Speech Friendly Packet Sniffer

by

Chin Chin SU
(12545069)

A thesis submitted for the degree of
Bachelor of Technology (Computer Systems and Networking)

Curtin
UNIVERSITY OF TECHNOLOGY

Project Documentation Sheet

TITLE	Speech Friendly Packet Sniffer		
AUTHOR			
Family Name:	SU		
Given Name:	Chin Chin		
DATE	4 November 2004	SUPERVISOR	Iain Murray
DEGREE	Bachelor of Technology	OPTION	Computer Systems & Networking

ABSTRACT:

The Speech Friendly Ethereal Packet Sniffer enables blind and visually impaired people to use Ethereal to troubleshoot network problems. The end product of this project is a version of Ethereal that integrates with Gnopernicus and GNOME libraries.

INDEXING TERMS:

	GOOD	AVERAGE	POOR
TECHNICAL WORK			
REPORT PRESENTATION			

EXAMINER:

CO-EXAMINER:

5/72 Kooyong rd.
RIVERVALE WA 6103

15 November 2004

Professor Syed Islam
Head of Department
School of Electrical and Computer Engineering
Curtin University of Technology
Kent Street
BENTLEY WA 6102

Dear Sir,

I am pleased to submit this thesis entitled “Speech Friendly Packet Sniffer” as part of the requirements for the degree of Bachelor of Technology in Computer Systems and Networking.

I declare that this is my own work and have not been provided to other people.

Sincerely Yours,

Chin Chin SU

SYNOPSIS

This thesis outlines the integration of the ethereal packet sniffer with Gnopernicus and GNOME libraries. It also outlines the testing and analysis of the final product, which is an ethereal packet sniffer that allows blind and visually impaired users to use it.

Ethereal packet sniffer was chosen because it is open source and it is used worldwide. Its source code is free and thus can be modified. Gnopernicus screen reader, which comes with Fedora Linux core 1 package, is incorporated with Festival speech synthetisers to allow sound output from the ethereal.

ACKNOWLEDGEMENTS

I would like to thank the following people:

- Mr. Iain Murray, my project supervisor, for his constant guidance, support and help throughout the course of this project,
- Mr. Alex Wong, my Cisco lab tutor, for his help in Gnopernicus and Fedora Linux installation, also in introducing Glade,
- Rudy Lee, my friend, for his help in integrating Ethereal source code with Gnopernicus and GNOME libraries,
- Silverio Armani, my boyfriend, for his help in debugging Ethereal source code,
- Leslie Riach, my good friend, and Lichen Su, my sister, for their help in editing my thesis.

I appreciate their patience, experience and understanding towards the completion of this project.

NOMENCLATURE

ABWA	Association for the Blind of Western Australia
CAVI	Cisco Access for the Vision-Impaired
GNOME	GNU Object Model Environment; it is a free desktop environment for UNIX and UNIX-like operating systems
GNU	GNU's not UNIX; refers generally to software distributed under GPL (Mcarty, 1999)
GPL	GNU Public License. It provides for free access to software published under its terms. Users are allowed to copy, modify, and redistribute GPL software (Mcarty, 1999)
GTK	GIMP Tool Kit. It is a multi-platform toolkit for creating GUI.
GUI	Graphical User Interfaces
TCP/IP	Transmission Control Protocol / Internet Protocol; a standard method of sending data packets across a computer network, focused on the routing and connection issues (Mcarty, 1999)
TTS	Text To Speech; it is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a web page. TTL is often used with voice recognition programs (Sticha, n.d)
Widgets	A data structure that normally includes information about a user-interface component (like a button or an edit box) and has pointers to the code needed to make the component work (USGS, 2004)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

Computer networking has advanced greatly and so has the technology used to access people’s internal networks. Nowadays, tools like packet sniffers, network analysers, packet replays and denial of service attack scripts are very common. As such, network administrators are faced with the challenge of trying to secure their networks from both internal and external intruders.

Packet sniffer, for instance, can be used to troubleshoot a network problem by capturing a packet and then analysing it. However, it can also be used to sniff a packet off from a network in order to obtain specific information, such as username and password to gain unauthorised access to the network.

With the increasing challenge to secure a network, it has become imperative to implement good network security policies and have regular network monitoring at all times. It is important for anyone to be able to use the network security tools available. This includes those with disabilities including visually impaired people.

Cisco in conjunction with Curtin University and the Association for the Blind of Western Australia (ABWA) have created a program called Cisco Access for the Visually Impaired, CAVI, to assist visually impaired people to gain skills in networking and computer technology (Waltner, 2003).

This thesis aims to developing a speech friendly packet sniffer so that visually impaired people are able to use it. It begins with an explanation of the existing assistive technologies available for visually impaired people. This is followed by an explanation of what packet sniffers are. A comparison of some of the freely available packet sniffers is provided as well.

An explanation as to why Ethereal and Gnopernicus are chosen is provided, and this is followed by a discussion on the integration of the Ethereal with Gnopernicus.

The thesis concludes with a description of the testing of the end product and the results.

2.0 COMPUTER TECHNOLOGY AVAILABLE FOR THE VISUALLY IMPAIRED

2.1 Overview

Today’s computer technology relies heavily on sight for information and communication. Therefore, visually impaired people may have trouble using technology and may feel isolated. For these reasons, tools have been created to assist them.

Braille presentation has become one of the dominant types of computer accessibility methods for the visually impaired. However, text information on computer screen needs to be translated into Braille alphabets and printed out. This process is slow and not efficient. A real-time process would be preferred.

Moreover, the computer cursor can be too small for visually impaired people and is not accessible for the blind. Therefore, the keyboard is an important tool for them for inputting data into the computer, navigating file directories and selecting icons. Hence, it is important to design software with keystroke accessibility; for instance, CTRL+O can be used to open a file in Microsoft Word.

