**Background**

In the coming years, cloud computing operators are expected to be prepared for a wide range of cloud services including video and music streaming, communication via text/voice/video, stock trading, and personal content storage, etc. By applying Network Function Virtualization (NFV), these services can be defined as a composition of multiple virtualized service functions that should be traversed by network flows in a specific order. The simplest case for such a service is a linear chain of at least one service function between two specific endpoints in the network. Inserting functions that can split network flows over different paths makes the structure of a service more complicated than a simple chain.

For resource allocation and network optimization purposes, network operators need precise representations that model the structure and requirements of these services. When the order of traversing the functions is fixed and given (e.g., for the traffic going through different application tiers), traditional graph representations can be used for modeling the services. But these representations quickly become ineffective when the exact order of traversing the functions is not specified or not relevant (e.g., for operational and administrative functions). Moreover, scaling properties of the service functions (e.g., number of required instances) might be unknown before the actual deployment. Therefore, deployment requests for complex services need to be composed as a template that shows the overall architecture of the service and specifies the requirements and scaling properties of its components.

**Thesis**

The initial step in this thesis is to investigate existing service specification models [1, 2] and design a flexible representation for describing the composition and requirements of complex services, including options and restrictions for scaling and reordering service functions. Based on the specification model, a placement algorithm with the following capabilities should be designed:

- Determining the number of required instances for each function and the final chaining structure for the functions
- Identifying the best location to instantiate application virtual machines [1] and virtualized network functions [2] in an integrated way
- Identifying the best routing paths among functions

**Milestones**

- Literature search
- Get to know the specification and placement models in Ref. [1] and [2]
- Extend the application template and placement optimization problems
- Evaluate the new approach

**Required Knowledge**

- Understanding of computer networks
- Understanding of optimization problems
- Python

**References**


Contact: Sevil Mehraghdam, s.mehraghdam@upb.de, Tel: 05251 / 60 1755, Room: O3.161