Acknowledgements

I wish to thank the following people for contributing to this project:

Shaun Bangay, the best supervisor anyone can hope for

Carrie Barrion, the best proof reader anyone can hope for

VRSIG, the best virtual reality support group anyone can hope for
Abstract

Networks are an integral part of computing systems and have, in recent years, been incorporated in all fields especially within the business and scientific areas. However, despite the increased interest and application of networks to business and other institutes network management still remains a complex task.

The ultimate goal of this project is to investigate how applicable virtual reality is to the subject of network management. Virtual Reality enables us to create a visual metaphor to represent our network. In addition, it also enables us the use of novel and ingenious input devices to provide a natural and intuitive form of navigation and interaction with the data. The visual metaphor allows network administrators to apply intuitive reasoning and perception to manage the network by observing and manipulating the visual representation. We will show that Virtual Reality can adequately represent a local area network and all its data.
**Table of Contents**

**Introduction**

1. Problem Statement and Motivation .................................................. 6

2. Goals for this project ........................................................................ 7

**Background & Related Research**

1. 2D Visualisation Systems .................................................................. 11
   1. Otter .......................................................................................... 13
   2. Network Animator (NAM) ............................................................. 17
   3. GXSnmp .................................................................................... 18

2. 3D Visualisation Techniques & Virtual Reality ................................. 20
   1. 3D Displays of Internet Traffic ...................................................... 21
   2. Virtual Environment for Network Monitoring (VENoM) .......... 22

**4. Gesture Recognition** ..................................................................... 25

5. Summary .......................................................................................... 26

**Design**

1. Network Management System .......................................................... 27
   1. Introduction & Design Goals ......................................................... 27

2. Visualisation system ......................................................................... 30
   1. Introduction and Design Goals ...................................................... 30
   2. Representation ........................................................................... 30
   3. Navigation & Interaction ............................................................... 33
   4. Creating Objects ......................................................................... 34
   5. Updating Objects ........................................................................ 34

4. Messenger system ........................................................................... 35

**Method**

1. Network Management System .......................................................... 37
   1. Device Detection ......................................................................... 37
   2. Error Detection ........................................................................... 39
3. Performance Measurement 40

2. Visual Representation 40
   1. Brownian Fractal Terrain Generation 40
   2. Fault Generation 42
   3. Erosion 43
   4. Colour Mapping 45

3. Navigation and Interaction 46
   1. Gesture Recognition 46

4. Messenger Service 48

Results and Discussion 49
   1. Results 49
   2. Discussion 52

Conclusion 55
   1. Future Directions 56
      1. Collaborative Virtual Reality 56
      2. Animation 56
      3. Navigation and Interaction 57

References 58
List of Figures

2-1 Circular Layout 13
2-2 Co-ordinate Based Layout 13
2-3 Zoomed View 15
2-4 Colour Mapping 16
2-5 Node Reduction 16
2-6 NAM Screenshot 17
2-7 NAM Cluttered Display 18
2-8 GXSnmp Screenshot 20
2-9 Global Internet Usage 21
2-10 VENoM Screenshot 23
2-11 Gestures Recognised 24
4-1 Midpoint Displacement Algorithm 38
4-2 Unfiltered Image 40
4-3 Filtered Image 41
1. Colour Mapped Terrain 42
2. Pose Gestures 43
5-1 Device Representation 45
3. Virtual Lake 46
4. Network Overview 47
Introduction

1. Problem Statement and Motivation

There is no doubt that the invention and development of computer systems has led to the simplification of the processes involved in collecting, sorting and recording data. Nonetheless, it still remains a formidable task to analyse and interpret the data gathered. Network Visualisation, being a subset of the Scientific Visualisation, faces similar difficulties.

Thus far, network traffic has often been managed and produced by management protocols such as Simple Network Management Protocol (SNMP). However, in order to function effectively, SNMP is often used in tandem with network management tools. This combined system is referred to as a Network Management System.

Data visualisation became popular during the 1990s, due to advances in computer graphical hardware and software. Despite its popularity, however, the system has disadvantages as well. One shortcoming involves the inability of the system to generate an accurate picture of the data collected, because data analysis is often restricted to a 2D graphical representation of the information gathered. As a means to alleviate this problem, this project aims to investigate the applicability of virtual reality environments to Network Visualisation. In succeeding with this goal, it is thus possible to overcome the inadequacies associated with traditional methods of visualisation.

As already mentioned before, conventional data visualisation systems consist of a network management protocol combined with a 2D graphical representation. The majority of Network Visualisation systems use a link-node paradigm to represent the network. The nodes and links represent networked devices such as computers, switches, routers, etc., and these in turn are interconnected to other network devices with another set of links. The colour and thickness of the links indicates the state in which the current connection can be
found. Despite its comprehensibility and simplicity, link-node displays only work well within small networks. As the size of the network increases, the link-node display becomes progressively more cluttered and cumbersome, resulting in a jumble of lines of all colours and sizes connecting nodes. One method of avoiding the confusion of bulky link-node diagrams is to represent the data in 3D form.

Because of the extra dimension, 3D graphical representations have proved that it can adequately eliminate the limitations of associated with 2D visualisation systems. Furthermore, data represented in the 3D format is aesthetically more pleasing and comprehensible. Since most systems are hardly infallible, 3D graphical representation, too, has its own glitches. Possible drawbacks could result from the use of a 2D input device, such as a mouse, to navigate data represented in 3D format and difficulty in interacting with objects in the 3D model. This research into Virtual Reality Network Visualisation attempts to address these problems associated with 3D visualisation systems.

Virtual Reality provides solutions to the limitations found in existing Network Visualisation systems. It provides the user with an interactive three-dimensional world in which the user can move around in the data set with a novel range of 3D input and output devices, such as trackers and head-mounted-displays (HMD). Unlike 2D visualisation systems, the user is immersed and made a part of the virtual world, which greatly enhances the user’s understanding of the data. This understanding and insight into data is the primary focus of any Visualisation System.

2. Goals for this project

The primary goal for this project is to investigate the suitability of virtual reality to the subject of data visualisation and more specifically, Network Visualisation. The following question will be answered to evaluate the applicability of the virtual environment to aspects of the Network Visualisation:

- **Representation**
  How do we represent the network components in such a manner that it will enhance the user’s understanding of the network?
Navigation
How can users move freely around the virtual world?

Interaction
How are computer commands issued in the virtual world?
Network Visualisation Systems became popular in the 1990s and are now regarded as one of the more important tools for network management. The bulk of interest regarding these systems was invested in the management capabilities, leaving the visual interface much to be desired. Very little research has been done that focuses primarily on visualisation methods for network data. However, aspects that pertain to Visualisation in general are very applicable to Network Visualisation and all systems make use of these techniques in one form or the other. Needles to say, even less research focuses on 3D visualisation and the applicability of virtual reality to the subject. Again, general research into the fields of 3D visualisation and visualisation in virtual reality is very relevant and has proved to be an invaluable resource for this project. They define the areas that we wish to visualise as well as emphasise the inherent shortcomings we want to overcome in this project. We also look at gesture recognition systems to resolve issues relating to navigation and interaction in a 3D environment. In this section, an attempt will be made to highlight and discuss similar studies within the data visualisation field and to compare them to the study at hand. It is important to note, however, that many inconsistencies may result, since some of the papers published do not necessarily reflect an academic format and, in some cases, even fail to acknowledge the weaknesses of the study done.