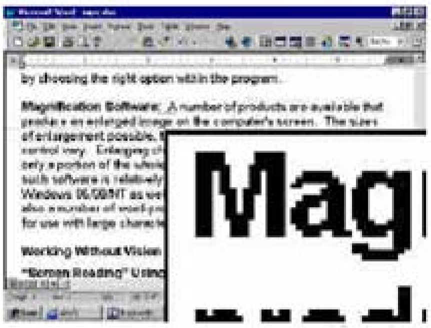
As a result, many companies such as Microsoft, Adobe and Macintosh have integrated accessibility, such as magnifier, narrator and on-screen keyboard, into their products. These allow increased accessibility for visually impaired people.

2.2 Assistive Technology for the Visually Impaired

Assistive technology refers to technology that has been modified to assist individuals who have physical or cognitive difficulties, impairments or disabilities (http://www.microsoft.com/enable/at/types.aspx).

Some examples of assistive technology for the visually impaired are screen magnification, Braille presentation, screen readers and speech synthesisers.

Screen magnification works like a magnifying glass. It enlarges screen text and graphics to a size between 2 and 20 times their original sizes. Common examples are Zoomtext, InLarge, Magic, Magnum Deluxe and Luna.



by choosing the right option within the program.

Magnification Software: A number of products are available that produce an enlarged image on the computer's screen. The sizes
of enlargement possible,
control vary. Enlarging ch
only a portion of the whole
such software is relatively
Windows 95/98/NT as we
also a number of word p
for use with large charact

Working Without Vision

"Screen Reading" Using

Figure 1: Screen Magnification

Braille embossers print Braille on paper. Common brands are Everest, Versapoint, Thiel, Romeo, Bookmaker and Braille Express embossers.

Figure 2: Braille Embosser

Braille displays are devices that convert text from the screen into Braille. They use a series of electronic “pins” that are either in the up or down position, to represent the Braille characters so that blind and visually impaired people can read them. Common models of Braille Display are Alva, Navigator, PowerBraille and Tieman ranges (DATWA, nd).

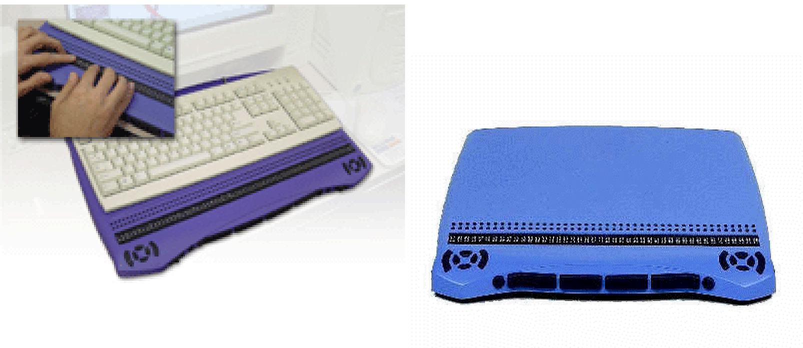


Figure 3: Braille Display

A screen reader is software that works together with a speech synthesizer to read aloud information such as icons, menus, text, punctuation, and control buttons from a computer screen. Table 1 lists a few screen readers running under Windows and Linux platforms.

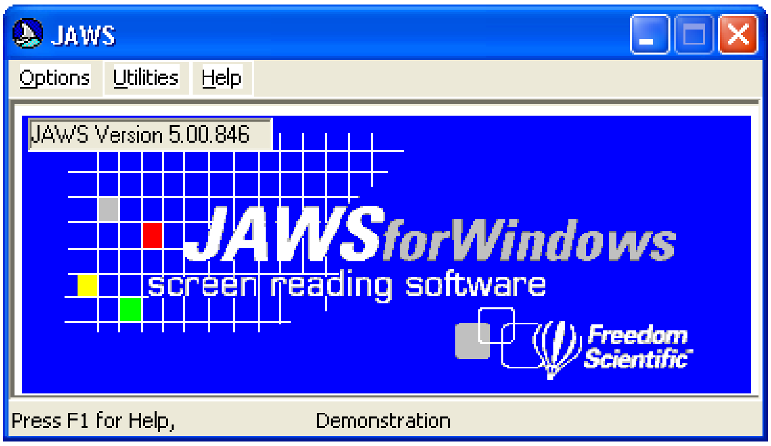
Operating System	**Screen Reader**	**Availability (Price quoted in September 2004)**	**Website/Company**
Windows	HAL95, HAL NT	Commercial (1,218 DM, 1,856 DM)	Dolphin Computer Access
Windows	Insight	Commercial (around 2000 DM)	ECO, Berlin
Windows	Blindows	Commercial (around 1000 to 4000 DM)	Franc Audio Data (http://www.audiodata.de)
Windows	JAWS	Commercial (2100 AU)	Freedom Scientific (http://www.freedomscientific.com) Demo version can be downloaded with restricted timing usage.
Windows	Virgo	Commercial	Demo version can be downloaded from www.virgo4.com
Linux	Gnopernicus	Free	Baum (http://www.baum.ro)

Table 1: Screen readers available for Windows and Linux

A brief description of the screen readers that relate to this project is listed as follows:

- JAWS

  JAWS stands for Job Access With Speech. It is a screen reader that reads and magnifies information in the computer screen. It not only outputs refreshable Braille displays but also provides unmatched Braille support to all screen readers on the market (Freedom Scientific, 2003). It also supports many different languages and most standard Windows applications without the need to do special configuration, including Microsoft Office XP.



