Antraff traffic analysis software User Manual

Ron Addie and Mostfa Albdair

June 11, 2018

Abstract

This manual describes how to use the `Antraff` traffic trace analysis software, and how to make use of the scripts and procedures associated with it.

1 Introduction

This document provides basic knowledge about how to use the USQ HPC computer to make use of the *Antraff* system for analysing packet trace data (pcap files). This includes an explanation of how to run the software, including its use for generating plots which display the underlying features of a pcap file, and also special information about how to use the software on the USQ High Performance Computer (HPC).

In many instances it will be possible to set one or two environmenetal variables, then issue an `rmake` command, of the form `rmake target`, as described in Subsection 3.5, to produced the results that are needed with Antraff. However, sometimes the desired target has not been formulated as one of the existing targets in the `Antraff` makefile. In this instance it will often be best to create a new target in this makefile which can then be used by the user who needs this target, and by all others.

Adding a target to the makefile is easy, and can be done by anyone in the Antraff group. Usually it will only require a few lines of script. However, these lines of script do require an understanding of the `Antraff` program. This is where it will be necessary to consult Section 3 of this manual, where all the options of the `antraff` program are described.

2 Operation on the HPC

2.1 Setting up an ssh key

For convenient use of the USQ HPC it is highly recommended to set up ssh-key authentication, as follows:

1. `scp .ssh/id_rsa.pub userid@sgihpc:`
2. log in to sgihpc in the usual way (`ssh -X userid@sgihpc`)
3. `cat id_rsa.pub >> .ssh/authorized_keys`
4. `rm id_rsa.pub`

Under windows, these steps should still work so long as they are carried out in a bash shell, as provided by Git-for-windows.

2.2 Modules

Quite a bit of the software on the HPC has to be activated by means of the module command before it can be used. In particular, this applies to the alglib and gnuplot packages. These can be activated by means of the commands

```
module add alglib
module add gnuplot
```

To avoid having to enter these commands every time one logs in, they should be added to the end of the `.bash_profile` file.

To find out the full range of modules which are available, the following command may be used: `module avail 2> modlist`. This will put a list of the available modules into the file `modlist`.

2.3 Libraries

The antraff software makes use of a library for reading of pcap files (files captured from internet devices) and a library for carrying out linear algebra on numerical data. In order to use the latter library, it is necessary to record the location of the dynamic library in an appropriate environmental variable. This can be achieved as follows:

```
     export LD_LIBRARY_PATH=
        /usr/local/opt/software/alglib/alglib-3.10.0-gnu/lib
```

This command should be added at the end of the file .bash_profile so that in future the command will not need to be manually executed again.

2.4 Scripts

Two scripts, overwrite and rmake are used by the Antraff system. Access to both of these, and the antraff program itself, can be gained by modifying the PATH environmental variable. the following command will make the necessary change:

```
     PATH=$PATH:/home/antraff/bin
```

This command should be placed near the end of the user's .bash_profile file.

2.4.1 Local installation of scripts

In rare instances, users may prefer to locally install the scripts overwrite and rmake. This subsection explains how to do that.

The makefile used in the antraff software, which is used to generate certain key results, makes use of a shell script overwrite, which is shown in Figure 1. This script should be placed in a file called overwrite which is stored in /bin, and made executable. The .bash_profile file should have the lines

```
PATH=$PATH:$HOME/bin
export PATH
```

at or near the end. These lines might already be present, in which case they do not need to be added.

Another script (rmake) which users will want to store in their ~/bin directory is described in Subsection 3.5 and the code provided in Figure 2. The rmake and overwrite scripts are also stored in the Antraff area on the HPC, and in the Antraff repository, and it is preferable to copy them from an Antraff directory than to cut and paste them from this document (the Antraff manual).

```
#! /bin/sh
#  overwrite:   copy standard input to output after EOF
#               version from "The Unix Programming Environment"

opath=$PATH
PATH=/bin:/usr/bin

case $# in
0|1)  echo ’Usage: overwrite file cmd [args]’ 1>&2; exit 2
esac

file=$1; shift
new=/tmp/overwr1.$$; old=/tmp/overwr2.$$
trap ’rm −f $new $old; exit 1’ 1 2 15      # clean up files

if PATH=$opath "$@" >$new                  # collect input
then
    cp $file $old     # save original file
    trap ’’ 1 2 15    # we are committed; ignore signals
    cp $new $file
else
    echo "overwrite: $1 failed, $file unchanged" 1>&2
    exit 1
fi
rm −f $new $old
```

