The Design and Implementation of a Network Simulation Platform

S. von Solms
Defence, Peace, Safety and Security
Council for Scientific and Industrial Research
Pretoria, South Africa
svsolms@csir.co.za

S.W. Peach
Defence, Peace, Safety and Security
Council for Scientific and Industrial Research
Pretoria, South Africa
speach@csir.co.za

***Abstract*—Network security risks are becoming an increasing threat as new network attack methods are constantly being developed by hackers to compromise secure networks and devices. The use of a network simulation environment that can realistically replicate these events and their effects can enable researchers to identify these threats and find ways to counter them. In this paper we present the design of a network simulation platform which can enable researchers to study dynamic behaviour of networks, network protocols, and emerging classes of distributed applications in a controlled setting under real-world conditions.**

Keywords— emulation, networks, simulation, virtualisation

I. Introduction

Numerous South African corporate networks are exposed to security risks due to a variety of issues, including poor configured and unprotected network infrastructure [1]. These security risks can expose organisations to attackers that can misuse resources, gain unauthorised access to confidential data and introduce viruses and spyware programs. These security risks do not only put an organisation at risk, but also the individuals utilising the network as personal information can be obtained and exploited.

Networks and networking infrastructure cannot be protected when one does not know what to protect it against. Only after the risks are known, policies and plans can be developed to reduce those risks. One of the best approaches to understanding risks in a network is to attack the infrastructure of the network to attempt to breach the security. These tests, however, are destructive activities that could compromise network stability or cause service failure.

A realistic network test platform can enable researchers to study cyber security risks and relevant countermeasure strategies without compromising real live networks. Due to the lack of secure and realistic test environments, however, these research activities are difficult to perform [2]. The use of approximated networks and estimated user behaviour can lead to results that are most often questionable or incorrect altogether.

The development of a network simulation environment that is a realistic representation of a known address space can provide cyber researchers with a secure, contained and controlled environment to address the growing cyber threat. Such a platform can offer researchers the opportunity to reproduce these cyber attacks under contained and controlled circumstances, where the events can be replicated and possible countermeasures produced. In addition to threat evaluation, the network simulation environment can provide a development platform for various network-oriented tasks, such as network related product acquisition, testing and evaluation.

In order to develop an environment where realistic results can be obtained, all the relevant features, which include routing dynamics, connectivity, background traffic etc., must be included [3]. This will provide a realistic platform which is isolated, more controlled and more predictable than implementation across live networks [4].

In this paper we discuss the development of such a network simulation environment, called a network simulator (NS). The aim of the NS is to create a research and development platform, capable of emulating a small portion of the Internet address space to provide a usable environment for development, testing and evaluation of network related products and tasks.

There exist a wide range of systems developed to simulate the Internet address space. Some of these systems, also called cyber ranges, have large CPU and memory requirements. On the other hand, there exist simulators where relevant features and characteristics are simplified or estimated, compromising the fidelity of the developed environment. This paper describes the development and implementation of a network simulation environment which offers a realistic replication of acceptable fidelity of Internet address and user network space without unreasonable hardware requirements.

II. Related Work

This section discusses the various approaches taken by researchers in developing existing cyber ranges. The construction of a cyber range requires knowledge regarding features such as topology, connectivity, link bandwidth, traffic patterns and mixes as well as congestion levels. These relevant features must be obtained from real-life Internet observations and measurements which must be implemented with the right amount of detail in order to produce an accurate model of the Internet regarding scale and speed, while utilising an acceptable level of hardware resources, thus finding a balance between two extremes [3]:

1. simulating all aspects of the Internet at too fine granularity resulting in a costly project regarding CPU resources and memory usage; and
2. over simplification and heavy approximation regarding the Internet environment leading to inaccurate or incorrect results.

Our work on the network simulator is related to mainly three areas of network research, namely Internet simulation, node emulation and traffic modelling.

A. *Internet Simulation*

The United States National Cyber Range (NCR), developed by Defence Advanced Research Projects Agency (DARPA) in the USA, provides an experimentation platform which can be used to address the growing threat of cyber security incidents. The NCR is a secure, self-contained facility developed for activities which include cyber research, development of new capabilities, the analysis of malware, cyber training and exercises [2]. The NCR aims to gain knowledge and insight in order to develop technologies and procedures to strengthen cyber security.

The NCR requirements include realistic node replication regarding connections, hardware and endpoints, which includes firmware, software applications and hardware. The NCR is designed to enable a variety of node configurations as well as enabling the network to run any required protocol or service. In addition, the NCR is developed to be scalable so that anything from a single device or several thousand devices can be instantiated [2], [5]. Due to the scale of this high-fidelity environment, the hardware requirements are massive.