JAWS Version 5.00.846

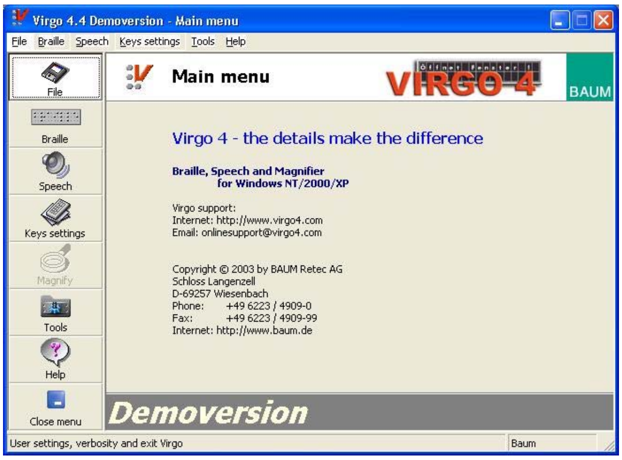
Press F1 for Help,

Demonstration

Figure 4: JAWS 5.0 (demo version)

- Virgo

This screen reader reads text from the screen and outputs it as speech via a speech synthesizer or a sound card. It runs, however, only on Windows platforms. The information can be presented on a Braille display and output using a speech synthesizer. The display screen can be magnified to display the information on the screen using the size and colours to meet the users' requirements. Figure 5 shows the main menu of Virgo 4.4 (demo version).



Virgo 4 - the details make the difference

Braille, Speech and Magnifier
for Windows NT/2000/XP

Virgo support:
Internet: http://www.virgo4.com
Email: onlinesupport@virgo4.com

Copyright © 2003 by BAUM Retec AG
Schloss Langenzell
D-69257 Wiesenbach
Phone: +49 6223 / 4909-0
Fax: +49 6223 / 4909-99
Internet: http://www.baum.de

User settings, verbosity and exit Virgo

Baum

Figure 5: Virgo 4.4 (demo version)

- Gnopernicus

This screen reader enables blind and visually impaired users to use GNOME desktop and Gnome/GTK+ applications effectively. GTK+ stands for GIMP Tool Kit. It is a toolkit for creating graphical user interfaces (GUI) and offers a complete set of widgets. It is free software and part of the GNU project.

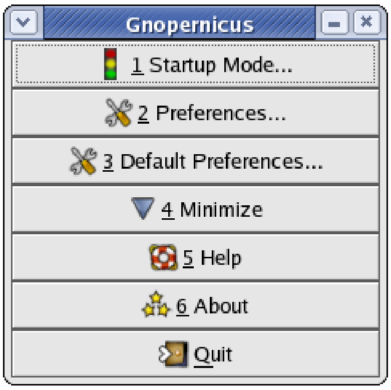


Figure 6: Gnopernicus screen reader for GNOME desktop

Table 2 provides more information about Gnopernicus as extracted from Gnopernicus website.

Name	Gnopernicus
Current version (October 2004)	0.3.4
Operating System	Linux, Solaris
Language versions	English, German, Swedish
Speech Synthetisers	Festival, Via Voice, FreeTTS, Dec-Talk (through gnome-speech)
Braille devices supported	VARIO20, VARIO40, VARIO80, DM80p, INKA ALVA 380, ALVA 544, ALVA 570 BRLTTY supported drivers
Sound output device	All sound cards (supported by TTS)

Table 2: Gnopernicus screen reader product information

Speech synthesisers may be software or hardware and are used to convert screen contents into spoken words using synthetic speech. Some examples are:

- Festival

  It is a speech synthesizer written in C++ that offers full text-to-speech conversion through a number of APIs. It was developed by the Centre for Speech Technology Research at University of Edinburgh. It provides a binding to the Java Speech API and is currently available in British and American English, Spanish and Welsh. Other languages are in development and documentation on how to build new languages is available through Carnegie Mellon's FestVox project (http://festvox.org). Festival is free and its speech tools are distributed under an X11-type licence allowing unrestricted commercial and non-commercial use as such it is constantly being improved.

- FreeTTS

  This is another speech synthesizer that is written entirely in Java^{TM} programming language. It is based upon a small run-time speech synthesis engine developed at Carnegie Mellon University called Flite. Flite is derived from the Festival and FestVox projects. Building, running, and testing FreeTTS requires Java^{TM} 2 SDK, Standard Edition, 1.4. FreeTTS is mainly in English; however, it also supports a number of languages such as Spanish, Welsh and other languages, which can be imported from FestVox project website.

- ViaVoice

  This is an IBM speech synthesizer. It is a commercial product running under Windows, Macintosh and handheld computer platforms. IBM offers many different editions of ViaVoice to suit user's needs such as ProUSB, Advanced, Standard and Personal edition for Windows platform, and MacOS X edition and Simply Dictation MacOS X edition for Macintosh platform. ViaVoice Standard edition offers features such as direct dictation into Microsoft Word products, Internet browser voice-command and creation of customised dictation words, addresses and acronyms.

3.0 PACKET SNIFFERS

3.1 Overview

Packet sniffer is a dedicated device or software package designed for monitoring network traffic in order to recognize and decode certain packets of interests (ATIS Committee T1A1, 2001). It is generally used by system administrators for network management and diagnostics. However, it can also be used by hackers to steal usernames, passwords and credit card numbers to make illegal purchases, capture and replay Voice over IP telephone conversations, map networks and so on.

The issue of network security is beyond the scope of this thesis. This thesis mainly focuses on how to develop a speech friendly packet sniffer so that visually impaired people can use it to troubleshoot network problems and help to maintain networks.

Packet sniffers are available freely and commercially. Descriptions to the freely available packet sniffers are described next.

