Measuring Accuracy and Performance of Network Emulators

Robert Lübke, Peter Büschel, Daniel Schuster, Alexander Schill
Faculty of Computer Science
Computer Networks Group
Technische Universität Dresden
Email: {robert.luebke, daniel.schuster, alexander.schill}@tu-dresden.de

***Abstract*—Network emulators are often used tools for different kinds of experiments, for example testing network protocols and studying application behavior under certain network conditions. It is crucial for network emulators to work according to their specifications in order to ensure reliability and reproducibility of the performed experiments. This paper therefore evaluates the accuracy and performance of current well-established network emulators, including hardware and software solutions. The results show that no emulator reproduces all effects perfectly. Although the accuracy is acceptable in general, we finally analyzed strengths and weaknesses of the different solutions. Other researchers can use our measurement results to choose the right emulator for their experiments.**

I. Introduction

Network emulation is used in research as well as industry to test and evaluate the behavior of applications under various network conditions. It is an alternative approach to network simulation and using real-world systems. Emulation uses real network traffic and degrades it according to a given configuration of bandwidth limitation, packet delay, jitter, loss, duplication and reordering. It is essential for emulators to reproduce the desired network conditions accurately to prevent unreliable experiments with misleading results. Therefore, it is crucial to test and evaluate them intensively. Many different network emulators have been developed by researchers and industry. Unfortunately, comparative studies regarding features and precision of the different approaches are scarce.

Therefore, this paper contributes with a comparison of the accuracy and performance of current network emulators. We focus on well-established software emulators, but also include hardware solutions into our comparative measurements. Furthermore, we do not focus on one single aspect, but try to cover all functionalities of the respective emulator in our evaluation. After presenting the network emulation solutions we compared, existing comparative studies and their shortcomings are highlighted. To enforce reproducibility of our measurements, we present our measurement concepts. Finally, we discuss the measurement results and conclude the paper.

II. Related work

In research there are many software-based solutions available for network emulation. In our comparative study we choose Dummynet, KauNet and NetEm as well-established representatives of software emulators because of their current popularity.

NetEm [8] is part of the Linux kernel and supports the emulation of packet delay, loss, duplication and corruption. The configuration is done via the Linux Traffic Control (tc) tools. This tool set also include a Token Bucket Filter (tbf) for bandwidth limitation. NetEm was evaluated in [4] and found to emulate most parameters accurately. Only when performing jitter emulation the authors observed deviations of configured and measured values. Another well-established emulator is Dummynet [1]. It is part of the FreeBSD kernel, but also available for Linux and Windows. The basic concepts of Dummynet are *rules* and *pipes*. Similar to a firewall configuration, the rules decide which incoming or outgoing packets must pass which pipes. These pipes can then be attached with certain network characteristics, including a maximum bandwidth, packet loss, delay and reordering. KauNet [2] extends Dummynet with the emulation of bit errors and improves its precision and reproducibility by employing a pattern-based approach. These patterns define which packets get degraded with certain network characteristics. The emulation can then be performed time-driven and data-driven using the patterns.

Besides the discussed software-based solutions, there are also hardware emulators offered by different manufacturers. They can show a very high precision even under heavy load. Unfortunately, research work evaluating this is still missing. Compared to software emulators a clear disadvantage of hardware-based solutions is the high purchase price. In this paper, we use the *Linktropy 7500 PRO* by *Apposite Technologies*[1] as a representative of hardware emulators for our measurements.

There are existing comparative studies of multiple network emulators [6] [7] [5], but they have different shortcomings. No related study covers all aspects that are relevant when choosing a network emulator for an experiment. They only focus on bandwidth and delay, but omit packet loss, reordering and duplication. Furthermore no comparative study takes hardware network emulators into account. Another problem is that some of the performed measurements are too old. New versions of the emulators as well as the underlying operating systems have been released. Furthermore the used hardware is also out-dated after about five years.

In the following, we therefore discuss our concepts for extensive comparison measurements of network emulators. The main concepts of the experimental setup and the measurement procedures are based on ideas from the discussed related work.

