantifiable.

2. Related Work

Rule reordering has been considered to decrease latency associated with packet classification. Studies highlighted through experimental evidence that ordered ACLs could reduce packet processing time [5]. The study did not however, consider the conflicts that may exist between different rules in an ACL. A subsequent paper does consider rule reordering, however only a simplistic treatment is given by organizing similar rules into classes, individual rule reordering and conflicts are not considered [6].

Anomalies in firewall databases using algorithmic techniques have been identified [3] and subsequent work presented a method to introduce early rejection rules for the most commonly matched traffic, providing dynamic updates as traffic flows change [7].

Several schemes have been proposed for storing filtering rules in alternative data structures which facilitate faster lookup times than linear lists. This is achieved by representing the rules as a decision tree [8,9]. Hash tables are also considered for packet classification using a single memory lookup however such schemes exhibit worst-case exponential space complexity which limits their use in devices with limited memory capacity [10].

Hardware solutions to the latency problem have been developed using Ternary Content Addressable Memory (TCAMs). These evaluate all rules in the packet filter in parallel and return the rule first matching in a single memory lookup [11]. Due to their low density they are only able to handle a small number of rules [12]. TCAMs are typically only found in expensive high-end core routers [13].

Most of the work has been carried out on individual routers and there has been comparatively little research undertaken into optimization of packet filters in a single domain. Algorithms have been proposed for identifying anomalies and implementing these in the form of a software tool which allows a network administrator to provide anomaly free policy editing and creation [14–20].

Anomalies present in multiple packet filters traversed by a packet within a domain have been studied and several types of anomalies identified as being similar to those found for single sets of filtering rules [19]. The use of binary decision diagrams (BDD's) to search for anomalies in distributed firewalls using static analysis techniques resulted in a firewall analysis tool being produced utilizing these techniques[20].

A protocol was developed for the detection and elimination of redundancies between two adjacent firewalls located in neighboring domains under different administrative control [21]. The protocol allows the firewalls to cooperate to exchange filtering information without revealing the content of the neighboring firewall, a potential security risk. The protocol was implemented with a variety of real-world and synthetic firewalls which showed that up to 49% of redundant rules can be safely removed.

The significance of the definition of a security policy as the basis for an implementation was shown in [22]. Applications have been created to automate the conversion of a security policy into a set of

rules for use in routers e.g., Guarddog [23] but other than manufacturer's recommendations [24] little work has been carried out on optimization within a domain.

3. Packet Delays within a Router

When considering the packet delay through a domain there are a number of factors that need to be considered. These factors include the route selected by the routing protocol, the bandwidth of the links along the selected route and the internal delays within the equipment [25]. Routing Protocols optimize the route selection using a shortest path algorithm based on cost functions for each path [26]. The delays experienced within equipment e.g., routers and switches are often ignored since the link bandwidth has generally been considered as the dominant factor [27]. However as technology has improved the link speeds have increased and so the equipment delays have become more significant [28].

Analyzing the delay within a domain will therefore depend on the route selected, which can be expressed as, the summation of delays through the components in the route [29]. The link delays are easily calculated since they are proportional to the bandwidth. However the equipment delays are more difficult to quantify. From a theoretical point of view it is possible to identify the causes of delays within a router since it is basically a specialized computer system. A further complexity is introduced due to the real-time operation of the router OS [30]. A practical approach was used to help identify the variation in delay caused by the nature of the processing used in routers. To this end delays have been measured under laboratory conditions.

3.1. Delays within a Router

A simple laboratory network was set up with the use of a dual ported Linux machine running Wireshark as a method of measuring delays across a router. An initial experiment was conducted to identify the accuracy of the measuring system by passing packets into a 100 Mbps hub and measuring the delay experienced on two of the outputs. Since a hub is a very simple device with no buffering, results from the experiment show that the average delay was only 9 μsecs. This would be the error bar for a 100 Mbps network and should be applied to all results but is not significant in the overall values obtained.

3.2. Delay Caused by Packet Routing

Packets which enter a router via its network interface card are filtered by their destination network address using its routing table. The delay of this process is dependent on the software to setup the process and the hardware components e.g., memory access time. Performance of router hardware is highly variable since it is dependent on its underlying technology, including its processing power and memory capacity. Additionally, high throughput hardware can be purchased which exhibits performance improvements due to its specification. Networks typically contain equipment of varying ages which results in performance variations. In this work, to enable other factors to be compared, consistent typical performance hardware has been used.

Router operating systems (OS) are optimized for routing of packets. Routers are also required to perform many other tasks which will be dependent on its feature set. The Basic OS has all the normal

routing functionality but the Advanced OS has an additional feature set that includes more than 950 additional functions which include support for VoIP, SNMP, SIP, AAA Server, PIM MIB Extension for IP Multicast, Netflow, IPSec Network Security, TCP–TCP Congestion Avoidance *etc.* [31]. This functionality can be invoked by applying the appropriate configuration to the router. A comparison of OS size and number of supported/running processes was undertaken using an OS with basic functionality and another with advanced functionality. For all the measurements reported the hardware used was a Cisco 2600 series routers, basic OS was c2600-js2-mz.122-13.T5 and the advanced OS c2600-adventerprisek9-mz.124-19. Applying a configuration that only enabled the interfaces and OSPF routing protocol to the router, the show processes command provided the information shown in Table 1. The third column indicates the total number of processes that the router had run from startup. Many of the processes are only run once or twice and so do not take part in the normal packet handling so the fourth column was introduced to indicate the processes that were being run continually to give a better understanding of the overhead associated with the routing process. Clearly the advanced OS runs many more processes for a given configuration compared to a basic OS which will have an effect on the responsiveness of the CPU and the amount of memory required.

Table 1. Operating System (OS) Comparison.

	OS Size	**Number of Processes**	**Active Processes > 2**
Basic Functionality	12 MBytes	73	32
Advanced Functionality	29 MBytes	184	51

If a core part of the OS is enhanced with additional functionality e.g., HTTP or DHCP Servers it can have an additional adverse effect on the size of the OS and its performance to that seen in Table1.