Boeing Intelligence and Security Systems developed a Cyber Range-in-a-Box (CRIAB) [6] for the modelling and simulation of cyber threats to develop security solutions. This developed system is advertised as a system that can be used to support the development, evaluation and experimentation of cyber tools and techniques [7]. In addition, the CRIAB is built to be scalable, enabling the capacity of a high fidelity environment without the hardware and facility requirements of large cyber ranges. The networks that can be emulated have various scales of fidelity and the modelling and network configuration can be done via one interface.

As Internet-scale security becomes increasingly important, it can be seen that researchers require tools to replicate and study the cyber threats. The abovementioned projects are just two examples of research companies developing tools to virtually replicate the Internet to support the development of cyber security.

B. *Node Characterisation*

A network node can be replicated in four different methods of varying fidelity: physical, virtual, emulated and simulated [8]. The fidelity of the node influences the number of nodes that can be created on a physical host, due to CPU and memory constraints.

A physical node can be added to the network environment by physically connecting the device to the network, which gives optimum fidelity, but requires hardware. The virtualisation of network nodes provides high fidelity nodes by partially allowing access to the host hardware which requires a substantial amount of memory and hardware resources. Emulation enables the creation of more nodes as less resources are required, but has a lower fidelity. Simulation has the lowest fidelity, but can enable the instantiation of a great deal of nodes on a physical host. The selection of software will depend on the scale and node fidelity of the cyber range to be developed.

There exist a wide variety of software that allows for the development of a natwork simulator. To replicate nodes with acceptable fidelity and hardware requirements, two categories of tools can be used [9]:

1. Simulation tools, including OPNET [10], ns-3 [11], and QualNet [12], which creates a simulation model of the network nodes and runs on a single host machine.
2. Network emulation tools, which include PlanetLab [13] and NetBed [14], which involves a physical testbed where real systems are connected for testing.

The emulator tool developed by Boeing, Common Open Research Emulator (CORE), combines these two categories by emulating the network nodes through virtualisation and simulating the network links. This combination between virtualisation and simulation enables CORE to virtualise a large number of machines on a single physical host, making it lightweight and scalable and enabling high performance and efficiency [9]. Thus CORE can provide applications running in real time on an emulated network where the hardware requirement is relatively small [9]. CORE can be used for network and protocol research, demonstrations, application and platform testing, evaluation of network scenarios, security studies and increasing the size of physical test networks [15].

C. *Traffic Modeling*

The generation of realistic user traffic is essential for the realistic simulation of a user networks connected to a portion of the Internet address space. Various researchers working on Internet simulation commented that statistically representative Internet communication patterns are lacking [16], [3] and the process of measuring network traffic and then generating synthetic Internet communication patterns can be complex and time-consuming. In addition, the model must be updated when the protocols or applications change [16].

BreakingPoint devices [17] allow for detailed simulation of various aspects of network traffic. It offers cyber tomography machines (CTM) which are network simulation devices that simulate users browsing, emailing, texting, talking, and spreading malware. BreakingPoint also includes an extensive library of applications that can be simulated, including Facebook and Youtube, which is updated and maintained on a regular basis [17]. A Markov chain approach is used to increase the level of realism in user simulation and dynamic recreation of real world traffic flow in computer networks.

III. DESIGN

Instead of attempting to simulate or emulate the Internet as a whole, the network simulator design focuses on emulating the behaviour of users inside a corporate network with traffic from within the network to the Internet and vice versa, emulated using dedicated hardware. The design include entry points into the emulator to connect external physical networks for functional testing. Figure 1 illustrates the conceptual design of the NS. A discussion on the various aspects of the NS is discussed subsequently.

A. *Topology*

It can be seen from Figure 1 that the developed NS comprises of multiple network sections, namely Internal User Networks/Local Area Networks (LANs) connected to foreign networks/the Internet through a demilitarised zone (DMZ).

1) Internal User Network/ Local Area Network

The NS enables a sandbox environment where clients' networks can be replicated from scratch, new topologies created or existing/default networks changed. The default topology of each Internal User Network space are simplified topologies from real-life operational network environments.

These topologies are designed in order to accurately represent realistic real-life networks for the NS. The user network topologies can be described in terms of topological importance [18], [19]. The topological importance of nodes can be described through two basic metrics, namely node degree and node eccentricity [18]:

1. Degree of node - captures the quantity of the node's neighbours. The degree of a node is equal to the number of incident edges of the node. The higher the degree, the higher the importance of the node.

