Performance Measurement Tool for Packet Forwarding Devices

Tamás Kovácsházy[1] and Róbert Szabó[2]

[1]Budapest University of Technology and Economics
Department of Measurement and Information Systems
H-1111, Műegyetem rkp. 9, Budapest, Hungary
khazy@mit.bme.hu

[2]Budapest University of Technology and Economics
Department of Telecommunication and Telematics
H-1117, Pázmány Péter sétány 1/D, Budapest, Hungary
robert.szabo@ttt.bme.hu

Extended Abstract

Keywords

Performance Measurement, Packet Forwarding Device, Workload generation, Time Synchronization

I. Introduction

Packet forwarding devices, switches, routers, proxy servers, firewalls, and others are the backbone of the modern TCP/IP based computer networks, such as the global Internet. These devices are under a fast evolution, because the provided functionality is increasing to accommodate the new applications and needs, and the traffic is growing exponentially parallel with the number of users. As we rely more and more on the infrastructure provided by these networks we need to measure the real word performance of the networks and their components.

To evaluate the performance of such devices expensive measurement equipments are offered by vendors, but these equipment are not optimal in most of the applications because of their price and limited flexibility. For example, they are inaccessible for university research groups and small companies because of their price. In addition, they can not be used in new research areas because they support a well-defined set of stable network protocols and feature sets.

A loosely coupled tool built from various software components is presented here, that allows the performance evaluation of packet forwarding devices and computer networks. The tool used with a proper measurement methodology allows the performance evaluation of Linux based routers as demonstrated in the example. Furthermore, the distributed and modular approach of the tool promises scalabilty to larger scale measurements.

II. PRELIMINARIES

The IP Performance Metrics Working Group (IPPM) and Benchmarking Methodology Working Group (BMWG) of the Internet Engineering Task Force (IETF) have published memos on various topics of benchmarking TCP/IP based networks and devices, see RFC1242, RFC1330, RFC2285, RFC 2544, and RFC2889 for details. These RFCs define fundamental concepts, common terms and vocabulary, and they deal with a wide variety of issues of performance evaluation. The systematic steps of computer performance evaluation procedure were defined by Jain in [1]. There are no identical performance analysis problems but these steps act as a framework which helps to avoid common mistakes.

No public domain tool was found which addresses the complex requirements of packet forwarding device performance measurements. Based on case studies found it can be stated that these kinds of measurements are done manually in practice. Public domain software tools are available to generate workload on packet forwarding devices and computer networks like DSB [2], netspec, or the widely used but outdated ttcp and netperf. Unfortunately, none of them are appropriate for complex, repetitive measurements, and they provide no or very minimal measurement data processing and presentation capabilities. There are other software tools to translate, process, and present data. Text processing tools (e.g., sed) and scripting languages (e.g., perl) can translate file and glue together other software components such as workload generators (e.g., DBS and netspec), data analysis and presentation programs (e.g., gnuplot, Matlab, and MS Excel). LaTeX and html tables are also required output forms to present numerical results in publications.

Delay measurement in computer networks requires time synchronization of the participating devices. The possible solutions are global radio systems such as the Global Positioning Satellite (GPS) system, packet based time Network Time Protocol (NTP), and dedicated time/clock distribution networks [3].

III. RESULTS

There are no identical performance measurement tasks, so creating a monolithic but flexible enough tool was not considered. As a solution, a set of simple tools was built based on available components, and they are glued together for the specific measurement task by perl scripts. Design patterns, i.e., complete measurements created as examples, are provided to help customizing the tool for specific needs.

The list of activities supported by the tool are the following:

- Generation of individual measurements from a high level description of system and workload parameters using a perl program;
- Execution of the individual measurements using a perl program and DBS or netspec as workload generator;
- Conversion of the collected data to matrices stored in binary Matlab files or in TAB separated text files using Matlab;
- Consistency check of the data and repeating the measurement in case of inconsistency problem using Matlab;
- Data analysis using Matlab;
- Presentation, generation of figures and tables using Matlab.

