TRPR User's Guide Version 2.1b5

Abstract

The TRace Plot Real-time (TRPR) is open source software by the Naval Research Laboratory (NRL) PROTocol Engineering Advanced Networking (PROTEAN) group that analyzes output from the *tcpdump* packet sniffing program and creates output suitable for plotting. It also specifically supports a range of functionality for specific use of the *gnuplot* graphing program. *trpr* can operate in a "real-time" plotting mode where *tcpdump* `stdout` can be piped into *trpr* and *trpr's* `stdout` in turn can be piped directly into *gnuplot* for a sort of real-time network oscilloscope. *Trpr* can also parse *tcpdump* text trace files and produce files which can be plotted by *gnuplot* or imported into other plotting or spreadsheet programs. IPv4 and IPv6 traces from *tcpdump* are supported. *Trpr* can also perform the same functions with *mgen* log files (See http://cs.itd.nrl.navy.mil/work/mgen/ for more information on *mgen* and the MGEN test tool set) and *ns-2* (Berkeley's network simulator - see http://www.isi.edu/nsnam/ns) trace files.

By default, *trpr* creates a "data rate" versus time plot of the flows specified using the `auto` and `flow` (and `exclude`) filtering commands. The `auto` command is used to set filters to automatically detect and enumerate individual flows matching the `auto` filter parameters (protocol type, source addr/port, and destination addr/port) and the `flow` command aggregates flows matching its filter specification under a single data plot set. The `exclude` command is used to specify packet flows *trpr* should ignore. The `flow` , `auto` and `exclude` commands can each be used multiple times on the command line to specify different combinations of filters to produce different desired output. (In the future, an exclusion filter set will also be provided).

If the `interarrival` command is used, *trpr* creates a plot of the differential interarrival delay of packets for the specified flows. And for MGEN packets, the `latency` command can be used to create a plot of the transmission latency (*mgen*-logged rxTime - txTime) versus time for the flows. Also, for MGEN packets, the `loss` command can be used to generate profiles of packet loss over time. MGEN packet payloads contain sequence numbers and time stamps to facilitate these analyses. The `count` command simply produces counts of the indicate "send" and/or "recv" events for the specified flows. The `histogram` command causes *trpr* to output histograms of any of these statistics and the `window` command determines the averaging window interval to use (with "`window -1`" over the entire trace file and "`window 0`" for individual events). *Trpr* can also "play back" a *gnuplot* visualization of trace file content at real time rates with the `replay` command.

Table of Contents

1. Downloads

The *trpr* package is available at http://downloads.pf.itd.nrl.navy.mil/proteantools

Tcpdump can be found at http://ee.lbl.gov/

Gnuplot's official web site is http://www.gnuplot.info/

The *MGEN* web site is http://cs.itd.nrl.navy.mil/work/mgen/

The *ns* web site is http://www.isi.edu/nsnam/ns

2. Build Instructions:

Simply compile *trpr* with a C++ compiler. It has been primarily built with gcc on Unix platforms. For example, type:

```
g++ -o trpr trpr.cpp -lm
```

to build the executable binary.

On windows, use the provided Visual Studio Trpr.sln file to build the application. A windows binary file release is also available on the protean forge web site.

3. Quick Start

Here are a couple of examples illustrating use of *trpr* in simple ways. Note that *trpr* has a number of flexible command-line operations to get the results you want and understanding these is strongly recommended. And *tcpdump* has very flexible filtering options for paring down the data captured from the network so that your graphs can focus on the data of interest. The options of *tcpdump* and *trpr* can be coupled together in many different ways. And *trpr* supports options to command *gnuplot* to create Gif or Postscript files for hard output or use in other programs. Detailed usage instructions for *trpr* and hints for *tcpdump* and *gnuplot* usage are given later.