The link-node paradigm of Scientific Visualisation is well suited for visualising a network. Any network can be described in terms of this model [1]. Software services and hardware devices are depicted as nodes and the links represent the connectivity between these devices. Regardless to say, all the limitation associated with this type of representation manifests itself in visualising the network. The node-link model works well for small networks [1], but soon becomes infeasible for large networks. A large number of links and nodes results in a highly cluttered graphical representation and reduces the effectivity of the
visualisation. The tangled jumble of lines and nodes result in a confusing representation, which is often unusable for data analysis and interpretation, and is also aesthetically displeasing.

Creating an effective visualisation is more of an art form rather than a science. It involves, for example, aspects of graphic design and the touch of artistic intellect. However, there are guidelines that aid in visualising a system. As already stated in the introduction to the project, the primary purpose of visualisation, and Scientific Visualisation in particular, is to gain insight of complex data sets and to create visual representations of phenomenon that may often be unnoticed [2]. It is critically important that the visualisation accurately present the data without any distortions. Visualisations are created with the user in mind. The model should be simple enough to easily relate the concepts used in the model with the aspects of the underlying system.

There are two ways for mapping data to a visual representation: intuitive and non-intuitive data mapping.

An intuitive mapping represents data in a form that seems logical and almost self-explanatory at first sight. Examples of these include geographical information systems (GIS) that display geographical data as images that resemble specific landscapes. Intuitively, mapping data assists the users to gain familiarity with the system faster than non-intuitive techniques. Non-intuitive mappings, on the other hand, reveal interesting features of data by virtue of presenting an image that requires additional analysis from the user. An important aspect of mapping that is often neglected in discussions is that of aesthetics. Overloading the user’s senses with a bombardment of colours and shapes can render the visualisation ineffective, since the data becomes difficult to interpret and analyse.

A number of definitions exist for Network Visualisation, some of these are:

- “… visualising arbitrary network data that can be expressed as a set of nodes, links or paths.” [3]
In an attempt to compensate for the lack of a formal classification of Network Visualisation, this project defines it as a mechanism that represents a network and all its data in visual and non-visual form. As human beings, we do not limit ourselves to visual aspects of creating suitable representations, but acknowledge that other means exist that can equally or better depict network features. Hearing is one particular means. The word visualise may restrict to areas pertaining to the visible spectrum, but this paper defines it as "the power or process of forming a mental picture or vision of something not actually present to the sight" [5]

In an attempt to illustrate the concepts involved in Network Visualisation, a number of tools that are available for Network Visualisation are presented within this paper. A critical analysis of each tool will not be attempted, but rather this Chapter will merely strive to highlight the key points that pertain to the research. The systems analysed are Otter [3], Network Animator [6], GXSnmp [7], World-wide Internet Traffic [8], and VENoM [9].

1.

2. 2D Visualisation Systems

All 2D visualisation systems regard networks as an arrangement of links and nodes. The links and nodes are representations of real objects such as computers, routers and switches, while the links indicate the physical or logical connectivity between nodes. Both the links and nodes are the key components of the representation that draw attention to certain aspects of the network traffic. The colour, thickness and style of a link serve to denote its state of physical connection. This often indicates whether or not the node is overloaded, under-utilised or if any delay-time is expected between nodes. Likewise, the same above-mentioned characteristics of colour, thickness and styles of nodes indicate the state of the machines interconnected within the network.

There are several possible solutions to address issues of 2D-display clutter [1]:

…paint a pretty picture of your network…” [4]
Layout Algorithms

Layout Algorithms compute optimal node and links placement to reduce the overall cluttered representation. A number of layout algorithms exists (two of these are discussed at greater length in Section 2.1.1) but regardless of their quality they still fail to eliminate issues of clutter.

Display Reduction

Reducing the amount of information that is displayed gets rid of clutter but at the high cost of also creating a representation that is less informative. This can spell disaster if information important to analysis is removed.

Abstraction

Distant and less important information is displayed in less detail. This differs from Display Reduction in that the information is not eliminated, but can be retrieved using techniques of zooming and focus. Section 2.1.1 describes these principles of more detail.

Alternate Representations

Curves and splines provide an alternative to using straight lines. Using these line crossing can be reduced and thus resulting in a less jumbled image.

Although these techniques serve to reduce some of the problems associated with 2D representation, they do not provide any significant advantages for the network system on the whole. They still fail to completely eliminate the display clutter of network systems, which is the primary objective for good Network Visualisation. Several 2D visualisation techniques are currently being applied manage to networks. The following sections reflect a brief study of Network Visualisation applications and visualisation techniques, and serves to illustrate their respective limitations.
1. *Otter*

In the past few years, the Co-operative Association for Internet Data Analysis (CAIDA) has developed a number of tools, which allow one to visualise large sets of network topology information. As a result of many preluding projects, Otter was born. It is used to visualise multicast and unicast topology databases, routing tables, delay measurements, SNMP data, and web site directory structures.

Otter was designed with 4 main characteristics in mind:

- To use nodes, links and paths to allow graphical visualisation of data
- Efficient memory usage techniques for displaying network maps
- The ability to place nodes and links at geographical and topological defined positions
- Interaction techniques to modify the visual representation. These including zoom, focus and graph layout capabilities. Zoom and focus allows the user to select the amount of detail regarding network data present in the image. Graph layout techniques gives the user the ability to reposition nodes or links in the graph.

Otter makes use of elaborate graph layout algorithms for constructing the visual representation of the underlying network. These algorithms arrange the nodes in either a *circular* or *co-ordinate-based* layout. These are illustrated in *Figures 2-1 and 2-2.*

![Figure 2-1 Co-ordinate Based Layout](image1)

![Figure 2-2 Co-ordinate Based Layout](image2)
The *circular layout* arranges the nodes randomly along the circumference of a circle, while *co-ordinate based placement* uses an input file to place nodes at predefined geographical positions. In using *circular layout*, the image is generated on the fly and no user-interaction is required, as is in *co-ordinate based placement*. The network management capabilities of Otter have received a significant amount of attention and satisfy the initial design criteria. Yet, despite its obvious advantages, Otter suffers significant speed disadvantages. It takes several minutes to set up the visual representation.

Depending on the size of the network, the *circular layout* produces a graphical representation where many nodes overlap each other in the image. This is clearly visible in *Figure 2-1*, which is a representation of a moderate sized network. *Co-ordinate based placement* has been added to overcome difficulty in manipulating data sets that are too large to compute the layout of automatically (i.e. using the circular layout). Manual arrangement of the nodes is necessary to achieve a workable graphical representation (*Figure 2-2*). However, this does not eliminate the problem of data overlap that is normally associated with 2D displays. By increasing the number of nodes to be displayed, the effectiveness of the display is further reduced.