Figure 1: A script for overwriting a file by the output from a command

2.5 Copying files to and from the HPC

To copy files to or from the HPC, either the scp command should be used, or rsync. The latter utility automatically uses ssh to do the actual transport, but is more efficient since it will not transfer a file unless it is necessary to do so, and also it can compress files during transfer. Hence it is usually faster than scp.

For example, to transfer the Antraff manual to the HPC one could use the command

```
scp antraffman.pdf userid@sgihpc:antraffman.pdf
```

or

```
rsync antraffman.pdf userid@sgihpc:antraffman.pdf
```

By adding the -a or -r options to the rsync command, it can be induced to transfer directories, and their contents, as well as just files. A similar option exists for scp. Also, in both cases, if several files need to be transferred, these can be simply listed, separated by spaces, on the command line.

If the ssh key has been set up, as described in Subsection 2.1, it will not be necessary to enter the users password when using the scp or rscync commands. This is a further reason for setting up the ssk key.

3 Antraff

The antraff software is stored, on the HPC, at /home/antraff/antraff. It is called antraff, so the command is /home/antraff/antraff/antraff. An alias can be used to reduce this command to simply antraff by adding the line

```
alias antraff=/home/antraff/antraff/antraff
```

to the end of ~/.bash_profile.

In Subsection 3.2, the options of this program are set out. The options are used to specify which data files to read, what analysis to carry out, and where to place the results.

In most cases it will be necessary to conduct further steps on the data produced by antraff to create plots which display the results in a more easily understandable form. These steps are all stored in the makefile. The use of the makefile is explained in Subsection 3.4.

The antraff software is stored in a shared area on the HPC (`/home/antraff`) and in many cases the data on which it operates will also be stored in this shared area (normally in a subdirectory of `/home/antraff/data`). But it is undesirable for users to generate temporary and intermidiate files, or even their final results, in this shared area. Hence we need procedures for running the antraff software from a user directory, and referring to data stored in `/home/antraff`, while storing results in a local directory. These procedures are set out in Subsection 3.5.

3.1 Antraff Algorithms

The overall design of the `antraff` software is described in this subsection. Documentation of the C++ software is provided in [1].

The key processes undertaken in this software are as follows:

1. Read through the pcap file(s), processing each packet, or skipping them if sampling is specified.

2. Store statistics in relation to bytes sent by source IP addresses of packets, or sent to destination IP addresses, or O-D sent between pairs of IP addresses.

3. Accumulate mean, variance, $E(X^2)$, covariance, or $E(XY)$, for traffic between IP addresses in a matrix where the row and column are given by the index of an IP address (one of the most frequently occurring, by byte numbers), or an index of an O-D pair (one of the most frequently occurring, by byte numbers).

4. Carrying out an SVD analysis of one of the matrices which have been accumulated.

5. Extract and report the eigenflows as found by the SVD analysis in the previous item.

3.2 Running Antraff

The `antraff` program has a series of *options*, each of which is set by adding a string similar to `-a` or `-c 400000` to the command line. These options are described in Tables 1 – 3. Note: a brief summary of the options is also available by invoking `antraff` with the `-h` option.

Generally, `antraff` will undertake only one analysis task in each run, although the tasks of extracting eigenflows of the three different types can be conducted simultaneously. Conducting more than one analysis task in a single run of antraff is more efficient, but likely to lead to confusion, except in this instance. Hence, except for this case, only one analysis task can be conducted in each run.

For example, if the option to report frequency of source IP addresses is included, and also the option to report destination IP addresses, and the latter is included last, it will be only the latter option which is carried out.

The options for selecting which analysis task (or tasks) are carried out in a run of `antraff` are listed in Table 1. Note that all these options are indicated by upper-case letters. The options which vary the operation of the algorithms, rather than selecting the task, are indicated by lower-case letters.

