

Intuitive Real-Time Network Monitoring Using Visually Orthogonal 3D Metaphors

Warren Harrop, Grenville Armitage
Centre for Advanced Internet Architectures.
Swinburne University of Technology
Melbourne, Australia
{wazz, garmitage}@swin.edu.au

Abstract- With denial of service attacks and self propagating network viruses adding to today's network conditions, managing edge networks with the intention of providing a quality service requires vigilance by highly skilled network administrators. We believe the use of intuitive, interactive 3D representations of real-time network state information will improve the ability of non-specialist operations staff to understand and react to anomalous network behavior. We define a set of visual metaphors that represent dynamic activity metrics unique to IP networks in an easy to interpret manner. Our prototype software, 3VEN, implements many of the IP metrics to visual metaphor mappings suggested, using data collected from a darknet source.

Keywords- IP, Network Monitoring, Intrusion Detection, IDS, Darknets, Honeypots, Visualisation, 3D, OpenGL, Real-time

I. INTRODUCTION

One of the main challenges facing Internet Service Providers (ISPs) and other IP network operators today is the provision of robust, resilient and unobtrusive service in the face of external attacks and destructive actions by systems within their networks [1]. A key component of this process is the development of more cost-effective and accurate approaches to network monitoring and management.

In this paper we focus on a subset of the overall network management problem space. Our specific interest involves tools and techniques for detection, recognition and isolation of traffic that represents (or could represent) malicious attempts to undermine an IP network or the hosts attached to an IP network. Such traffic is becoming more and more prevalent - viruses, worms, and trojans exploit both technical vulnerabilities in network-attached hosts and socially-engineered vulnerabilities in the end-users, creating huge network loads (and repair bills) along the way [2]. In addition, human-directed denial-of-service (DoS) attacks have become a fact of life for network users. Reacting quickly to DoS attacks to minimise their impact on customers has become a fact of life for network operators.

Network operators face a technological and financial challenge. Current network management tools are only gradually evolving to provide (near) real-time, detailed analysis of network traffic (e.g. [3][4]), and yet the use of these tools often requires highly skilled employees. Employees cost money, and highly skilled employees both cost more and are harder to replace. They represent

the human central-point-of-failure in a network management system. A challenge for researchers in the network management field is to reduce the level of specialised knowledge required by an employee for that employee to become a valuable member of a network operations team.

It is our belief that next generation network monitoring and management tools must leverage visual metaphors that are familiar to relatively untrained and unskilled humans. In this paper we outline our preliminary design goals and decisions for a real-time network monitoring and management toolkit based around an interactive, three dimensional (3D) immersive environment (a 'virtual world'). Our immersive environment is filled with animated 3D objects whose visual characteristics and real-time behaviours coherently represent the current state of a network or of entities within a network.

The visual metaphors are not for fine-grained discrimination of network issues. They do however enable quick recognition of relevant, coarse-grained events occurring on a network. After visual detection of a network issue the operator could then, for example, move towards anomalous objects in the immersive world. This would reveal more information about the issue and if required, the operator could request the display of text based information giving more traditional and complex network statistics. This allows for detailed analysis by the skilled network administrator or allows the unskilled operator to pass on and escalate an issue they have found.

We have been inspired both by previous work on interactive system management using 3D game metaphors (e.g. [5]) and efforts at non-realtime rendering of network state in 3D space (e.g. [6]). Our proposal is a combination and evolution of these ideas, made practical by the fact that tools (both software and hardware) for rendering three-dimensional objects in real-time are now both inexpensive and powerful in consumer-grade, off the shelf equipment.

This paper continues in section II with an overview of existing network monitoring techniques and tools. Section III then describes previous work on network visualisation and outlines the context within which our visual metaphors will sit. In section IV we define our visual metaphors and present our proposed

implementation in section V. Future work is discussed in section VI before we conclude in section VII.

II. EXISTING NETWORK MONITORING SOLUTIONS

A. Network Intrusion Detection

Network Intrusion Detection Systems (NIDS) are tools that listen passively and promiscuously to network data, monitoring for anomalous activity. To be most effective, software of this type is generally placed at a privileged point on a network, but can also be run on a host by host basis depending on requirements. Careful analysis of the output of a NIDS setup can enable skilled administrators to take preventative measures on possible network issues, before the problems escalate to the point where they affect network hosts.