Another attempt at addressing the clutter and overlay issues is achieved by reducing the amount of information that is displayed. This is attained by simply showing less detail, which then produces a graph with less clutter.

Furthermore, the ability to *zoom* on certain data points allow the user the freedom to select a scale at which to view the network model, and to obtain a more detailed analysis of a particular area. Information that was eliminated during *Information Hiding* reappears when an area is viewed at higher detail. Likewise, using the zoom option can recreate some of the clutter that has previously been ‘eliminated’ by the high level representations, and one must undergo the procedures of hiding unnecessary information to accentuate the necessary data. (*Figure 2-3*)
Yet another technique that aims to alleviate the confusion of 2D visualisation is the use of spectrum colouring. Spectrum Colouring is used to indicate the status of links by assigning a unique colour to each value of a network characteristic. Links and nodes are then drawn using these colours. An accompanying colour legend is included with the graph (See Figure 2-4). Using Otter, however, it is difficult to distinguish more than 15 different colours. Many functionality improvisations have had to be made to accommodate this limitation. However, even these alterations have not had any significant effect on sufficiently employing Spectrum Colouring to visualise information. Human Computer Interaction considerations also play a role in choosing colours used in the representation [10]; thus guidelines to obtain standard and universally understood representations should be followed. Some of these guidelines are:

- Use a maximum of five colours to match the user’s short-term memory
When ordering items, follow the spectral order: red, orange, yellow, green, blue, indigo and violet.

- The use of familiar colour codings. Red for danger and green for safety.
- Make allowances for vision impairments such as colour-blindness.

One of the prominent limitations of Otter is that it has been designed solely to visualise networks with only 200-700 nodes. Above this range, the already limited visualisation ability of Otter is significantly reduced. This effect is demonstrated by Figure 2-5, which indicates the case wherein more than 30,000 nodes are displayed. Otter deals with this problem in two ways: First, by reducing a number of nodes to a single node and then displaying this single instance. However, this procedure eliminates some network data, which the user might want to view. Reduced visualisation is also countered by using Graph Layout Algorithms. These, however, have also proved to be highly insufficient in reducing the clutter problem.
2. \textit{Network Animator (NAM)}

Like other 2D-visualisation systems, Network Animator (NAM) represents the network as a set of nodes with interconnecting links. It employs the same principles applied in Otter and other 2D based visualisation systems, which make use of Spectrum Colouring.

However, the main characteristic of NAM that separates it from other 2D-visualisation systems, is its ability to animate network traffic on a packet switched network. The length, spacing, colour and width of connecting lines, links, represent the delay between connected devices.

Animated flowing rectangles are representations of the network traffic flowing through a link (see Figure 2.6). A number of animations have been designed to illustrate certain situations that may arise. For example, rectangles falling from the display indicate that the packets have been dropped. Although, animation is an effective means of visualisation, it is still inefficient in reducing the limitations imposed by the display of 2D characteristics of it.

![Figure 2 6 NAM Screenshot](image)

Network events usually arise suddenly and are usually of a short duration and, thus, are often unnoticed by administrators. To improve its clarity and usability, NAM has a set of

![Figure 2 7 NAM Cluttered Display](image)
controls that allows the user to adjust the speed and playback of animations. Individual components and traffic can also be tagged to display more information regarding the object during animation, as a means to overcome the display clutter problem. Another method used to counter clutter is through hiding information, and is obtained by adjusting the shapes and sizes of components or dropping components from the display.

Unlike Otter, NAM uses a highly ineffective layout technique. The nodes are all positioned with respect to the relative network delay between each other. This results in a very jumbled display of nodes, even within a very small network of less than 100 nodes. This trend of significant data overlay is demonstrated in Figure 2-7, which represents a small network consisting of eleven nodes.

Through its ineffectiveness, NAM proves to be a simple tool that highlights the limitations of 2D visualisation very clearly. Despite its drawbacks, however, NAM is a good educational method that has proves very useful in investigating aspects of animation in the 2D-visualisation realm.

3. **GXS\text{NMP}**

GXS\text{Nmp} is a network management application that was developed under the GNU Public License. The following issues were identified as the key components that GXS\text{Nmp} must address:

- SNMP Compatibility
- Network discovery functionality to automatically locate SNMP enabled devices
- Distributed network management
- Interface to network status monitor (the visual representation)
- Scripting language for remote administration of devices
- Performance and error reporting
Although the visual interface of GXSnmp receives much less attention than in the Otter and NAM projects, its strengths and capabilities, and relevance to this project lie in its network management capabilities.

The system architecture is composed of probes, monitoring agents, a presentation-, and database component. Probes are software tools that continually check the status of managed devices and reports their findings to the monitoring agent. There are two types of probes: polls and traps.

A Poll probes and checks the performance status of networked devices and Internet related services such as FTP, HTTP and DNS at regular time intervals. When a monitored event such as the failure of critical services occurs, it sends an unrequested message, a trap, to the Trap Handler notifying it of the event that has just occurred. These two services work in tandem, and if a faulty device is unable to send a trap to the Trap Handler, the Polling Probes will determine the error during the next polling interval. Unlike traps, polls are very network intensive and thus create a high level of network traffic.

The monitoring engine is found at the heart of the architecture. Its main responsibilities include maintaining a list of probes, receiving the trap notifications, accepting connections from the presentation layer and communicating with the users and the visual interface.

The graphical layout of GXSnmp is also composed of links and nodes. However, unlike Otter and NAM, GXSnmp does not try and present all network information in graphical form. Instead, information is stored within certain devices or links and can be obtained by selecting the respective devices or links. Information is then displayed in textual format. Minimal usage of the network is demonstrated in Figure 2-8, which provides nothing more than just a pretty picture of the network. The information that the Network Visualisation aims to present is still maintained in text form and, thus, does not simplify the process of analysis. It does not enhance the users understanding of the network in anyway.
A number of the techniques employed in 2D visualisation systems are very relevant to our project. Research into the use of colour to represent data is also applicable to 3D and virtual reality environment and animation adds significantly to the user experience and understanding of data. However, other issues of the visualisation system such as display clutter and overlap do not occur in 3D, and thus do not contribute to designing 3D visualisation systems.

The network management system in 2D visualisation systems is a separate entity from the visualisation system. Research relating to device discovery, error detection and performance management was adopted in principle and implemented in similar fashions.

3. 3D VISUALISATION TECHNIQUES & VIRTUAL REALITY

The added dimension of 3D displays eliminates many of the restrictive issues found in 2D visualisation techniques, such as display clutter and device overlap. Line crossings that are
found in 2D visualisation are eliminated along the added dimension of a 3D system, thereby reducing much of the clutter associated with 2D visualisation. The result yields an appealing representation that is not restricted to the link-node paradigm. It is also more open to all possible effective representations.