3.1. Non-real-time Operation

1. Capture IP packets with *tcpdump* with hexadecimal packet header output. Note you **must** use *tcpdump's* hexadecimal output option (-x):

   ```
   tcpdump -x > <traceFile>
   ```

   The *trpr* code parses the *tcpdump* lines and hexadecimal output for additional details (more consistent than the textual content through version updates of *tcpdump*). Alternatively, if you have a binary "pcap" file (e.g. created with "tcpdump -w <pcapFile>"), you can use tcpdump to convert this binary file to the text and hexadecimal output form with:

   ```
   tcpdump -lnx -r <pcapFile> > <traceFile>
   ```

2. Use *trpr* to process the captured <traceFile> to create a <plotFile> suitable for plotting with *gnuplot*, automatically creating lines on the graph for each unique "flow" of data discovered in the <traceFile>:

   ```
   trpr input <traceFile> auto X output <plotFile>
   ```

   Note you can optionally consolidate Step 1 and Step 2 here into a single step by pipelining the *tcpdump* STDOUT into *trpr* via:

   ```
   tcpdump -lnx -r <pcapFile> | trpr output <plotFile>
   ```

3. Use *gnuplot* to display a graph of *trpr's* analysis results (By default trpr puts appropriate headers in the <plotFile> for *gnuplot*:

   ```
   gnuplot -persist <plotFile>
   ```

   As examples, mgen log files can be processed with:

   ```
   trpr mgen input <mgenLogFile> auto X output <plotFile>
   ```

and ns-2 simulation trace files can be processed with:

```
trpr ns input <nsTraceFile> link <srcNode>,<dstNode> send auto X output <plotFile>
```

Note: The link command coupled with the send command specifies to process packets sent over the link from node <src> to node <dst> in the ns-2 simulation. The <src> and/or <dst> arguments can be wildcarded with the 'X' character to process multiple links to/from a particular or any simulation node.

Note: For ns-2 mobile trace files, the link command should be used in the form:

```
link <nodeId>,{AGT | RT | MAC}
```

to capture the corresponding set of packets (Agent, Router, or MAC) for a mobile ns-2 node).

We hope to provide more examples for using trpr with ns-2 soon.

3.2. Real-time Operation

1. Set up *tcpdump* to capture packets and direct hexadecimal output to *trpr*, in turn piping *trpr's* real-time output directly to *gnuplot* to get continuously updated plots of network traffic flow activity:

   ```
   tcpdump -lnx | trpr real auto X | gnuplot -noraise -persist
   ```

   Or for mgen operation:

   ```
   mgen flush output /dev/stdout | trpr mgen real auto X | gnuplot -noraise -persist
   ```

   Note that the "tail -f" option can also be used to pipe a *mgen* log file to *trpr* in parallel with logging. (The *mgen* "flush" option causes *mgen* to "flush" its output line by line for better real time performance. Note this may penalize system performance)

4. Usage

Usage:

```
trpr [version][mgen][ns][raw][key]

[real][latency][interarrival][loss][count]

[window <sec>] [history <sec>]

[flow <type,srcAddr/port,dstAddr/port>,flowId]

[auto <type,srcAddr/port,dstAddr/port>,flowId]

[exclude <type,srcAddr/port,dstAddr/port>,flowId]

[input <inputFile>] [output <outputFile>]

[link <src>[,<dst>]][send|recv][nodup]

[range [<startSec>][:<stopSec>]] [yrange [<min>][:<max>]]

[offset <hh:mm:ss>][absolute]

[summary][histogram][replay <factor>]

[png <pngFile>][post <postFile>][gif <gifFile>][multiplot]

[surname <titlePrefix>][ramp][scale]

[nolegend]
```

NOTE: *The type, addr, port , and flowId parameters can be "wildcarded" with an 'X' character. For the "auto" enumerated flow, these parameters can also be "trumpcarded" with a 'Y' character. In this case, the "trumped" fields are treated as a "don't care" case for flow enumeration. Thus, "auto Y" is functionally equivalent to "flow X".*