3.3. Measurement of Delays

Identical tests were undertaken using the ping command, which transmits ICMP packets, to quantify the delay across a router using an OS with basic and then with advanced functionality. The variation in times in the individual results, which can be as great as 100%, is due to the variation of the processes in the Operating System. Results were analyzed using histogram techniques and plotted in Figure 2 where the x-axis shows the measured delay in µsecs and the y axis shows the number of times this value was obtained in a set of 1000 pings *i.e.*, 5000 ICMP request and replies.

Figure 2. Delay through router with different OS.

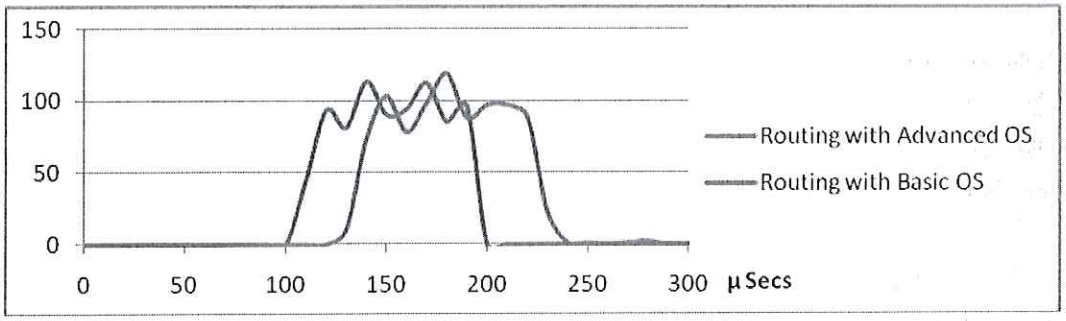


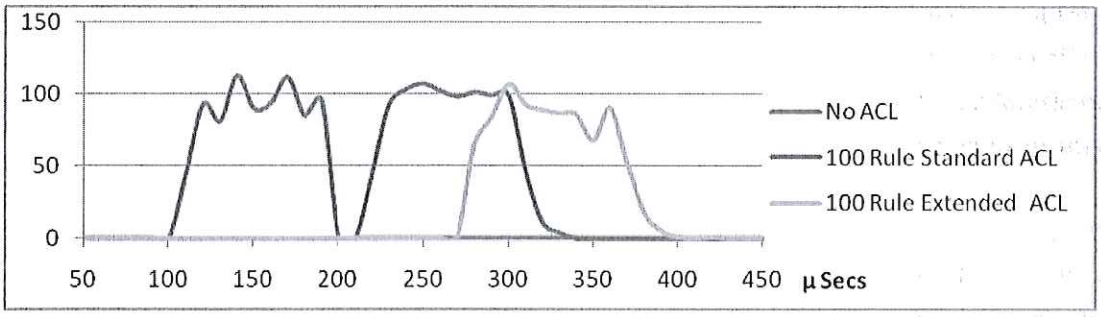
Figure 2 clearly shows the difference in delays attributed to the OS version where it can be seen that there is an increased delay introduced by the Advanced OS of about 25%.

3.4. Delay as a Result of Implementing Security

Security is typically implemented on a router using ACLs. Each rule is evaluated in turn until a matching rule is found. Standard ACLs only filter on the source IP address of a packet whereas extended ACLs provide the capability to filter on additional fields such as destination address, protocol and port numbers. A complete coverage of ACL design and formatting is not covered in this paper since there are many good texts available on this topic [32].

Measurements were taken to investigate the delays when the router, running the basic OS, was configured with 100 rule ACLs. This was seen as a minimum configuration and the results can be seen in Figure 3. As expected no ACL gave the least delay, a standard ACL increased the delay by approximately 110% and the extended ACL an increase of 270% over no ACL.

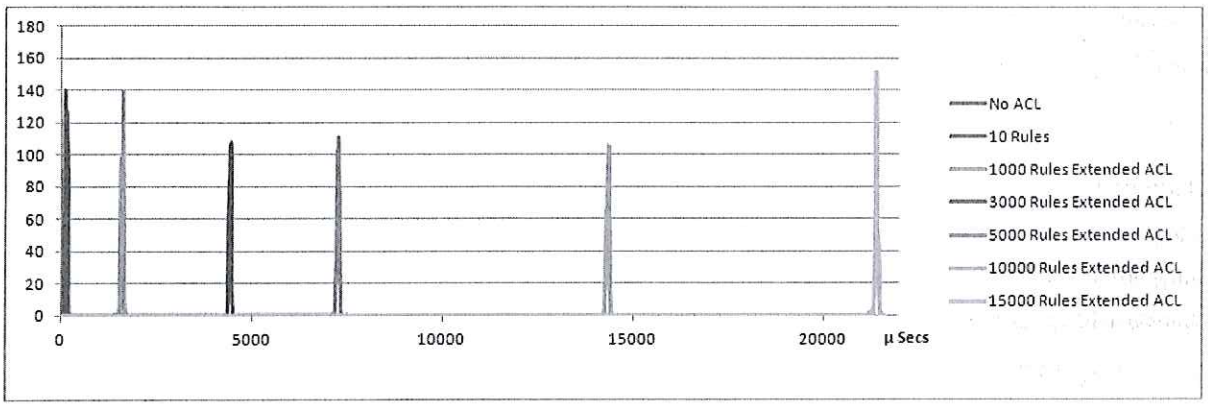
Figure 3. Delay through router with Access Control List (ACL) running Basic OS.



3.5. Effect of Number of Rules in ACL Using a Basic OS

Further work was carried out to investigate the delay experienced by packets matched against an increasing number of rules for both standard and extended ACLs. Figure 4 shows that for a Basic OS increasing the number of rules in the list has a significant effect on the delay. The effect of increasing the number of rules in an Extended ACL from 10 to 15,000 in a Basic OS has the effect of increasing the overall delay by approximately 1400% which is a pretty significant problem.

