Project Proposal

Christopher Schwagele
Supervisor: Barry Irwin
Computer Science Department, Rhodes University

20th February 2010

1 Principle Investigator

Christopher Zane Schwagele
czschwagele@yahoo.com
Supervised by: Barry Irwin

2 Project Title

The project is proposed under the title of:
dotNetVis: A re-implementation and enhancement of the InetVis network traffic 3D visualisation tool

3 Statement of the Problem

InetVis is a very useful tool for visualizing network traffic data-sets. This tool needs to be extended, and optimized to handle increasingly large data sets. This will involve optimisation of processing and memory usage techniques to allow full usage of the significantly improved hardware capabilities of today's computer systems. The dotNetVis system will need to be reimplemented and the User interface will need to be updated, and refined.

4 Objective of the Research

Based on the above problem statement, the objectives of this research are:

* **Primary goals**
  - Using Microsoft's Visual C#, re-implement the InetVis tool by porting over the existing C++ source
  - Study the source and refine the algorithms that are in use
  - Add a graphical overlay with useful information

- Improve on the visual representation in the 3D space

* **Secondary goals or future extensions to the work**

  - Move core functions and packet handling procedures from the CPU to the GPU through NVidia's CUDA framework
  - Use a different input source (such as a joystick) to navigate through the 3D representation of the network traffic

To re-implement the system, the current source code needs to be traversed and understood. The C# version will have to incorporate an openGL component (or an equivalent rendering framework such as Microsoft's XNA) to allow for the display of 3 dimensional graphs. The GUI needs to be improved and it should make efficient use of the real estate offered by today's monitors. Should the CUDA technology be needed, the CUDA.Net library needs to be investigated and integrated into the project as well.

5 History and Background

The multitude of problems that arise in the monitoring of networks can be classified as follows: the hardware sphere, where large volumns of data are being transported around networks at much higher rates; the security sphere where this data is vulnerable to an increasing number of vulnerabilities and exploits; as well as the availabilty and efficiency of tools developed to handle these unsolicited data transactions.

In response to these issues, a master's student at Rhodes University investigated the combined use of visualisation and dedicated sensor network monitoring methodolgies through the use of a network telescope [4]. The network telescope is able to filter network traffic by only capturing unsolicited activity. The InetVis tool was developed by van Riel as his master's project. The tool is a 3-D scatter plot concept which was originally adopted from Stephen Lau's Spinning Cube of Potential Doom [3]. InetVis plots the destination address, source address, ports and ICMP of packets captured in a network and displays the results in a 3-D representation [5]. This allows parallel inspection of the network and can identify port scans, network sweeps and step functions with ease [4].

Due to the high volume of network packets captured, processing in the InetVis tool has become an issue and can be improved. The utilisation of GPU processing power has become a feasible option on a standard machine [2]. By utilising the GPU, a parallelism approach can be taken to efficiently handle the volume of packets in the network. The reason behind the GPU's success is due to the difference in architecture that a GPU offers compared to the structure of a modern CPU [2]. Multithread support has become a valuable resource in application development and by implementing the right algorithms on the GPU,

performance can be significantly improved.

General Purpose Computing on Graphics Processing Units (GPGPU) is the utilisation of the GPU to perform computation in applications which are normally handled by the CPU. This approach is only effective when the computation requires a stream processing approach [1]. When there are set of records that require similar or identical processing, the stream processing technique applies. This is the case in the InetVis tool. Multiple packets are being collected, analysed and plotted [4]. The computation done on each packet is identical, thus GPGPU is an effective approach to solving the high volume of packets issue.

GPGPU tools have been developed by both AMD and NVidia for use with their GPUs. The more supported CUDA (an NVidia implementation) is to be used. The CUDA library was initially written for development in the C language, but recently, a CUDA.Net library has been released for development with CUDA in Microsoft's C# environment. Development in C# is less messy than it's rival, the C language as C# handles issues such as garbage collection behind the scenes.