4.1. Command-line Options and Parameters

`version`	Causes *trpr* to display program version number and exit.
`mgen`	*trpr* will expect to process a *mgen* log file instead of *tcpdump* hex output.
`ns`	*trpr* will expect to process a *ns* trace file instead of *tcpdump* hex output
`raw`	When this option is given, the <outputFile> will only include unlabeled sets of plotting data without the default *gnuplot* compatible headers. This is useful to get the "raw" plot data for importing into a spreadsheet or other plotting program
`key`	With this option, trpr will print a "key" to the data plot sets in the <outputFile>. This consists of one comma-delimited line with a leading "#" character. This line is printed when new flows of data are detected and another data set column is output. The first column is marked "Time". Subsequent columns are labeled with a description of the flow data being plotted.
`rate`	Causes *trpr* to create plots of data rate versus time. The window command can be used to set *trpr* 's rate averaging window. The rate command is the implicit default plot mode for *trpr*.
`interarrival`	Causes *trpr* to create plots of differential interarrival packet delays for detected flows instead of the default data rate versus time plot.
`latency`	Causes *trpr* to create plots of transmission delay for *mgen* flows instead of the default data rate versus time plot. This type of plot is only available for *mgen* operation.
`loss`	Causes *trpr* to create plots of packet loss based on received sequence numbers for *mgen* flows instead of the default data rate versus time plot. This type of plot is only available for *mgen* operation. The `window` command can be used to set *trpr* 's loss averaging window. The "window" specified should be large enough to encompass several expected packet events for desired results.
`count`	Causes *trpr* to create plots of packet counts versus time instead of the default data rate versus time plot. The window command can be used to set *trpr's* count accumulation window. The rate command is the implicit default plot mode for *trpr*.
`real`	When this option is given, *trpr* will output plotting commands and data to its `stdout`. This output is intended for the `stdin` of *gnuplot* for real-time plotting. However, note that this output can be redirected to a file for storage, and then later that file can be directed to the input of *gnuplot* for "playback". Note that the "real-time" mode can be used simultaneously with *trpr* 's cumulative "non-real-time" output option. Note that the "real

	time" graph update occurs once per window time. This option can also be used with pre-existing trace files. Use the replay command to limit the actual graph animation rate or the trace file will be parsed at "cartoon rate" (i.e. as fast as possible).
`gif <gifFile>`	This option commands *gnuplot* to create a "gif" (Graphics Interchange Format) file when it plots instead of the default X11 display. The <gifFile> parameter is the name of the file *gnuplot* will create when it processes *trpr's* output. This can be used in either real-time or non-real-time operation. In real-time operation, the <gifFile> will be periodically overwritten according to window setting.
`post <postFile>`	This option commands *gnuplot* to create a Postscript file when it plots instead of the default X11 display. The <postFile> parameter is the name of the file *gnuplot* will create when it processes *trpr* 's output. This can be used in either real-time or non-real-time operation. In real-time operation, the <postFile> will be periodically overwritten according to window setting.
`png <pngFile>`	This option commands *gnuplot* to create a .pngt file when it plots instead of the default X11 display. The <pngFile> parameter is the name of the file *gnuplot* will create when it processes *trpr* 's output. This can be used in either real-time or non-real-time operation. In real-time operation, the <pngFile> will be periodically overwritten according to window setting.
`surname <surName>`	Prepends "surname" to the plot's title.
`multiplot`	With *gnuplot*, *trpr* will create a "multiplot" graph with one graph per detected flow (stacked vertically). (This only works with the real-time updated (real command) graphing mode for now).
`ramp`	By default, *trpr* creates "stair step" plots of its averaging window results (i.e. 2 data points per window). The optional ramp command causes *trpr* to create plots with one data point per averaging window (at the window's end), thus "ramping" from one window to the next. This may be useful for alternative post-processing of *trpr's* output files or to reduce the number of data points on plots with an extremely large number of data points where the window start/stop points are indiscernible anyway.
`window <sec>`	This parameter sets the step size of *trpr's* window-based data rate and packet loss averaging algorithms. The step size unit is time in seconds. This algorithm counts the cumulative quantity of data (or packet loss) in each window of time and calculates the kilobits-per-second (kbps) (or loss fraction) value for each step. These discrete values of data rate (or loss fraction) versus time comprise trpr 's plot data. Two points are plotted, one at each time window's beginning and one at its end, to form a "stair step" plot. The window command also controls the *gnuplot* real-time graph update rate for real command operation. The window <sec> value can be specified as "-1" to cause *trpr* to average across the en-