The technical methods of implementing an NIDS are as varied as the networks they monitor. When it comes to matters of security, network operators will often extensively customise their systems. It is worth outlining the three main methods of NIDS - event based, darknets and honeypots. Real networks will often deploy a mixture of these techniques which have been implemented by a number of different open and closed source software packages.

B. Event-based detection

As the name implies, event based pieces of NIDS software are given fingerprints of network activity that are considered noteworthy and will create alert events when these fingerprints are matched. By intercepting traffic at the lower network layers and implementing the proper fingerprints, it allows for the detection of attacks that range from the network layer to the application layer. Snort [4] and Bro [3] are both popular open source packages capable of implementing event based NIDS.

C. Darknets

Darknets¹, also called network telescopes, are a form of NIDS that are made up of large continuous portions of unused IP address space. The IP addresses are still valid and routable, but any packets destined for the nonexistent hosts in the darknet range are silently captured. This can be achieved in a number of different ways technically (including using a program such as Bro), but all methods achieve the same outcome – logs of attempted connections to nonexistent hosts and “Internet backscatter” [7] (often the result of denial of service attacks occurring elsewhere on the Internet, where the attack packets use forged source addresses from the darknet address space).

Because darknets occupy space where no valid hosts exist, it can be assumed that all but a minuscule percentage of the inbound packets are of a dubious nature. Mis-configured clients, or mistyped URLs, do cause a small level of inbound traffic that is harmless, but the vast majority of packets are scans by other hosts attempting to locate hosts and services to infect through

¹The term Darknet has come to have two meanings in recent times, both “a network used for sinister, illegal or secretive purposes” and “undetectable monitoring network”. Here we refer to Darknet in the latter sense.

known exploits. Armed with this knowledge, darknet logs can be analysed to reveal information about potential and on-going network attacks.

D. Honeypots

Honeypots go one step beyond a darknet by actually implementing controlled instances of the network services that attackers typically look for on a foreign network. Instead of just passively monitoring a network's activity the honeypot NIDS attempts to attract those who wish to gain unauthorised access to network hosts. An open-source example of a honeypot toolkit is Honeyd [8] (released under the GPL).

Honeypots have a major advantage over the passive monitoring that a Darknet provides. By allowing an attack to progress a number of stages after the initial connection request, much more information can be gathered on the purpose and intent of the attacker. Whether or not this is information that the average system administrator is interested in is another matter as the intent of an intrusion may be of no concern, only the knowledge that an exploit is being tested.

While functional, all of the aforementioned systems gather information that still needs correct representation and interpretation before it can be acted upon. How this has been achieved in the past varies greatly.

III. VISUALISATION OF NETWORK STATE

A. Prior work

There are many existing tools that implement network state visualisation, but few of these have implemented anything that would meet our goals for a system allowing relatively unskilled individuals to glean network status at a glance. Visualisation of network state is, generally speaking, only approached from two angles: static visualisation of network structure being the goal [9], or simple dynamic visualisation of network monitoring data.

When static visualisation of network structure is the project goal, a scientific visualisation approach is often taken. Many areas of research create complex data sets that have value added when inspected visually and much research has gone into visualising many types of data sets in both 2 and 3 dimensions. Visualisation of data networks falls in this category and is often attempted as an exercise in scientific visualisation.

When this approach is taken, visualisations concentrate only on the representation of the static physical or logical layers of a data network. While this can be useful in design and provisioning, it is of limited use in presenting the dynamic transient nature of a running network. So while the representations of the networks may be highly visual and informative (even interactive), the data making them up is unchanging.

Approaching from the other direction, many of the open source and commercial tools that are capable of detailed live monitoring of a network only ever provide a minimal amount of visualisation. Most tools collect data about network state via SNMP [10] and (at best) display this information on a simple 2 dimensional

graphic of network topology. This style of implementation generally does not lend itself well to gleaning a detailed network status at a glance.

An extremely common tool to provide history information about link status is Multi Router Traffic Grapher MRTG [11] or its more flexible counterpart RRD [12]. These tools provide no visual information on network topology, but can provide graphs of network metrics over time, overlaid upon each other for comparison if necessary. When network monitoring packages are created, their visualisation components are usually, again at best, a combination of 2D network representations with time based graphs.