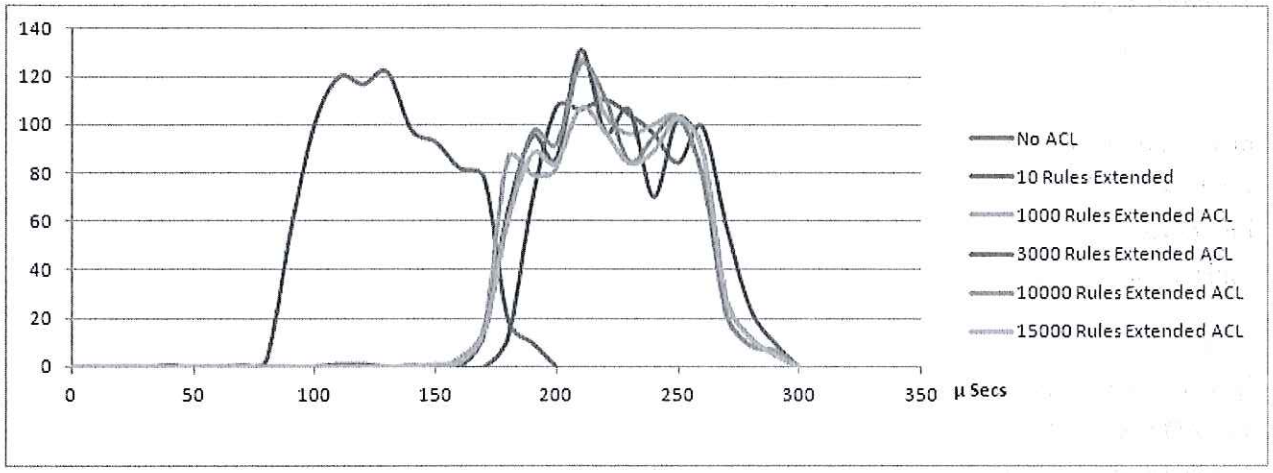
Figure 4. Delay through router with Basic OS.



3.6. Effect of Number of Rules in ACL using an Advanced OS

Repeating the experiment using the same 10 to 15,000 rules did not incur any additional delay using an Advanced OS (Figure 5). The performance improvement in the advanced OS could not be due to TCAMs since identical hardware was used in both tests so it must be due to a software enhancement in the router OS. Cisco do not release details of the operating systems however it is likely that a binary decision technique has been employed because the delay time is not dependent on the number of rules that are in the ACL.

Figure 5. Delay through router with Advanced OS.



4. Analysis of Delays within a Router

After considering the theoretical aspects of delays through routers, a model can be created. Additionally by quantifying the parameters, the model can be simplified and made to be more realistic.

4.1. Theoretical Approach to Delays through a Router

As discussed in section 3 a router is a specialized computer and therefore a basic equation can be defined by including delay parameters for the hardware (Dh) and the operating system (Dos). Applications such as NAT/PAT can be configured and is represented by (Da) and Services such as HTTP Server can also be used (Ds). Different protocols, e.g., IPv4, passed through the router will experience different delays and so a factor Dp has been included. When undertaking the practical work a great deal of time was spent reducing the number of packets that appeared on the network due to routing protocols *etc.* so that the queuing delays were kept to a minimum however a factor Dq is added to account for this. Earlier work has shown that when configuring ACLs delays are introduced based on the type of ACL (Dta) used and the number of rules in an ACL Dnr. The model can be described as shown in the equation 1. Some of the variations in the delay are due to factors of queuing, resulting in.

$$\text{Router Delay (Dr)} = Dh + Dos + Da + Ds + Dp + Dq + Dta + Dnr \quad (1)$$

4.2. Quantifying Parameters

The experiments provide results which were distributed over a large range of values. An average value of the range was calculated in order to provide a single value associated with each test.

Results show that some parameters in the equation have a greater significance than others. The average delay for each Dr is shown in Table 2 which indicates that there are significant differences in packet processing time depending on the OS version used.

Table 2. Average delays for all tests (times in μs).

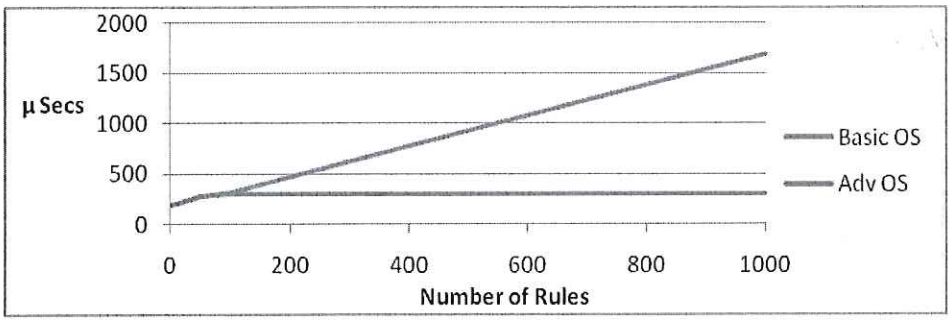
IOS version	No ACL	Standard	Ext 100	Ext 1000
Basic	150	271	320	1500
Advanced	172	239	300	309

By using a router with a basic OS rather than an advanced OS it can be seen that standard routing is faster by around 15%. This is even without configuring any extra services on the advanced OS which it is expected would further increase the latency. When ACLs are configured then for a basic OS the average delay is increased by around 80% for a standard ACL and 110% for an extended ACL. However, by replacing the basic OS with an advanced OS and configuring a standard ACL saving of around 12% can be made and for an extended ACL 6%.

4.3. Effect of Number of Rules in ACL Using Basic OS & Advanced OS

When using a router with a basic OS adding more rules to an ACL has a significant effect on the delays which can be of the order of 1400% for 1000 rules. The advantage of the advanced IOS functionality is that the number of rules used in an ACL does not have an effect on the delay. Figure 6 shows the average delay times when the number of rules in an ACL is varied.

Figure 6. Delay *vs.* number of rules.