	tire trace file (or the period specified by the range command). Note the negative window value should not be used in combination with the real command. Default = 1 second.
`history <sec>`	This parameter determines the range (in time units of seconds) of the X-axis of the graphs produced in *trpr's* real-time mode. As time progresses, the *gnuplot* graphs will scroll in "strip-chart" fashion to display the current history of network activity. Default = 20 seconds.
`auto <type,srcAddr:port,dstAddr:port,id>`	This command instructs *trpr* to automatically discover, enumerate, and plot "flows" of network data according to the matching (type,src,dst,id) criteria provided. Otherwise, *trpr* only plots "flows" given by the flow option described below. Valid values for <type> include "X", "Y", "UDP", "TCP", "ospf", "arp" or the numeric value of the IP protocol type of interest. The "X" value "wildcards" the <type> so that *trpr* will automatically create a plot on the graph for any type of IP protocol which meets the given <source,destination> criteria, if given. The "Y" value for a field (including the "type"), sets a "don't care" state with respect to "auto" enumeration. For example, if the "type" is set to "Y" and the address, etc is wildcarded with an "X" (i.e. "auto Y,X), then unique source/destination flows are enumerated but all the traffic, regardless of protocol "type" is consolidated for each source,destination tuple. The source and destination addresses (srcAddr & dstAddr) must be given in dotted decimal notation or may also be wildcarded with an "X" character. The <source,destination> portion may also be omitted and then will be automatically wildcarded. The optional "id" portion of the flow description corresponds to any "flow id" which may apply to the data analyzed. This currently only applies to *mgen* log files when the user wishes to additionally differentiate *mgen* flows by their "flow id". (See the *mgen* user's guide for more information). As an example, " auto udp" will cause *trpr* to enumerate individual plots for each unique UDP protocol flow detected regardless of source or destination. The source and destination port numbers can be explicitly specified or wildcarded with an "X" or implicitly through omission. Note that flows which match those given with the flow option (see below) will not be tested against the auto criteria. The auto option may be used multiple times on the *trpr* command line to establish multiple sets of automatic flow matching criteria (e.g. *trpr* auto udp auto tcp ... "). "Wildcard" and "trumpcard" flow specification may be abbreviated. For example "auto X" means all filter parameters are wildcarded while "auto X,Y" means the "type" is wildcarded for enumeration while other filter parameters are "trumped", meaning the "auto" enumeration will instantiate one flow per protocol "type". Note that if no flow or auto filters are provided, **trpr** runs with a default wildcard enumeration filter of "auto X"
`flow <type,srcAddr:port,dstAddr:port,id>`	This command instructs trpr to look for and plot specific "flows" which match the given (type,src,dst) criteria. All flows which match the given criteria are accumulat-