Tables 2 and 3 list the options which modify the way in which the analysis tasks are carried out.

The explanation of the options in Tables 2 and 3 should be adequate except in one case. The `-x` and `-b` options both refer to the number of distinct IP addresses which should be analysed. Two distinct options are necessary because when the data file contains a very large number of IP addresses, but a traffic matrix with number of rows and columns corresponding to distinct IP addresses is to be calculated, the byte-frequency of IP addresses will first be estimated using a large number of bins, specified by the `-x` option, and then a smaller number of distinct IP addresses, chosen according to byte-frequency (i.e. only the most important IP addresses will be chosen), will be used, as specified by the `-b` option.

For example, the options chosen might be `-x400000 -b1000`, meaning that 400000 distinct IP addresses will be found, and their frequency estimated, and then the 1000 most important IP addresses (according to byte-frequency) will be used when accumulating a traffic matrix with source and destination IP addresses. If the option chosen was `-x1000 -b1000`, the traffic matrices would still have 1000 distinct IP addresses, but they might not be the 1000 most important IP addresses, by byte frequency. If the total number of IP addresses is less than 400000, the options `-x400000 -b1000` will ensure that the 1000 IP addresses found are the most important. If the total number of IP addresses is a little larger than 400000, these options will still ensure that the selected IP addresses are approximately correct.

Option	Parameter	Explanation
-F		Estimate source IP byte frequencies
-G		Estimate destination IP byte frequencies
-P		Estimate OD byte frequencies (as opposed to IP address byte frequencies)
-U	output-datafile-name	Results of IP vs OD-pair byte frequencies should be stored in this file (this is sometimes referred to as community.dat)
-S	minf,maxf,numf,linerate	Estimate spectrum (for IP addresses, as sources and destinations), and for OD pairs, assuming linerate
-N	netml-file-name	Create a Netml file from the traffic, using `numberIPs` to determine the maximum number of nodes to create, and `numPairs` as the maximum number of traffic flows to create [not yet implemented].

Table 1: Options of the antraff system, Part 1 (options which generate reports on frequencies as measured by numbers of bytes)

3.3 Directories of traffic files

The `-d` option of `Antraff` can be used to specify either a file or a directory. If a directory is specified, `Antraff` will read all the pcap files in that directory. The makefile includes some cases where `Antraff` is run on the directory `$DATADIR`.

It may therefore be necessary to create and manage directories of trace files. Rather than copying trace files, to these directories which have been created so that `Antraff` can operate on collections of traffic files, the specially created directories should be populated by *links*. These are like *aliases*. Under windows, the term alias is used, in fact, rather than link, to refer to this way of placing files, or giving the appearance of placing them, in more than one location.

For example, if a new directory called `/home/addie/mydata` is to be populated by creating links to the files in `/home/antraff/data`, this can be achieved by commands like

```
ln -s /home/antraff/data/equinix.pcap /home/addie/mydata
```

-A	numpairs	Accumulate and analyse covariance; the number of OD pairs to be included in either frequency or SVD analysis is indicated by numpairs. The `-A` option is also used to indicate the number of OD pairs to use in the eigenvalue analysis which forms part of the `-C` option, and also in the OD-pair byte frequency analysis, which is initiated by the `-P` option, but no report from the eigenvalue analysis will be produced if the `-C` option is adopted after the `-A` option, and no eigenvalue analysis is carried out when the `-P` option is adopted.
-T		Accumulate and analyse mean traffic matrix
-V		Accumulate and analyse variance traffic matrix
-C	cov-num-flows	Extract and record eigenflows as derived from the covariance matrix of the traffic; the number of flows to extract and record is given by cov-num-flows. The first eigenflow to be extracted is given by the -S option. The remaining are in order of eigenvalue from this one.
-M	mean-num-flows	Extract and record mean-num-flows eigenflows from the mean traffic matrix
-W	var-num-flows	Extract and record var-num-flows eigenflows from the var traffic matrix
-J	mean-eigenflow-thresh	In the analysis of an individual mean-eigenflow into its component OD flows, all OD flows with weight greater than mean-eigenflow-thresh will be extracted and recorded to the data file
-K	var-eigenflow-thresh	In the analysis of an individual var-eigenflow into its component OD flows, all OD flows with weight greater than mean-eigenflow-thresh will be extracted and recorded to the data file
-E	cov-eigenflow-thresh	In the analysis of an individual covariance-eigenflow into its component OD flows, all OD flows with weight greater than cov-eigenflow-thresh will be extracted and recorded to the data file