The results averaged in Table 2 can enable values for the variables defined in equation 1 above to be estimated e.g., assuming the standard hardware running the basic operating is the bottom line for the measurements then the values in Table 3 can be calculated. For example Dh delay due to the hardware was taken as the shortest time observed with zero delay for the Operating System and extra time was added for the delay of the advanced OS. The spread in the values obtained were attributed to queuing delays which are positive and negative values from the mean.

Table 3. Calculating parameters from tests performed (times in μs).

		No ACL		Standard ACL		Extended ACL	
	Delay due to	Basic	Advanced	Basic	Advanced	Basic	Advanced
Dh	Router Hardware	150	150	150	150	150	150
Dos	Router Operating system	0	22	0	22	0	22
Da	Applications	0	0	0	0	0	0
Ds	Services	0	0	0	0	0	0
Dta	Type of ACL	0	0	121	67	170	128
Dnr	Delay per ACL Rule	0	0	0	0	1.37	0
Dp	Protocol	0	0	0	0	0	0
Dq	Queuing delay variation	+/− 50	+/− 50	+/− 50	+/− 50	+/− 50	+/− 50
Dr	Total delay thro' router	150	172	271	239	321	300

5. Delays within a Domain

Within a domain either static routes are configured or a routing protocol is used to select a route. Theoretically, the cumulative delay (D_d) for a given path can be calculated by the summation of the delays in equation 1 for each router (n) in the route. So the Domain delay (D_d) becomes

$$D_d = \sum_{i=1}^{n} D_{r[i]}$$

Therefore

$$(D_d) = \sum_{i=1}^{n} D_{h[i]} + \sum_{i=1}^{n} D_{os[i]} + \sum_{i=1}^{n} D_{a[i]} + \sum_{i=1}^{n} D_{s[i]} + \sum_{i=1}^{n} D_{p[i]} + \sum_{i=1}^{n} D_{q[i]} + \sum_{i=1}^{n} D_{ta[i]} + \sum_{i=1}^{n} D_{nr[i]}$$

5.1. Calculation for Example Route

The optimized route shown in red in Figure 1 contains 6 nodes and can be used to investigate the implications of the values in Tables 2 and 3.

For a typical security policy, the domain gateway routers (ingress & egress) may have thousands of rules configured. In addition, each internal router may have 100 rules configured, possibly on both ingress and egress interfaces within each router. Using these figures, it is possible to calculate the cumulative delay for a packet which is required to traverse all routers shown in the example.

Figure 7 shows a simplified version of the route and Table 4 shows the expected delay for a packet using both a basic and advanced version of the OS. Analyzing the delay within a domain will depend on the route selected, which can be expressed as the summation of delays through the components in the route. A number of options are calculated, the worse case being all routers configured with ACLs and the basic OS utilized.

Figure 7. Simplified Route through Domain.

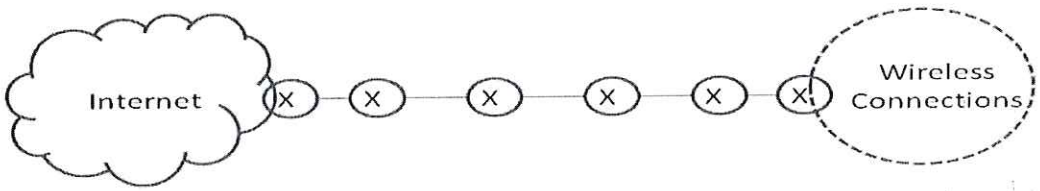
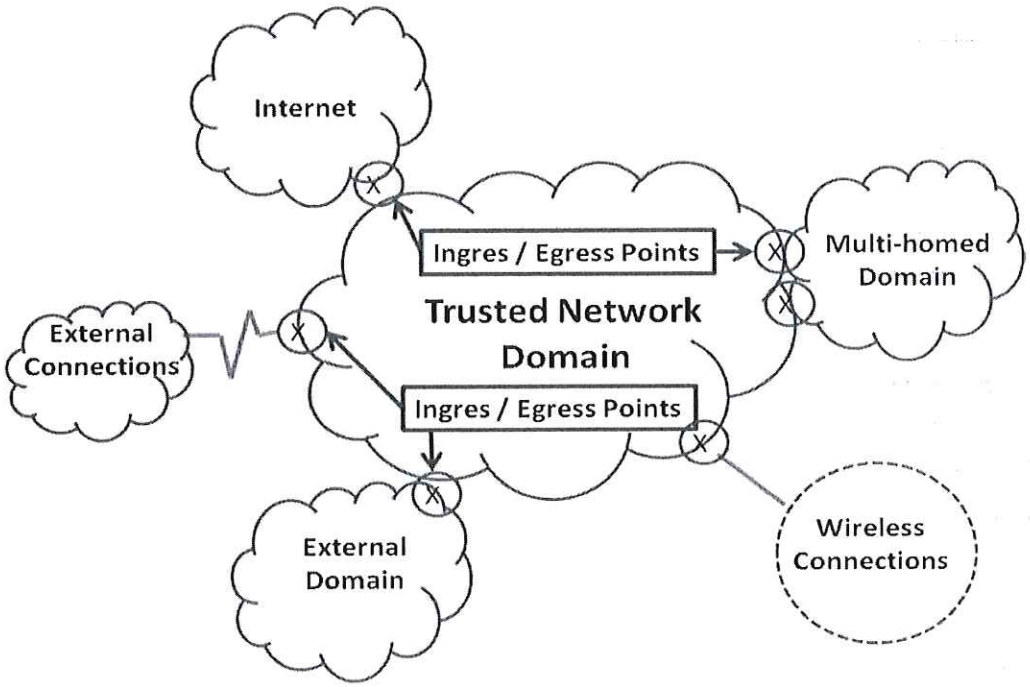


Table 4. Calculated Delays for example network route within Domain.