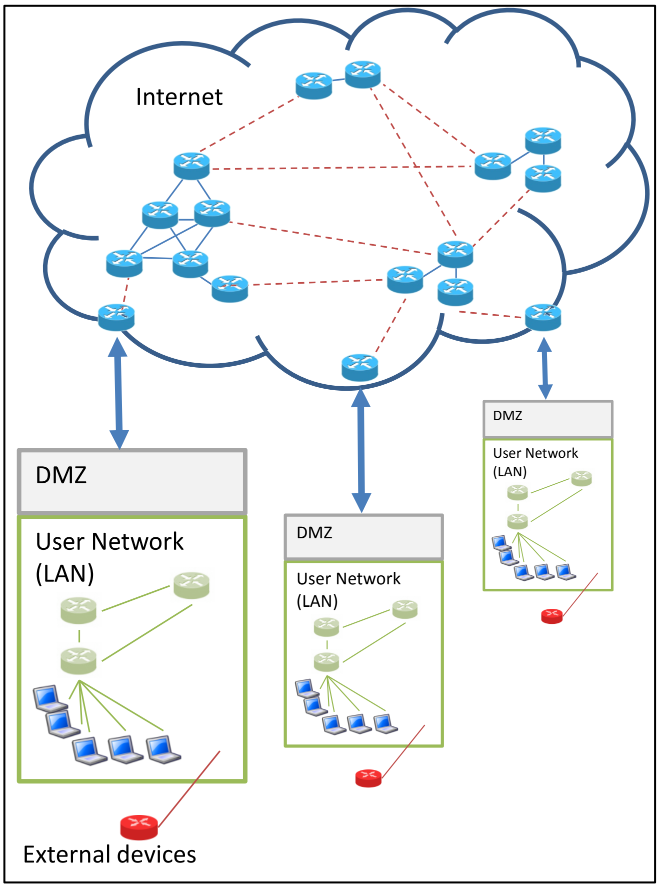

Figure 1: NS conceptual design

2. Node eccentricity - minimum number of hops required to reach at least 90% of the nodes in the network. The lower the eccentricity, the higher the importance of the node.

The Internal User Network topologies used for the NS contain both nodes with high degrees, and several nodes have a low eccentricity as most of the network nodes can be reached with a low number of hops. According to our definition, these nodes are of topological importance and described in [19] as the core of nodes that from a "clique" in the centre if the network. The nodes further from the central clique mostly have a higher eccentricity and in some cases a lower degree, which can be seen as nodes of lower importance.

The above description of an Internal User Network is used to create multiple realistic, representative Internal User Networks. At the edge of each network connected to the Internet address space, or backbone, an edge router can be found. This router communicates to its provider's autonomous system (AS).

2) DMZ

A DMZ can be seen as a subnetwork, consisting of a router or collection of routers that sits between firewalls or off one leg of a firewall. The DMZ acts as a middle ground between the trusted, protected internal network and the untrusted, external network, like the Internet [20], [21], [22]. The default DMZs through which the Internal User networks communicate to the Internet can be altered or built from scratch.

3) Internet

The Internet address space portion that is emulated in the NS can be deconstructed into subnetworks which are under different administration authorities, called Autonomous Systems (ASes) [23], [24]. The various ASes contain border routers which interconnect ASes to each other and to user networks or LANs. The autonomous systems also contain core routers which functions as a backbone to route traffic between edge routers. Figure 2 shows the structure of the emulated Internet address space at router level.

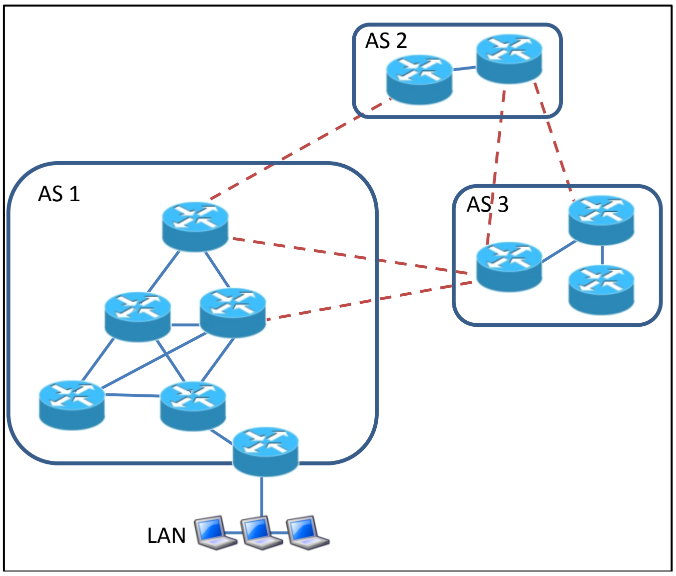

Figure 2: Internet structure overview

B. *Node Emulation*

A single server/ host machine is not able to instantiate all the virtual nodes that must be created for the NS, due to hardware limitations. CORE, described in Section II B, enables the seamless construction of a network over multiple hosts, where each host independently emulates a section of the network. The topologies are configured on the separate host machines which are then connected via physical network cables to allow the seamless transmission of data between nodes on different host machines [9].