[1] Apposite Technologies: `http://www.apposite-tech.com`

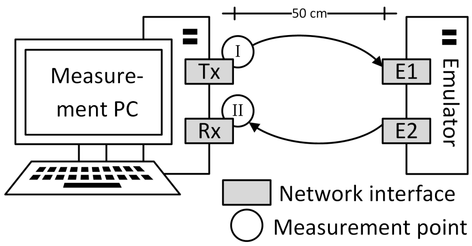

Fig. 1. Experimental setup used for comparative measurements

III. Measurement concept

Experimental setup: The physical experimental setup we use for our measurements is illustrated in Figure 1. It consists of one measurement PC and the respective emulator, each containing two network interfaces. Probe packets are sent via *Tx*, arrive at the emulator at *E1*, get degraded and leave the emulator via *E2* to finally arrive at *Rx*. Each network interface of the measurement PC acts as a measurement point (MP). Packets traversing a MP are captured for later evaluation. Adding additional MPs on the emulator itself (see [3]) is not possible in our scenario as we want our setup to be independent from the employed emulator. Hardware-based emulators for example cannot be attached with MPs.We choose a setup with only one measurement PC in order to be able to accurately measure one way delays. This allows us to use the local PC time for measuring the time span between sending (MP-I) and receiving (MP-II) packets. Other studies just omit one way measurements or use time synchronization to adjust the timers of all involved measurement points. Time synchronization means like NTP fail to provide the desired accuracy of microseconds.

The emulator PC runs different operating systems for the emulators due to their requirements: FreeBSD 10.1 for Dummynet, FreeBSD 7.3 for KauNet and Debian 7 (3.2 Kernel) for NetEm. The hardware specifications of our experimental setup are: Measurement PC with Intel i7 820 @2.93GHz, 8GB DDR3; Emulator with Intel Core 2 6600 @2.4GHz, 4GB DDR2; all network interfaces are Intel Pro GT/PT 1000.

Procedure: For performing the actual experiments we use a previously developed network measurement tool called NORA (*Network-Oriented Rates Analyzer*). It is able to measure all required network parameters in one measurement, including bandwidth, delay and jitter as well as packet effects like loss, duplication and reordering. It further provides a fine-grained configuration of all required parameters of the probe streams.

At first, there is a preexamination phase for each emulator in which we measure the emulators' buffer sizes, base delay values and maximum loss-free packet forwarding rates with emulation turned off. After that, the concrete measurements are performed. We send probe streams with different characteristics, regarding packet size (64, 512, 1024 and 1500 Byte), inter departure time (20, 120 and 1000 μs) and number of packets. If support by the emulator is given, we configured presetting values for bandwidth (7.2, 54, 61 and 160 MBit/s), delay (15, 50, 100 and 1000 ms) and the three packet effects (0.001, 0.01, 0.1 and 1 %). If the emulator is able to emulate network paths, this feature is evaluated with a path length of 10.

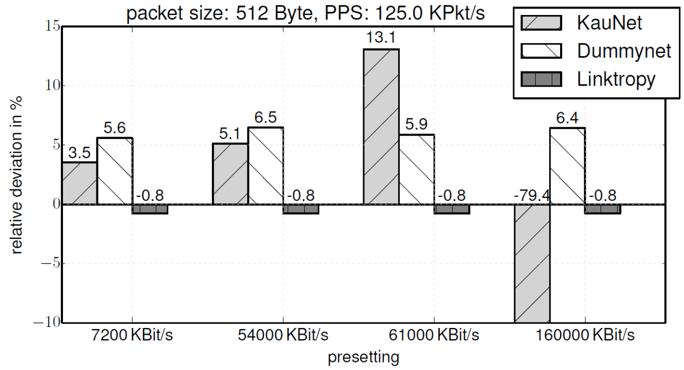

Fig. 2. Relative deviation of configured and measured bandwidth

IV. Measurement results

Due to limited space this section discusses only selected results of the accuracy measurements, regarding bandwidth, delay and the three packet effects loss, reordering and duplication as well as path emulation.

