Scenario Analysis using Out-of-line Firewall Evaluation Framework

L. Saliou, W.J. Buchanan, J. Graves, and J. Munoz
Centre for Distributed Computing and Security, Napier University, Edinburgh, United Kingdom
l.saliou@napier.ac.uk
w.buchanan@napier.ac.uk
j.graves@napier.ac.uk
j.munoz@napier.ac.uk

Abstract: Distributed Denial-of-Service (DDoS) attacks against corporate networks and assets are increasing, and their potential risk for future attacks is also a major concern. These attacks typically aim at disabling computer network infrastructure, and, since there is no one method to mitigate this type of threat, organisations must deploy adequate solutions, and assess the adequacy of their choices against their network requirements, through analysis, such as a simulation, or through network device modelling. A key factor is that DDoS is a dynamic type of attack, and thus device performance is a key parameter, especially for intermediate devices, such as network firewalls. Most of the modelling, though, for firewalls is focusing on static and logical performance attributes, such as whether traffic is denied or permitted. Thus existing models typically cannot deal with dynamic issues when related to intermediate devices. Simulation tools might be possible, but it is often difficult to cover a whole range of devices, thus this paper outlines a novel method of modelling the dynamic performance of network firewalls, and in measuring if they can cope with varying network loads.

Keywords: Dynamic evaluation, network firewall, analysis, security, out-of-line evaluation.

1. Introduction

Increased reliance on networked systems poses several challenges to organisations, especially in terms of: protecting data from unauthorised access or corruption; guarantying accessibility; and; resilience of network services among others. Safeguarding assets, such as Web servers, customers' databases, and so on, is often achieved by means of security policies. However, this is a non-trivial task (Eloff *et al.* 2003), and these policies often do not reflect organisational aims, objectives, or work practices (Danchev 2003a). There could be also an issue with policy interpretation by technical staff (Saliou *et al.* 2005 2006) as they might not fully understand legal requirements (Barton *et al.* 2003). In addition, the discrepancy between the expression of policies and the actual deployment on live systems, often plays an important role in security breaches (Danchev 2003b, and Ioannidis *et al.* 2000). Furthermore, security solutions, such as network firewalls, are often deployed with little testing (Danchev 2003b), or succinct investigation, regarding their performance in a production environment. This is a serious issue, as, of all the security solutions that an organisation could use to defend its assets (Saliou *et al.* 2006) with, the network firewall is the element whose configuration is the most closely linked to the organisation's security policy.

Many threats exist in networked systems and most systems can cope with known threats, such as worms and viruses, as they have well-known traffic and activity signatures, and also have reliable mitigation procedures. Unfortunately, DDoS is one of the most difficult attacks to cope with, as it is almost impossible to differentiate between legitimate and non-legitimate traffic when they are requesting resources, such as from a Web server, or networked device. It is thus a growing threat, as Yegneswaran *et al.* 2003 highlight, and comes in many forms (Glenn 2003, and Paxson 2001). Along with this, servers can be hardened against DDoS, however there is an effect on intermediate devices, such as network firewalls, and is a major threat. Hence, a strong understanding of how intermediate device cope with traffic flows is just as important as how an end-device, such as a server, might cope with dynamic attacks.

One of the methods used to enhance the network infrastructure resilience is to deploy rules against a complete network domain, such as for an Internet Service Provider (ISP) domain, however this would also ban hosts within that domain which were non-malicious. Thus, an alternative is to apply firewall rules that deny known offending nodes access to the corporate network (Glenn 2003). This approach is likely to results in large number of firewalls rules for a large number of attack nodes (Yegneswaran *et al.* 2003). However, creating firewall rule sets is often error prone, thus researchers have developed the means to analyse rule sets for contradictions, and other logical defects (Al-Saer *et al.* 2004, Guttman *et al.* 2003), in turn producing near-optimal rule sets. This reduced rule set is important, as it has been shown that large rules sets strongly affect network firewall performance (Saliou *et al.* 2006). Unfortunately, along with understanding the effect of rule set sizes, there is also a lack of research on assessing the feasibility of the chosen policy on a given system. At present, network performance is not taken into account when implementing security factors, such as rule set size and robustness. This paper outlines a system which allows improved judgments to be made on whether a policy is achievable, before it is actually deployed, and whether a device can cope with a dynamic attack, such as with DDoS.