For the NS, CORE was implemented on a Linux platform. The Linux CORE utilises the Linux network namespace virtualisation to build the virtual nodes, where all network namespaces share the same file system in CORE. Linux network namespaces are the primary virtualisation technique utilised by CORE. CORE combines these namespaces to form virtual networks through the use of Linux Ethernet bridging [15].

Network topologies can be created through the graphical user interface (GUI). The user is presented with a blank canvas where nodes of various types and functions can be placed and linked. An example of a created network topology is shown in Figure 3. It can be seen that the created network in Figure 3 contains routers, a range of end-devices, as well as physical ports. Each physical port placed on the canvas can be configured to link to a physical Ethernet port on the host machine which can be connected to an external hardware device or network on another host. In Figure 3 it can be seen that the emulated network is connected to two external devices: "Server2" hosting another network and a physical laptop, "USER LAPTOP".

When the network environment is created, it can be switched from editing to execution mode where the network, including all routers and end-devices, are virtualised and emulated. The user can double click on any node to obtain a Unix shell on that virtual node to invoke commands in real-time. The physical ports placed on the canvas allow real-time connectivity between the virtual nodes inside the real-time emulation and external devices or networks. This feature allows the connection of external physical nodes to the SSIS.

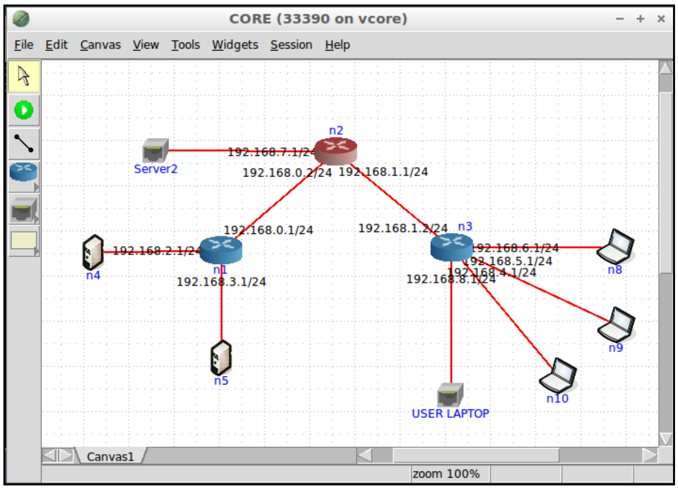

Figure 3: CORE user interface

C. *Routing and Forwarding*

Routing can be described as the process of moving packets over a network from the source node to the receiver node [25]. The path from source to receiver node can consist of a single, direct connection or a series of hops through routers and other network devices.

In order to obtain an accurate NS model, the routing and forwarding over the Internet address space, DMZ and Internal User Network must be accurately constructed. The routing protocol selected for the network devices in the NS is able to select the best routes around a network based on predefined metrics, such as cost and hop count, and is able to prevent loops from forming [26].

For the Internet portion of the NS, each AS were set up to route traffic inside the AS and between different ASes. Different intra-routing protocols are employed within each AS, like Open Shortest Path First (OSPF) and Internal Border Gateway Protocol (iBGP), where routing between ASes are implemented by an inter-routing protocol, External Border Routing Protocol (eBGP) [24]. The routing setup within each Internal User Network is dependent on the topology of the designed LAN. In most cases OSPF and Routing Information Protocol (RIP) are employed to route data between the end-users and the DMZ. As stated in Section III A 1, each LAN is connected to its AS with an edge router, where BGP routing is implemented.

D. *User traffic*

Realistic user traffic is generated through the use of BreakingPoint. The traffic profile used generates realistic application traffic in a distribution representative of an enterprise network. Due to the fact that the user traffic is generated on a separate hardware platform of the BreakingPoint device, a program was written to enable the incorporation of the BreakingPoint generated traffic to CORE. This program allows the generated traffic of each end-user to flow directly from the emulated end-user to the specific destination.

IV. Performance Testing

This section presents the benchmarking tests performed on the NS hardware platforms running CORE. Benchmarking testing is used to measure and evaluate the performance of the physical machines running a well defined workload [27]. Thus the benchmarking tests are used to measure the limiting performance metrics for a selected set of tests with varying test parameters [27], [28]. According to [9], the performance of CORE is largely hardware and scenario dependent.