Bandwidth: For bandwidth measurements NetEm was omitted, because it does not directly support this feature. Although, the Linux *tc* toolkit provides different means for bandwidth emulation that can be used in combination with NetEm. Figure 2 clearly shows that the hardware solution achieves the best accuracy in bandwidth emulation. Its relative deviation of configured and measured values is under 1 % in all measurements. The reasons for these very good results are the large buffer sizes and the high processing speed of the hardware. KauNet and Dummynet do show much higher deviations, because of smaller buffers in the default configuration. KauNet even produces faulty results when emulating bandwidths higher than 64MBit/s. We found an integer overflow to cause this wrong behavior that leads to a deviation of -79.4 %. Furthermore, we were able to observe, that all candidates emulate more accurately with small probe stream rates and big packet sizes. Many small packets produce more management overhead than few big ones. High packet rates and small packets lead to high load and inaccuracy.

Delay: The results of the one way delay measurements in Figure 3 show that especially Linktropy and NetEm can achieve very accurate emulations. The reason for the measured deviations of Dummynet and KauNet is the default timer frequency of 1000 Hz. Netem can make use of high resolution timers by default. Linktropy clearly has the advantage that its hardware timers have been designed for precise emulations. Figure 3 also illustrates that the accuracy of delay emulation depends on the packet size. Large packets lead to more inaccurate results on all emulators.

Packet effects: Due to its deterministic pattern-based approach, KauNet is able to emulate the three packet effects loss, reordering and duplication perfectly and achieves stable results. In all measurements with different probe stream characteristics the various expected values can be determined exactly. Concerning packet loss the other emulators do show differences of configured and measured values with a maximum of $\pm$3% deviation. As Linktropy, NetEm and Dummynet use pseudo random number generators, the measurement interval must be wide enough to produce stable results. Regarding

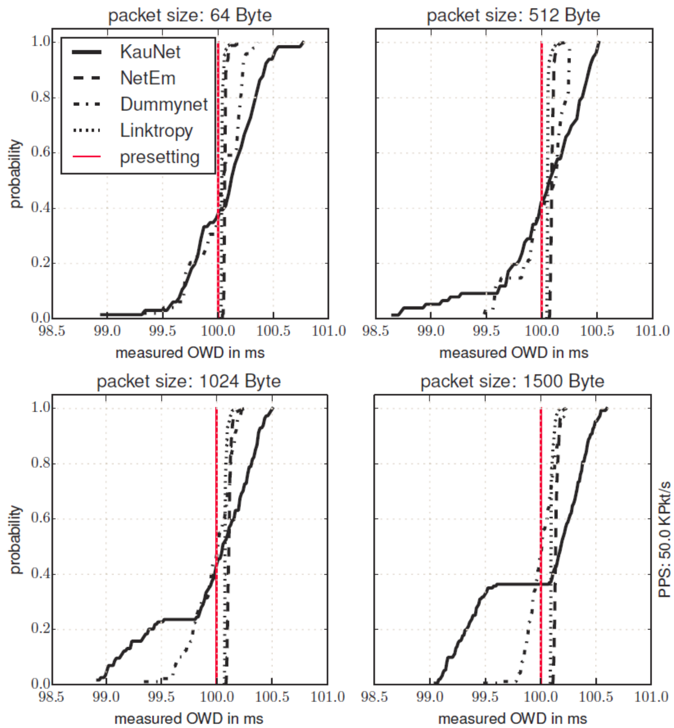

Fig. 3. Distribution functions for one way delay using a presetting of 100ms

the emulation of packet reordering, Linktropy's measured reordering rates comply with the expected values, but they show higher variations. NetEm finally reorders too many packets for high packet rates. When using low packet rates NetEm does not apply reordering at all. Unfortunately, only NetEm and Linktropy support the emulation of duplication. Both use pseudo random number generators and produce very similar results. Large packet rates again tend to worsen the emulation accuracy.