2. Background

Research in the field of network firewalls often focuses on logical attributes such as: what is authorised and what is not; and whether it fits within the organisational requirements. However, there is little research in attempting to assess whether chosen policies are achievable, especially in terms of varying network traffic levels. Indeed, detailed and granular security policies are likely to translate into extensive and complex rule sets, which can degrade network performance.

The issues include the central position in the security infrastructure of network firewalls, and their performance capabilities. The fact that network firewall constitute a central point-of-failure has already been highlighted in Ioannidis *et al.* 2000. They propose an alternative to network firewalls with a software suite which is managed across the network, and in a distributed manner, however it has hardware requirements that cannot be always met by organisations. Thus, it is important to provide organisations with means to identify when they need to consider an alternative technology for performance reasons; a point which is not addressed in Ioannidis *et al.* 2000.

One argument against increased security is that it might have an affect on network performance, to such an extent that it outweighs the benefits of the security implementation (Rodgers 2003), and is often used as an argument for reduced security standards (Smetters *et al.* 2002). Yet, some researchers, such as Lyu *et al.* 2000, claim that their evaluations prove otherwise. Unfortunately, there is currently no scientific model that can actually predict the actual effect that security policies have on network firewall performance. Such a model is required as solutions are more likely to be deployed, if their benefit can be fully demonstrated (Bajcsy *et al.* 2004).

This research thus presents an evaluation environment which evaluates performance in regards to varying security policies, in an objective manner. The evaluation of network security equipment must thus be carried out in an environment which is as close as possible to the production environment in which the device is going to be deployed, in order to enhance the relevance of the data gathered. This research investigates a novel integrated security framework which aims to highlight the discrepancies: between organisational requirements; deployment on live equipment; and the audit of the overall security posture (Saliou *et al.* 2005). This research argues the benefit of such a framework, as it incorporates an environment which evaluates network devices under conditions similar to actual production environment in order to establish the device's strengths, and weaknesses. The gathered data could then serve to assess the feasibility of a chosen security policy, evaluate the resilience of the network infrastructure against emergent network threats, such as infrastructure-based DDoS. Hence, this paper demonstrates the ecology between the chosen security policy, and the equipment that is tasked to enforce it.

3. A novel dynamic evaluation environment for firewall analysis

This section highlights some of the challenges involved in network firewall testing, and how these obstacles are overcome. A network firewall performance model is also described along with experimental procedures used to create it.

3.1 Environment Design

Networks performance modelling and evaluation for firewalls suffers from four main problems:

- **Lack of component data on devices.** Individual security devices are typically not tested in isolation, whereas in most other engineering or scientific disciplines, a system is normally split into subcomponents, each of which are tested independently from the others, and verified from a component point-of-view. Thus, a model of each component is created, and evaluated to be able to build a complete understanding of the whole system, and how the components fit together. Unfortunately, little work has been done on this, as most organisations simply evaluate the complete networked system, without actually understanding the dynamic performance of the network components. A key focus is thus to evaluate performance using network metrics, such as latency and available bandwidth across the Device Under Test (DUT), as opposed to evaluate application layer protocol, such as with HTTP and FTP (Lyu *et al.* 2000). Typically, programs implementing these protocols introduce overheads, and different program versions can also influence evaluation results.
- **Non-repeatable and verifiable test evaluation.** It is often difficult to test a firewall for every known situation, as there are an almost infinite number of different traffic streams, and firewall settings. Thus, as highlighted by Al-Tawil *et al.* 1999, out-of-line evaluations are often time-consuming, and true scientific rigour demands the exploration more than one configuration.
- **Non-automated test environments.** Al-Tawil *et al.*'s evaluations were, unfortunately, done mostly by-hand, as was the research in Saliou *et al.* 2006.
- **Lack of relevance with production environment.** Most researchers, in the past, have only tested firewalls with the traffic which exercised the key logical parameters of the firewalls, and did not contain any other background traffic. This is unfortunate, as real-life firewalls must cope with both background, and filtered traffic (Al-Tawil *et al.* 1999).