This section details the benchmark testing of the CORE emulation software on various servers used in the NS. The hardware acquired for the NS includes:

1. DELL R320: 2.3GHz, 16 logical CPU core servers
2. DELL R420: 1.9GHz, 24 logical CPU core servers
3. Virtual machines:

a. 2.6GHz dual core

b. 1.9GHz dual core

A. Test Objectives

As CORE is hardware dependent, hardware platforms with different CPU frequencies and CPU cores are utilised for the benchmarking tests. The acquisition of these performance characteristics are used to ensure that the emulated network of the NS is efficiently distributed over the multiple physical machines, eliminating possible bottlenecks in the system.

The objectives of the benchmark testing is to determine the scalability of CORE on a server with regards to the following metrics:

1. CPU frequency
2. CPU utilisation
3. Hop count

B. Test Environment

The performance testing is done on the hardware platforms as described in Section IV. On each host device, a single CORE emulation is run consisting of routers connected in a chain. This topology is shown in Figure 4 and represents the worst case routing scenario where each transmitted packet are routed via all the routers in the network (number of hops).

Each emulation also runs the Iperf utility. Iperf is a network performance measurement tool which can create UDP and TCP data streams and measure the throughput of the network over which the data streams are transmitted [29]. Iperf has a client and a server functionality, where the throughput between the two are measured. The first and the last router in the chain will be will host the Iperf server and client, respectively, as shown in Figure 4. The Iperf client generates TCP packets to transmit over the chain of emulated routers to the Iperf server, running the Iperf benchmarking utility.

C. Test methodology

To determine the performance of the system in terms of the abovementioned metrics, the test methodology, based on that in [9], [30], will be implemented. The network configuration for the test setup shown in Figure 4 are the following:

1. The number of routers configured in a chain topology varies from 2 to 50.

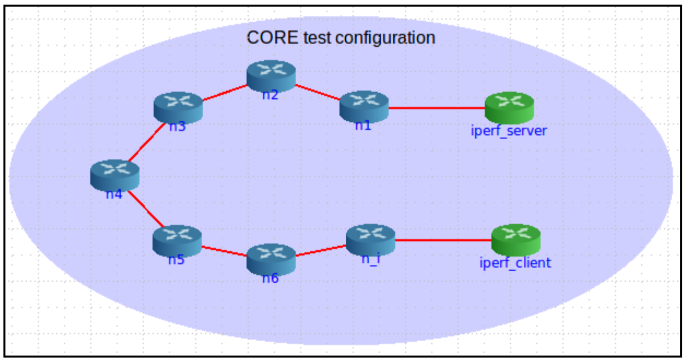

Figure 4: Test Configuration

2. Virtual links between CORE routers are configured with no link restrictions, including bandwidth and delay.
3. CORE routers are running the default Quagga routing suite configured with OSPFv2 and OSPFv3 routing.

Test parameters in Iperf:

1. The TCP packets are transmitted as fast as possible.
2. Data is transmitted for 60 seconds and the test iterated 10 times.
3. For each set of iterations, the maximum transmission unit (MTU) value, which determines the size of the transmitted packets, are set to different values: 1446, 1052, 156.

D. Results

1) Hop count

For the first test, the maximum throughput available for a TCP application are measured. For each test iteration, the number of routers in the chain (hops) are increased from 2 to 50 and the throughput is measured. These tests are run on the R320 and R420 servers.

Figure 5 shows the resulting average throughput measurements, in Mbps, of the R320 and R420 servers in solid and dotted lines, respectively.

It can be seen that a throughput of approximately 1000 MB over 5 hops can be sustained with a segment size between 1446 MB and 1052 MB. When the server is emulating 50 nodes, the network is able to sustain the routing of approximately 100 Mbps of data.

2) CPU Frequency

For the second set of tests, the average throughput over a 10-hop CORE network was determined for each of the four hardware platforms mentioned in Section IV.

Figure 6 shows the resulting average throughput measurements in Mbps for a single Iperf session. The dotted lines represent the throughput measurements of the two dual core machines and the solid lines that of the two servers. The results in Figure 6 show that the throughput performance of a CORE network is influenced by the CPU frequency of the