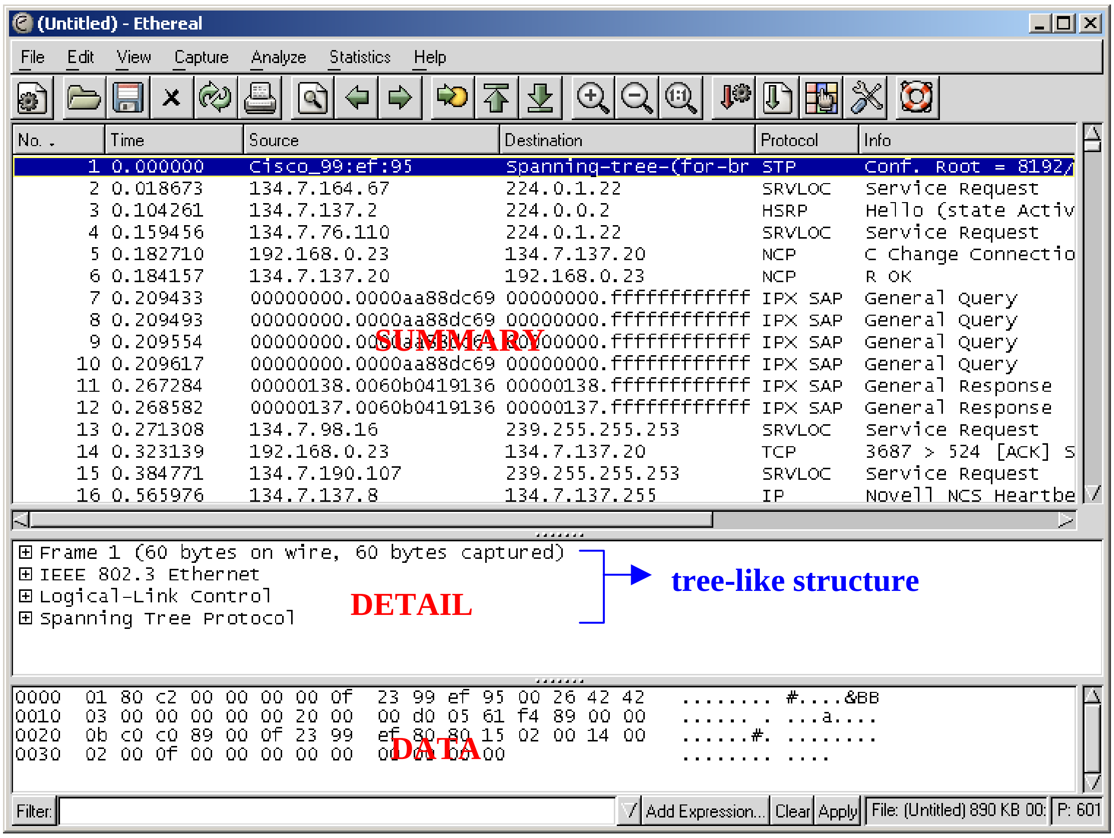
3.2 Ethereal

Ethereal is an open source packet sniffer that captures packets off the network, decodes them and presents them in a format that is easy to understand. It is mainly used for analysing networking problems, such as unreachable network devices, denial of service, and so on.

Ethereal has an easy to read and very configurable Graphical User Interface (GUI) with a rich display of filter capabilities such as tcpdump format capture filters. It is available in precompiled binaries and source code. It can be run on over 20 UNIX platforms such as IBM AIX, Linux, HP UX, OpenBSD and Solaris 8, and Windows platforms such as Windows XP, Windows 2000, Windows NT and Windows Me/98/95. However, it requires GTK+, GLIB and libpcap to run.

Ethereal supports over 480 protocols. Because the source code is freely available, it is very easy for programmers to add new protocols to Ethereal, either as modules, or built into the source. The version used for this project is 0.10.5.

Figure 7 shows the Ethereal packet sniffer display window. This display window consists of three panes: summary, detail and data. The summary pane displays a one-line summary of the capture. The detail pane provides all of the details of each of the layers contained inside the captured packet in a tree-like structure and the data pane displays the raw data captured in hexadecimal and ASCII format (Orebaugh et.al., 2004).



```
No. .  Time      Source                     Destination                Protocol Info
    1 0.000000  Cisco_99:ef:95             Spanning-tree-(for-br      STP      Conf. Root = 8192/
    2 0.018673  134.7.164.67               224.0.1.22                 SRVLOC   Service Request
    3 0.104261  134.7.137.2                224.0.0.2                  HSRP     Hello (state Activ
    4 0.159456  134.7.76.110               224.0.1.22                 SRVLOC   Service Request
    5 0.182710  192.168.0.23               134.7.137.20               NCP      C Change Connectio
    6 0.184157  134.7.137.20               192.168.0.23               NCP      R OK
    7 0.209433  00000000.0000aa88dc69 00000000.ffffffffffff IPX SAP  General Query
    8 0.209493  00000000.0000aa88dc69 00000000.ffffffffffff IPX SAP  General Query
    9 0.209554  00000000.0000aa88dc69 00000000.ffffffffffff IPX SAP  General Query
   10 0.209617  00000000.0000aa88dc69 00000000.ffffffffffff IPX SAP  General Query
   11 0.267284  00000138.0060b0419136 00000138.ffffffffffff IPX SAP  General Response
   12 0.268582  00000137.0060b0419136 00000137.ffffffffffff IPX SAP  General Response
   13 0.271308  134.7.98.16                239.255.255.253            SRVLOC   Service Request
   14 0.323139  192.168.0.23               134.7.137.20               TCP      3687 > 524 [ACK] S
   15 0.384771  134.7.190.107              239.255.255.253            SRVLOC   Service Request
   16 0.565976  134.7.137.8                134.7.137.255              IP       Novell NCS Heartbe
```

```
⊞ Frame 1 (60 bytes on wire, 60 bytes captured)
⊞ IEEE 802.3 Ethernet
⊞ Logical-Link Control
⊞ Spanning Tree Protocol
```

```
0000  01 80 c2 00 00 00 00 0f  23 99 ef 95 00 26 42 42   ........ #....&BB
0010  03 00 00 00 00 00 20 00  00 d0 05 61 f4 89 00 00   ...... . ...a....
0020  0b c0 c0 89 00 0f 23 99  ef 80 80 15 02 00 14 00   ......#. ........
0030  02 00 0f 00 00 00 00 00  00 01 00 00              ........ ....
```