Table 2: Options of the antraff system, Part 2 (options which produce reports in relation to SVD analysis of traffic matrices)

Option	**Parameter**	**Explanation**
-a	avwidth	the time-interval over which results should be averaged (so that plots of eigenflows will be smoother and therefore easier to interpret); this option is only used for eigenflows.
-b	numberIPs	The maximum number of distinct IP addresses which will be found and for which frequency statistics will be collected
-d	datafilename	The name of the data file name (or directory name) to be analysed; when datafilename is the name of a directory, all the pcap files in this directory will be analysed
-f	flowfilename	Information about flows will be recorded in flowfilename
-g		Print debugging information (extra information which may help in testing and debugging)
-h		Print brief help with antraff software and its options
-i	interval-length	Use interval-length as the sampling interval length when collecting statistics which require calculations to be repeated for every separate interval of a certain length, e.g. .e.when calculating the variance and covariance matrices
-m		Merge all the files in the specified directory, rather than reading them one after the other
-n		Read and analyse the first n entries of the provided pcap file(s); stop reading at the n-th packet
-o		Print the bytes associated with each IP address, but not the IP address itself, or column-headings
-p		Report accumulated quantities of bytes as bytes, rather than Mbytes

Table 3: Options of the antraff system, Part 3 (options which vary the way in which reports are generated or produced)

Option	**Parameter**	**Explanation**
-q	comparison-interval	The interval of time at which frequency of IP addresses and frequency of OD-pairs is compared, successively, in the “community analysis”
-r	output-datafile-name	Results will be stored in this file name (only applies to certain analysis types)
-s	sampling-interval	sample only 1 in every sampling-interval packets
-u	startflow	Eigenflows starting from this one should be extracted and reported. This enables the first so many eigenflows to be skipped. It applies to mean, variance, and covariance eigenflows.
-x	numbins	Maximum number of bins for storing statistics about IP addresses is set to numbins
-z		subtract the mean byte value when calculating variances and covariances

Table 4: Options of the antraff system, Part 4 (further options which vary the way in which reports are generated or produced)

Target	**Description**
bytes.dat	File containing the number of bytes from each source IP address (only including the most frequent so many, as indicated by the parameter of the -b option)
destbytes.dat	File containing the number of bytes to each destination IP address (only the most frequent)
odbytes.dat	File containing the number of bytes in the largest flows
revpropbytes.dat	bytes.dat in reverse order, as proportions of total bytes
propbytes.dat	bytes.dat, as proportions of total bytes
propdestbytes.dat	destbytes, as proportions of total dest bytes
propodbytes.dat	odbytes, as proportions of total od bytes
community.dat	the number of IP addresses in the first k intervals, and the number of OD pairs in the first k intervals, for k=1, 2, ...
bytes.pdf	Plot of bytes.dat
odbytes.pdf	Plot of odbytes.dat
cumbytes.pdf	Plot of cumulative bytes vs cumulative proportion of IP addresses
cumdestbytes.pdf	Plot of cumulative destination bytes vs cumulative proportion of IP addresses
cumodbytes.pdf	Plot of cumulative OD bytes vis cumulative proportion of OD pairs
communityplot.pdf	Plot of the file community.dat (on a log-log scale)

Table 5: Targets in the antraff makefile, Part 1

3.4 Making results from Antraff

As already indicated, the `antraff` program is able to generate numerical results only. The steps required to convert these results into graphs are undertaken, under the control of a makefile and scripts, by scripts, gnuplot, and R. All of these procedures are recorded in the makefile (`/home/antraff/antraff/makefile`). The targets in the makefile are listed in Tables 5–7. Any of these targets can be *made* by entering the command `make` *`target`* in the directory where the `Antraff` source code is located. These targets can be made in a *different* directory from the `Antraff` source code – which will almost always be preferable – by using the script `rmake` which is described in Subsection 3.5.