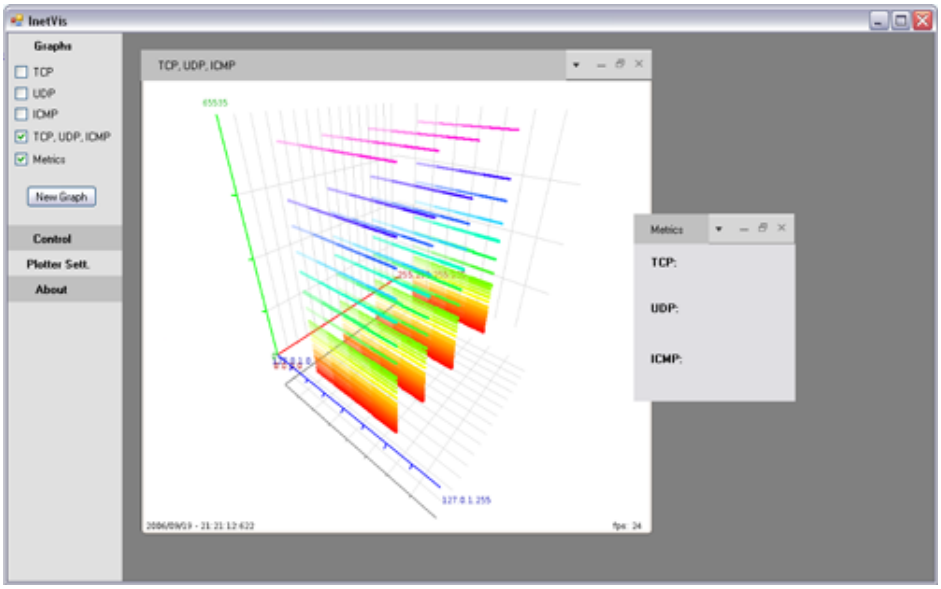
6 Approach

The first phase involves gaining familiarity and experience with the various aspects of the project. This will require that a literature survey be conducted to evaluate the currently available documentation. During this step, an understanding and proficiency in C# as well as C++ will be acquired. Familiarity with packet capturing using WinPCap will be necessary and a good understanding of Networks will be a requirement.

The second phase involves the traversal of the current InetVis source code. The algorithms need to be noted and all the components need to be outlined. Investigation into how the algorithms can be ported to C# needs to be done. While analysing the source, algorithms must be judged based on efficiency and any improvements that can be implemented need to be included into the algorithm descriptions.

Once the source has been fully examined and understood, InetVis needs to be implemented in C#. An understanding of and familiarity with packet capture in C# would also be necessary at this point.

The next phase, once InetVis has been reimplemented onto a C# platform is to enhance and extend the GUI representation of the network traffic visualisation. A careful analysis of the current GUI interface needs to take place and the problem areas need to be identified. Any issues must be researched and the best solution needs to be implemented. These enhancements will deal with packet representation, colour schemes etc. A rough layout is shown below:



Based on the understanding of InetVis, CUDA capabilities using CUDA.Net will improve the efficiency of the tool tenfold. Research into how CUDA.Net can be incorporated into InetVis needs to be done at this stage. Basically, an understanding of how GPGPU processing works is the main focus. Algorithms that run over a large amount of the same type of data can be broken down and run on the GPU instead [2]. Traffic volume should essentially be handled by the GPU at this stage.

The final stage is to complete the InetVis reimplementation and run tests. Using the original C++ implementation, results can be obtained to show whether or not this project achieved what was proposed - a more efficient implementation of InetVis to visualise higher volumes of network traffic.

7 Requirements/Resources

On Windows, the original InetVis implementation depends on the following library support:

* WinPcap (the windows equivalent of Libpcap)
* OpenGL graphics library
* MinGW (Minimalist GNU for Windows)

This project will require (in addition):

* Microsoft Visual Studio (C#)

* CUDA Driver
* CUDA SDK
* CUDA Toolkit

Minimum Hardware Requirements for Development:

* Pentium 4
* 512MB RAM
* A NVidia 3-D graphics accelerator with OpenGL hardware support.

8 Progression Time-line

Deadline	Activity
22 February	Formal Written Proposal
23 February	Presentation of Project (Seminar Series 1)
29 March	Refreshed in C++ and C# (5 weeks)
19 May	Literature review complete
24 May	C# Implementation complete (8 weeks)
19 July	C# InetVis GUI Enhanced (8 weeks)
20 July	Presentation of Project (Seminar Series 2)
17 August	Short Paper Submission
6 September	CUDA integration (7 weeks)
27 September	Testing and Evaluation (3 weeks)
4 October	Paper First Draft Handed in (1 week)
5 October	Presentation of Project (Seminar Series 3)
18 October	Final Paper Submission (2 weeks)
1 November	Project Hand-in
8 November	Website Complete

References

[1] David Luebke, Mark Harris, N. G. A. L. M. H. J. O. M. S. M. P. I. B. Gpgpu: general-purpose computation on graphics hardware. In *Conference on High Performance Networking and Computing* (2006).

[2] David Tarditi, Sidd Puri, J. O. Accelerator: using data parallelism to program gpus for general-purpose uses.

[3] Lau, S. The spinning cube of potential doom. *Communications of the ACM Archive 47* (2004), 25–26.

[4] van Riel, J.-P. Inetvis, a visual tool for network telescope traffic analysis.

[5] van Riel, J.-P. Toward visualised network intrusion detection.