Figure 7: Ethereal packet sniffer display

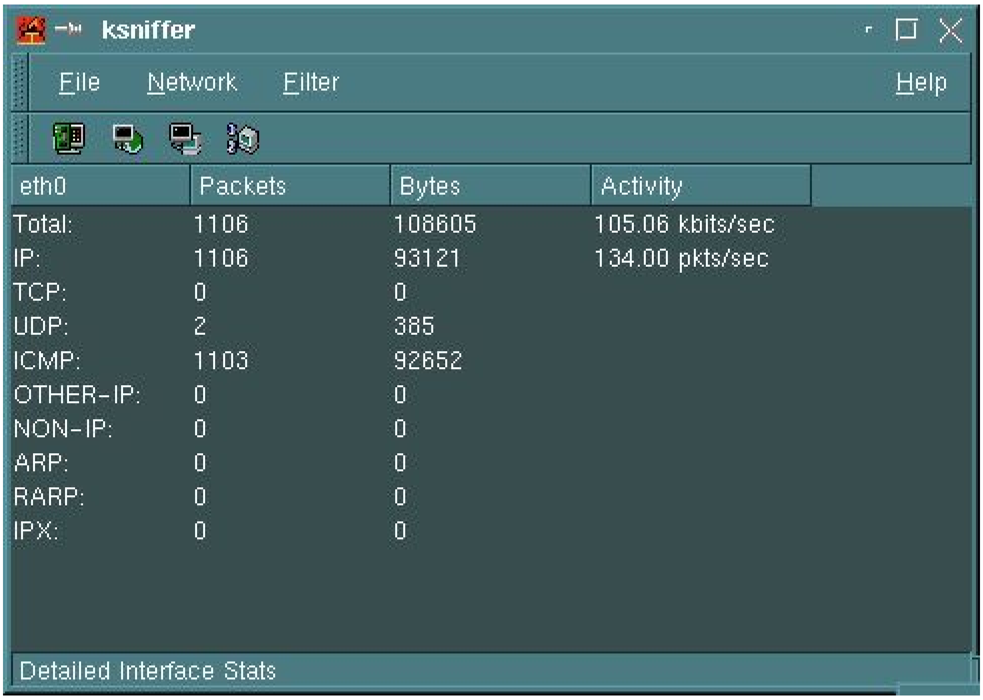
The detail pane provides detailed information on a captured packet. One packet is selected in the summary pane in Figure 7, one at a time, by using a down arrow key. Its detailed information can be obtained by using a ‘TAB” key to switch to the next pane. A “SHIFT+TAB” key is used to switch to the previous pane. This is useful for blind and visually impaired people who cannot use mouse.

A plus sign (+) in the detail pane is used to maximize the nested texts in the detail pane and a minus sign (-) to minimize them.

3.3 KSniffer

KSniffer is a network statistics collector (Donahoo, n.d.). It collects the number of packets and number of bytes for each protocol and displays the activity in terms of Kbits/s, Kbytes/s and packets/s. KSniffer supports most TCP/IP protocols such as TCP, IP, UDP, ICMP, ARP, RARP as well as minimal IPX. It can also be used to watch specific port traffic in monitoring network services such as http, ftp, and telnet.

A snapshot of the Ksniffer display with detailed interface statistics is provided in Figure 8. It contains fewer menu items and captures fewer protocols than Ethereal. In general, it is simpler than Ethereal.



eth0	Packets	Bytes	Activity
Total:	1106	108605	105.06 kbits/sec
IP:	1106	93121	134.00 pkts/sec
TCP:	0	0	
UDP:	2	385	
ICMP:	1103	92652	
OTHER-IP:	0	0	
NON-IP:	0	0	
ARP:	0	0	
RARP:	0	0	
IPX:	0	0	

Detailed Interface Stats

Figure 8: KSniffer display

3.4 IpGrab

IpGrab is a verbose packet sniffer that displays a great amount of detail on each of the packet it reads, including the application layer fields (SourceForge.net, 2004). A sample output is shown below (SourceForge.net, 2004).

```
----------------------------------------------------------------
                        Ethernet header (961445334.490653)
----------------------------------------------------------------
Hardware source:        00:10:4b:96:1d:a8
Hardware destination:   08:00:02:25:29:77
Protocol:               0x800 (IP)
Length:                 68

----------------------------------------------------------------
                        IP Header
----------------------------------------------------------------
Version:                4
Header length:          5
TOS:                    0x10
Total length:           54
Identification:         6795
Fragmentation offset:   0
Unused bit:             0
Don't fragment bit:     1
More fragments bit:     0
Time to live:           64
Protocol:               6 (TCP)
Header checksum:        37890
Source address:         149.112.60.156
Destination address:    149.112.36.168
----------------------------------------------------------------
                        TCP Header
----------------------------------------------------------------
Source port:            2692 (unknown)
Destination port:       23 (telnet)
Sequence number:        2876130028
Acknowledgement number: 3994633468
Header length:          8
Unused:                 0
Flags:                  PA
Window size:            32120
Checksum:               58743
Urgent:                 0
Option:                 1 (no op)
Option:                 1 (no op)
Option:                 8 (timestamp)
  Length:               10
  Timestamp value:      181028495
  Timestamp reply:      44432019
-----------------------------------------------------------------
0D 00                                                 ..
```

```
And of a minimal mode TCP session:
961445601.933843 00:10:4b:96:1d:a8->08:00:02:25:29:77 IP
149.112.60.156->198.186.203.44 (72,DF) TCP 2690->22
(PA,2794909852,2764779739,31856)

961445602.086258 08:00:02:25:29:77->00:10:4b:96:1d:a8 IP
198.186.203.44->149.112.60.156 (52,DF) TCP 22->2690
(A,2764779739,2794909872,32120)

961445602.086374 08:00:02:25:29:77->00:10:4b:96:1d:a8 IP
198.186.203.44->149.112.60.156 (96,DF) TCP 22->2690
(PA,2764779739,2794909872,32120)

961445602.097106 00:10:4b:96:1d:a8->08:00:02:25:29:77 IP
149.112.60.156->198.186.203.44 (52,DF) TCP 2690->22
(A,2794909872,2764779783,31856)
```

Unlike Ethereal and KSniffer that are GUI-based, IpGrab is text-based and runs on FreeBSD and Linux platforms only.