In the realm of 3D visualisation of NIDS data, a small number of groups have made moves to represent various metrics of network state in three dimensions. In [13], software that combines the visualisation of recorded NIDS data with haptic interaction is introduced. In [14], general models for mapping network metrics to virtual world objects are introduced with prototype software and [15] introduces a framework for designing systems that visualise intrusion detection data, based on the feedback of human intrusion detection analysts.

At the present time, there has been limited exploration into live three dimensional representations of IP networks. Nagios, [16] a popular open source network monitoring tool contains a 3 dimensional visualisation component but this feature is implemented in VRML and is really just a 3D expansion of the same features available in 2D.

The most unique visualisation of NIDS data currently is the "Spinning cube of potential doom" (SCPD)[6]. For each packet received into its associated Darknet, the SCPD places a dot within a 3 dimensional cube, where the axes relate to source IP address, destination IP address and port number. The dots representing packets have a lifetime, so that portscans of many different kinds can be easily detected by eye. Vertical lines appear representing a port scan on a single host, flat 2 dimensional planes appear for port scans upon multiple continuous host IP addresses, other port scans trying to avoid detection produce spiral "Barber Pole" like patterns. Even in this software's early development stages, it is possible to leverage the human ability at pattern recognition to detect port scans that can elude certain NIDS software.

B. Context for our proposal

The current problems that the network administrator faces in keeping a network running often leads to the use of NIDS that allow for detection of anomalous network activity, followed by an automated response. While these system's response time is quick enough to stop even the most prevalent Internet worm, it comes at a cost. It is a trade off between a quick, limited intelligence response prone to false positives or an intelligent, all be it slower, human response.

What are the major factors that separate the more desirable human response from an automated one? The cycle that an automated NIDS goes through includes

detection, interpretation and reaction. With human involvement this expands to detection, representation, interpretation and reaction. For a human response to be an effective response, the representation of detected data is all important. We believe that this is one of the keys to the effective human response to network issues.

We believe that through the use of unique 3D visual metaphors, monitoring and (even controlling) of IP data networks can be taken away from the box centric view most administrators are currently forced to have of their network. With the right visual metaphors, administration need not even be confined to the highly trained network specialist any longer. Intuitive interfaces and metaphors can leverage human's natural spatial and pattern recognition abilities to speed the human reaction to anomalous network activity. In short, if the representation stage of the human reaction cycle is of high quality, the interpretation and reaction stages will be much more timely and effective.

Implementation of complex, live, 3D representations of networks has been possible for quite some time due to the availability of cheap 3D hardware acceleration and software interfaces such as OpenGL [17] and DirectX [18]. We have chosen to utilise OpenGL because of its cross-platform potential (e.g. being able to run under Windows, Linux, FreeBSD, etc...) Using these tools we are able to implement visualisations of network metrics collected from various sources that update in real time.

IV. VISUAL METAPHORS

Today's real time 3D rendering technology gives us a huge number of visual metaphors that collected network metrics can be mapped onto. The possibilities are near infinite, but we define relatively simple, low polygon count objects as a starting point to minimise the resources needed for implementation and display.

A. Being visually orthogonal

A key goal for our visual metaphors is that they be 'visually orthogonal' - in other words, the visual metaphors for distinct network metrics should themselves be visually distinct. The value of using visually orthogonal metaphors is that we can leverage the typical human's ability to quickly develop intuitive responses to information presented in three-dimensional form. Our choice of visual metaphors should also bear some intuitive (or easily learned) relationship to the underlying network metrics that each metaphor will represent.

An example of visual orthogonality would be shape and rotation - given a suitably restricted set of 3D shapes we would be extremely unlikely to confuse one shape with another just because it was rotating. (Clearly we would choose shapes that had distinct profiles at different angles, such that the visual orthogonality was met.) Effective mapping of metaphor to underlying network metric can be illustrated with a counter-example. Imagine mapping IP packet type (e.g. TCP, UDP or ICMP) to (a) object size or (b) object colour. Our ability to recognise distinct values in a colour space is usually far better than our ability to correctly identify the absolute size of an object being rendered into a 3D

space on screen. Thus mapping (b) would be superior to (a).

B. Our first cut at appropriate visual metaphors

Given the preceding design goals we have established a preliminary mapping of visual metaphors to commonly important network metrics (Table 1). Our first demonstration is in the context of a small darknet established on our university network and exposed to the outside world. Later work will enable the output from NIDS such as Snort or Bro to be mapped and displayed.