Target	**Description**
eigenbytes.dat	File containing the eigenvalues of the covariance matrix generated from the traffic
propeigenbytes.dat	eigenbytes.dat, as proportions of the total squared bytes
cumeigenbytes.pdf	Plot of propeigenbytes, i.e. cumulative proportion of bytes vs cumulative proportion of eigenvalues
eigenplot.pdf	plot of eigenbytes.dat, the eigenvalues from SVD analysis of the covariance matrix
svd.pdf	Plot of the mean, variance, and covariance eigenvalues, from the datafile
svd.meandat	Data from mean traffic analysis from the nominated dataset
svd.covdat	Data from covariance analysis from the nominated dataset
svd.vardat	Data from variance analysis from the nominated dataset
svd.xcovdat	Like svd.covdat, but with the mean^2 subtracted appropriately
svd.xvardat	Like svd.vardat, but with the mean^2 subtracted appropriately
svdx.pdf	Plot of the mean, variance, and covariance eigenvalues, subtracting the mean in the calculation of the variance and covariance matrices
pcapdata.pdf	Plot the mean, variance, and covariance analysis of the pcap files in $DATADIR
eigenflowplot	Plots the first three eigenflows in the sense of the mean matrix, the variance matrix, and the covariance matrix
svd.epdf	Plot of the first three eigenflows in the sense of the mean matrix, the variance matrix, and the covariance matrix
netml	Create netml files corresponding to all the datafiles in $DATADIR, and store them in a new directory called `netml`

Table 6: Targets in the antraff makefile, Part 2

Target	**Description**
svdcum.pdf	Create a plot of cumulative bytes^2 explained vs eigenvalue, for mean, variance and covariance eigenvalues
svdcumproportion.pdf	Create a plot of cumulative *proportion* of bytes^2 explained vs eigenvalue, for mean, variance and covariance eigenvalues
svdvarsamp.{meandat |covdat |vardat |pdf}	Create a plot of eigenvalues based on arbitrary sampling interval (Specified in `$SAMPLING_INTERVAL`)
svdcumvarsamp.pdf	Create a plot of cumulative bytes^2 explained vs eigenvalue, for mean, variance and covariance eigenvalues, with the sampling interval specified in `$SAMPLING_INTERVAL`
svdcumproportionvarsamp.pdf	Create a plot of cumulative *proportion* of bytes^2 explained vs eigenvalue, for mean, variance and covariance eigenvalues, with the sampling interval specified in `$SAMPLING_INTERVAL`

Table 7: Targets in the antraff makefile, Part 3

3.5 Remote operation

Placing the targets of analysis of a traffic datafile in a makefile has some advantages, especially during testing, but once it is desired to undertake the same analyses on many different files it becomes more convenient to be able to carry out these steps not just on specific files, but on a variety of files, and it is also desirable to be able to carry out the process in different locations. In particular, users need to conduct the analysis in a directory of their own choosing, with their own traffic data. For this purpose a script, `rmake` has been developed.

The `rmake` script is shown in Figure 2. This script should be saved to the `bin` directory and made executable (`chmod +x rmake`). Brief help in using `rmake` is given by the command `rmake -h`.

After the `rmake` script has been installed any of the targets of the makefile, listed in Tables 5–6, can be made in any directory, and applied to any datafile, or directory of datafiles, by means of the `rmake` command. To generate the target from data file `dfile`, the rmake command should be run with option `-d dfile`.

For example, to generate the plot `cumodbytes.pdf` using the data file `../data4.pcap`, we enter the command `rmake -d ../data4.pcap cumodbytes.pdf`. To generate `cumodbytes.pdf` from all the `.pcap` files in the directory `datadir`, we enter the command `rmake -d datadir cumodbytes.pdf`.

The rmake script also takes an option, -D datadir, for setting a directory where datafiles may be found by default. Some targets automatically search for their data files in the default location, and this option assists in these cases. The rmake script also has an option -r reportfile for indicating the name of the reportfile. Some targets use the reportfile setting, and others store their output in a file with a hard-wired name no matter what the reportfile is set to.