3.5 IpLog

IpLog is a TCP/IP traffic logger. It is used to monitor and log TCP, UDP and ICMP traffic. It is also able to detect TCP port scans, TCP null scans, FIN scans, UDP and ICMP "smurf" attacks, bogus TCP flags, TCP SYN scans, TCP "Xmas" scans, ICMP ping floods, UDP scans and IP fragment attacks (McCabe, 2001).

Similar to IpGrab, IpLog is text-based, but it can run on other platforms such as POSIX BSD, FreeBSD, NetBSD, OpenBSD, POSIX IRIX, Linux and Sun OS / Solaris.

4.0 DEVELOPING A SPEECH FRIENDLY PACKET SNIFFER

4.1 Packet Sniffer Selection

Ethereal was chosen among KSniffer, IpGrab and IpLog for this project because it is open source, GUI-based, supports multiple protocols and platforms, provides more features and is widely used.

4.2 Screen Readers Analysis

JAWS is unable to read Ethereal because it is a commercially developed screen reader. Its source code is not available and thus it is almost impossible to know what libraries are required to be integrated into Ethereal to make it useable by JAWS screen reader.

Virgo is the Windows version of the Gnopernicus. It can read Ethereal but is not able to read the Ethereal menus.

Simultaneously running JAWS 4 and Virgo 4.4 causes the system to crash because Virgo 4.4 intercepted the video card and would not allow JAWS to use it. However, running a more recent version of JAWS (version 5) and Virgo 4.4 does not result in system crash.

Both Virgo and JAWS are not capable of reading Ethereal. JAWS could not read Ethereal at all, while Virgo could not respond to the Ethereal menu widgets. Due to unavailability of both of the source codes, further investigation could not be carried out at the time of this thesis.

Gnopernicus is free and runs under Linux. The 0.3.4 version is used in this project with the Festival screen synthesizer. It is quite complicated to configure and build because it requires many files with many dependencies. One file depends on many other files. Gnopernicus and Festival installation steps and notes can be found in the Appendix.

Gnopernicus is unable to read Ethereal as well. However, it could read other GNOME/GTK+ applications, such as the Glade application that is part of Fedora Linux package. Next section will discuss how to make Gnopernicus reads Ethereal.

4.3 Ethereal Source Code Integration

Ethereal source code is freely downloadable from its website (http://www.ethereal.com). Ethereal version 0.10.5 is used for this project.

Gnopernicus has many dependencies/libraries. In order to read one application, its libraries must be imported into the application. The CFLAGS in both Ethereal-0.10.5/Makefile and Ethereal-0.10.5/gtk/Makefile must be included in the following:

```
CFLAGS   =   -DORBIT2=1   -pthread   -I/usr/include/libgnomeui-2.0   -
I/usr/include/libgnome-2.0     -I/usr/include/libgnomecanvas-2.0    -
I/usr/include/gtk-2.0           -I/usr/include/libart-2.0           -
I/usr/include/gconf/2        -I/usr/include/libbonoboui-2.0         -
I/usr/include/glib-2.0         -I/usr/lib/glib-2.0/include          -
I/usr/include/orbit-2.0        -I/usr/include/libbonobo-2.0         -
I/usr/include/gnome-vfs-2.0    -I/usr/lib/gnome-vfs-2.0/include     -
I/usr/include/bonobo-activation-2.0     -I/usr/include/pango-1.0    -
I/usr/include/freetype2         -I/usr/lib/gtk-2.0/include          -
I/usr/include/atk-1.0 -I/usr/X11R6/include -I/usr/include/libxml2\
      -pthread  -Wl,--export-dynamic  -L/usr/X11R6/lib  -lgnomeui-2  -
lSM -lICE -lbonoboui-2 -lxml2 -lpthread -lz -lgnomecanvas-2 -lgnome-
2  -lpopt  -lart_lgpl_2  -lpangoft2-1.0  -lgtk-x11-2.0  -lgdk-x11-2.0  -
latk-1.0  -lgdk_pixbuf-2.0  -lpangoxft-1.0  -lpangox-1.0  -lpango-1.0  -
lbonobo-2  -lgconf-2  -lgnomevfs-2  -lbonobo-activation  -lORBit-2  -
lgobject-2.0 -lgthread-2.0 -lm -lgmodule-2.0 -ldl -lglib-2.0
```

Makefile is used to define a set of compilation rules. It contains comments, dependency lines, directories, macros, response files, rules and shell lines in order to create the executable file.

A long line in a Makefile can be broken up by putting “\”as the last character of the line. For example,

first_part_of_line second_part_of_line

is the same as:

```
first_part_of_line \
second_part_of_line
```

The gnome header file, gnome.h, and gnome initialisation function (gnome_program_init) must also be included in main.c file located in ethereal-0.10.5/gtk.

Gnome_program_init function is used to initialise the current GNOME libraries. Each application needs to call this function early in its lifetime. Users have to specify which modules should be initialised and if there is any special parameter value that needs to be passed on to the modules. The modules in this context refer to libraries needed for the application, for example, libgnome_module or libgnomeui_module in Gnopernicus.

The following code is integrated into the Ethereal source code:

in main.c file located in Ethereal-0.10.5/gtk/main.c:

```
#include <gnome.h>
```

in int main(int argc, char *argv[]) function:

```
gnome_program_init ("ethereal", NULL, LIBGNOMEUI_MODULE,
                      argc, argv,
                      GNOME_PARAM_APP_DATADIR, NULL, NULL);
```

As long as the above two lines of codes in bold are integrated in any GNOME applications, the program will be readable by Gnopernicus.