	ed together onto a single plot line. The address and port criteria are given in the same way as for the auto command and may be wildcarded in the same way. Note the "trumpcard" has no distinct effect for the "flow" command at this time and is equivalent to the "wildcard". For example, the option "flow udp" will cause trpr to accumulate all detected UDP traffic (regardless of source and destination since they are implicitly wildcarded here) into a single plot. Thus the command "*trpr* flow UDP flow TCP ..." will produce a graph with two lines, one plotting cumulative UDP traffic and the other plotting cumulative TCP traffic detected by *tcpdump*. As with the auto option, the flow option may be used multiple times on the command line and may be used in conjunction with the auto option. Flows of network traffic matching the criteria specified with the flow option will be accumulated into a matching flow plot and are also tested against the sets of auto option criteria so redundant plot lines may result depending on the criteria used.
`exclude <type,srcAddr:port,dstAddr:port,id>`	This command instructs *trpr* to ignore specific "flows" which match the given (type,src,dst) criteria. The address and port criteria are given in the same way as for the auto command and may be wildcarded in the same way. For example, the option "flow udp" will cause *trpr* to ignore all detected UDP traffic (regardless of source and destination since they are implicitly wildcarded here). The exclude command filters are evaluated before the auto and flow command filters.
`input <inputFile>`	This option instructs *trpr* to use the file name given by <inputFile> for input. Otherwise *trpr* looks for input from stdin . The expected input format is text output from the *tcpdump* program run with its hexadecimal option (-x) given and properly filtered so that only IP protocol data is captured. Non-IP data from *tcpdump* will result in errors in *trpr's* output.
`output <outputFile>`	This option instructs *trpr* to save cumulative data into the file name given by <outputFile> for later (non-real-time) plotting. The plot data stored here contains data from the entire *tcpdump* run (as opposed to the trpr real-time mode's limited history of data). By default (i.e. unless the raw option is given), the output file contains text header information at its beginning so that *gnuplot* can be used to create a nicely-labeled graph.
`link <src>[,<dst>]`	This causes *trpr* to process only packets associated with the identified "link" or "node". For ns trace files, the <src> and <dst> values correspond to simulation node identifiers. For *tcpdump* operation, the MAC address is used. Note that <src> and/or <dst> values can be wildcarded by omission or by designating 'X' as the value. For ns simulations using the wireless/mobility extensions, the <dst> value may be "AGT" or "RTR" corresponding to the wireless transmission type (By default, both "AGT" and"RTR" are counted by trpr) since the notion of "links" is not used in the trace files. Wildcarding the <src> or <dst> values allows the user to analyze all traffic arriving to and/or leaving from a specific sim-

	ulation node or MAC address. The send and recv commands may be optionally used in combination with the link command to specify whether only arriving packets (recv) or departing packets (send) are processed. By default, only departing packets are processed.
`nodup`	Causes *trpr* to discard duplicate packets.
`send`	Specifies that only "sent" packets are to be processed. In **ns**, this corresponds to 's' events for traced links or nodes. In *tcpdump*, this corresponds to packets whose source MAC address correspond to the <src> value given with the link command. For *mgen* logfiles, this corresponds to packets sent by *mgen*. By default, only "received" packets are counted by trpr . The send and recv commands are generally useful only for *ns* simulations but may be applicable to *tcpdump* trace file analysis in some situations.
`recv`	Specifies that only "received" packets are to be processed. In *ns*, this corresponds to 'r' events for traced links or nodes. In *tcpdump*, this corresponds to packets whose destination MAC address corresponds to the <dst> value given with the link command. By default, only "received" packets are counted by *trpr* . The send and recv commands are generally useful only for *ns* simulations but may be applicable to *tcpdump* trace file analysis in some situations.
`range <min>[:max]`	Causes *trpr* to skip ahead to the "start time" (in seconds) from the first packet event in the trace file and end processing at the optional "stop time" (in seconds). Setting the "stop time" to -1 causes *trpr* to process until the end of the trace input. Note the range command may be used in combination with the offset and/or absolute commands to perform analysis for a specific time period in the trace file. NOTE: the deprecated "xrange" command is still supported.
`yrange <min>[:max]`	Will override TRPR's auto-yrange behavior
`offset <hh:mm:ss>`	This allows the user to specify an absolute analysis start time using a time-of-day reference. The time given is in 24-hour clock time format and must be within 12 hours of the time of the first packet event in the trace file.
`absolute`	Causes *trpr* to use the absolute time given in the trace file in its output instead of "normalizing" the time values (generally the plots' x-axis) to zero time for the first packet event or optional offset time.
`summary`	This causes *trpr* to output summary statistics of results to `stdout` at the end of analysis. These summary results are available with or without the production of data intended for plotting. This options is useful for commanding or scripting *trpr* to collect statistics in addition to or instead of plots.
`histogram`	This causes *trpr* to output a histogram of the values of analyses intervals (intervals determined by the window command) for each flow to `stdout`. Some percentile information of the histogram content is also provided in the output. The histograms are comma-delimited tables