Rather than using options of the rmake command, a user can set environmental variables (DATAFILE, DATADIR, and REPORTFILE) to set these options. this has the advantage that these settings will be consistently used during a session where these envirnmental variables have been set. For example, to set DATADIR one can enter

```
export DATADIR=/home/addie/antraff/pcapdata.pcapdir
```

If this is added to the .bash_profile file it will automatically be set in all new sessions henceforth.

A good practice when using rmake is to create a script file in the directory where a collection of experiments will be undertaken, in which the environmental variables which specify the details of the experiment are specified. This provides documentation and also makes it easier to run experiments repeatedly. For example DATAFILE or DATADIR and SAMPLING_INTERVAL might be set this way.

The default values for these options are also set in the rmake script itself, and this script can easily be changed. In this way, the default settings can be set up to suit an individual user. However, if a user is simultaneously undertaking a collection of different experiments, it is better practice to set the values of key parameters in a script file, as the values of environmental variables, as suggested in the previous paragraph.

4 Examples

In this section four additional examples are provided. These are shown in Figures 3–8. As a result of the execution of the rmake command in Example 1, the pdf file displayed in Figure 4 is created and displayed.

In Example 2, the data is provided from a *directory* rather than a single file. In effect, all the pcap files in this directory (of which there are two in this instance) are merged. The plot produced in this case is a plot of the eigenvalues of three different singular value decompositions derived from the merged traffic data: from the mean traffic matrix, the variance traffic matrix, and the covariance matrix. The plot produced in this case is shown in Figure 4.

```
#!/bin/bash
if [[ xxx$DATADIR == "xxx" ]]
then export DATADIR=/home/addie/usq/pg/antraff/pcapdata.pcapdir/
fi
if [[ xxx$DATAFILE == "xxx" ]]
then export DATAFILE=/home/addie/usq/pg/antraff/pcapdata.pcapdir/caida.pcap
fi
if [[ xxx$REPORTFILE == "xxx" ]]
then export REPORTFILE=report.dat
fi
export ANTRAFF_DEBUG=" "
datafile=""
reportfile=""
debug=""
norun=""
ANTRAFFMAKEFILE=/home/addie/usq/pg/antraff/makefile
TEMP=‘getopt -o d:r:hgD:n --long datafile:,reportfile:,help,datadir,norun: -n ’rmake’ -- "$@"‘

if [ $? != 0 ] ; then echo "Terminating..." >&2 ; exit 1 ; fi

eval set -- "$TEMP"
while true ; do
case "$1" in
-d|--datafile)
datafile=$2
shift 2;;
-D|--datadir)
datadir=$2
shift 2;;
-r|--reportfile)
reportfile=$2
shift 2;;
-n|--norun)
norun="TRUE"
shift 1;;
-h|--help)
echo "./rmake [-d datafile | --datafile datafile] [-r reportfile] \
[--reportfile reportfile] [-n | --norun ] target"
shift 1
exit 1 ;;
-g)
debug="TRUE"
export ANTRAFF_DEBUG="-g "
shift 1;;
--) shift ; break ;;
*) echo "Internal error!" ; exit 1 ;;
esac
done

if [ ! $datafile == "" ]
then export DATAFILE="$datafile"
fi
if [ ! $datadir == "" ]
then export DATADIR="$datadir"
fi
if [ ! $reportfile == "" ]
then export REPORTFILE="$reportfile"
fi
export WORKDIR=‘pwd‘/
echo ’$WORKDIR = ’ $WORKDIR
echo ’$DATAFILE = ’ $DATAFILE
echo ’$DATADIR = ’ $DATADIR
echo ’$REPORTFILE = ’ $REPORTFILE

if [ xxx$norun == ’xxxTRUE’ ]
then
    make -f $ANTRAFFMAKEFILE -e -n $1
else
    make -f $ANTRAFFMAKEFILE -e $1
fi
```

