

MOONGEN PERFORMANCE EXPERIMENTS

Generalized Methodology for Emulating Complex Network Functions

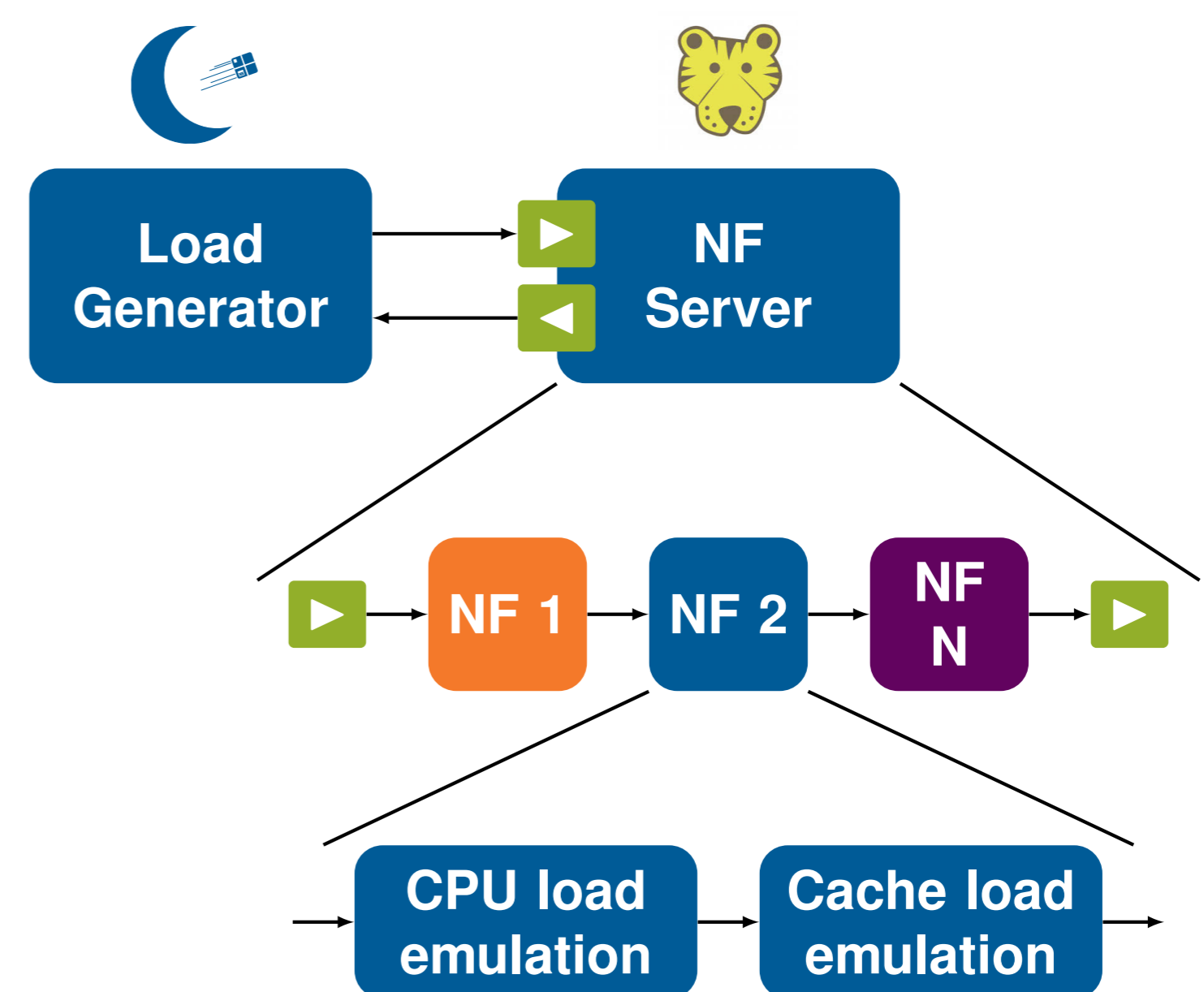
Network Function measurements

The Demo shows a measurement setup for network functions. The setup replicates complex network function chains with the following methodology:

