MoonGen: Software Packet Generation for 10 Gbit and Beyond

Sebastian Gallenmüller, Paul Emmerich, Daniel Raumer, and Georg Carle
Technische Universität München
{gallenmu | emmericp | raumer | carle}@net.in.tum.de

1 Introduction

Packet generation is crucial to network measurements. Despite its importance, software packet generators lack the accuracy and precision to model realistic test scenarios [1]. Hardware traffic generators provide the necessary precision and accuracy but tend to be inflexible and expensive.

MoonGen presents a novel approach to provide flexible, accurate, and precise traffic generation in software on inexpensive commodity hardware. Moreover, it features latency measurements in the sub-microsecond range. We offer MoonGen[1] as free software under MIT licence. For more information, we refer the interested reader to the draft of our full paper [2].

2 Architecture of MoonGen

Fig. 1 shows the layered structure of MoonGen. The *hardware* is managed by DPDK[2] to support bandwidths of $\geq$ 10 Gbit/s. This allows packet generation at line rate, 14.88 million packet/s (*Mpps*), with 64 Byte packets per CPU core; a scenario where traditional approaches are doomed to fail [3]. We have benchmarked MoonGen with up to 89 Mpps, line rate at 60 Gbit/s.

The *MoonGen core* takes care of the initialization process of DPDK and provides a simple interface for userscripts.

A *userscript* contains the entire packet generation logic. Each single packet is crafted in real-time by a user-provided script. Other packet generators with scripting APIs, e.g. Pktgen-DPDK[3], offer basic configuration through scripts. Packets can only be generated by predefined patterns with these tools.

MoonGen scripts are written in Lua running on LuaJIT[4]. This engine is able to provide necessary performance and offers a powerful foreign-function API to interface with the underlying DPDK library.

The *master* task of a userscript contains the configuration data to set up a measurement. Userscripts also incorporate *slave* tasks generating the test traffic for measurements. Running slave functions in parallel allows complex processing logic with linear multi-core scaling and support for multiple NICs and queues.

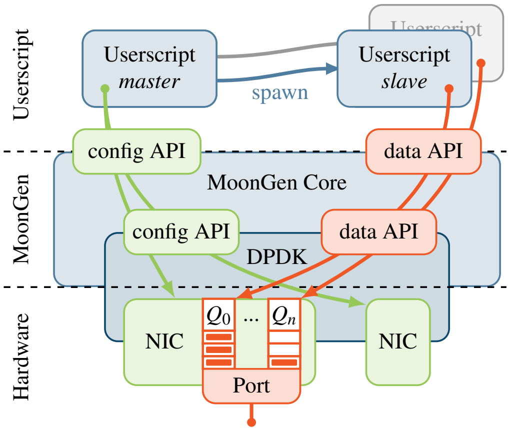

Figure 1: Architecture of MoonGen

3 Precise Timestamping

MoonGen (mis-)uses hardware features on modern commodity NICS meant for clock synchronization (via the Precision Time Protocol) for delay measurements. This achieves sub-microsecond accuracy and precision.

4 Live Demonstration

We will show that MoonGen reaches 14.88 Mpps (line rate) running on a single CPU core on a recent Intel CPU with a clock speed of 2 GHz while crafting each packet in real-time in a user-controlled Lua script. Moreover, we will show a live demonstration of the timestamping accuracy by determining cable lengths via time-of-flight measurements. Additionally, we present live delay measurements of a device under test with detailed histograms.

References

[1] Botta, A., Dainotti, A., and Pescapé, A. Do you trust your software-based traffic generator? *IEEE Communications Magazine 48*, 9 (2010), 158–165.

[2] Emmerich, P., Gallenmüller, S., Wohlfart, F., Raumer, D., and Carle, G. MoonGen: A Scriptable High-Speed Packet Generator. `http://go.tum.de/276657`, 2015. *Draft, conference tbd.*

[3] Rizzo, L. netmap: a novel framework for fast packet I/O. In *USENIX Annual Technical Conference* (2012), pp. 101–112.

[1] https://github.com/emmericp/MoonGen/
[2] http://dpdk.org/
[3] https://github.com/Pktgen/Pktgen-DPDK/
[4] http://luajit.org/