Zhang *et al.* 2002 recommend creating network component models from data collected in a live environment. However, accurate models can only be created when environmental variables, such as network load, or firewall rule sets’ position (Saliou *et al.* 2006), can be controlled. In order to address the shortcomings of Al-Tawil *et al.* 1999 and Saliou *et al.* 2006, this research has created a framework which is: capable of configuring network equipments according to the specifications of evaluation scenario; capable of controlling network load; and, is able to collect chosen metrics, in an automated manner. The orchestration of the overall system is inspired by the work of Donat 2005, and thus the combination of these features enables repeatable evaluations. Figure 1 illustrates the model using this methodology; and Figure 2 illustrates the evaluation framework.

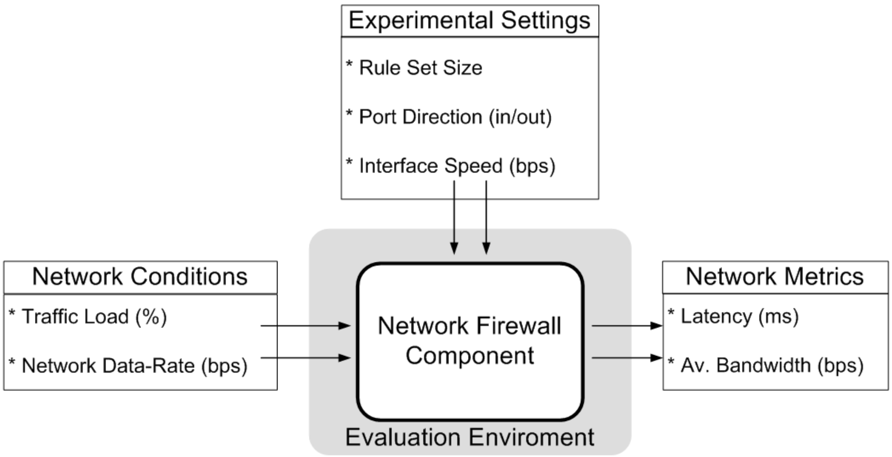

Figure 1: Network firewall performance model

3.2 Evaluation procedure

This section outlines the usage of an out-of-line dynamic evaluation environment by studying the deployment of a firewall rule set designed to prevent 1000 known offending network nodes from accessing corporate network assets. The rule-set is designed in such a way that the firewall must examine every item in the rule-set before allowing traffic through. Hence, this evaluation uses stateless firewalling. This type of firewalling does not require the device to keep track of previous communications. Furthermore, in order to ensure that the data gathered for tested equipments is comparable, the rule-set is expressed using a generic form based on Al-Saer *et al.* 2004 model for firewall rules, of which a snippet is given in Appendix B. The rule-set is then converted into the syntax specific to the firewalling engine used by the device, such as for a Cisco Access Control List (ACL), or a Linux netfilter statement (Netfilter Core Team 2006). Hence the evaluation environment is not tied to a specific equipment vendor.

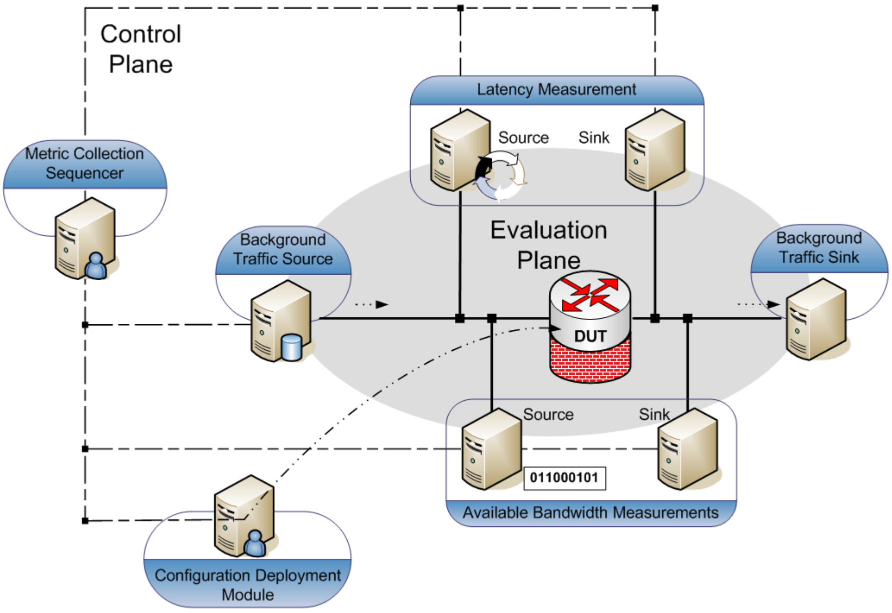
Control Plane
Latency Measurement
Source
Sink
Metric Collection Sequencer
Background Traffic Source
Evaluation Plane
Background Traffic Sink
DUT
Source
Sink
011000101
Available Bandwidth Measurements
Configuration Deployment Module

