



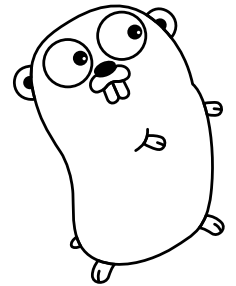
Thesis  
M.Sc.

IDP,  
Guided  
Research

## Efficient high-speed packet processing in high-level languages

### Motivation

Efficient packet processing on high-speed links (e.g. 10 Gbps) is mostly implemented in low-level languages such as C or C++. The reason for that is, that high-level languages lack performance and predictability for time-critical tasks such as packet acquisition, flow matching etc. However, code written in high-level languages tends to be easier to extend and maintain. It is desirable to have a small code-base in C or C++ handling the performance critical parts of the packet processing and on top the logic written in a high-level language.



An example for such an approach can be found in the stenographer project<sup>a</sup>. The flow monitoring tool Vermont<sup>b</sup> predates such a design. It is a monolithic binary written in C and C++ and does not scale well on multi-core CPU architectures. To enhance the maintainability and extensibility of the code base and exploit modern CPU architectures, part of Vermont's code needs to be redesigned and rewritten.

Gopher: CC BY <https://github.com/keygx/Go-gopher-Vector/blob/master/svg/gopher03bw.svg>

<sup>a</sup><https://github.com/google/stenographer/blob/master/DESIGN.md>

<sup>b</sup><https://github.com/tumi8/vermont>

### Your Task

- Investigate which parts of the capturing logic need to be written in C/C++
- Use modern multi-core friendly packet capturing frameworks, e.g. PF\_RING ZC
- Implement configuration and high-level processing tasks in a high-level language
- Evaluate the performance characteristics of such a modular architecture

### Prerequisites

- Knowledge of a high level programming language (Go, Python or Lua)
- Understanding of C/C++ and modern computer architecture
- Willingness to learn and understand the architecture, characteristics and trade-offs of the chosen tools
- Interest and basic knowledge in network measurement such as IPFIX
- Familiarity with GIYF-based work approaches to avoid LMGTFY-scenarios

### Contact

Oliver Gasser [gasser@net.in.tum.de](mailto:gasser@net.in.tum.de)  
Johannes Naab [naab@net.in.tum.de](mailto:naab@net.in.tum.de)

<http://go.tum.de/306574>