IOS version	No ACL	Standard	Ext 100	Ext 1000	Total
Basic	0	0	1280	3000	4280 µsecs
Basic optimized	600	0	0	3000	3600 µsecs
Advanced	0	0	800	400	1200 µsecs
Advanced Optimized	688	0	0	400	1088 µsecs
Advanced/Basic Optimized	600	0	0	400	1000 µsecs

By simply using an advanced OS in all the routers, an improvement of 250% can be obtained over a basic OS. Optimizing the route based on the conditions derived from Tables 2 and 3, the best case is using advanced OS everywhere an ACL is applied and a basic ACL where no ACLs are applied. If this principle is applied to a trusted domain as described in the introduction then routers loaded with the advanced OS would be used at the ingress/egress points and routers with basic OS and no ACLs within the rest of domain as seen in Figure 8. The best case shows an improvement of 320% over using a basic OS everywhere in the domain and 30% over using an advanced OS everywhere.

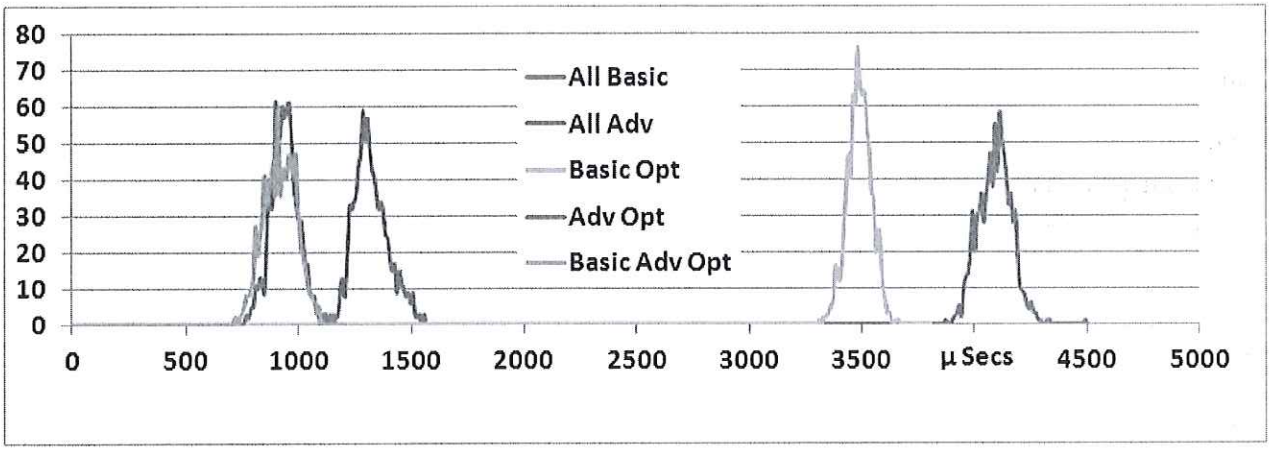
Figure 8. Trusted domain.



5.2. Conformation of Calculation by Measuring Delays

To confirm these calculated results a network was set up containing 6 nodes and measurements were taken as previously described. Tests varying the OS and the ACLs were run and the results shown in Figure 9 were obtained. These are very similar to the calculated values found in Table 4.

Figure 9. Measured values over the Domain.



5.3. Condition Controlling Optimization

If all the traffic within a domain is trusted *i.e.*, from a controlled environment then it should be possible to eliminate the ACLs from routers within the Domain on condition that the security is catered for in the ingress and egress points of the domain.

It is imperative that any optimization process that is undertaken preserves the security policy for the domain. Therefore it is required that all ingress/egress points in the domain are capable of identifying and removing all packets which are not permitted by the security policy.

Similarly, the list of rules on the egress port of all border routers must only pass traffic which is defined as authorized to leave the domain. If the border routers are configured with an advanced OS the number of rules generated due to the elimination process is insignificant since they are capable of processing a large number of rules without a significant impact on packet latency.

5.4. Protocols and the Placing of ACLs

Most routers OSs provide the ability to configure ACLs using different routed protocols. In addition to IP, ACLs can also be configured for IPv6, IPX, AppleTalk *etc.* and each protocol is configurable on each interface. Usually routers only allow a single ACL to be configured on a router interface for a given direction and protocol. IP in its IPv4 guise is the predominant protocol used in network communications and on the Internet, therefore this is the only protocol considered in this study. The ongoing emergence of IPv6 justifies consideration of the optimization of IPv6 ACLs for future work.

5.5. Effect of an ACL

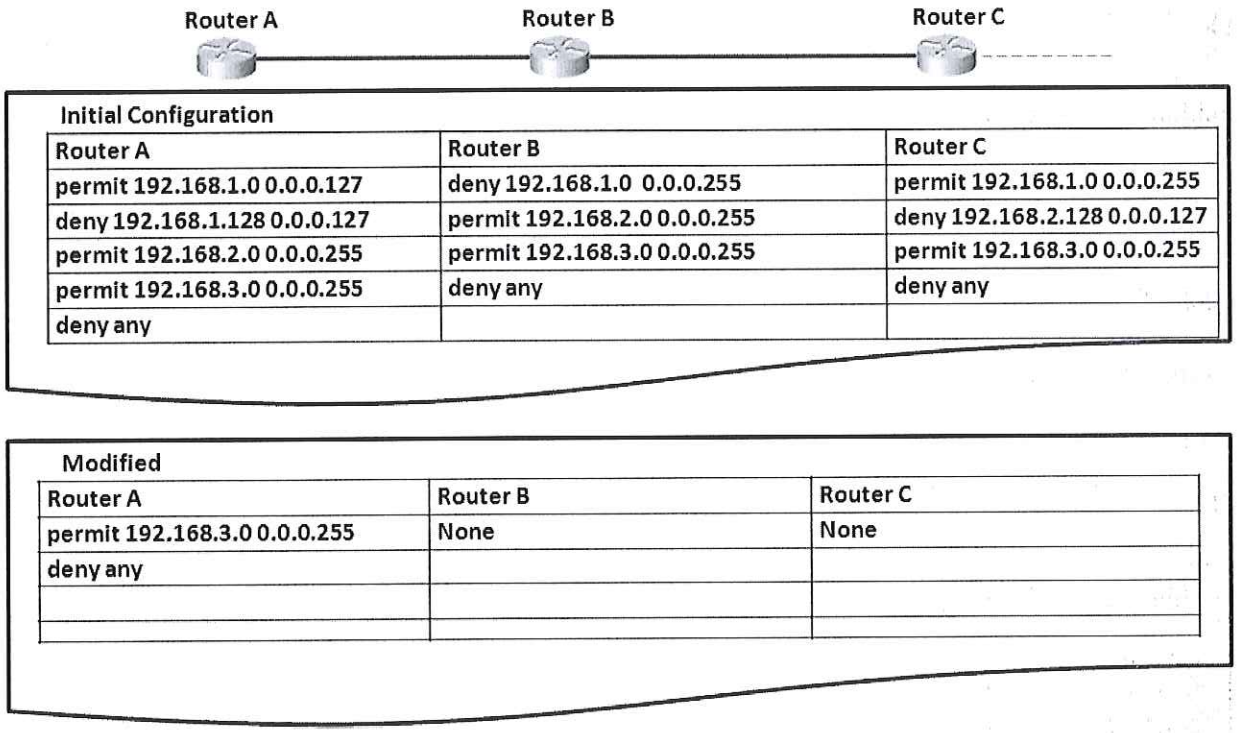
It is not a simple process to replace all the ACLs in routers internal to the domain with rules at the ingress and egress points. An investigation on the feasibility of carrying out this process has been considered for the simplest case of a standard ACL. There is a possibility that anomalies such as redundancies may exist within an ACL which could be removed without affecting the semantics of the ACL. This principle can be extended to consider subsequent routers along a path. Although