Figure 2: Evaluation environment

The evaluation was carried out on two distinct network firewalls. The first device is manufactured by Cisco Systems (DUT 1), and the second is built using standard PC hardware elements and specialised software applications running on a Linux kernel (DUT 2). The full specification of these firewalls is given in Appendix A. The evaluation was then repeated for both low and high network data rate. Low data rate networks have a maximum rate of 10 Mbps, and high data rate networks have a maximum rate of 100 Mbps.

The metrics are then collected using widely available network measurement tools: HPing for the latency (Sanfilippo 2005); and Netperf for the available bandwidth (Hewlett-Packard 2005). As advised in the Netperf documentation, each instance of sample collection lasts for 60 seconds. Every result items presented in this paper is the average of at least 10 samples. The orchestration of the sample collection is realised through a specially-developed software interface between the scheduling module (cf. Figure 2), and the required software applications are distributed across nodes of the evaluation environment. The same interface is also used to control the traffic generator, which reproduces realistic network conditions (Turner 2005). This traffic generator is designed to replay traffic at different rates, typically as a percentage of the maximum network data rate. Henceforth, the network-load refers to the bandwidth throughput of the traffic generator.

The DUT is thus used as the entry point of all traffic crossing the network, as illustrated in Figure 2. It interconnects two distinct networks, where one is setup to initiate, the main traffic flow, and is illustrated on the left-hand side of Figure 2, whereas the second network acts as a sink for the traffic. The evaluation begins with metrics being collected before the rule-set is deployed onto the DUT. These readings serve as a baseline to measure the impact of rule-set's size on firewall performance. The collection of metrics is then repeated each time the traffic load is increased by 10% of the maximum network data rate. Next, the rule-set is deployed on the device, and the cycle of metric collection is repeated. The successful deployment of a rule-set on Cisco equipment requires both network interface, and communication flow direction to be specified (Saliou *et al.* 2006), hence Section 4 reflects this characteristic by presenting two sets of results for DUT 1. As far as the logic of firewalling is concerned, it makes no difference where the rule set is applied, however, for scientific rigour the variation in position is investigated.

4. Evaluation Results

This section presents an outline of the results obtained during the experiments.

4.1 Static Results

Table 1 and Table 2 present the results in a static manner. These tables suggest that, for low data rates, the impact of deploying the rule-set is negligible for both devices. High data rate readings indicate that high bandwidth usage is not achievable with DUT 1, however, there is hardly any loss in performance once the firewalling functions are enabled. This contrasts with DUT 2 which offers, in baseline configuration, excellent performance that is dramatically reduced once the rule set is deployed.

Table 1: DUT 1 static results

Base line	Low data rate		High data rate	
	Latency (ms)	Av. Bandwidth (Mbps)	Latency (ms)	Av. Bandwidth (Mbps)
	1.15	9.39	0.67	54.75
Evaluation rule set	Low data rate		High data rate	
	Latency (ms)	Av. Bandwidth (Mbps)	Latency (ms)	Av. Bandwidth (Mbps)
Incoming interface	1.18	9.39	0.70	52.13
Outgoing interface	1.17	9.39	0.67	55.10

Table 2: DUT 2 static results

Base line	Low data rate		High data rate	
	Latency (ms)	Av. Bandwidth (Mbps)	Latency (ms)	Av. Bandwidth (Mbps)
	0.74	9.39	0.30	87.99
Evaluation rule set	Low data rate		High data rate	
	Latency (ms)	Av. Bandwidth (Mbps)	Latency (ms)	Av. Bandwidth (Mbps)
	1.11	9.39	0.63	38.06

4.2 Dynamic Results

Figures 3 to 6 plot the collected metrics against increasing network load for both high and low data rate network settings. Figure 3 presents the results for network latency for a high data rate setup. DUT 2 offers promising results with baseline configuration as readings vary from 0.3 ms to 1.6 ms when traffic load reaches 50 %. Hence, these readings cannot be visualised in Figure 3. However the impact of the rule set is evident when the traffic load exceeds 10% of the maximum data rate. DUT 1 is visibly sensitive to network load, as the latency noticeably increases after a 10% network load in all of the evaluation configurations. When deploying the rule set it is possible to mitigate the effect on latency by applying the rule set on the outgoing interface.