1. **3D Displays of Internet Traffic**

One of the more popular representations for mapping Internet Traffic is through the use of a globe with interconnecting links. This representation uses an input file with nodes indicating the geographical locations of several sites, which are then mapped on a 3D globe. The links between the nodes are represented as arcs. Again, a link's visual appearance indicates the status of the connection between the two nodes. Stephen Cox created one such example, in his representation of ‘The Usage of the Internet on a global scale’ (*Figure 2-9*)

The end effect of 3D visual representation is concise and very appealing. The ‘Globe Metaphor’ used to represent global Internet usage also adds to analysis process. We can immediately identify the locations and the intensities of traffic between two such sites. Even the crossing-overs of links are not difficult to follow when compared to 2D displays. The combination of the colour-coding scheme indicates where a large volume of traffic is being generated, and how these areas are geographically related. The globe appears to be lit from a light source that represents the current position of the sun with respect to earth. This is a very effective representation of the time reference associated with network usage. Upon detailed analysis, 3D images certainly convey well-presented and concise information.
Since most systems are hardly infallible, 3D graphical representations, too, have their own anomalies. One possible drawback could result from the use of a two-dimensional input device, such as a mouse, to navigate data represented in 3D format. Users are initially confused by the limitations of a 2D input device in 3D, which stems from the use of a device (the mouse) with only 2 degrees of freedom, in an environment (the 3D environment) that has 6 degrees of freedom.

**Figure 2-9 Global Internet Usage**

Another 3D-visualisation technique, that of virtual reality, is applied today through the virtual environment for network monitoring (VENoM) system. VENoM is a suite of models and applications that are based on collaborative virtual reality. It has been designed to specifically visualise large high speed networks such as ATM based networks. It uses Simple Network Management Protocol (SNMP) for monitoring tasks.

The ever-increasing sizes of networks have meant that the management of networks has ceased to be the sole responsibility of a single administrator, and is now shared with other network administrators. Many administrators now have to face the challenge of managing several networks simultaneously. This task is made possible through the use of VENoM. By use of this visualisation technique, network administrators can see the representations of their peers within the network.

Normally, when one visualises a network, an image of the entire network consisting of all its different working components can be drawn together, as shown in the previous text illustrations. This allows one to envision the entire network, as well as approximate the location and interaction of all components relative to each other.

However, when creating a virtual reality room, one’s graphical representation is limited solely to that space. In the VENoM model, this space is normally considered where one
keeps all the necessary equipment. Thus, little insight into the overall functioning of the network is gained. Many administrators are already familiar with the layout of equipment within the respective rooms and can find their way around the networks more easily. However, because the information viewed at any given time is restricted to a single virtual room enclosed within virtual walls, one fails to see how machines outside the room are interconnected to each other. It is also difficult to determine how the network traffic flows between them. The latter restriction carries far more weight in terms of Network Visualisation. It is important to observe the interactions between the machines and to also observe where hotspots occur. The VENoM project has tried to compensate for this handicap by creating additional views of the network that give logical layouts, instead of physical layouts. This is illustrated in Figure 2-10.

![Figure 2-10 VENoM Screenshots](image)

By obtaining multiple views or representations of the network system, it is possible to counter the initial inaccurate intuitive representation generated. Within the virtual reality environment, it is possible to simultaneously expose the networks limitations and employ another representation technique to extract the data or information needed.
VENoM uses a comprehensive set of virtual reality input and output devices such as trackers and head mounted displays. The immersive virtual environment enhances the user experience and leads to a better understanding of data. In this instance, users become a crucial part of the data, and are no longer considered as external observers viewing the data set from the outside.
4. Gesture Recognition

In this project, we also considered Gesture and Speech Recognition to provide the navigation and interaction mechanisms needed in virtual reality. To achieve these results, we consulted past research studies done by the Rhodes University Virtual Reality Special Interest Group (VRSIG) in the field of Gesture and Speech Recognition. The following list is a compilation of the papers used to answer questions regarding Navigation and Interaction in virtual reality:

- "Investigation into gestures as an input mechanism – A simple 3D modelling application in a virtual reality environment" by Holger Winnemöller [12]
- “A Distributed Virtual Reality Interface for use in Articulated Figure Animation” by Luis Casanueva [13]

Shaw and Winnemöller both describe a Feature Based Gesture Recognition System, that recognises and performs functions defined by certain shapes. The system operates by specifically recognising 3D shapes drawn with Polhemus trackers. The system analyses the shape of the object drawn and, if it is recognised, an appropriate action will be performed. The following diagram shows some sample objects that these systems are able to recognise.

![Figure 2-11 Gestures Recognised](image)

With this gesture-recognition ability of the system, the user can then specify certain actions associated with each of these shapes.
One possible limitation to this system is the restriction of various actions to single shapes. Thus, the same gesture cannot be used in a different context within the same application. Another drawback is that these systems are capable of recognising only gestures that involve movement. Static body gestures such pointing in a direction cannot be recognised as valid gestures.

Unlike Shaw and Winnemöller, Casanueva describes a system that captures and animates user motion with the use of inverse kinematics. Given the position of the Polhemus trackers attached to the body of a user, the pose of the body can be determined and in turn a gesture can be formed based on the pose. The implementation of the system does not provide any means for gesture recognition as it only animates body motion.

5. SUMMARY

Network Visualisation is an evolving science. To date, there has not been an adequate proposal that addresses all the issues of visualisation. The 2D-visualisation systems admittedly pioneered the field, but were later superseded by 3D and virtual reality techniques. Current research indicates that the latter mentioned techniques are capable of suitably defining visualisation frameworks. The aim of this project and other related research is to satisfy this claim.
1. NETWORK MANAGEMENT SYSTEM

1. INTRODUCTION & DESIGN GOALS

The network management system collects information, such as bandwidth usage, about the network, and reports it to the user. In this project, the description of a network management system offers functionality similar to that defined by the ISO FCAPS model [14]. Fault Configuration, Accounting, Performance and Security Management, or FCAPS defines the main functionality that should be addressed by any network management. Within this project, the main functional areas of the management system, as well as its design objectives include error detection (Fault Management), performance measurement and remote administration (Configuration Management) of a network.

- Error detection
  Error detection provides tools for locating and isolating malfunctions within the network. However, it does not specify any corrective measures for these breakdowns, and thus is only considered as a simple warning system.

- Performance Measurement
  This functionality aspect provides the network with mechanisms for measuring and monitoring the performance of devices located within the network. It also
provides an overview of the network usage that allows one to uncover the source of bottlenecks.

## Remote Administration

Remote administration enables one to remotely modify the settings on a network device, and is one of the most highly desirable features for an effective network management system. Since many networks span a large geographical area, it has become increasingly more important to have a means to access remote devices from distant locations, thereby allowing one to perform administrative tasks.

Yet another important area of network management is network security. However, due to the broad scope of issues pertaining to this subject, and to the time constraints imposed upon this project, it will not be covered in this discussion.