<i>Visual Metaphor</i>	<i>Network Metric</i>
Location	IP address, port number
Shape	Representation of object type (subnet, host or connection)
Size	Time aggregate of unique connections
Colour	IP packet content type
Alpha channel	Time since packet arrival
Rotational Velocity	Throughput

Table 1 Mapping visual metaphors to network metrics

We chose to map the following characteristics of 3 dimensional objects onto network metrics - location, shape, size, color, alpha channel, and rotational velocity. The network metrics being represented are packets per second, source and destination port, IP address, protocol and unique connections.

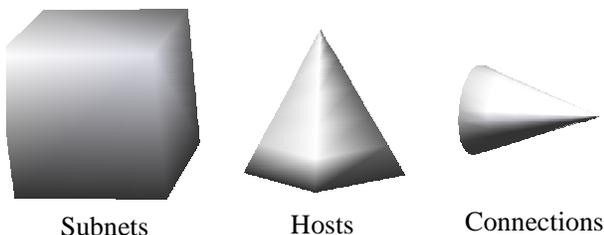


Figure 1 3D objects representing subnets, hosts and connections

Figure 1 shows the three “primitive” shapes chosen to distinctly represent (sub)networks (“subnets”), hosts and connections - squares, pyramids and cones respectively. We chose not to use circles or globes because it is visually difficult to ascertain whether such an object is spinning without applying textures. We are still considering the use of textures to convey additional network state information.

As mentioned earlier, points in a colour space are usually easy for people to distinguish when the points are far apart (e.g. bright red vs blue) and there are not too many distinct values required. Thus we are using colour to represent IP packet type - it efficiently encodes a small range of packet types without consuming additional visual real-estate on screen.

Reducing alpha channel (transparency) is easily recognisable as analogous to the “fading out” or death of an object. For trends to become apparent in the

visualisation, connections (or partial connections) cannot simply flash up in the moments they occur and disappear instantaneously. Connection objects linger, fading out over a user defined time that gives the best possible chance of detecting trends that may be present.

We link the number of unique incoming connections to an object's size and the rate of arrival of packets to the objects spin rate. We are not usually interested in the precise number of incoming connections, thus an imprecise relative metaphor such as object size is acceptable. Object size also conveys an intuitive sense of “weight” or “gravity”, which is easy to relate to the “work load” implied by lots of connections coming in from a wide variety of remote IP addresses. The orthogonal related metric is how rapidly IP packets are arriving - spin rate of each object maps intuitively to something like “speed of packet arrival”.

Putting these two mappings together on an object means a large item spinning slowly is an object serving a large number of unique connections, but at a slow data rate. A small object spinning quickly is an object serving a small number of hosts a large amount of data. A large object spinning fast, most likely needs your attention.

C. Representing network hierarchy through aggregation

For the overall structure of the visualisation, we propose a three-tier approach that is akin the hierarchy already found in IP networks.

At the the highest level, visual objects represent entire subnets. Virtual world camera movement allows the controller to move within these objects representing networks and view further objects that represent the individual hosts that comprise that particular subnet. Further movement into these host objects allows the individual connections of the host to be viewed.

The three tier approach allows network health to be gleaned quickly at any level. It gives the ability to quickly expose details not present in the top level views by moving in, or just as quickly allows movement out back to a view of the entire network.

When using this three tier representation, we attempt to preserve as much consistency as possible at each level of the hierarchy. When viewing an object representing an entire subnet, size is akin to the unique connections leaving and entering that subnet, the same as when viewing an object representing a host.

D. Non-linear scaling of each visual axis

It is clear that most of the network metrics mentioned do not lend themselves well to a simple linear graphing. For example, consider the range of possible TCP or UDP port numbers. If shown graphically in a linear manner, the most active range of ports, the reserved ports 1-1024, only take up around 1.5% of the possible range. A visualisation where the majority of your data is displayed in 1.5% of the available viewing area is not optimal. One solution would be to use static nonlinear scales on this sort of metric. Unfortunately, predefined non-linear mappings are unlikely to be generally useful in all cases.

