Software Defined Resource Allocation for Service-Oriented Networks
Nan Zhang, Ya-Feng Liu, Hamid Farmanbar, Tsung-Hui Chang, Mingyi Hong, and Zhi-Quan Luo

Motivation
- Today’s networks must support diverse service requirements, each service consists of a predefined service function chain (SFC).
- Traditional specialized network hardware provides dedicated network services — it is costly and inflexible.
- Network function virtualization (NFV) [1]: intelligently integrate a variety of network resources to establish a virtual network (VN) for each request.
- Joint VN embedding and resource allocation [2, 3, 4]: select function nodes for service function instantiation.
- Route traffic such that each flow gets processed at function nodes in the order defined in the corresponding SFC.

Main Contribution
- Perform joint VN embedding and traffic engineering for service-oriented networks.
- Propose a novel problem formulation taking practical network constraints into consideration.
- Show NP-hardness of the formulated problem.
- Develop an efficient penalized successive upper bound minimization (PSUM) algorithm with convergence guarantee.

System Model
- Flow $k$ shall be transmitted from $S(k)$ to $D(k)$ with rate $R(k)$.
- SFC of flow $k$: $F(k) = f_1 \rightarrow f_2 \rightarrow \ldots \rightarrow f_M$.
- The set of function nodes that can provide function $f$: $V_f$.
- Binary variable indicating whether function node $i$ provides function $f$ for flow $k$: $x_{i,f}(k)$.
- Rate of virtual flow $(k,f)$ over link $(i,j)$: $r_{i,j}(k,f)$.
- Each function node provides at most one function for each flow: $\sum_{f} x_{i,f}(k) \leq 1$.

Problem Formulation and Analysis
- Link capacity constraint: $\sum_{i} r_{i,j}(k,f) \leq C_{ij}$.
- Node capacity constraint: $\sum_{f,j} x_{i,f}(k) R_{i,j}(f) \leq \mu_i$.
- Network flow conservation constraints: $L(k) x_{i,(k,f)}(k) = \sum_{j \neq i} r_{j,i}(k,f) - \sum_{j} r_{i,j}(k,f) - \sum_{j} x_{j,(k,f)}(k)$.
- Joint VN embedding and resource allocation.
- The total link rate optimal avoids cycles in choosing routing paths.
- The problem of checking the feasibility of (P) is NP-hard (Proved).

PSUM Algorithm
- Relax binary variables $(x_{i,j}(k,f))$ to be real and add a penalty term to the objective function.
- Solve for penalized problem (P1) with the objective function $g_p(x)$.
- Convergence analysis: Suppose the positive sequence $(\sigma_p)$ is monotonically increasing and $\sigma_p \rightarrow \infty$, and $\sigma$ is a global minimizer of the penalized problem (P1) with the objective function $g_p(x)$. Then any limit point of $(\sigma_p)$ is a global minimizer of problem (P).
- Successive Upper Bound Minimization (SUM) [6]: solve a sequence of approximate objective functions which are lower bounded by $g_p(x)$.

Simulation Results
- Simulation scenario: a mesh network.
- 100 nodes and 684 direct links.
- 5 service functions, $|V_s| = 10$ candidate nodes for each function.
- $F(k) = (f_1 \rightarrow f_2 \rightarrow f_3)$ and $S(k), D(k)$ are uniformly randomly chosen for each flow $k$.
- Parameter setting: $C_{ij} \sim [0.5, 5.5]$, $\mu_i \sim [0.5, 8]$, $K = 30$, $R(k) \leftarrow 1/k$.
- Compare with the modified heuristic algorithm in [3].

References