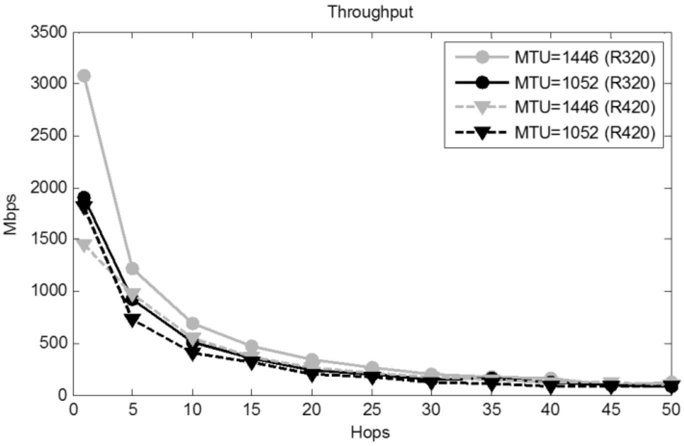

Figure 5: Iperf measured throughput

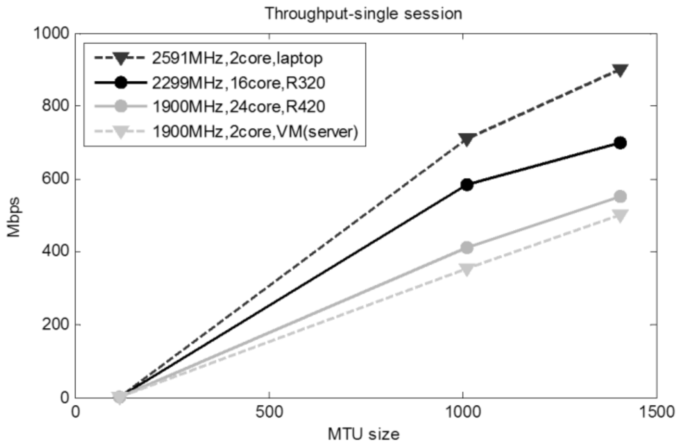

Figure 6: Iperf measured throughput for a single session

hardware platform. The throughput performance of a 10-hop network is the highest when running on a hardware platform with the highest CPU frequency, which is the dual-core virtual machine running on a 2.591MHz laptop.

It can be seen from Figure 7 that the performance of 2 parallel sessions follows the same trend as the performance of a single Iperf session. As in the first test, the CORE network run on the host machine with the highest CPU frequency has the best throughput performance.

The relationship of the throughput results, shown in Figure 7, and the CPU frequency of the hardware platforms are shown in Figure 8. The graph in Figure 8 shows that the throughput obtained from a CORE network scales to the CPU frequency of the hardware platform. From these results it can be determined that CORE utilises a single CPU core for each networking session run. Due to the fact that all the hardware platforms contain at least two CPU cores, CORE was able to run two parallel sessions without constraint, limited by the CPU frequency of the hardware platform.

3) CPU utilisation

To determine the CPU utilisation of CORE, we ran 8 parallel Iperf sessions on each hardware platform. The total throughput obtained from each CORE network are shown in Figure 9.