1. Determination of load profile for real network functions, e.g., routing, VPN, firewall
2. Imitation of real network functions with *synthetic* network functions which forward packets while simultaneously imitating the original CPU/cache load profile
3. Measurement of throughput and latency utilizing the synthetic network function chain

The advantages of such a synthetic setup are:

- ▶ Simple setup imitating complex network functions
- ▶ CPU-intensive network functions are emulated without requiring a real implementation (if the CPU load profile is known)
- ▶ Throughput and latency prediction possible by combining different network functions



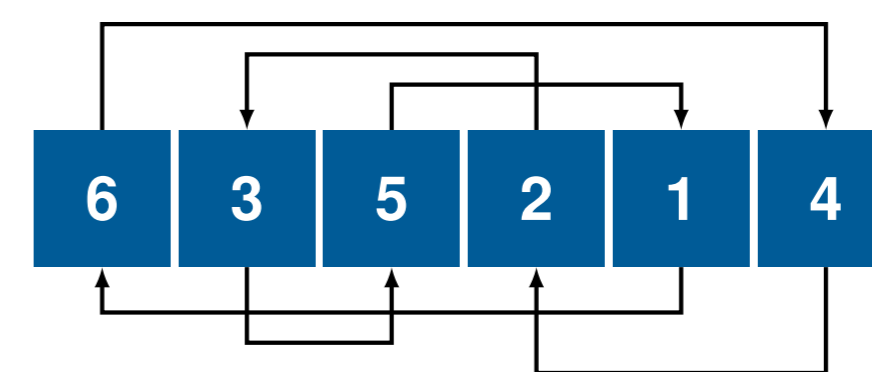
CPU load emulation

```
mov %0, %%ecx
inc %%ecx
1: dec %%ecx
cmp $0, %%ecx
jnz 1b : r (cycles)
```

CPU processing time is a resource consumed in software-based packet processing.

- ▶ CPU processing time can be emulated by executing specific code such as the assembler loop shown above
- ▶ CPU load can be emulated precisely (specifying the `cycles` variable in the assembler loop above) [2]
- ▶ Looping for a certain amount of `cycles` emulates constant or predictive costs, e.g., checksum calculation or encryption

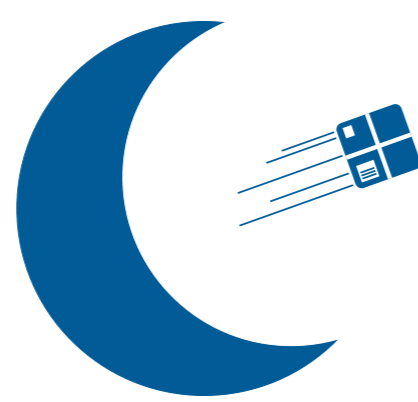
Cache load emulation



A randomly linked list is used to emulate cache accesses.

- ▶ Size of the linked list can be adjusted by configuring the size of the each element and the length of the list
- ▶ Using different sizes influences memory being used, e.g. L1 – L3 cache or RAM
- ▶ Number of accesses to linked list can be configured
- ▶ Good for emulating tasks with memory dependent costs, e.g., routing table lookups, firewalls, DPI

MoonGen



MoonGen [1] is a packet generator developed at the Chair of Network Architectures and Services:

- ▶ Latency measurements in the nanosecond range
- ▶ 10 Gbit/s packet generation on a single core
- ▶ High-precision generation of traffic patterns
- ▶ Easy extendability using the Lua scripting language

Snabb



Snabb is a high-performance packet processing framework (similar to DPDK, netmap):

- ▶ Packet processing entirely in user space
- ▶ High-Performance packet processing on commodity hardware
- ▶ Network functions written in Lua scripting language

[1] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle. MoonGen: A Scriptable High-Speed Packet Generator. In *Internet Measurement Conference 2015 (IMC'15)*, Tokyo, Japan, Oct. 2015.
 [2] S. Gallenmüller. Simulation algorithms for Empirical Evaluation of Processor performance (SHEEP). <https://github.com/gallenmu/sheep>.