Figure 4 shows the degradation in performance is almost linear in all situations, and it is noticeable that the gradient is greater than 1. For the Cisco firewall, this gradient is around 1.6 for traffic load between 0% and 50%, and 3, thereafter. The gradient is about the same for the Linux firewall, when the rule-set is applied, for approximately traffic load between 0% and 20%. Figure 4 clearly shows that, DUT 2, with far less available bandwidth with no traffic, has limited resilience against high traffic load.

Figure 5 shows that for DUT 1, readings do not permit to distinguish between the incoming or outgoing interface configurations, unlike in the high data rate situation. The effect is more noticeable with DUT 2 from a device point-of-view, but delays only become greater than DUT 1's for traffic loading above 40%. Arguably, for low data rates, it is difficult to measure the footprint on network performance of deploying the rule-set, as Figure 6 illustrates.

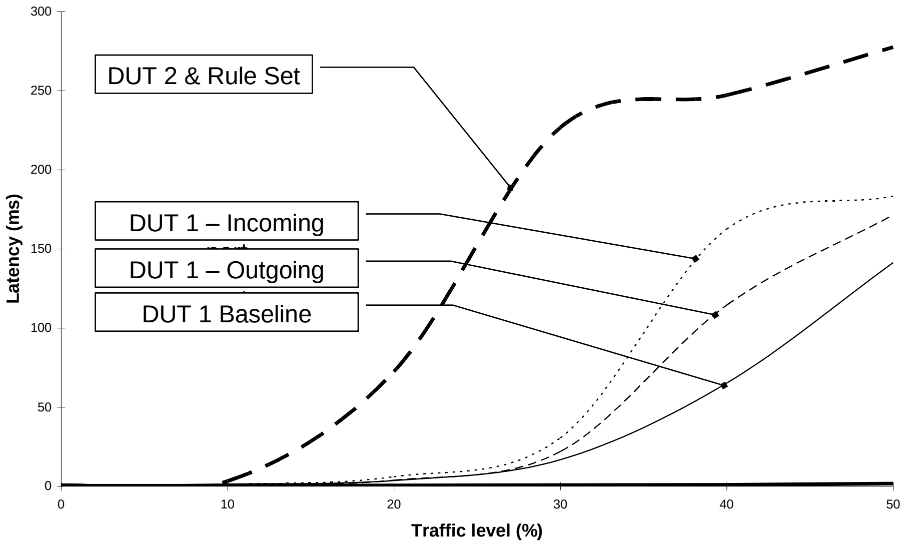

Figure 3: High data rate latency

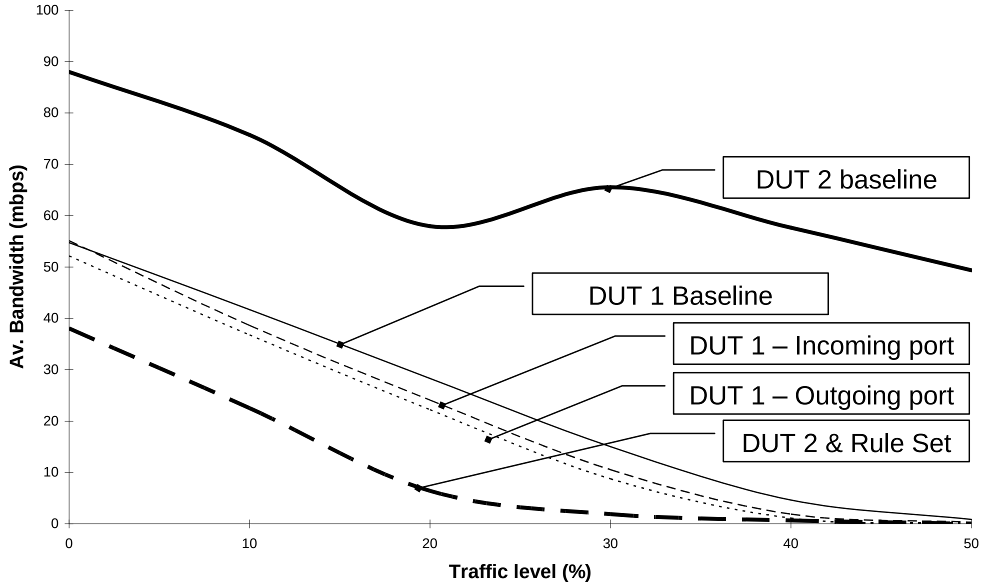

Figure 4: High data rate available bandwidth

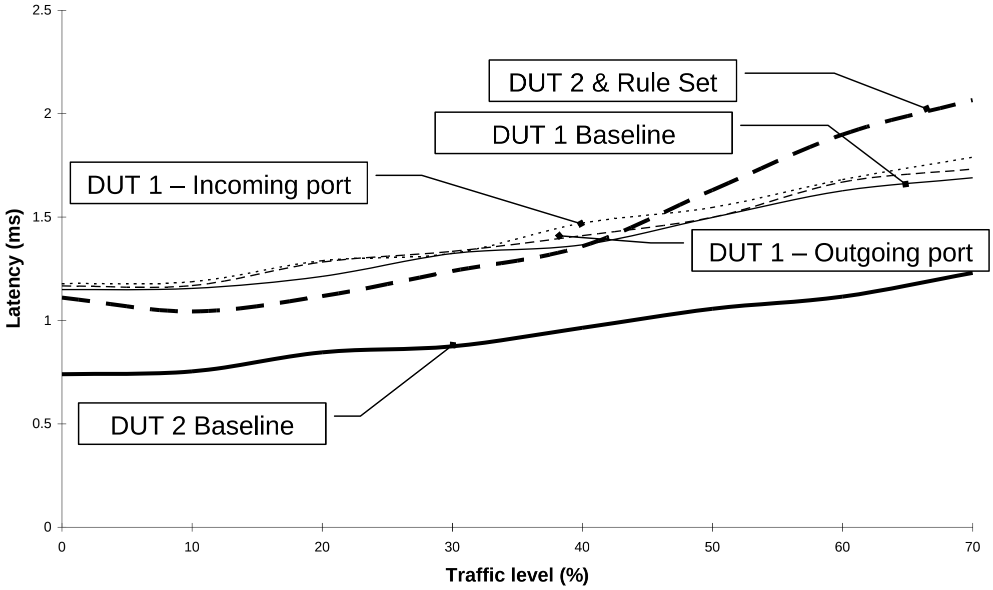

Figure 5: low data rate latency

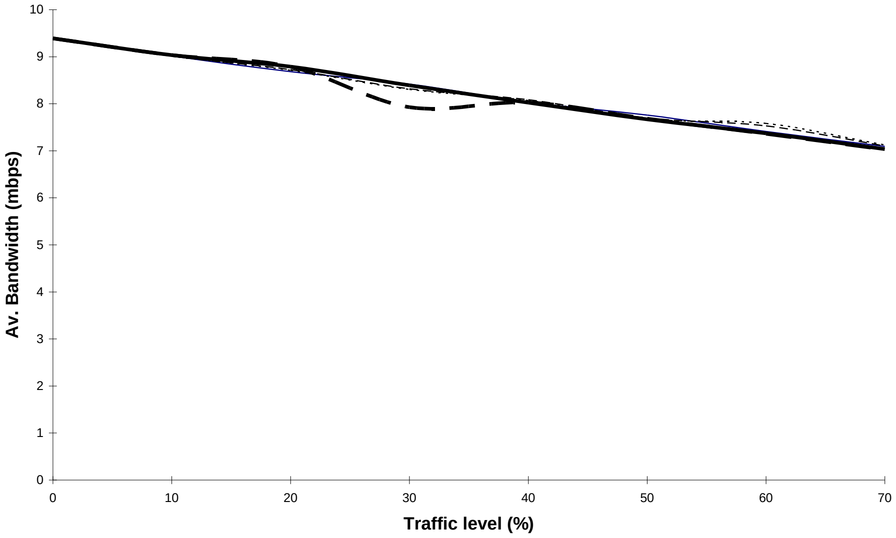

Figure 6: Low data rate available bandwidth

5. Conclusions and future work

This paper has shown that a better understanding of the effect on network performance of increased security can be obtained when network firewall equipments are evaluated in a dynamic manner. It has also highlighted that environments permitting such evaluation can be created outside the production environment, in other words out-of-line, and can still produce data relevant to the environment in which the device is to be deployed. This is achieved by combining the scheduling of metric collection, and control over network characteristics, such as traffic load and device configuration. In addition, the automation of such a system allows experiments to be repeated rapidly, and thus enables the testing of a variety of firewall systems.