Within the network management system, the manager is the central component that provides cohesiveness within the system. As the name implies, its tasks are to monitor and manage the devices in the network. The devices that are being managed are referred to as nodes, a term used to support the already-mentioned Visualisation models. The nodes run software packages that allow the manager total control over them. Standardised channels for communication between the manager and the nodes, such as those provided by Simple Network Management Protocol (SNMP), serve as swift information pathways between the two locations on the network.

The manager can send SNMP requests (polls) to a node for information regarding its status. This data is stored within the management information base (MIB) of the node. After receiving the SNMP request, the appropriate value data is extracted from the MIB of the node and is then sent to the manager.

In contrast to polling, trapping takes place when a node, under certain predefined conditions sends the manager unsolicited SNMP data. However, these are actions are mostly restricted to extraordinary circumstances, such as when errors occur.
Often, an error may go unreported if it affects the node’s ability to notify the manager of the error. For this reason, the manager polls critical devices continuously in the network, thus ensuring high activity of the unit, data on the node’s performance, and/or the conditions that lead to errors. Polling devices continuously tax both the network and the node under scrutiny, by creating additional network traffic load. To lessen the heavily monitored and burdened network, carefully selected time intervals between consecutive polls are established.

To locate all the nodes within the network, the manager automatically sends SNMP requests to all the devices within its control. A node that is capable of SNMP communication will respond to the manager’s request with a reply, while devices that are not SNMP-enabled will ignore the message. The replies received by the manager are used to construct a list of the devices in the network. These devices are then continuously polled to check their activity and performance. If the manager receives a trap from a device that is not in the list, it will create a new entry in the list for it.

In addition to monitoring and requesting information from the MIBs of nodes, the manager can also set new values for some of the variables stored within the MIB of the node. In SNMP terminology, this action is referred to as a *set request*. The changes in the values of the MIB result in a rippling-effect that passes upward through to the operating system, and causes the appropriate alteration of all the corresponding values. SNMP provides a very limited set of variables that can be configured, and thus does not provide true remote administration capabilities. The scope of this project does not make any provisions for other existing remote administration capabilities apart from those presented by SNMP.

The network manager in a database captures some of the details of the messages that are passed between the manager and the nodes. Among the many benefits of utilising a database, the following points remain crucial for this project:

- To automate the network configuration journal kept by most administrators;
Applying data-mining algorithms to sufficiently large databases that may reveal interesting and important aspects of the network, and may also contribute gainful insight into the management of the network;

And most importantly, to detect events such as significant errors that occur at odd hours when little supervision of the network is in effect.

2. VISUALISATION SYSTEM

3.

1. INTRODUCTION AND DESIGN GOALS

The Visualisation system creates a suitable graphical representation that reflects the current operational state of the network. In addition, it also presents the user with a platform or means to monitor and control devices within the network. The three key areas that are normally considered when creating a visual system are:

2. REPRESENTATION

The representation should aim to enhance the user's understanding of the current network system. It is a complex topic, which cannot simply be summarised, because many factors are usually combined to create an effective model for representation. However, to construct an effective representation model, the following factors should be taken into consideration:

The representation should be able to model and describe all aspects, such as errors and bandwidth utilisation of the network adequately. The model should be simple enough to easily relate the fundamental concepts applied in the model to the functional aspects of the underlying network. The representation should also be able
to provide as much information as is possible, without overwhelming the user’s senses. However, one should allow for varying levels of detail, in order to accommodate the information demand of the user. For example, if the user wants more information about a particular area within the network, all information pertaining to that area should be available for display. On the other hand, if the user wishes simply to render a superficial accounting of the network status and functionality, the system should have the ability to present only the significant data pertaining to the information request.

The mode of representation should not be restricted solely to the visual senses, and should take the other forms of sensation into consideration. Incorporating hearing into the representation is one such example.

1. Nature Metaphor

Inspired by the challenge to explore yet unstudied applications of 3D visual representation and Virtual Reality environments, we used a landscape metaphor for our visual representation. We selected this approach because it offers a means to fulfil the above-mentioned attributes of a good visual representation, and offered a challenge to our imaginations.

Generally speaking, Nature teems with the mixtures of sound, colour, shapes and motion, all important aspects that should be considered when creating and transforming data to an image. By associating a network aspect with each of one of these characteristics, we should be able to convert our network data into an image that contains elements from nature.

Mountains & Trees

Within our system, the mountains represent network switches. All objects found on the virtual mountain represents the networked components that are connected to the switch. The computers connected to the particular switch are denoted as the trees.
that grow on the slopes of the mountain. In real situations, it is possible to have many different types of computers connected to a single switch. These computers often vary in terms of architecture, manufacturer, operating system, or even on the types of network services provided to each unit. These varying computer categories are represented differently on the mountain. These differences are designated by different species of trees or different plants.
**Water**

Every virtual mountain has a body of water (for example, a lake) located near it. The water level of the lake indicates the overall performance of each networked switch. When the lake overflows, it indicates that the attached switch is handling too much network traffic.

**Rivers**

Apart from being associated with lakes, the mountains in the visualisation are also connected to rivers or streams. The width of the river indicates the amount of traffic between the connected devices and, thus, they give one a representation of the backbone bandwidth utilisation between the various switches and hubs.

**Thunder**

Because Virtual Reality offers an immersive environment, the user often becomes part of the visualisation. When users are located in one part of the landscape, they may be unable to observe events that occur in distant areas. With the incorporation of Audio into the representation, the system has a means to notify the user of an event that has just taken place. Thunder, a natural phenomenon that is generally associated with trouble, indicates the source of network errors. The use of 3D-surround sound audio gives the user the opportunity to instinctively identify the direction from which the errors occur.

### 3. Navigation & Interaction

The model represents a large physical area, and the information relating to the network and its devices is scattered throughout the model. Thus, in order to obtain the information they may need, it is a necessity to move from one part of the model to the other. Navigation is, therefore, imperative.
The model is more than a representation of the network. It also provides the functionality of a network control centre, where certain aspects of the network can be adjusted.

As already mentioned, the Visual Manager is at the heart of the Visualisation system, and its functionality applies to all aspects of the Visual system. It receives the network data collected by the network manager, and generates a suitable representation depending on the type of data it receives. The network data received is generally classified into two categories: data used to generate objects, and data used to update already existing objects in the model.

4. **Creating Objects**

When the Visualisation Manager receives information about the presence of a new network device, it creates a new visual object in the model to represent it. Depending on the type and location of the network device, the data collected and specified by the network manager is relayed to the visualisation manager, which draws a suitable graphical object in the model.

Normally, the switch connected to the new device is used to determine a device’s location. After the appropriate metaphorical representation for the network device has been generated, the new device will be relocated and displayed on the right mountain. If the new device is a switch, a new mountain will be generated in an unpopulated area of the landscape.

5. **Updating Objects**

Updating an object refers to altering the appearance or location in the model to represent the current status of a network device. For example, if a computer is responsible for
creating a lot of network traffic, it will be relocated closer to or even in the river to indicate that it is generating some about of network traffic.