Consider the display of throughput mapped to spin rate. For a particular host, above a certain user definable data rate the network administrator is no longer concerned with the exact rate at which packets are impacting on the network object, only that action needs to be taken to rectify a problem. It would be useless to continue accelerating the object past a certain point, no extra information is being given. This exact 'panic' level will vary from object to object but should not vary greatly with time. In this case a user definable nonlinear scale set at program startup would suffice.

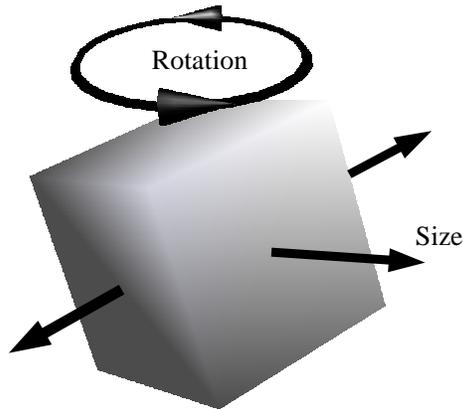


Figure 2 A subnet spinning slightly off axis to accentuate the motion

However this is still somewhat inflexible. It does not easily allow for metrics where a prior knowledge about a range of interest is not known and will not be known until well into program runtime.

For example, for much of the time the vast majority of the possible IPv4 address space is of little concern to the network administrator, but if an attack were to originate from a particular remote /24 subnet involving multiple hosts, it is less than helpful if the display scale of the visualisation is linear. The fact that multiple hosts are involved in the attack is hidden because the representing objects are effectively drawn on top of each other. The subnet of the attacking hosts was not known prior to the commencement of the attack so static nonlinear scales would have been of little help as well.

Our approach is to dynamically resize the scale focus on regions (e.g. address space, port number space, etc) containing the greatest activity. Contour lines (like those on topographical maps) will allow the viewer to be shown that space is being warped to bring out detail where it is required.

V. IMPLEMENTING OUR PROTOTYPE

Our first prototype has been implemented to monitor a darknet established on our University network. The darknet covers an entire /24 address space (an old 'class C' network), and is open to the outside world. The traffic generated by port-scans from internal and external hosts has provided us with a good starting point for demonstrating our animated space of 3D objects.

A. Tapping the network

Our darknet consists of a single VIA motherboard running FreeBSD 4.10 [19], configured with aliases to

every IP address in a particular /24 subnet. Out a different port this machine relays information about all captured traffic to a remote desktop machine with hardware 3D acceleration. This desktop computer then runs the application 3VEN (3D Visualisation Environment for NIDS) pronounced "even". 3VEN was created to explore data collected by our darknet, refine visualisations and provide a starting point for future software development.

3VEN is a C program created under FreeBSD 4.10 and KDevelop [20] using the freeGLUT [21] library. FreeGLUT speeds the creation of small to medium size OpenGL [14] applications by implementing cross platform windowing and allowing the mouse and keyboard to be read easily. It is an excellent library for implementing prototype OpenGL software quickly and is in common use. These factors made it a natural choice for building 3VEN.

3VEN implements many of the mappings and functions mentioned previously. 3VEN can display and update network objects in realtime or replay network events from a stored log file. It can replay this data from 1 to 20 times normal speed using the '+' and '-' keys to control the replay speed in real-time.

3VEN implements the network hierarchy, shape, size, colour and rotational velocity visual metaphors described previously.

B. Example output

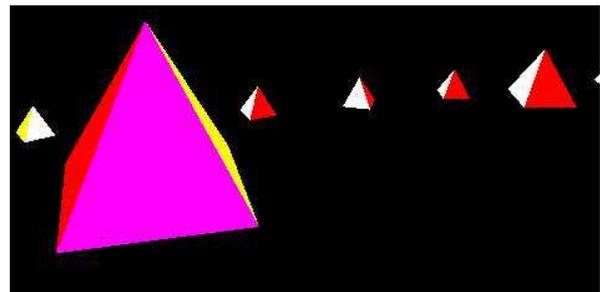


Figure 3 Inside a subnet showing hosts, one under attack. Multiple colours not normally present have been added to visualise rotation on the page

The implementation and use of 3VEN has guided development of our visual metaphor to network metric mappings and vice versa.

Figure 3 shows 3VEN displaying the hosts inside our darknet. It is impossible to see on the page but the hosts have a spin rate based on the number of attempted connections received in the last 10 minutes. They all have roughly the same size as many of the scans across the darknet IP range tend, although random, to be exhaustive.