A. Example

The measurement of the packet forwarding performance of Linux based software routers was selected as an example to demonstrate the capabilities of the developed tool. The performance of a router is primarily defined by how many packets it can forward in a second. If the router receives packets close to this limit

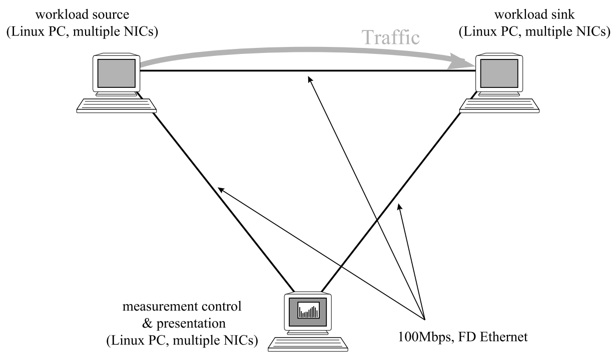

Fig. 1. Measurement setup without router.

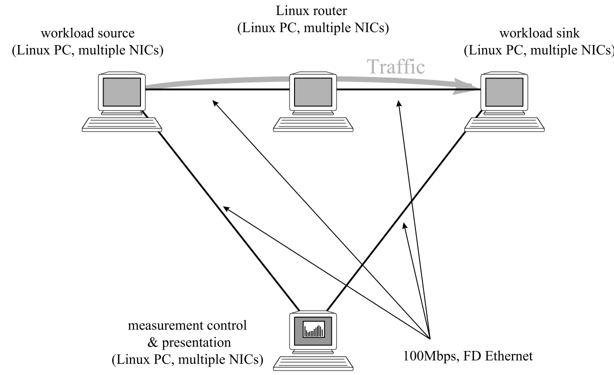

Fig. 2. Measurement setup with router.

it starts to drop packets. Dropped packets may be retransmitted, which increases the traffic even further. Finally congestion can occur, which is observed by users as the visible degradation of the service.

The most prominent change in TCP/IP network traffic is the continuous decrease of average packet size due to real-time traffic, such as voice transmissions (Internet telephony), real-time team games, real-time networked measurement and control, and other interactive applications. Therefore it is interesting to investigate that how Linux can handle routing of small packets. Because the network interfaces have defined bandwidth, the number of packets at 100% utilization is inverse proportional to the packet size. User Datagram Protocol (UDP) packets were generated using DBS from the payload size 64 bytes to 8192 bytes as workload. The UDP packets are encapsulated in IP packets in the network layer, than in Ethernet frames in the data-link layer. The encapsulation procedure guarantees defined IP packet and Ethernet frame sizes.

To verify the performance of the traffic source and sink machines a measurement setup without a router is used, see Figure 1 for details. Later all the measurements are repeated with the router in place, as shown in Figure 2. By comparing these two measurements the performance of the router can be determined if DBS can send and receive more packets than the router can forward, in essence, if the router is the bottleneck in the system. This provision is satisfied in our case.

The measurement done without routers shows that the packet sending performance of DBS is approximately 35,000 packets/sec on the used Linux configuration. The maximum achievable utilization of the 100Mbs link dramatically falls as packets smaller than 512 bytes are sent. The packet loss is approximately 0% at all packet sizes, which means that the DBS sink is capable to handle all packets send by the source.

The performance with router changes very significantly in the region of small packets, because the router can forward only the 50% of the packets sent. The interrupt handling time of Linux is identified as the main bottleneck defining achievable packet forwarding performance. Network Interface Cards (NIC) ask for interrupts both at receiving and sending of packets, and therefore the router needs to service twice as many interrupts as the traffic source or sink.