Figure 2: A script for running make in a different directory

```
> rmake plot
$WORKDIR =  /home/addie/usq/pg/antraff/test/
$DATAFILE =  /home/addie/usq/pg/antraff/pcapdata.pcapdir/caida.pcap
$DATADIR =  /home/addie/usq/pg/antraff/pcapdata.pcapdir/
$REPORTFILE =  report.dat
/home/addie/usq/pg/antraff/antraff -d /home/addie/usq/pg/antraff/pcapdata.pcapdir/caida.pcap -n0 -x400000 -b400000 -F -o > bytes.dat
overwrite bytes.dat sort -rn bytes.dat
/home/addie/usq/pg/antraff/genfreqplot /home/addie/usq/pg/antraff/test/ `wc bytes.dat | awk "{print $1}"`
gnuplot freqplot.gnu
ps2pdf bytes.ps bytes.pdf
evince bytes.pdf &
```

Figure 3: Example 1. The bytes associated with each IP address, as a source

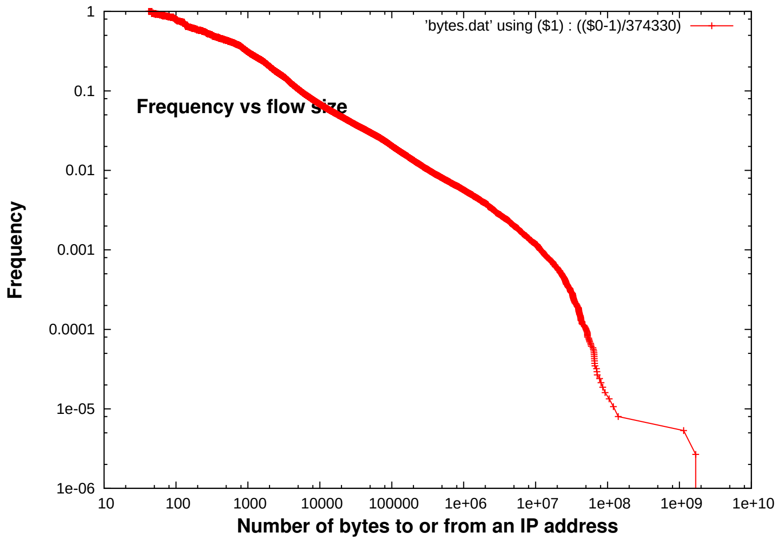

Figure 4: Example 1. The plot produced.

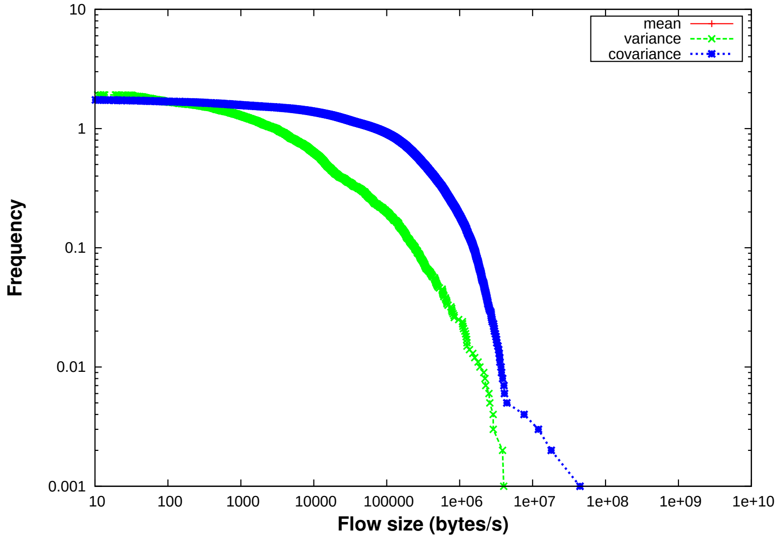

Figure 5: Example 2. The plot produced.

```
> rmake pcapdata.pdf
$WORKDIR =  /home/addie/usq/pg/antraff/test/
$DATAFILE =  /home/addie/usq/pg/antraff/pcapdata.pcapdir/caida.pcap
$DATADIR =  /home/addie/usq/pg/antraff/pcapdata.pcapdir/
$REPORTFILE =  report.dat
/home/addie/usq/pg/antraff/antraff -d /home/addie/usq/pg/antraff/pcapdata.pcapdir/ -T
Antraff: analyse IP traffic.
Mean will be analysed.
Data is broken into intervals of length 0.05 in the var and cov analyses.
A report on this run of antraff will be stored in /home/addie/usq/pg/antraff/pcapdata
Opened pcap file /home/addie/usq/pg/antraff/pcapdata.pcapdir//caida1.pcap for process
............... [the output continues for many lines] .....................
```