Figure 4 shows inside one of the hosts in Figure 2, where the individual connections can be seen. The connections are spinning based on their data rate.

At this point due to 3VEN being in the early development stage, it is not possible to read specific information such as host addresses directly off the visualisation. Where as previous network visualisation work displayed only raw data, 3VEN has gone in the opposite direction and shows none of the raw metrics

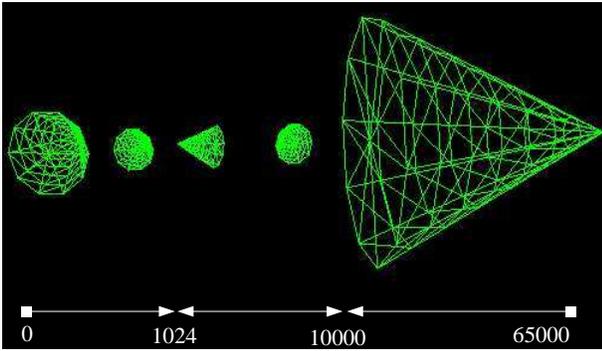


Figure 4 Inside a host looking at individual spinning connections and the nonlinear scale that is implemented.

from the collected data. A compromise must be found between these two extremes in future versions.

Many expansions to the 3VEN package are planned. Although our mappings are a start, it is clear that other mappings can be found, and our current mappings will no doubt continue to be refined. For example, we plan to add the notion of 'mass' to objects in the visualisation and implement a force that resembles gravity to naturally group items that require attention.

Future effort will be given to quantifying the visual differences between "normal" and "anomalous" network activity within 3VEN and its future revisions. Results from this research will feedback into the project and influence what proportion of detection should be technical (non linear curves, forms of automated detection etc) and what proportion should be left to human interpretation.

A. Interacting with the network

A key line of future research is inspired by the development of a unique process management tool known as psDoom [5]. psDoom modified the mid-1990s first person shooter game Doom so that while playing the game, "monsters" appear for each process running on the host computer. Moving around the virtual world represented moving around the computer's process space. Shooting a monster dead literally killed its associated process, wounding a monster reduced their process priority. psDoom demonstrated a key principle we plan to explore - if you translate complex ideas into intuitive interaction metaphors, you reduce the level of system-specific technical training required of people who manage your system. This applies whether your system is a host, a group of hosts or an entire network. An appealing consequence is a reduction in staffing costs (by lowering the threshold at which a new employee becomes useful) and increasing the fault-tolerance of your business (as replacement staff are easier to find and train).

We plan to extend our representation of network metrics to include styles of interaction that are metaphors for managing the underlying network and its components. The human operator will take on a role within the immersive environment containing the visual network objects, and interact in ways consistent with

their avatar's characteristics. For example, upon seeing a host object spinning rapidly a first person shooter metaphor might involve navigating down to the connection level and then 'shooting' a virtual weapon at the connection object (or objects) most responsible for the traffic load. The action of shooting would be translated behind the scenes into a set of updated firewall rules that act on the IP address and port pairs representing the offending traffic. Different interaction metaphors could be chosen to suit the kinds of employees you have - instead of shooting objects, perhaps the system operator takes on the role of a 'medic' in the virtual world, 'healing' overloaded networks, hosts or connections.

Regardless of the actual metaphor for interaction, the operator thinks in terms of "see activity and interact with objects using real-world metaphors" without needing to understand or absorb the intricacies of "addresses port numbers, firewall configuration languages", etc. Our prototype will be built on a testbed based around FreeBSD firewalls [22].

B. Increasingly sophisticated visual representations

As a longer term goal, we realise that visualisations need not stop at primitives for the display of metrics. The possibility of using off the shelf 3D game engines allows for many advanced metaphors to be used. A number of complex and visually impressive 3D engines have been released over the past few years, and many engines allow modification of the "upper layers" of their software (for example the Quake3 [23] and HalfLife [24] engines). In some cases the software has its entire source code open to the public for complete modification and extension as is the case with the Cube game engine [25].

Finally, when we move to representing full NIDS data, we can additionally represent the output of signature based system alerts as objects. We believe it will be possible to, as with firewall rules, also implement new signatures using the virtual world.