	of values. The hcat program provided in the TRPR distribution can be used to query and manipulate these histogram files or they can also be plotted with a graphing tool (e.g. *gnuplot*). The hcat program also allows multiple histogram files from multiple *trpr* analysis runs to be combined together for cumulative statistics collection. Currently the quantization size and curve of the histogram is fixed and adapts in range with data. The histogram output may be useful for packet latency analyses or other kinds of statistics compilations.
`replay <factor>`	This limits *trpr's* rate of real-time *gnuplot* graph generation to a <factor> of real time when parsing a pre-existing trace file. When the replay command is given, *trpr* generates the same *gnuplot* output as for the real command. The <factor> parameter scales the playback rate with respect to real time. For example, <factor> = 1 is actual real time, while <factor> = 2 is double speed playback. Note that real time update occurs once per window time.
scale	Autoscales the plots y axis.
nolegend	No key/legend will be created in the gnuplot output. This is particularly useful for smaller displays as well as on certain live displays.

5. *tcpdump* Hints

1. By default, *trpr* expects to process the payload of the captured Ethernet frames from *tcpdump* and can identify IPv4 and IPv6 payload protocols such as UDP, TCP, etc with port number information when applicable. The ARP protocol is also identified by *trpr* and protocol names identified by *tcpdump*. The *tcpdump* "-e" option can be invoked at which point *trpr* uses the Ethernet MAC source and destination addresses for trpr flow identification. This is useful for getting cumulative packet rates and/or counts based on MAC addresses. In this case, the protocol types embedded in the Ethernet frame payloads are ignored. With additional scripting, using *trpr* as a helper command, one could first learn the source and/or destination MAC addresses for flows within a *tcpdump* trace file and then use *tcpdump* filtering in conjunction with *trpr* analysis to identify flows for specific MAC source and/or destination addresses.

2. Always use the "-x" option when using *tcpdump* with *trpr*. (*trpr* looks for and parses the hexadecimal output)

3. Use *tcpdump's* "-n" option to skip DNS lookups and speed up *tcpdump's* performance (*trpr* only uses dotted decimal numeric IP addresses).

4. Use *tcpdump's* line buffering option ("-l") to get output with minimal delay for real time plotting.

5. Read and learn *tcpdump's* man page for the extensive set of filtering options *tcpdump* provides. Uses these filter options in conjunction with *trpr's* own filters to get the graphical results you wan

6. Leverage *tcpdump's* ability to store captured data in a binary file (use *tcpdump's* "-w" option) and then post-process it with *tcpdump* 's filter's (using *tcpdump* to process the stored binary file with its "-r" option and redirecting its output to *trpr*).

6. *gnuplot* Hints

1. Use *gnuplot's* "-noraise" option when using with *trpr* in "real-time" mode if you don't want the updated plots to continually pop to your display's top level.

2. Use *gnuplot's* "-persist" option if you wish the last plot to remain displayed after exiting.

3. *trpr's* output files for *gnuplot* are in text format and easily edited to customize output. *Gnuplot* is a very flexible program with lots of options to get the graphs into almost any format you would like. It is also lightning fast.

7. Examples of Use

To pipe *mgen* output directly into a real-time *gnuplot* display and create new plots for each src/dst pair:

```
mgen flush event "LISTEN TCP 5000" | trpr mgen window 5 history 300 real auto X multiplot rate | gnuplot -noraise -persist
```

To pipe *tcpdump* output directly into a real-time *gnuplot* display and create new plots for each detected network flow:

```
tcpdump -lnx -i eth0 | trpr window 5 history 300 real auto X multiplot rate | gnuplot -noraise -persist
```

To plot cumulative transmission rates from distinct Ethernet sources:

```
tcpdump -elnx -i eth0 | trpr window 5 history 300 real auto X,X,Y multiplot rate | gnuplot -noraise -persist
```

To plot cumulative transmission rates to distinct Ethernet destinations:

```
tcpdump -elnx -i eth0 | trpr window 5 history 300 real auto X,Y,X multiplot rate | gnuplot -noraise -persist
```