Furthermore, data obtained with this environment allow the creation of a network firewall performance model, which could be used in combination with logical firewall models, such as of Al-Saer *et al.* 2004, and Guttman *et al.* 2003, in order to assess the feasibility of a chosen security policy. Arguably, not all types of firewall devices can be thoroughly tested as there are an almost unlimited number of different infrastructure, however, with enough devices sampled key the attributes, such as interface speed, firewalling engine, and so on, can be identified and, in turn, help determine a best fit, for an unknown device.

This paper has demonstrated that there is thus an ecology between: the desired security; the device chosen to enforce it; and, the environment in which the device will operate. Indeed, results show that network performance can be severely affected, by up to 50%, depending on the type of firewall equipment employed to protect the network.

As the tested rule set required the device to examine all items against the incoming traffic, it would be interesting to investigate the difference in performance when the rule set is arranged so that threats are examined last, thus the firewall prioritises filtering corporate traffic. Furthermore, the rule set uses addressing information only, whereas finer control can be exercised using the information on the requested services, for instance. Hence, the additional overhead, and additional cost in performance, should be investigated.

Appendix A: List of equipment used during the experiments

Cisco device (DUT 1):

- Cisco c2600MX series
  Cisco IOS 12.3 (27), release software fc3.

Linux netfiler (DUT 2):

- Operating System: Debian Ubuntu Linux 6.06 Dapper Drake
  Routing software : quagga 0.99.2-1ubuntu3
  Firewalling engine: netfilter
  Pentium III 700MHz, 256MB of RAM, 3 Com 10/100Mbps compatible Ethernet cards

Network switches:

- Cisco Catalyst 2950 series
  Cisco IOS 12.1(13) EA1a, release software fc1.

Traffic generation nodes:

- Pentium IV 2.4GHZ, 512MB of RAM, 3 Com 10/100Mbps compatible Ethernet cards
  Operating System: Debian Ubuntu Linux 6.06 Dapper Drake
  TCPReplay version 2.2.0, 1
  DARPA trace from the 1st Monday of the 1st week of the year 1998 without ARP preamble

Latency measurement, available bandwidth measurement, scheduling module, configuration module, nodes:

- Pentium III 700MHz, 256MB of RAM, 3Com 10/100Mbps compatible Ethernet card
  Operating System: Debian Ubuntu Linux 6.06 Dapper Drake
  Hping version 2.0.0r3, 1
  Netperf version 4a

Scheduling module, configuration module, software details:

- Microsoft .NET Framework 1.1
  C# programming language for .NET

Appendix B: Rule set sample

The following outlines the generic rules format used for generating the test rule sets, such as for the conversion from these generic rules into Cisco ACLs and also Linux netfilter syntax.

Rule Nb.	Protocol	source address	action
1	ip	177.129.1.56	deny
2	ip	118.226.112.42	deny
3	ip	124.84.62.221	deny
4	ip	189.119.171.195	deny
...			
1000	ip	10.0.0.0 255.0.0.0	accept

Figure 7: Tested rule set snippet expressed using Al-Saer *et al.* 2004 model

Reference

Al-Saer, E. S. and Hamed, H. H. (2004) "Modelling and Management of Firewall Policies", *IEEE Transactions on Network and Service Management*, Vol. 1, No. 1, pp 3 - 13

Al-Tawil, K. and Al-Kaltham, I. A. (1999) "Evaluation and testing of internet firewalls", *International Journal of Network Management*, Vol. 9, No. 3, pp 135 - 149