The use of off-the-shelf 3D game engines would come into their own when rendering the alerts of signature based NIDS. It is not hard to imagine, instead of a 12 foot tall multi-eyed beast coming at a Doom 3 player, it could just as easily be a 12 foot tall multi-eyed copy of the Sasser virus coming at a network administrator. It's just fortunate that the administrator has their firewall rule dispensing rocket launcher handy.

VI. CONCLUSION

Network Intrusion Detection Systems that allow for detection of anomalous network activity, often followed by an automated response, are tools used by the network professional to detect problems. These tools are often less than helpful at giving an overall impression of network health due to the fact they only implement visual aids in 2 dimensions.

Work into 3D representations of IP network data has been limited in scope. Attempts have been made that are simple extensions of 2D representations, without

leveraging the full potential of real time 3D representation.

We have proposed a set of unique visual metaphors using primitive objects that allow IP network data to be displayed in a manner that speeds the interpretation and thus the response to this data. To test aspects of our proposed method of representation, we have developed a prototype called 3VEN using OpenGL and freeGLUT under FreeBSD.

Results from our software are encouraging. Our future work will extend 3VEN to incorporate interaction with the objects in a virtual world, where the interactions lead to real-world reconfiguration of network entities such as firewalls and flow rate-limiting. Our longer term goals are to leverage fully implemented 3D software engines to implement visualisations of substantial complexity and interactivity.

REFERENCES

- [1] G. Armitage, "Making the internet go away," IEEE Internet Computing Vol.8, No.2, pp. 94-96, March-April, 2004
- [2] J. Nguyen, "The impact of Microsoft Windows infection vectors on IP network traffic patterns," CAIA Technical Report 040804A, August 2004 (<http://caia.swin.edu.au/reports/040804A/CAIA-TR-040804A.pdf>)
- [3] "Bro", <http://www.icir.org/vern/bro-info.html>, August 2004
- [4] "Snort", <http://www.snort.org/>, August 2004
- [5] "psdoom", <http://psdoom.sourceforge.net>, August 2004
- [6] S. Lau, "The Spinning Cube of Potential Doom," Communications of the ACM Volume 47, Issue 6 June 2004
- [7] D. Moore, G. Voelker, S. Savage, "Inferring Internet Denial-of-Service Activity," 2001 USENIX Security Symposium August 2001
- [8] "Honeyd Honeypot project", <http://www.honeyd.org/>, August 2004
- [9] "Visualizing Internet Topology at a Macroscopic Scale", http://www.caida.org/analysis/topology/as_core_network/, August 2004
- [10] J. Case et al. "Simple Network Management Protocol (SNMP)", RFC 1157 Internet Engineering Task Force, May 1990
- [11] "MRTG: The Multi Router Traffic Grapher", <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>, August 2004
- [12] "RRDtool", <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>, August 2003
- [13] K. Nyarko et al, "Network Intrusion Visualization with NIVA, and Intrusion Detection Visual Analyzer with Haptic Integration," Information Visualization Vol.2, No.2, pp. 82-94, June 2003
- [14] P. Abel et al, "Network management and virtual reality," International Scientific Workshop on virtual reality and prototyping, June 1999
- [15] A. Komlodi, J. R. Goodall, W. G. Lutters, "An Information Visualization Framework for Intrusion Detection," Proceedings of the ACM Conference on Human Factors in Computing Systems, 2004
- [16] "Nagios", <http://www.nagios.org/>, August 2004
- [17] "OpenGL", <http://www.opengl.org/>, August 2004
- [18] "Microsoft DirectX: Home Page", <http://www.microsoft.com/windows/directx/default.aspx>, August 2004
- [19] "The FreeBSD Project", <http://www.freebsd.org/>, August 2004
- [20] "KDevelop - an Integrated Development Environment", <http://kdevelop.org/>, August 2004
- [21] "The freeglut Project", <http://freeglut.sourceforge.net/>, August 2004
- [22] "FreeBSD Handbook, 14.8 Firewalls ", http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls.html, August 2004
- [23] "id Software: Quake III Arena", <http://www.idsoftware.com/games/quake/quake3-arena/>, August 2004
- [24] "The Official Half-Life Web Site", <http://games.sierra.com/games/half-life/>, August 2004
- [25] "Cube (Game/3D Engine)", <http://cube.sourceforge.net>, August 2004