optimization of each individual ACL would provide some reduction in packet latency, it is unlikely that this will have the desired effect of removing all the rules in a router.

5.6. Example of Eliminating the Requirement for a Standard ACL

An example of how a standard ACL could be optimized to eliminate the ACL in 2 out of 3 routers is seen in Figure 10.

Figure 10. Optimizing rules across a domain.



The basis for optimization is that if some addresses are denied access by subsequent routers then these can be moved to routers earlier in the route. Additionally the range of rules can be extended to be incorporated into earlier rules by modifying the mask. The effect would be the reduction of the overall delay across the domain. This has to be done with great consideration to ensure that the security policy is not violated. In this simple example with 3 routers the effect would be a reduction in the delay over the domain by around 65%.

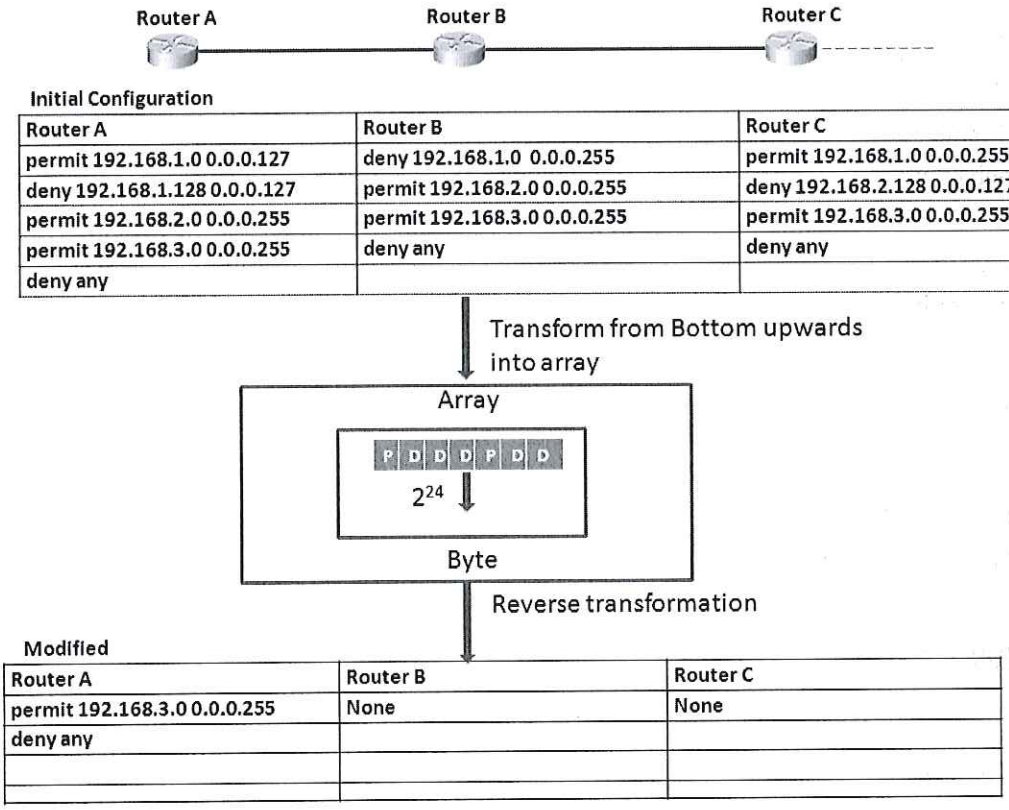
5.7. Processing of Rules

To prove the possibility of this being applied in practice the processing of the rules has been investigated. ACLs provide a very simple decision process since for every packet tested there are only 2 possibilities, permit or deny which are defined by the first rule match.

When considering all the rules in an ACL it is possible to represent the result for all possible IPv4 packets by constructing an array which contains a bit representing each possible IP address Figure 11. As there are 2^{32} possible IP addresses then such an array will require approx 512 Mbytes of memory. This was done by developing a custom-built data structure to store the bit values for each possible IP

address. The contents of the array is initially set to reflect all addresses being denied which corresponds to the implicit "deny any any" statement found at the end of all ACLs. If a "permit any any" statement specified before the "deny any any" then it would be prudent to populate the array in the opposite sense, all addresses being permitted instead of all denied, hence increasing the time taken for the array to be populated.

Figure 11. Building array.