Network path emulation: Unfortunately, Dummynet was the only emulator able to emulate network paths with each node having unique network characteristics. KauNet should also support this feature, but it failed and produced system crashes when we tried to use it. We tested all supported network parameters and effects of Dummynet with a path length of 10. In general the results turned out satisfactory. Measuring bandwidth in a network path produced similar results as discussed before. When we configured each node of the path with 10 ms delay, we were able to measure the expected value of 100 ms. If each node is set to emulate 1 % loss, the expected overall loss rate is about 9.6 %. Dummynet was able to emulate this behavior as well.

Final comparison: The major strengths and weaknesses of each analyzed emulator are shown in Table I. Obviously, there is no solution covering all aspects perfectly. If accuracy in bandwidth and delay emulation are crucial, then a hardware solution like Linktropy is the best match. For these parameters Dummynet and KauNet produce inaccurate results. Dummynet clearly has the advantage of network path emulation. KauNet has the unique feature of pattern-based emulation that makes it especially suitable for packet effects. NetEm itself cannot be used for bandwidth emulation, but it achieved very accurate

TABLE I. RECOMMENDED APPLICABILITY OF THE EXAMINED NETWORK EMULATORS

	Linktropy	NetEm	Dummynet	KauNet
bandwidth	+	x	o	−
delay	+	+	o	o
packet effects	o	o	−	+
network path	x	x	+	x

+: most suitable, o: applicable, −: not recommended, x: not supported

results for the delay values. In combination with the Linux *tc* toolkit NetEm definitely is a powerful emulator as well.

V. Conclusion

This paper gives a comparison of accuracy and performance of current network emulation solutions. Previous comparative studies only focused on certain aspects, but we evaluate the full feature set of network emulation that is necessary to reproduce real-world network conditions precisely. We focused our comparative measurements on three well-established software emulators and one professional hardware emulator. But the measurement concept can easily be applied to any other network emulator as well. The measurements results have shown various differences concerning accuracy of the emulators. Therefore, choosing a suitable emulator carefully is crucial for the quality of the experimental results. Our final comparison can help other researchers in this process.

Acknowledgment

This work results from the NESSEE project that has been financed by Citrix Systems, Online Services Division. We would like to thank Citrix for letting us use their hardware for test measurements and evaluation.

References

[1] M. Carbone and L. Rizzo, "Dummynet Revisited," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 12–20, April 2010. [Online]. Available: http://doi.acm.org/10.1145/1764873.1764876

[2] J. Garcia, P. Hurtig, and A. Brunstrom, "KauNet: A Versatile and Flexible Emulation System," in *Proceedings of the 5th Swedish National Computer Networking Workshop (SNCNW 2008)*, Karlskrona, Sweden, April 2008.

[3] T. Hall, P. Hurtig, J. Garcia, and A. Brunstrom, "Performance evaluation of kaunet in physical and virtual emulation environments," Karlstad University, Department of Computer Science, Tech. Rep. 2012:32, 2012.

[4] A. Jurgelionis, J. Laulajainen, M. Hirvonen, and A. Wang, "An empirical study of netem network emulation functionalities," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, July 2011, pp. 1–6.

[5] T. Minhas, M. Fiedler, J. Shaikh, and P. Arlos, "Evaluation of throughput performance of traffic shapers," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, July 2011, pp. 1596–1600.

[6] L. Nussbaum and O. Richard, "A Comparative Study of Network Link Emulators," in *Proceedings of the 2009 Spring Simulation Multiconference*, ser. SpringSim '09. San Diego, CA, USA: Society for Computer Simulation International, 2009, pp. 85:1–85:8. [Online]. Available: http://portal.acm.org/citation.cfm?id=1639809.1639898

[7] J. Shaikh, T. Minhas, P. Arlos, and M. Fiedler, "Evaluation of delay performance of traffic shapers," in *Security and Communication Networks (IWSCN), 2010 2nd International Workshop on*, May 2010, pp. 1–8.

[8] Stephen Hemminger, "Network Emulation with NetEm," in *linux.conf.au*, Canberra, Australia, April 2005.