Additonally, Ethereal must be configured and built every time it is modified. The commands “./configure”, “make” and “make install” are used for this purpose.

5.0 ETHEREAL TESTING

5.1 Results of Testing

Gnopernicus is able to read some portions of Ethereal only if Festival speech synthesizer has been installed into the system. It reads the menus and the detail pane as shown in Figure 7. However, it cannot read the summary and data panes.

A few debugging messages are generated as shown:

```
[root@localhost ethereal-0.10.5]# ./ethereal
Bonobo accessibility support initialized
GTK Accessibility Module initialized
Atk Accessibilty bridge initialized

(lt-ethereal:3166): Gtk-CRITICAL **: file gtkwindow.c: line 2973
(gtk_window_resize): assertion `height > 0' failed

(lt-ethereal:3166): Gtk-CRITICAL **: file gtkwindow.c: line 2973
(gtk_window_resize): assertion `height > 0' failed

** (lt-ethereal:3166): CRITICAL **: file gailclist.c: line 816
(gail_clist_get_column_header): assertion `GTK_IS_BIN
(return_widget)' failed

** (lt-ethereal:3166): CRITICAL **: file gailclist.c: line 816
(gail_clist_get_column_header): assertion `GTK_IS_BIN
(return_widget)' failed

 [root@localhost ethereal-0.10.5]#
```

The first three lines notify that Gnopernicus has been initialised and is ready to read Ethereal. The "assertion height > 0 failed" message is a bug in Ethereal-0.10.5 and 0.10.5a, which has been fixed in ethereal-0.10.6 and later versions (Ethereal, 2003). When running Ethereal-0.10.6, this message does not occur but the "GTK_IS_BIN" message was found. It is believed that this may be a bug in ethereal-0.10.6.

5.2 Effectiveness of the speech-friendly Ethereal

The speech-friendly Ethereal is very useful for blind and visually impaired people as it enables them to utilise Ethereal to troubleshoot network problems. It is not yet very effective because not all parts of the Ethereal are readable by Gnopernicus screen reader.

Gnopernicus can only read certain GTK widgets but cannot read customised ones. Ethereal uses customised widgets to display a list of sniffed packets therefore Gnopernicus cannot read the items in the list widget.

Gnopernicus only reads widgets that can be focused on. It does not read static widgets such as labels therefore all descriptive labels are ignored by the screen reader.

Ethereal’s online help uses a read-only text list widget to display the manual pages therefore it cannot be focused on via keyboard. However, the user can highlight the text in the list by using mouse so Gnopernicus will read it out but it would be difficult for a visually impaired person to do so.

Moreover, inactive widgets are partially readable by Gnopernicus. It tells whether the menu items are inactive but do not read their captions. It does not read inactive combo boxes either.

6.0 CONCLUSION

Ethereal packet sniffer is open source; it is free, modifiable, performs better and runs under Linux. It is written using GTK+ libraries that are partly readable by Gnopernicus screen reader.

The source code of Ethereal has to include Gnopernicus libraries/dependencies, as well as GNOME initialisation function in order to make it speech friendly.

7.0 FUTURE WORK

As mentioned in section 5.1 and 5.2, Gnopernicus could only read some portions of the Ethereal. It is suggested to alter Gnopernicus modules in order to read the customised components and descriptive label of the widgets used in Ethereal.

8.0 REFERENCES

AlltheScripts, 2004, ‘EtherDetect Packet Sniffer’, retrieved: 25 October 2004 from http://www.allthescripts.com

Angela O., Greg M., Ed W., and Gilbert R., 2004, ‘Ethereal Packet Sniffing’, Syngress Publishing, Inc., USA

ATIS Committee T1A1, 2001, ‘Packet Sniffer’, retrieved: 25 October 2004 from http://www.atis.org/tg2k/_packet_sniffer.html

Baum Engineering SRL, 2003, ‘Gnopernicus Information’, retrieved: 9 July 2004 from http://www.baum.ro/gnopernicus.html

Baum Engineering SRL, 2003, ‘Virgo 4’, retrieved: 6 June 2004 from http://www.virgo4.com

Black, A.W., Clark R., Richmond K., King S. & Zen H., 2004, ‘The Festival Speech Synthesis System’, retrieved: 14 July 2004 from http://www.cstr.ed.ac.uk/projects/festival

DATWA, nd, ‘What is Adaptive Technology’, retrieved: 25 October 2004 from http://www.abwa.asn.au/datwa/what_is_adaptive_technology_.html

Degioanni L., 2004, ‘WinPcap: The Free Packet Capture Architecture for Windows’, retrieved: 27 April 2004 from http://winpcap.polito.it

Donahoo, M. J., nd, ‘KSniffer’, retrieved: 25 October 2004 from http://cs.ecs.baylor.edu/~donahoo/tools/sniffer/ksniffer.htm

Ethereal, 2003, ‘Ethereal: A Network Protocol Analyzer’, retrieved: 20 March 2004 from http://www.ethereal.com

Fedora Project, 2003, ‘What is the Fedora Project?’, retrieved: 25 October 2004 from http://fedora.redhat.com/

Freedom Scientific, 2003, ‘JAWS for Windows’, retrieved: 25 October 2004 from http://www.freedomscientific.com/fs_products/software_jaws.asp

GNOME CVS Repository, nd, ‘Gnopernicus Changes’, retrieved: 25 October 2004 from http://cvs.gnome.org/viewcvs/gnopernicus/NEWS?rev=1.50 and http://cvs.gnome.org/viewcvs/gnopernicus/README?rev=1.80

IBM, 2003 ‘ViaVoice’, retrieved: 14 July 2004 from http://www-306.ibm.com/software/voice/viavoice/

McCabe, R., 2001, ‘IpLog’, retrieved: 25 October 2004 from http://freshmeat.net/projects/iplog/

Microsoft, 2004, ‘Types of Assistive Technology Products’, retrieved: 25 October 2004 from http://www.microsoft.com/enable/at/types.aspx