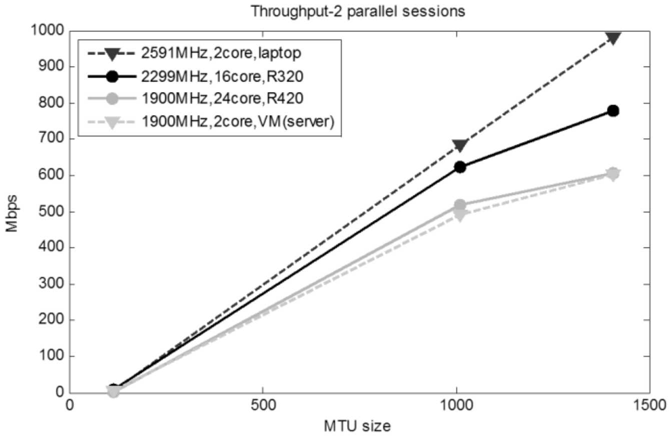

Figure 7: Iperf measured throughput for 2 parallel sessions

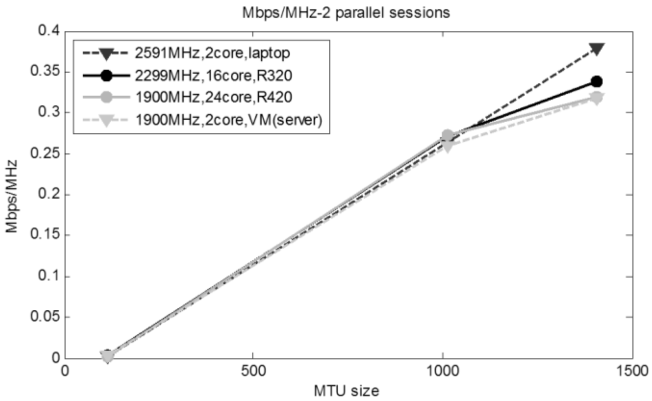

Figure 8: Mbps/MHz relationship of two parallel Iperf sessions

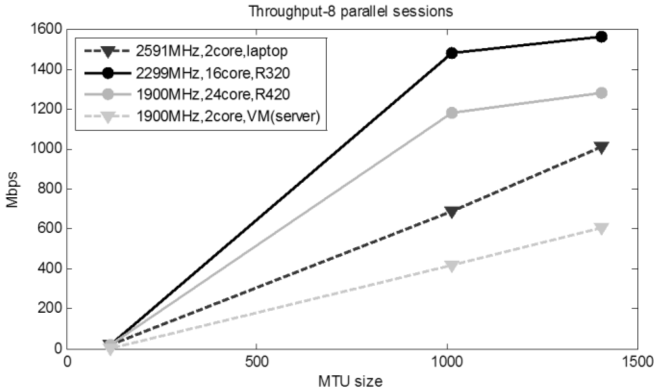

Figure 9: Iperf measured throughput for 8 parallel sessions

It can be seen from Figure 9 that the overall throughput measured on the Dell servers, containing 16 and 24 CPU cores respectively, outperforms the two dual-core hardware platforms. This behaviour differs from the result obtained for a single and two parallel Iperf sessions shown in Figure 6 and Figure 7, respectively.

It can be seen that the overall throughput of the two dual-core platforms did not increase from two to eight parallel sessions. The overall throughput of the two servers, however, continued to increase as the number of parallel sessions increased. The change in the overall throughput for various Iperf parallel sessions, with an MTU-size of 1446 are shown in Figure 10.

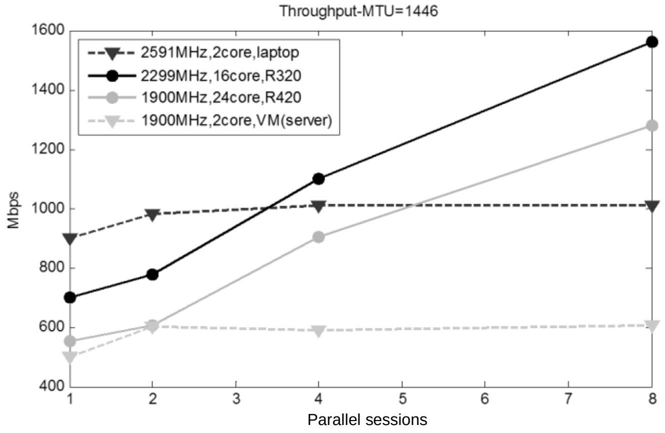

Figure 10: Iperf measured throughput for parallel sessions

From the results obtained, it can be seen that the performance of multiple CORE sessions are dependent on the number of CPU cores available for utilisation. For each networking session performed by CORE, a single CPU core is utilised. Thus the number of parallel CORE sessions that can be run without constraint, is limited to the number of CPU cores available.

4) Discussion

From the benchmark tests run in the above section it can be determined that the limiting factor of this system the processor speed, as confirmed in [9]. In addition, the number of CPU cores of the host machine determine the number of parallel networking sessions that can be run without constraint.

These tests, however, can only be seen as an approximation of actual network scenarios, as the different processes run on the routers would also influence the performance of the CORE network.

V. Conclusion

In this paper we discussed the design and development of a network simulator. The conceptual network simulator model was discussed and detail was given on what techniques, software and hardware was used to enable the realistic construction of network topologies, emulation of end-users as well as traffic generation and routing setup. The performance tests results provides information regarding the bottlenecks of the system, which is important to consider when networks are constructed and virtualised.

The network simulator provides a controlled real-life environment that allows networks to be emulated through a mixture of physical and virtual devices. This virtual environment allows for the results of tests to be viewed in real-time, recorded and analysed to improve the overall resilience of the target network, software or device. In addition, the network simulator can be used as an interactive environment to train and educate users in computer and network security related tasks.

The network simulator will enable corporations to adopt a comprehensive approach to network security through preventative threat mitigation solutions for the securing of hardware, applications, systems and networks, forming an integral part in the evaluation and improvement of South Africa's corporate network infrastructure.

References

[1] IDGCONNECT, “South Africa: Spotlight,” *Africa 2013: Cyber-Crime, Hacking and Malware,* pp. pp10-11, 2012.

[2] Lockheed, Martin and Wynstone, National Cyber Range, Flexible Automated Cyber Test Range (FACTR), 2012.

[3] S. Wei and J. Mirkovic, “A realistic simulation of internet-scale events,” in Proceedings of the 1st international conference on Performance evaluation methodolgies and tools, New York, NY, USA, 2006.

[4] P. Sanaga, J. Duerig, R. Ricci and J. Lepreau, “Modeling and emulation of internet paths,” in Proceedings of the 6th USENIX symposium on Networked systems design and implementation, Berkeley, CA, USA, 2009.

