NAME

`gengraph.pl` - Transforms Traffic Generator(TG)'s binary output file data into formats suitable for viewing with xplot, xgraph, or gnuplot.

SYNOPSIS

```
gengraph.pl [-h | --help] [-c client-file] [-s server-file]
      [-o output-file] [-f xplot | xgraph | gnuplot]
        [-SQ] [-DR] [-D] [-AD] [-S] [-R] [-IS] [-IR] [-J] [-AJ] [-L]
```

DESCRIPTION

`gengraph.pl` is a perl script that converts TG binary output file data into a format suitable for viewing using graphical tools.

OPERATION

`gengraph.pl` invokes `dcat` (part of the TG distribution) to first convert log file data into ascii form. The script parses `dcat`'s output and populates internal data structures. The script extracts the datasets selected by the user and outputs this data in the required output format.

Input. Both the client-side and the server-side log files must be provided to gengraph. When the input file names are not specified, `gengraph.pl` assumes the TG log files on the client and the server sides to be *client.log* and *server.log* respectively.

Output. The output file name can be specified using the *-o* option to the script. When the user does not specify any output file name, `gengraph.pl` uses the name *stats.xpl*. In case of gnuplot two output files are created by the script: one containing the gnuplot *plot* command in the file with *.dem* name extension, and another containing the datasets in the file with *.dat* name extension. The graphical output format can be specified on the command line using the `-f` option. The default output format is *xplot*. `gengraph.pl` supports xgraph, xplot, and gnuplot tools.

Viewing the plots. The commands to view the plots are as follows:
   xgraph: xgraph -M -nl -tk -p -bb *<output-file>*
   xplot: xplot *<output-file>*
   gnuplot: gnuplot *<output-file>*.dem

The various graphical tools are available at the following URLs:
   Xgraph: *http://jean-luc.ncsa.uiuc.edu/Codes/xgraph/*
   Xplot: *http://www.esrf.fr/computing/scientific/xop/xplot/*
   Gnuplot: *ftp://ftp.ucc.ie/pub/gnuplot*

Jitter calculation. For this, we first compute the average delay over the entire duration of the experiment. We use this average delay computation to compute the deviation of the delay corresponding to individual data packets, i.e., jitter. The experimenter must modify the implementation

to reflect his/her notion of jitter, which could be the difference in the delay measured for successive packets.

Plots. All plots are generated with time as the x-axis. The specific data set to be extracted from the TG log files can be specified using the command line options identified below in the OPTIONS section. The arguments identify the specific parameter of interest such as the inter-arrival time or the jitter. Multiple data sets can be specified, but only certain combinations yield meaningful graphs: those whose units match and whose values are comparable.

Title. The default title of any plot is the start time contained in the client-side log file's header. However, if a file named *readme* exists in the current working directory, the first line in this file is concatenated with the above timestamp to form the title.

Computation of Averages. The script computes the average delay, jitter, and send and receive rates, over user-controllable time windows. The following variables in the script correspond to each of the above mentioned windows: *$average\_delay\_computation\_window,$ $average\_jitter\_computation\_window,$* and *$rate\_computation\_window$* respectively. The default values for these variables are 1 sec, 0.5 sec, and 0.5 sec respectively.

Log Errors. TG records any errors in packet transmission or receipt in the log file. The script informs the user of such errors, displaying both the name of the file in which the error was observed and the line containing the error.

Script Errors. The script assumes that the perl interpreter is located in the */usr/bin* directory. If perl interpreter is unavailable in that directory, a platform and shell specific message such as *"bash: gengraph.pl: No such file or directory" or "ntcsh: gengraph.pl: bad interpreter: No such file or directory"* may be obtained. If the perl interpreter is available in a different path, for example /local/bin, then the first line of the script must be modified to reflect the correct path. The script also assumes that `dcat` is available in the path. If dcat is not available, a message similar to *"Can't open client log: No such file or directory"* is displayed.

NOTES

The script is still under development. The user should take note of the following:

1. The script ignores excess log entries in the server-side log. This is possible when the multiple clients send data to the server in either interleaved or sequential fashion. `gengraph.pl` does not handle the case of single-server and multiple clients. The script displays an appropriate warning in this case.
2. The output generated by the script cannot be piped to another command.
3. There is a mismatch between the views from various graphical tools. We tried to keep the colors/graph formats consistent across graphing tools. However, we could not control some aspects, such as the black background in xplot.
4. To improve clarity, packet drops are displayed on the negative y-axis i.e., a sequence number x dropped at time t will be indicated by a mark at coordinates *(t, -x)* instead of *(t, x)*. This behavior can be modified by changing the value of the variable *$reflect\_drops$* in the script from -1 to 1.

5. TCP might combine packets and/or perform retransmissions to achieve reliable delivery. This can mask/distort certain network parameters such as the end-to-end delay and loss ratio.

OPTIONS

Table 1: Command line Options

Option	Semantics	Combinations[1]
-SQ	Plot of Packet Sequence Numbers[2]	**-SQ & -DR**
-DR	Plot of the Sequence Numbers of the Dropped Packets[3]	**-SQ & -DR**
-R	Plot of the Receive Rate[4]	**-R & -S**
-S	Plot of the Send Rate	**-R & -S**
-IS	Plot of the Inter-Departure times at the Sender	**-IS & -IR**
-IR	Plot of the Inter-Arrival times at the Receiver	**-IS & -IR**
-D	Plot of the End-to-end Delay of Packets (UDP)[5]	
-AD	Plot of the Average End-to-end Delay (UDP)	
-J	Plot of Jitter at the Receiver	
-AJ	Plot of the Average Jitter[6]	
-L	Plot of Lengths of the packets at the Sender side (UDP)	
-h, --help	Prints this list of options	
-c *client-file*	Client-side log file name	
-s *server-file*	Server-side log file name	
-o *output-file*	Name of the output file	
-f *format*	Target graphical tool	

1. Only certain combinations are valid due to different scales as well as units.
2. The sequence numbers refer to the cumulative byte counts in case of TCP traffic, and packet IDs in case of UDP traffic.
3. Useful only when UDP protocol is selected.
4. The rates are calculated over a time window, whose default size is 1.0 sec. It can be modified by adjusting the *$rate_computation_window* variable in the **gengraph.pl** script.
5. The rates are calculated over a time window, whose default size is 0.5 sec. It can be modified by adjusting the *$average_delay_computation_window* variable in the **gengraph.pl** script.
6. The rates are calculated over a time window, whose default size is 0.5 sec. It can be modified by adjusting the *$average_jitter_computation_window* variable in the **gengraph.pl** script.