The process starts by using the ACL found at the ingress point of the domain the array is populated with the filtering action for each possible IPv4 packet. The rules are evaluated starting from the bottom of the list since the priority of a rule increases for rules higher in the list. Figure 10 shows the process, for each router hop containing an ACL along the network path the array is rewritten based on the rules defined in each ACL. When the first rule in the ACL has been reached then the array reflects all the rules of the ACL. It is now necessary to modify the contents of the array created to reflect the rules found in the ACLs of the next router in the path. This process is continued until the final point in the domain is reached. The final content of the array represents the security policy for that route through the domain.

In this example, Router A represents the final router for the network path and Router C is the ingress router for the domain. Using the techniques described an equivalent rule set is created which is placed on Router A. The set of packets which can be forwarded by Router A is identical. However the removal of ACLs on Routers B and C considerably reduces the latency for a packet traversing that particular route. The security policy is represented as a byte array of 2^{24} elements. Since this example is based on standard ACLs this provides sufficient space to store the action taken for every possible

IPv4 packet ($2^{24} \times 8$). The IPv4 address is used as an array index when accessing a particular IP address value. For example the first element in the array maps to the IP address 0.0.0.0 and the last element to the IP address 255.255.255.255. The reverse transformation of the array is relatively trivial, using the same mapping technique in reverse it is possible to construct a rule set which consists of individual rules for each address value.

An intermediate step can be performed to merge rules into larger ranges providing they have the same filtering action. Appropriate masks are generated to represent the merged address ranges for all rules within the ACL. Due to the use of contiguous masks in most packet filtering schemes there are restrictions which limit the number of rules that can be merged. Each time a change is made in the security policy for a particular path through the domain it would be necessary to repeat the above steps to obtain a new ACL which reflects the contents of the rewritten array for each path affected by the change.

5.8. Creation of New Rules

Having created the array it is necessary to do a reverse transformation to produce the rules for an ACL which replicates the functionality of the distributed ACLs for a given filtering direction. The resulting ACL can be further optimized to remove any anomalies such as redundancies which may be present. It may also be possible to consolidate similar rules using wildcard masks and range commands for port numbers for adjacent or overlapping ranges. Once fully optimized, the ACL can be applied to the ingress/egress. The final list may be considerably longer than the initial ACL but based on the work carried out in section 3 by implementing this in a router with an advanced OS then the additional delay is insignificant.

One of the main concerns that network administrators have about this technique is that the final list can bear little relationship to the original ACL created. To alleviate this concern the original ACL is kept and a cross reference list is provided showing the correlation between the rules in the ACLs *i.e.*, each rules in the initial ACL is cross referenced with the rule in the modified ACL that a packet would match. Changes made by the Network Administrator are, of course, applied to the original list before further efficiencies are applied.

6. Conclusions

Utilizing routers within a domain to provide security does have an impact on the performance of the network since it introduces significant delays due to the equipment. There are some relatively simple steps that can be taken to improve the performance.

By investigating the theoretical aspects of delays through routers and carrying out a series of measurements it has been possible to improve the mathematical model of delays encountered by a packet as it transverses a domain. It has also been possible to quantify the delays to understand which components are more significant which leads to a series of rules that can be used as best practice when designing large networks.

There are significant differences in the delays experienced using different versions of the OS in the router. A more advanced OS adds delays to the basic routing process but if other functionality is required then an advanced OS has to be used.

Optimal performance can be gained by not having ACLs enabled in a router. Clearly it is not possible to remove the ACLs from all routers within a domain but there are gains to be made by reducing the number of routers that have ACLs enabled. When using an Advanced OS the number of rules in an ACL is insignificant. Since a domain has a common security policy then it should be possible to optimize the placement of ACL rules to ensure that the minimum number of routers in a domain use an ACL.

Having completed optimization on the number of routers requiring an ACL then using a basic OS for a router without ACLs and using advanced OS for the routers that do require an ACL will show an overall improvement of performance.

The results of this paper have demonstrated that it is possible to optimize the performance of a network without modification to its underlying hardware or security policy. The most significant finding is to ensure that the OS loaded into routers is appropriate for the functionality required and so understanding the requirements of the individual routers in a domain is essential.

By investigating a domain which has a single security policy and therefore allows functionality of routers to be allocated this paper considers how it can be optimized. It shows that by ensuring the OS with the appropriate functionality is used an improvement of performance will be gained. This is particularly important when the security is provided by ACLs. By utilizing an OS with advanced functionality an improvement around 130% is possible.

Furthermore this paper shows that by moving the ACLs to only the egress/ingress points of a domain that a performance improvement of the order of 320% can be gained over using a basic OS with no optimization or in excess of 30% over using an advanced OS. The justification for doing so is that within a domain there is a common security policy and so it is possible to reduce the number of routers that have to be configured with an ACL and therefore reduce the delays through them.

7. Future Work

These results have been produced for a fixed hardware configuration which was a typical router and so further investigations can be carried out to understand the effect of more advanced hardware.

The effect of using additional functionality/services to the network within a router e.g., DHCP, HTTP were not studied. It would be expected that these could have considerable effects.

In the example of showing the process of optimizing across domain this paper only considered standard ACLs for simplicity to prove the principle; however in a typical domain extended ACLs would be used to provide more granular filtering for security purposes. Figure 9 (Optimizing rules across a domain) in section 5 gives an example of how standard ACLs can be optimized. Future work will provide a similar example applicable to extended ACLs.

As the Internet gets moved to utilizing IPv6 it will be necessary to consider ACLs which use this protocol for addressing. Clearly this is a far more complex issue due to the size of the parameters involved.

References

1. Davies, J.N.; Grout, V.; Picking, R. Improving the Performance of IP Filtering Using a Hybrid Approach to ACLs. In *Proceedings of the 8th International Network Conference (INC2010)*, Heidelberg, Germany, 6–8 July 2010.