Bajcsy, R., Benzel, T., Bishop, M., Braden, B., Brodley, C., Fahmy, S., Floyd, S., Hardaker, W., Joseph, A., Kesidis, G., Levitt, K., Lindell, B., Liu, P., Miller, D., Mundy, R., Neuman, C., Ostrenga, R., Paxson, V., Porras, P., Rosenberg, C., Tygar, J. D., Sastry, S., Sterne, D. and Wu, S. F. (2004) "Cyber defense technology networking and evaluation", *Communications of the ACM*, Vol. 47, No. 3, pp 58-61

Barton, P. and Nissanka, V. (2003) "Cyber-crime - Criminal offence or civil wrong?" *Computer Law and Security Report*, Vol. 19, No. 5, pp 401-405

Danchev, D. (2003) "Building and Implementing a Successful Information Security Policy", [online], Window Security, http://www.windowsecurity.com/pages/article_p.asp?id=1218

Danchev, D. (2003) "Reducing "Human Factor" Mistakes", [online], WindowSecurity, http://windowsecurity.com/pages/article_p.asp?id=1261

Donat, M. (2005) "Orchestrating an automated test lab", *Queue*, Vol. 3, No. 1, pp 46-53

Eloff, J. and Eloff, M. (2003) *Information security management: a new paradigm*, 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, Gauteng, South Africa, pp 130-136

Glenn, M. (2003). *A summary of DoS/DDoS Prevention, monitoring and Mitigation Techniques in a Service Provider Environment* (White Paper): SysAdmin, Audit, Network, Security Institute, pp 1-30

Guttman, J. D. and Herzog, A. L. (2003) "Rigorous Automated Network Security Management", *International Journal of Information Security*, Vol. 4, No. 1-2, pp 29-48

Hewlett-Packard (2005) "Netperf", [online], http://www.netperf.org/netperf/NetperfPage.html

Ioannidis, S., Keromytis, A. D., Bellovin, S. M. and Smith, J. M. (2000) *Implementing a distributed firewall*, 7th ACM conference on Computer and communications security, Athens, Greece, pp 190-199

Lyu, M. R. and Lau, L. K. Y. (2000) *Firewall Security: Policies, Testing and Performance Evaluation*, 24th International Computer Software and Applications Conference, pp 116-121

Netfilter Core Team (2006) "The Netfilter.org Project", [online], Noris Network, http://www.netfilter.org/

Paxson, V. (2001) "An Analysis of Using Reflectors for Distributed Denial-Of-Service Attacks", *Computer Communication Review*, Vol. 31, No. 3, pp 38 - 47

Rodgers, D. H. (2003). *Implementing a Project Security Review Process within Project Management Methodology* (Practical Assignment): SANS, pp 1 - 23

Saliou, L., Buchanan, W. J., Graves, J. and Munoz, J. (2005) *Novel Framework for Automated Security Abstraction, Modelling, Implementation and Verification*, 4th European Conference on Information Warfare and Security, Glamorgan, United Kingdom, pp 303-311

Saliou, L., Buchanan, W. J., Graves, J. and Munoz, J. (2006) *Analysis of Firewall Performance Variation to Identify the Limits of Automated Network Reconfigurations*, 5th European Conference on Information Warfare and Security, Helsinki, Finland, pp 205-214

Sanfilippo, S. (2005) "Hping Security Tool", [online], http://www.hping.org/

Shimonski, R. J. (2004) "Threats and your Assets – What is really at Risk?" [online], WindowSecurity, http://windowsecurity.com/pages/article_p.asp?id=1354

Smetters, D. K. and Grinter, R. E. (2002) *Moving from the design of usable security technologies to the design of useful secure applications*, 2002 workshop on New security paradigms, Virginia Beach, Virginia, USA, pp 82 - 89

Turner, A. (2005) "Tcpreplay: Pcap editing and replay tool for *Nix", [online], http://tcpreplay.sourceforge.net/

Yegneswaran, V., Barford, P. and Ullrich, J. (2003) "Internet intrusions: global characteristics and prevalence", *ACM SIGMETRICS Performance Evaluation Review*, Vol. 31, No. 1, pp 138 – 14

Zhang, Y., Breslau, L., Paxson, V. and Shenker, S. (2002) *On the characteristics and origins of internet flow rates*, the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, Pittsburgh, Pennsylvania, USA, pp 309-322