Mcarty, B., 1999, ‘Learning Debian GNU/Linux – Glossary’, retrieved: 4 November 2004 from http://www.oreilly.com/catalog/debian/chapter/book/glossary.html

SourceForge.net, 2004, ‘IpGrab’, retrieved: 25 October 2004 from http://ipgrab.sourceforge.net/

Sticha, D., n.d., ‘VoiceXML - Glossary’, retrieved: 4 November 2004 from http://filebox.vt.edu/users/dsticha/vxml/glossary.html

USGS, 2004, ‘Xtme Documentation Glossary’, retrieved: 4 November 2004 from http://geology.usgs.gov/tools/metadata/tools/doc/xtme/glossary.shtml

Sun Microsystems, Inc., 2002, ‘FreeTTS 1.2beta2 - A speech synthesizer written entirely in the Java^{TM} programming language’, retrieved: 11 July 2004 from http://freetts.sourceforge.net

Waltner C., 2003, ‘Cisco Networking Academy Instructor Pioneering Classes, Tools for Visually Impaired Students’, News @ Cisco

APPENDIX

Gnopernicus Installation

The following libraries are required to install Gnopernicus on the system:

GNOME libraries:

- cspi-1.0 (at-spi package)
- gconf-2.0
- gtk+-2.0
- gtk -doc (optional-necessary just for docs generation)
- libglade-2.0
- libgnome-2.0
- libgnomeui-2.0
- libxml-2.0
- gnome-speech
- gnome-mag (optional - if not installed, there will not be magnifier support in gnopernicus)

Non-GNOME packages (optional):

- Text-to-Speech engines (used for speech):
    - Festival
    - ViaVoice
    - FreeTTS
- Braille drivers (used for Braille):
    - BRLTTY

GNOME libraries must be installed first before installing Gnopernicus, otherwise the following GNOME packages must be installed and built to provide the libraries above: ***at-spi, gconf , gtk+, [gtk -doc], libglade, libgnome , libgnomeui, gnome-xml, gnome-speech, [gnome- mag]***. Note that “[]” means optional.

In addition, Gnopernicus must have a speech synthesizer installed in order to read out information on the screen. It is recommended to use one type of speech synthesizer because Gnopernicus has not supported speech markers (GNOME CVS Repository, n.d.). The lack of the markers implicates that not the entire presentation chunk will be spoken. A better support for voices that do not support speech markers had been added to Gnopernicus 0.3.3 (GNOME CVS Repository, n.d.).

Installation Notes:

- Installing from scratch

  Installing Gnopernicus from scratch requires a lot of work because Gnopernicus has a lot of dependencies. When running "./configure" on the system, it produced an error stating that installation was not able to proceed because there were no complete libraries. It advised to try adjusting the $PKG_CONFIG_PATH environment variable by typing “set $PKG_CONFIG_PATH=/usr/lib/pkgconfig”, but it still would not work. The command pkg-config is used to check if the system has all the necessary libraries required to compile Gnopernicus. The value for pkg_config_path environment variable was set to be the directory that we wanted pkg-config to search for. That is, the directory that has all the *.pc files. However, most of the *.pc files were already in the default directory.

- Installing from Fedora Linux Package

  Fedora Linux is an openly developed project designed by Red Hat. It is an open source project. There are 3 cores available, i.e. core 1, 2 and 3. Core 1 is used in this project because it is more stable than the second one. The third one has just been released and is still a test version. There are 3 CDs for Fedora Linux Core 1 installation. The third CD contains Gnopernicus package. A full Linux installation was used.

Festival Installation

Festival has been tested and run on Sun (SunOS and Solaris), FreeBSD, Linux, SGIs, HPs and DEC Alphas but should be portable to any standard UNIX machine. It will also compile on MS Windows platforms with a little work and patience. Any C++ compiler can be utilised. However, those that have been tested are as follows:

- Sun Sparc Solaris 2.8/2.9: gcc-2.95.3, gcc-3.2, gcc-3.3
- Intel Solaris 2.5.1: GCC 2.7.2
- FreeBSD for Intel 3.x, 4.x: GCC 2.95x, GCC 3.0
- Linux 2.x for Intel (including Redhat 6 and 7)
- Windows NT 4.0, Windows95/98: Cygwin 1.1, Visual C++ 6.0.

Building and running Festival requires a UNIX-like environment, a C++ compiler, GNU Make (any recent version) and an audio hardware.

Packages needed to get Festival working are as follows:

- Main system packages:
  - speech_tools-1.2.2-release.tar.gz
  - festival-1.4.2-release.tar.gz
- Main voice packages:
  - festvox_kallpc16k.tar.gz: American English male voice “Kal” 16kHz
  - festvox_kallpc8k.tar.gz: American English male voice “Kal” 8kHz
  - festvox_rablpc16k.tar.gz: British English male voice “Rab: 16kHz
- Lexicon packages:
  - festlex_POSLEX.tar.gz: “Part of Speech” lexical models
  - festlex_CMU.tar.gz: American English lexicon (requirement for American English voices).

Running Festival with the default British voice requires the following packages:

- speech_tools_libs
- festival
- festlex_POSLEX
- festlex_CMU
- any festvox.

Festvox_kallpc16k.tar.gz: American English male voice “Kal” 16kHz, was selected in this project.

Ethereal Installation on Windows

Installing and running Ethereal for Windows requires the following files:

- ethereal-setup-0.10.2.exe (14,048 KB)
- WinPCap_3_0.exe (431 KB)

Ethereal will not be able to capture packets without **WinPcap (wpcap.dll)**. So, in order to capture packets, WinPcap must be installed.

Ethereal Installation on Linux

Building and installing Ethereal for Linux are explained in Chapter 2 of the Ethereal documentation of the Ethereal website (http://ethereal.planetmirror.com/docs/user-guide/ChapterBuildInstall.html)