2. Grout, V.; McGinn, J.; Davies, J.N.; Picking, R.; Cunningham, S. Rule Dependencies in Access Control Lists. In *Proceedings of International Conference WWW/Internet (IADIS)*, San Sebastian, Spain, 25–28 February 2006.
3. Al-Shaer, E.S.; Hamed, H.H. Modelling and management of firewall policies. *IEEE Trans. Netw. Serv. Manag.* **2004**, *1*, 2–10.
4. Hari, B.; Suri, S.; Parulkar, G. Detecting and Resolving Packet Filter Conflicts. In *Proceedings of the 19th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM00)*, Tel Aviv, Israel, 26–30 March 2000.
5. Bukhatwa, F.; Patel, A. Effects of Ordered Lists in Firewalls. In *Proceedings of International Conference (IADIS) WWW/Internet 2003*, Algarve, Portugal, 5–8 November 2003.
6. Bukhatwa, F. High Cost Elimination for Best Class Permutation in Access Lists. In *Proceedings of International Conference (IADIS) 2004*, Madrid, Spain, 6–9 October 2004.
7. El-Atawy, A.; Hamed, H.; Al-Shaer, E. Adaptive Statistical Optimization Techniques for Firewall Packet Filtering. In *Proceedings of IEEE INFOCOM 2006*, Barcelona, Spain, 23–29 April 2006.
8. Gupta, P.; McKeown, N. Classifying packets with hierarchical intelligent cuttings. *IEEE Micro* **2000**, *20*, 34–41.
9. Singh, S.; Baboescu, F.; Varghese, G.; Wang, J. Packet classification using multidimensional cutting. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, 25–27 August 2003.
10. Varghese, G. *Network Algorithmics*; 1st ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2005; p. 75.
11. Meiners, C.R.; Liu, A.X.; Torng, E. TCAM razor: A systematic approach towards minimizing packet classifiers in TCAMs. *IEEE/ACM Trans. Netw.* **2010**, *18*, 490–500.
12. Meiners, C.R.; Liu, A.X.; Torng, E. *Hardware-Based Classification for High-Speed Internet Routers*; 1st ed.; Springer: Berlin/Heidelberg, Germany, 2010; p. 2.
13. Liu, A.X.; Meiners, C.R.; Yun, Z. All-Match Based Complete Redundancy Removal for Packet Classifiers in TCAMs. In *Proceedings of IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, Phoenix, AZ, USA, 13–18 April 2008.
14. Alfaro, J.G.; Cuppens, F.; Cuppens-Boulahia, N. Complete analysis of configuration rules to guarantee reliable network security policies. *Int. J. Inf. Secur.* **2008**, *7*, 103–122.
15. Al-Shaer, E.; Hamed, H.; Boutaba, R.; Hasan, M. Conflict classification and analysis of distributed firewall policies. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 2069–2084.
16. Kim, S.; Lee, H. Classifying rules by in-out traffic direction to avoid security policy anomaly. *Trans. Internet Inf. Syst.* **2010**, *4*, 671–690.
17. Alfaro, J.G.; Cuppens, F.; Cuppens-Boulahia, N. Analysis of Policy Anomalies on Distributed Network Security Setups. *Lecture Notes Comput. Sci.* **2006**, *4189*, 496–511.
18. Alfaro, J.G.; Cuppens, F.; Cuppens-Boulahia, N. Aggregating and Deploying Network Access Control Policies. In *Proceedings of The Second International Conference on Availability, Reliability and Security (ARES 2007)*, Vienna, Austria, 10–13 April 2007.
19. Al-Shaer, E.S.; Hamed, H.H. Discovery of Policy Anomalies in Distributed Firewalls. In *Proceedings of INFOCOM 2004, the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, Hong Kong, China, 7–11 March 2004.

20. Yuan, L.; Chen, H.; Mai, J.; Chuah, C.N.; Su, Z.; Mohapatra, P. FIREMAN: A Toolkit for Firewall Modeling and Analysis. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 21–24 May 2006.
21. Chen, F.; Bruhadeshwar, B.; Liu, A.X. A Cross-Domain Privacy-Preserving Protocol for Cooperative Firewall Optimization. In *Proceedings of IEEE INFOCOM 2011*, Shanghai, China, 10–15 April 2011.
22. Grout, V.; McGinn, J. Optimisation of Policy-Based Internet Routing using Access Control Lists. In *Proceedings of 9th IFIP/IEEE Symposium on Integrated Network Management (IM 2005)*, Nice, France, 15–19 May 2005.
23. Guardog. Available online: http://www.simonzone.com/software/guarddog/ (accessed on 5 April 2012).
24. Cisco Systems, User Guide for ACL Manager 1.5, Optimizing ACLs. 2003. Available online: http://www.cisco.com/en/US/products/sw/cscowork/ps402/products_user_guide_chapter09186a008017addf.html (accessed on 5 April 2012).
25. Choi, B.Y.; Moon, S.; Zhang, Z.; Papagiannaki, K.; Diot, C. Analysis of point-to-point packet delay in an operational network. *Comput. Netw.* **2007**, *51*, 3812–3827.
26. Moy, J. *RFC 2328 OSPF Version 2*; The Internet Society: Reston, VA, USA,1998.
27. Resende M.G.C.; Pardalos, P.M. *Handbook of Optimization in Telecommunications*; Springer Science + Business Media: New York, NY, USA, 2006.
28. Hohn, N.; Papagiannaki, K.; Veitch, D. Capturing router congestion and delay. *IEEE/ACM Trans. Netw.* **2009**, *17*, 789–802.
29. Lai, K.; Baker, M. Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '00)*, Stockholm, Sweden, 28 August–1 September 2000.
30. Bollapragada, V.; White, R.; Murphy, C. *CCIE Professional Development: Inside Cisco IOS Software Architecture*; 1st ed.; Cisco Press: Indianapolis, IN, USA, 2000; p. 13.
31. Cisco Tools. Available online: http://tools.cisco.com/ITDIT/CFN/jsp/compareImages.jsp (accessed on 19 March 2012).
32. Sedayao, J. *Cisco IOS Access Lists*; 1st ed.; O'Reilly & Associates, Inc.: Sebastopol, CA, USA 2001; pp. 22–31.