IV. Novelties

A tool built from primarily public domain software components is presented to evaluate the performance of packet forwarding devices. Using the tool it is possible to specify complex traffic patterns based on a high level definition, then to execute the designed measurements, to process the results of measurements, and finally to generate high quality figures and tables for presentation. As an example, the packet forwarding performance of a Linux based router is investigated, and the limitations of such a device is shown. The

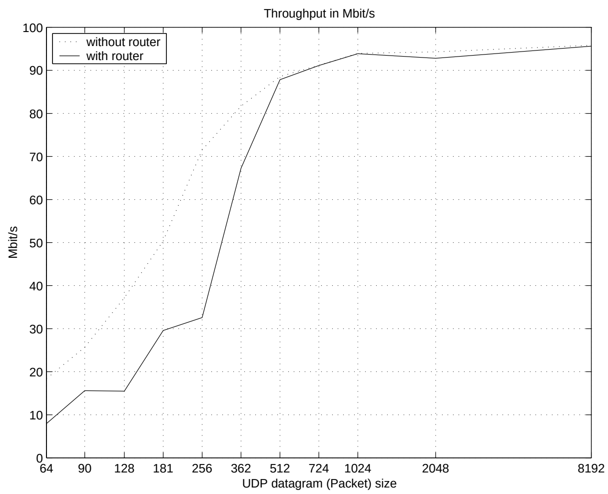

Fig. 3. Throughput achieved in Mbit/s with and without the router.

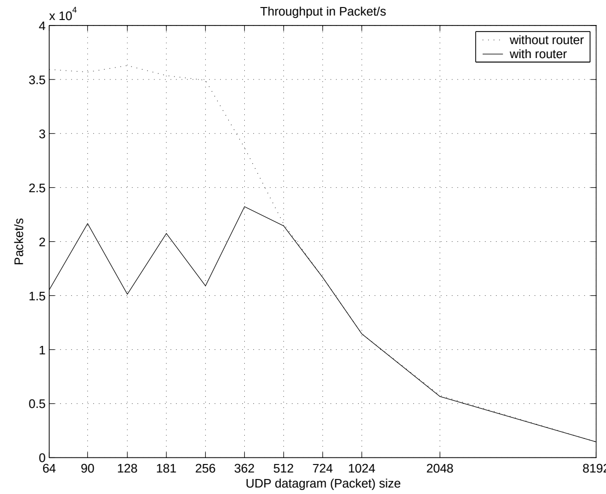

Fig. 4. Throughput achieved in packet/sec with and without the router.

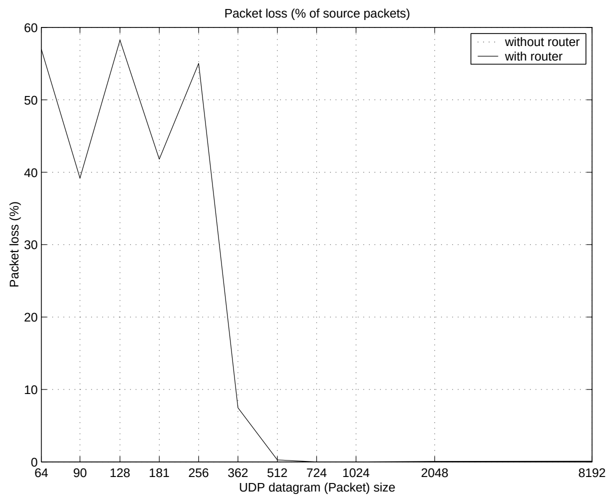

Fig. 5. Packet loss with or without the router.

tool promises to scale to more complex, higher speed devices by using synchronized set of traffic sources and sinks.

References

[1] Raj Jain, *The Art of Computer Systems Performance Analysis*, John Wiley & Sons, INC., 1991.

[2] Yukio Marayama and Suguru Yamaguchi, "DBS: a powerful tool for TCP performance evaluation," in *Proceedings of the 1997 International Conference on Performance and Control of Network Systems*, November 1997, pp. 570–581.

[3] Cliff Hodge David W. Allan, Neil Ashby, "Fine-tuning time in the space age," *IEEE Spectrum*, pp. 41–50, March 1998.