Figure 6: Example 2. Plotting the mean, variance, and covariance eigenvalues from the merged traffic files

```
> rmake -n -d ~/usq/pg/antraff/pcapdata.pcapdir/caida.pcap cumodbytes.pdf
$WORKDIR =  /home/addie/usq/pg/antraff/test/
$DATAFILE =  /home/addie/pg/usq/antraff/pcapdata.pcapdir/caida.pcap
$DATADIR =  /home/addie/usq/pg/antraff/pcapdata.pcapdir/
$REPORTFILE =  report.dat
/home/addie/usq/pg/antraff/gencumodfreqplot /home/addie/usq/pg/antraff/test/ `wc odby
gnuplot cumodfreqplot.gnu
ps2pdf cumodbytes.ps cumodbytes.pdf
```

Figure 7: Example 3. Finding out what steps will be needed to make a target from a specific data file

In Example 3, the -n option has been added to the rmake command so that instead of executing the required steps, we are merely shown what steps would be carried out in order to make the target.

In Example 4, the same command tried in Example 3 is re-tried without the -n option. This time the steps are carried out, and the plot shown in Figure **??** is produced.

References

[1] R. G. Addie. Antraff software documentation. Technical report, 2016.

```
> rmake -d ~/usq/pg/antraff/pcapdata.pcapdir/caida.pcap cumodbytes.pdf
$WORKDIR =  /home/addie/usq/pg/antraff/test/
$DATAFILE =  /home/addie/usq/pg/antraff/pcapdata.pcapdir/caida.pcap
$DATADIR =  /home/addie/usq/pg/antraff/pcapdata.pcapdir/
$REPORTFILE =  report.dat
/home/addie/usq/pg/antraff/antraff -d /home/addie/usq/pg/antraff/pcapdata.pcapdir/cai
overwrite odbytes.dat sort -rn odbytes.dat
totalbytes=`awk -f /home/addie/usq/pg/antraff/totalbytes.awk odbytes.dat` ;\
awk -v totalbytes=$totalbytes -f /home/addie/usq/pg/antraff/accum.awk odbytes.dat > p
/home/addie/usq/pg/antraff/gencumodfreqplot /home/addie/usq/pg/antraff/test/ `wc odby
gnuplot cumodfreqplot.gnu
ps2pdf cumodbytes.ps cumodbytes.pdf
```

Figure 8: Example 4. Plotting the cumulative proportion of OD pairs vs the cumulative proportion of bytes generated by these OD pairs, as evidenced in a specific data file

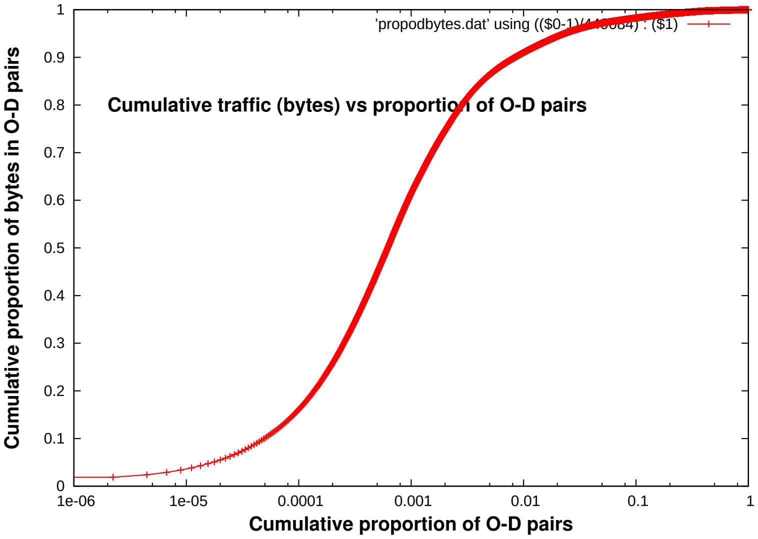

Figure 9: Example 4. The plot produced.