Virtual Reality has many innovative input devices, which allow one to both navigate and interact with the model. This project has been developed using the CoRgi Virtual Reality toolkit. CoRgi has support for Polhemus trackers, virtual hand devices and audio input. These devices are used in interacting with and navigating the representation. The audio capabilities of CoRgi can also be used for speech input by making use of CoRgi’s speech recognition capabilities. Similarly, the trackers and virtual hands of CoRgi can be used to make gestures that will serve as input, since CoRgi also has gesture-recognition capabilities. Commands that are generated from gestures and speech input are sent to the network management system by the visualisation manager, where the network manager performs the appropriate network related procedures.

4. MESSENGER SYSTEM

The information interchange between the Visualisation system and the network management system is provided by the messenger system. The network data collected by the network manager is normally found in SNMP format, and will not be understood by the Visualisation system. Likewise, the commands issued in the Visualisation system are not in the SNMP format and will thus not be understood by the network manager. Thus, the messenger system is crucial communication tool for the two systems, because it possesses a common set of standards that allows these two systems to communicate with it and, hence, with each other.

For example, when the network manager receives an SNMP message, it passes a copy of this message to the messenger system. The messenger system then ‘translates’ the message to a common format understood by both the visualisation- and network manager systems. The database described as part of the network management system stores its data in this commonly understood format. Thus, technically speaking, the Visualisation system does
not actually receive its data from the network Manager system, but rather from the messenger system.

Since the messenger system is connected to both the Network Management System’s database and the network manager, it can extract and send data from either of these sources to the visualisation system. If the messenger system extracts the data from the database, past network events can be reconstructed and ‘replayed’ by the visualisation system.
As mentioned in Chapter 3, the complete system is composed of two distinct sections: a Network Management System and a Visualisation System, and is implemented as such.

1. Network Management System

Simple Network Management Protocol is, to all intents and purposes, the standard chosen for network monitoring and management. A publicly available SNMP suite was developed at the University of California, Davis (UCD) and was used in gathering information on network devices. The UCD implementation of SNMP consists of a manager application `snmptrapd`, a client agent application `snmpd`, and a C language application-programming interface (API). The `snmpd` application is installed on monitored computers, while the `snmptrapd` is found on the managing computer. Most hardware manufacturers also provide their own SNMP implementation on their devices to allow easy access to them.

The key components of the Network Management System are device detection, error detection, and the measurement of devise performance.

1. Device Detection

The manager automatically detects the networked devices that are placed under its control. Additionally, it also identifies the switch that the device is connected to. The network
manager is limited to managing only those devices that are connected to switches capable of SNMP communication, and do not manage devices that are connected to non-SNMP switches. This is an artificial restriction we placed on the system and can be easily rectified to cover all SNMP devices. Due to the time constraints imposed on this project, we restricted our study to devices that are connected to SNMP-enabled switches.

A list of all the devices attached to a particular switch is maintained on the network manager. The information stored in this list is then used during the visualisation process. The structure for the list is as follows:

```c
struct Port {
    int usage;
    int errors;
    int operational_status;
    int capacity
    char hw_address[17];
};

struct Switch {
    struct Port port[24];
    struct Switch *next;
};
```

The device list is implemented as a linked-list of the type struct Switch. Every switch can support a maximum of 24 ports. Each of these ports, in turn, hold information such as the usage, errors generated, operational status and Ethernet address of the attached device.

The system has been designed in such a way that additional requirements such as protocol usage or any other network statistic can be easily incorporated by adding appropriate variables to the Port structure.
The network manager automatically locates the devices attached to the switches through the use of the following two methods: Scanning and Trapping.

- **Scanning**
  The Manager inspects every port of every switch to determine if a device is connected to the port. The SNMP variable `interfaces.ifTable.ifEntry.ifSpeed` indicates the capacity of the connection between the switch and the connected device. If the value attained is zero, then one can either deduce that there is no device connected to the port, or that the connected device is offline. For non-zero values of this variable, the information regarding the connected device is added to the appropriate variable in the linked-list. The device’s hardware address is also determined from a variable in the switch’s MIB.

- **Trapping**
  After the network manager has scanned all the ports on the connected switches, it terminates itself. A device that is switched on any time after the scan will not have an entry in the device list. To compensate for this a device it sends a start-up trap to the manager when it is started up. Upon receiving the trap, the manager determines the Ethernet address of the device by means of an address resolution. It then looks into the MIB for all switches with a matching Ethernet address. If a match is found, the manager adds the information of the device to the appropriate variable of the linked-list.

Device Detection requires a minimal amount of user input. If a specific switch is to be monitored, the IP address of the switch needs to be specified in a configuration file. Normally, this configuration file is read when the system starts up.

**2. Error Detection**
If a device experiences an error, it reports the situation to the network manager by means of sending a trap. The content of the trap is then examined and the appropriate action is taken. For example, if a device sends a shutdown-trap, the manager determines the device’s Ethernet address, locates the corresponding variable in the linked list, and sets the operational-status variable to indicate that the device is offline.

3. Performance Measurement

For reasons already mentioned elsewhere, the network manager continuously polls the devices listed for network usage information. The network usage is then determined by obtaining values for the number of bytes received and sent. This information is revealed in the SNMP variables: `interfaces.ifTable.ifEntry.ifInOctets.x` and `interfaces.ifTable.ifEntry.ifOutOctets.x` (Where x indicates a port number in the range 1-24.) After obtaining data from the device, the manager pauses for five seconds. Thereafter, it begins extracting and gathering values for these variables again. If the device were generating any network traffic, the second set of values would be larger than those of the first. The difference between these two sets of numbers is then divided by the capacity of the port and the 5-second delay to give a measure of the network usage.

Performance measurements are implemented by using posix threads, which enable the manager to measure the performance on all ports simultaneously.

2. Visual Representation

1. Brownian Fractal Terrain Generation

Robert Brown first observed Brownian Motion, the random movement of particles, while he was looking at pollen particles. Since then, the concept of Brownian Motion has made its
impact on computer graphics. It is used as powerful tool for generating landscapes through the use of fractals. Thus, for this project, we also made use of fractals and *Brownian Motion* to generate our visualisation landscape. To obtain this, we used a particular technique called Midpoint *Displacement*.

*Midpoint Displacement* is a well known and widely used technique that makes use of *Brownian Motion* to create natural looking landscapes [15]. The Midpoint Displacement algorithm can be summarised as follows:

**Step 1:**
- Select a set of 4 points to form a square

**Step 2:**
- Calculate the midpoints for each side of the square, as well as the midpoint for the square. Displace each midpoint vertically by a random amount (*Brownian Motion*).

**Step 3:**
- For each of the resulting squares (A, B, C and D in *Figure 1*) repeat steps 1 to 3, each time decreasing the amount of the vertical displacement.

![Figure 4-1 Midpoint Displacement Algorithm](image)
The data structure that is used to describe the terrain is a simple 2D array. The value at any given \([x][y]\) position in the array represents the height at that point in the landscape. We refer to the array as a height map.

