September 2002

The Software Tools for Networking column contains brief presentations of software tools that are freely available on the Internet and could be useful for the readers of this magazine. Each presentation is based on an extended abstract submitted by the authors of the tools that was copy edited and checked for accuracy against the version of the tool available on the Internet. Authors willing to have their tools presented in this manner should send a 300-word description of their tool in ASCII format with the URL of the tool by email to Olivier Bonaventure with an indication that the description is submitted for the IEEE Network Software Tools for Networking column. Appropriate tools will be presented in this column.

Olivier Bonaventure
Dept. of Computing Science and Engineering
Université catholique de Louvain (UCL), Belgium

TCP Satistic and Analisys Tool

Marco Mellia
http://tstat.tlc.polito.it

Tstat is able to analyze traces in real time, using common PC hardware, or to start from previously recorded traces in various dump formats (libpcap, DAG, etc.). Besides the more common IP statistics, derived from analysis of the IP headers, Tstat is also able to rebuild each TCP connection status looking at the TCP header in the forward and backward packet flows. Indeed, the software assumes to receive as input a trace collected on an edge node, such that both data segments and ACK segments can be analyzed. When used on a core link, Tstat will ignore flows whose backward path does not cross the measurement point. The TCP flow analysis allows the derivation of novel statistics, such as the congestion window size, out-of-sequence segments, and duplicated segments.

Statistics are collected distinguishing between clients and servers, but also identifying internal and external hosts (i.e., hosts located inside or outside the edge node used as a measurement point). Instead of dumping single measured data, for each measured quantity Tstat builds a histogram, collecting the distribution of that given quantity. Every 15 min it produces a dump of all the collected histograms. A set of companion tools are available to produce time plot or aggregated plot over different time spans. Moreover, a Web interface is available that allows the user to travel among all the collected data, select the desired quantity, and directly produce graphs, as well as retrieve the raw data that can then later be used for further analysis.

More than 80 different types of histograms are available, comprising both IP and TCP statistics. They range from classic measures directly available from packet headers (e.g., percentage of TCP or UDP packets, packet length distribution, TCP port distribution) to advanced measures related to TCP (e.g., average congestion window, RTT estimates, out-of-sequence data, duplicated data).

Tstat derives from S. Ostermann's tcptrace and is written in standard C. It was developed on Linux Redhat systems with 2.2 and 2.4 kernel versions, but should also work on other UNIX variants. The Web site provides many examples, and there is a mailing list for technical support.

Mtools

S.Avallone, M.D'Arienzo, M.Esposito, A.Pescape, S.P.Romano, G.Ventre
http://www.grid.unina.it/grid/mtools

Mtools is a collection of tools for analyzing network performance through the generation of UDP packets flows that traverse the network and the measurement of the one-way delay and round-trip time of each packet. Some of its possible applications are evaluation of the capacity of a network to serve incoming traffic (by loading it in different ways), verification of the accuracy of an analytic model that estimates throughput and packet transmission time based on some assumptions on the traffic source (this raises the need to simulate that particular traffic source), and comparison between different configurations of the same network.

The main features of Mtools compared to other packet generators are the possibility of modeling both packet interdeparture times and packet sizes as random processes (actually constant, uniformly distributed and exponentially distributed), the possibility of specifying the seed value for these random processes (in

order to exactly repeat a particular pattern), and the possibility of recording both packets arrived at the receiver and those transmitted by the sender. Moreover, thanks to R. Davies' random number generator library (included in Mtools), Mtools is easily extensible to support a wide variety of traffic source models (e.g., the last version allows modeling the packet inter-departure times as a Pareto distributed random variable, in order to generate self-similar traffic).

Mtools is written in C++ and currently can be compiled on the Linux and FreeBSD operating systems, requiring only the standard C++ libraries (a Windows version will be released as soon as possible). Installation instructions are provided, and limited support is available by email.

Python Routeing Toolkit

R. Mortier
http://www.sprintlabs.com/ipgroup/PyRT

The Python Routeing Toolkit (PyRT) is a tool written in Python that supports the passive collection and offline analysis of routing protocol data. It collects data by forming minimal router peering sessions and dumping the routing PDUs received over these sessions. PyRT currently supports BGPv4 and IS-IS. Among its advantages is the fact that no routing information is ever advertised and only a minimal amount of information is injected into the network.

The collected data is dumped in the standard MRT format as supported by the RouteViews and RIPE/RIS projects. PyRT also provides a library of routines for parsing, displaying and manipulating MRT format data files. A few example applications using PyRT are included in the distribution.

PyRT is open source (under the GPL licence) and was developed on RedHat 7.1 Linux with Python v2.1.1 and tested against GNU Zebra v0.91a and Cisco 7xxx routers. PyRT has been deployed on a tier 1 ISP for over six months without incident. Data collected has been used to analyze BGP churn, IP level link failures, and IGP/EGP interaction.

REMOTE

K. Christensen
http://www.csee.usf.edu/remote

Parallel independent replications (PIR) is one means of speeding up the execution phase of a simulation modeling project. Simulation modeling is a much used method for performance evaluation of computer networks. In contrast to most existing PIR tools that are UNIX-based, REMOTE tool allows for automatic remote execution of Windows console mode programs (i.e., a .exe or .com "DOS mode" program) on network-connected Windows PCs. This enables idle PCCPU cycles to be used for compute-intensive tasks such as executing simulation models. The REMOTE tool consists of a master PC with a host list and job list. The host list contains all Windows PCs running the remote program. The job list contains a list of executable files and their associated input and output files. The master program sends out the executable and input files to the remote PCs and then receives the output files sent back from the remote PCs. No changes are required to programs to be exectued remotely. To a user, it appears as if all programs executed on the single mster PC. The host and job lists are manually built, but remote PCs running the remote executor can be automatically detected.

The REMOTE tool consists of two executable files: master.exe and remote.exe. The remote.exe file can be emailed to colleagues who wish to "loan" CPU cycles over a weekend (or other period of non-use of their PCs). No special configuration or installation of the master or remote PCs is required. Security features minimize the chance of anyone other than a designated master accessing files or other resources on the designated remote PCs. The REMOTE tool is written in ANSI C as a Windows console mode program (and can be built with both Borland and WindowsC/C++ compilers). The Winsock interace is used for communications. The REMOTE tool is a lightweight remote program executor. It does not have capabilities to redistribute jobs based on load, recover jobs from failures, and so on as do much more complex remote program executors, such as Condor. Support is provided by the author, and ideas for future revisions are solicited.