[5] G. Warner, What does a National Cyber Range do?, 2009.

[6] B. O’Donnell, “CRIAB: Powerful Cyber Personnel Training Solution,” The Boeing Company, 2013.

[7] D. Garlick, “Game On. Huntington Beach lab test scenarios to help defend agains cyberattacks,” Frontiers, Vols. XI, , Issue VIII, pp. 18-19, 2012.

[8] J. Liu, R. Rangaswami and M. Zhao, “Model-driven network emulation with virtual time machine,” in Proceedings of the Winter Simulation Conference, 2010.

[9] J. Ahrenholz, C. Danilov, T. R. Henderson and J. H. Kim, “CORE: A real-time network emulator,” in Military Communications Conference, 2008. MILCOM 2008. IEEE, 2008.

[10] “OPNET Modeler: Scalable Network Simulation,” OPNET, [Online]. Available: http://www.opnet.com/. [Accessed 10 July 2013].

[11] “ns-3 Project,” ns-3, 2012. [Online]. Available: http://www.nsnam.org/. [Accessed 10 July 2013].

[12] Scalable Network Technologies: QualNet Developer”, 2013.

[13] “Planetlab. An open platform for developing, deploying, and accessing planetary-scale services,” PlanetLab, 2007. [Online]. Available: http://www.planet-lab.org/. [Accessed 11 June 2013].

[14] “Emulab - Network Emulation Testbed Home,” Netbed, 2013. [Online]. Available: http://boss.netbed.icics.ubc.ca/. [Accessed 14 June 2013].

[15] “CORE Online Manual,” coreemu: Common Open Research Emulator, 2012. [Online]. Available: http://pf.itd.nrl.navy.mil/core/core-html/. [Accessed 24 March 2013].

[16] M. C. Weigle, P. Adurthi, F. Hern\andez-Campos, K. Jeffay and F. D. Smith, “Tmix: a tool for generating realistic TCP application workloads in ns-2,” SIGCOMM Comput. Commun. Rev., vol. 36, no. 3, pp. 65-76, #jul# 2006.

[17] “BreakingPoint 2.0 User Guide, Release 3.0”.

[18] G. Siganos, S. L. Tauro and M. Faloutsos, “Jellyfish: A conceptual model for the as Internet topology.,” Journal of Communications and Networks, vol. 8, no. 3, pp. 339-350, 2006.

[19] S. L. Tauro, C. Palmer, G. Siganos and M. Faloutsos, “A simple conceptual model for the Internet topology,” in Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE, 2001.

[20] Cisco, “Cisco Craft Works Interface User Guide,” Cisco IOS XR Software Release 3.5, 2007.

[21] M. Murfitt, “The case for a Tiered Internal Network,” Network Security, pp. 17-21, 2006.

[22] E. Cohen, S. Cohen and M. Schrul, Norton Personal Firlewall 2005: Advantages and Disadvantages, Florida: Florida Atlantic University, 2005.

[23] L. Gao, “On inferring autonomous system relationships in the Internet,” Networking, IEEE/ACM Transactions on, vol. 9, no. 6, pp. 733-745, 2001.

[24] Y. He, G. Siganos and M. Faloutsos, “Internet Topology,” in Encyclopedia of Complexity and Systems Science, R. A. Meyers, Ed., Springer New York, 2009, pp. 4930-4947.

[25] F. T. Leighton, B. M. Maggs and S. Rao, “Packet Routing and Job-Shop Scheduling in O(Congestion + Dilation) Steps,” Combinatorica, vol. 14, no. 2, pp. 167-186, 1994.

[26] F. Baker, “RFC 1812 Requirements for IP Version 4 Routers,” 1995.

[27] U. Krishnaswamy and I. D. Scherson, “A Framework for Computer Performance Evaluation Using Benchmark Sets,” IEEE Trans. Comput., vol. 49, no. 12, pp. 1325-1338, #dec# 2000.

[28] D. A. Menasce and V. Almeida, Capacity Planning for Web Services: metrics, models, and methods, 1st ed., Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[29] “Iperf,” NLANR/DAST, 2012. [Online]. Available: http://iperf.sourceforge.net/. [Accessed 10 July 2013].

[30] J. Ahrenholz, “Comparison of CORE network emulation platforms,” in MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010, 2010.

[31] S. Wei and J. Mirkovic, “A realistic simulation of internet-scale events,”

in Proceedings of the 1st international conference on Performance evaluation methodolgies and tools, New York, NY, USA, 2006.

[32] E. Condon, E. Cummins, Z. Afoulki and M. Cukier, “How secure are networked office devices?,” IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), vol. 0, pp. 465-472, 2011.