The algorithm is implemented as a set of recursive function calls. A square area of the two dimensional array is selected for fractalisation and is subdivided into 4 new squares. The vertical displacements of the new midpoints are used as the height values for the landscape. The subdivision is continued until the resulting squares have a size of one unit. The recursive nature of the algorithm implies that the dimension of the 2D array must be a power of two.

The prototype for the terrain generation is:

```plaintext
void generateterrain(int x1, int y1, int x2, int y2, int level, float random_height)
```

The variables \(x1, y1, x2, y2\) indicate the area in the 2D array that will be fractalised. Level is decremented in each recursive call. When it reaches zero, the recursive function returns. Random_height specifies the range of numbers from which the random displacements is chosen. Each recursive call decreases the range of random numbers.

With this recursive function, one can generate both the mountains and flat landscapes. By altering some of the parameters for the function, different types of landscapes are generated. Specifying a large value for random_height and a small square area results in the creation of a mountain, while using small random_height values and large square areas creates a flat landscape.

**2. Fault Generation**
In his paper “Generating Realistic Terrain”, Robert Krten [16] describes the process of Fault Generation, an alternative method for generating realistic terrain. The algorithm is summarised as follows:

An imaginary straight line is drawn through the array.

All the height values on one side of the imaginary line are incremented by a value x, while the height values on the other side of the imaginary line are decremented by x.

Fault generation can be obtained by decreasing the value of x, and choosing a new imaginary line through the array. These steps are repeated until a satisfactory result is obtained.

Combined with Brownian Fractal Terrain Generation, Fault Generation generates a very rugged course terrain. Because the terrain has a high degree of sharpness, it does not reflect a very realistic portrait of Nature. To enhance the visual appearance of the terrain, we simulated natural erosion by means of a digital filter. This technique is furthered discussed in the following section.

3. Erosion

We view the high degree of sharpness as the result of high frequencies, also known as noise. To create a smoother terrain, this noise is removed by passing the terrain through a low-pass filter. Thus, high frequencies are removed, and only the low frequencies are allowed to pass through the filter.

Fast Fourier Transforms (FFT) are then used to transform the height map from the spatial domain to the frequency domain [17]. All frequencies above a certain cutoff value are removed. An Inverse Fast Fourier Transform is then applied to transform the image back to the spatial domain. This results in a smoother image.
Because the image has two dimensions, it is necessary to filter it in two steps. First, all the columns in the array are filtered followed by the rows.

![Figure 4-2 Unfiltered Image](image1)

![Figure 4-3 Filtered Image](image2)

Figures 4-2 & 4-3 illustrate Midpoint Displacement and Erosion. The wireframe model of the terrain in Figure 4-2, produced by Midpoint Displacement has a very sharp appearance and is not a good representation of a landscape. Figure 4-4 is result obtained after Figure 1
is passed through a low pass filter. It produces a much more realistic simulation of a landscape.

4. Colour Mapping

The different features in the terrain are identified based on their colour. Mountains are a dull blue-purple colour, while the flatland is a mixture of green and brown, and rivers are blue. The colours are associated with various heights of the terrain, i.e. at a certain high cut-off value for height; the mountains are drawn in a blue-purple colour. On the other hand, height values that fall below a defined low cut-off height value are blue to signify water. Any values that fall between the two boundaries are assumed to be for flat landscapes, and are assigned either brown or green colours. These are illustrated in Figure 4.4.

The shading information is calculated for each vertex and interpolated across the surface of the polygon. This interpolated shading, or Gouraud Shading, produces a smooth transition of colours in the image and enhances the visual appearance of the terrain.
3. NAVIGATION AND INTERACTION

1. GESTURE RECOGNITION

Within our project, there are two sets of gestures that we wish to use when issuing commands to our system: pose gestures and motion gestures.

- Motion Gesture
  Moving certain parts of the body such as the hand when waving forms these types of gestures.
Pose Gesture

Unlike motion gestures, pose gestures require a body remain in a static pose. For example, static bodies pointing in a certain direction are considered a gesture.

Two pose gestures point and stop, are illustrated in Figure 4-5.

![Figure 4 5 Pose Gestures](image)

The Feature Based Gesture Recognition systems are useful in distinguishing *Motion Gestures*. We can create an object representation of a gesture such as a hand waving and add this to the library of gestures. In the virtual world, the simple task of waving can be easily recognised and associated with a valid command.

By applying the techniques used in inverse kinematics we can create a system that can recognise pose gestures such as the pointing gesture. We can determine the position of the hand as well as the other joints in the arm and based on these determine the gesture that is made.

Once the gesture has been identified, the appropriate action can then be taken. Gestures we are considering in this paper are those illustrated in Figure 1. The outstretched denotes a point gesture and the bent arm indicates a stop gesture.
In life, gestures are normally used to emphasise speech when interacting with people. They are used to clarify the spoken commands and, thus, they allow the same gesture to be used within different contexts. For example, the point gesture can be used in indicating a direction. However, when combined with a spoken command “move”, the point gesture moves the user in the indicated direction. But, if the same gesture points at a certain device and is associated with a different spoken command, the system will not move the user in the direction of the device. Rather, it will perform the command on the device that is pointed at by the user.

4. MESSENGER SERVICE

The messenger service is responsible for providing the connectivity between the Network Management System and the Visualisation System.

It is implemented as a client-server application. The server is attached to TCP port 16001 on the Visualisation System. Every 5 seconds the network management system transmits the complete device list and all its data to the Visualisation System via the messenger service. Upon receipt of the device list, the Visualisation System changes the appearance of the objects in the representation according to the new values obtained from the device list.
Results and Discussion

1. Results

With the application of our designed virtual reality system, we succeeded in fulfilling our objective of graphically representing the network by using the nature metaphor. In the following diagram (Figure 5.1), we generated an image that exhibits all the essential components of a network system. At this high resolution, it is possible to see the different computers and network devices within the system. These are indicated by the vegetation (trees) in the landscape. We also demonstrated the association between the network devices and the appropriate switch as indicated by the proximity of the trees to the mountain, where the mountain represents the switch.

Figure 5.1 Device Representation
By navigating through the trees, it is also possible to determine the approximate locations and distances of all the computer network devices with respect to each other and to the switch.

From another perspective, it is possible to determine the state of the network in terms of its traffic load. By our pre-defined conditions, the traffic load is normally associated with a particular switch and is represented by a body of water found near the mountainside. Depending on the water level of this water body, one can determine the traffic/work load demands on a particular switch. For example, in Figure 5.2, the lake found near the mountain shows that the traffic/work load associated with that particular switch is not really heavy, since the water level remains below the level of the lakeshore. To indicate heavy traffic demands on a particular switch, the water level would overflow on the banks of the lake. Through this representation, it is possible to alert the network administrator to the condition at hand.

![Virtual Lake](image)
As already mentioned, one of the advantages of virtual reality representation is its ability to afford the user/viewer different perspectives of the network system. By altering the viewpoint, varying details can be observed. In the previous figures, we have already presented examples of high-resolution representations of the system in which one can differentiate each of the individual network components. However, in our program we also succeeded in allowing the viewer to examine the overall network system. This is indicated in the following illustration (figure 5.3).

Figure 5-3 Network Overview

Although it is difficult to distinguish between the individual network devices such as the different computers this perspective allows the viewer/ user the means to examine the network in its entirety. Thus, one can study all the switches and their respective traffic/work loads, as represented by the number of mountains and lakes displayed. By our definition,
the water level of the lakes near the mountains indicate the respective traffic/ work loads associated with the switches. Thus, this representation pinpoints so-called potential ‘hot-spot’ switches. With this information and the ability to view the data at different perspectives, the network administrator can then zero in on the particular computer(s) or device(s) that is/are creating heavy network traffic loads. An example of this is illustrated in figure 5.3.

In this instance, the presence of two mountains and two lakes with normal water levels allows us to discern that there are two switches within the system, each with a fairly reasonable traffic/ workload.

Gestures and speech must complement each other and together form a unit of communication. The point gesture and the voice command move is a good example of this. One generally points at an object or in a direction and says something like ‘What is that’ or ‘I want to go there’. After a thorough research on the types of gestures that are relevant in managing a network we were unable to form a set command involving Motion Gestures that are relevant to network management. Pose gestures however, do hold a lot of possibility. Although it was our intent to include gesture recognition as part of our navigation and interaction function, it was not possible to utilise the application since the study on gesture recognition is, as yet, incomplete. Thus, no results regarding that aspect of 3-D representation in network management have been obtained.

2. Discussion

In order to establish the effectivity of our network management system, it is imperative that we compare our model to already pre-existing models. In chapter two, a list of the ideal characteristics for a network management system was outlined. Several existing network management systems were then introduced and examined, and their strong and weak points established. Based on these points and the method in which we addressed the problematic areas, we can determine the effectivity of our network management system.
Based on our research mentioned in Chapter Two, one of the problems associated with 2D visualisation systems was that they were mainly characterised by line-crossing and display clutter. However, the images generated in our project are done by a virtual reality visualisation system, which offers an additional visualising dimension that allows us more freedom to specify the location of our devices. Thus, our project is not limited by the line-crossing and display clutter factors. Furthermore, our visualisation system uses a perspective projection that allows for depth perception; i.e. it allows the user to perceive objects that are far away as small objects, while objects that are close to the user are perceived as large objects. With this depth perception, objects located some distance away from the user are displayed in less detail; thus, this information does not interfere with the detailed image representation of objects within the proximity of the user.

By our definition, a limited set of representations for the visualisation system yields the appropriate information regarding the various components of the system. This allows us the ability to derive the necessary information about the system components and to locate them within the network. Through the ability to alter the resolution of the graphical representations, it is also possible to study the system at various levels. Thus, we are able to examine the interconnection of switches within the network to pinpointing a particular device in the system. We have already demonstrated that it is also possible to deduce which network devices are associated with a specific switch. Our program allows us to distinguish between the computers, switches and the network traffic between these components. These properties further allow us to observe all the data pertaining to the system. Unlike 2D visualisation techniques that require one to hide some data in order to examine other data points, no information is hidden or omitted in our system.

To further distinguish between the components of our system, we also made use of a system-defined colour scheme. For example, a mountain is given a blue-purple colour to distinguish it from the rivers that are a bright blue. However, aside from just differentiating between the components of our system, the use of colours also indirectly offers subtle information about the system. For example, the intensity of a colour (such as the blue of the lakes or rivers indicates) a potential problem, which is worth investigating further. In this
Network Visualisation and Virtual Reality

respect, our use of colour differs from that of the Spectrum Colouring technique used in 2-D visualisation representations, which are used for indicating the status of several links in a network.

It is important to bear in mind that the main focus of our project is to represent the data collected from the network in a virtual reality representation, hence the emphasis is on the graphical aspect of managing a network system. However, some analysis is done by the network manager, which performs simple calculations that determine the usage of several computer devices. These calculations further aid in the graphical representation of the data.

Although we proposed a means to represent certain events with 3D-surround sound, our application has been hindered by the limitations of the virtual reality toolkit implemented. The CoRgi virtual reality toolkit does not allow one to make provisions for 3-D surround sound, and thus we could therefore not investigate this form of data representation.

As mentioned elsewhere, one weak point of other 3-D visual representation systems is that they are limited by the use of 2-D navigation devices such as computer mice. However, within our system we did not experience this limitation, since CoRgi supports a wide range of input devices. These include trackers, virtual hands and audio input. As already explained, we were unable to implement the gesture and speech recognition interaction and navigation techniques described in chapter four, because the research and development of those aspects were still in process at the time this project was written.
We have shown how virtual reality can be used in presenting network administrators with valuable information regarding the network, thereby greatly simplifying the tasks of network administration. By first introducing and investigating the limitations of already existing 2D and 3D visualisation systems, we were able to develop a system that counters these difficulties head on, and provides a virtual reality alternative to overcome these obstacles.

Through our virtual reality application, we were able to reduce the display clutter and line crossings associated with 2D representations. We were also able to apply a metaphor for graphically representing the network data; a concept that should be standard for other 3D visualisation techniques. The nature metaphor used in the system is well suited to visualise local area networks. It can model network components and network traffic efficiently, and it provides varying levels of abstraction. This allows several network aspects to be displayed in greater detail by moving closer to an object, or offers a ‘bird’s eye’ view of the entire network. From the latter, we can deduce the overall network performance. It is through these properties that we have overcome the limitations of spatial arrangements that allow limited views of the network, e.g. limited visualisation in VENoM.

Although we were not able to implement gesture and speech recognition techniques within our project, we still maintain that by combining virtual reality input devices with these techniques, we can devise a reasonable means of interaction in the virtual world. The combination of speech and gestures defines a type of body language that we can use to move around and interact with the components in the network. Thus, we can generate a better simulation of the real world, where people can communicate not only by the spoken word, but by means of body language as well.
Admittedly, this research project is incomplete, as there are so many other aspects that can be explored. However, given the constraints of time and the amount of technical knowledge required, it was not possible to pursue the other aspects. However, for further development of our project, we would like to point out the following aspects that could be examined in future research.

1. **Future Directions**

1. **Collaborative Virtual Reality**

As already mentioned, network management is no longer the sole responsibility of a single administrator. These days multiple administrators have to often simultaneously manage various aspects of the network. Through the development of a collaborative virtual reality extension within this project, administrators will be able to observe the tasks and progress of other systems administrators. Thus, enhancing the capabilities of the network management system.

2. **Animation**

Animation is a key component for creating realistic environments, as well as enhancing the user-experience of the virtual world. The use of animation effects can be efficiently employed to demonstrate several monitored events in the network. This allows room for more observations, and the ability to detect potential problems within the network. Thus, animation indirectly will increase the overall effectivity of the network system.
3. Navigation and Interaction

Research relating to navigation and interaction in virtual reality must be an ongoing project to find better and more efficient ways to interact and move around the virtual world. This would further enhance the accessibility and efficiency of a network system, thereby assisting administrators in maintaining their networks.
References


