On the Placement of Security-related Virtualised Network Functions Over Data Center Networks

by Abeer Ali

Supervised by
Prof. Dimitrios Pezaros , University of Glasgow
Dr. Christos Anagnostopoulos , University of Glasgow

Agenda

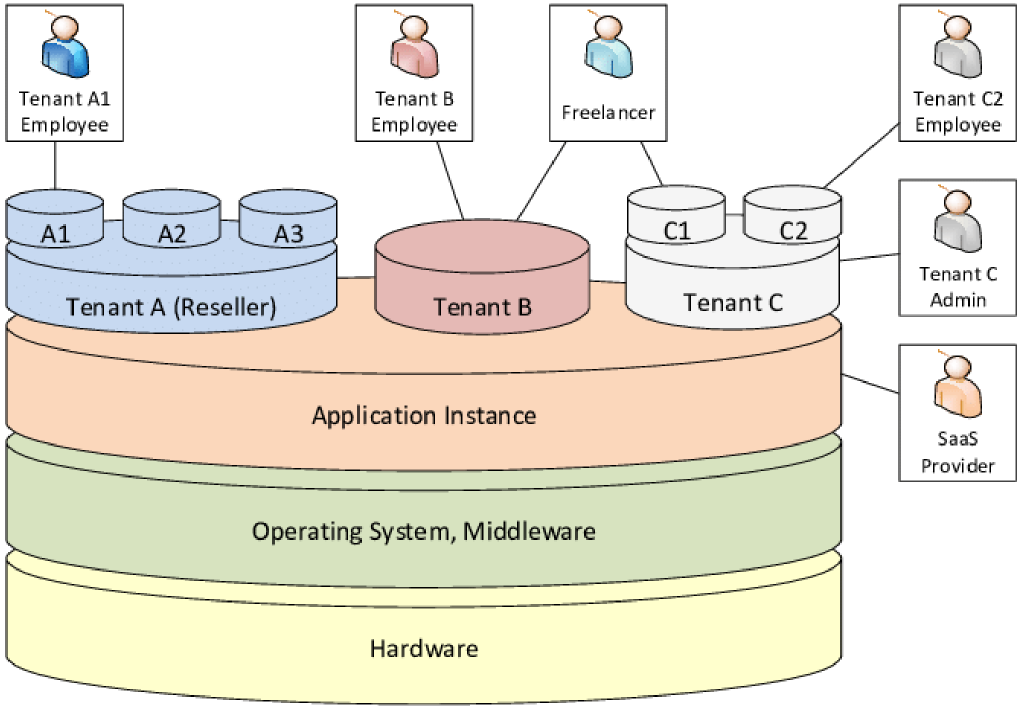
- Background
- Research objectives
- Literature Review
- Proposed System
- Evaluation

Background

- Multi-tenant Environments
- Security
- Recent trend in Virtualised functions management (VNF and SDN)

Background:Multi-tenant Environments

- Public Cloud
  - Amazon AWS
  - Microsoft Azure
  - IBM's Blue Cloud
  - Sun Cloud
  - Google Cloud



Background:Network Security Solutions

Hardware-based Middleboxes
Legacy implementation

- Fixed allocation
- Centralized & Monolithic systems
- Limited extent of functionality
- Vendor lock-in
- Expensive

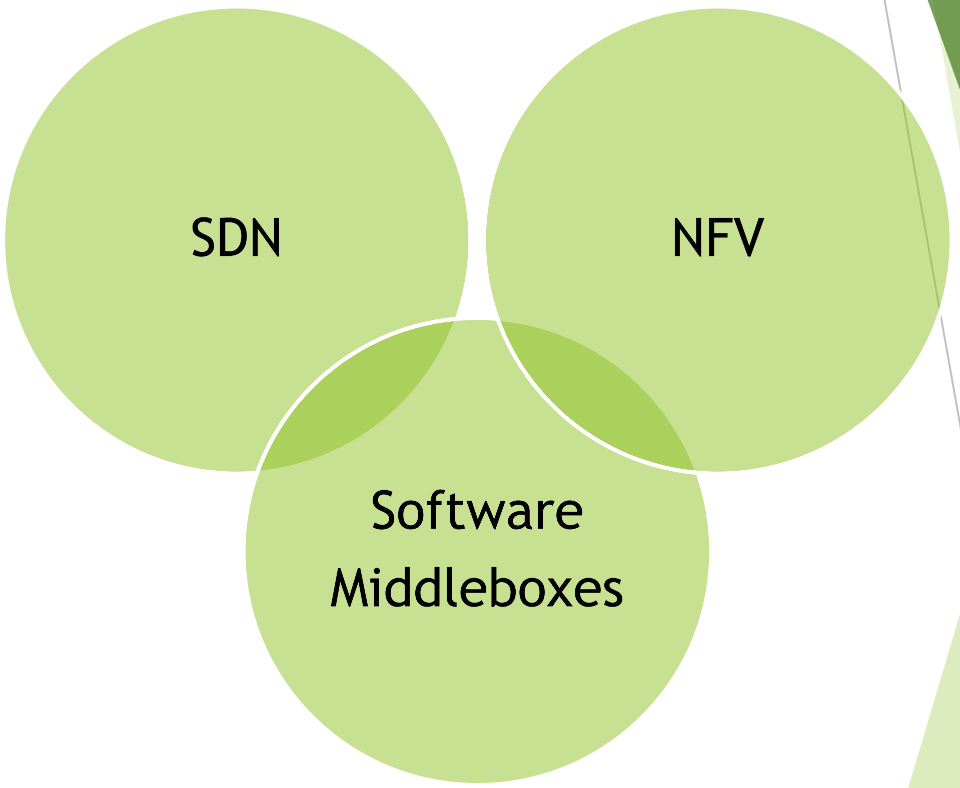
Software-based Middleboxes

- Rapid and Flexible deployment
- Scalable resources
- Allow extension of functionality
- No Vendor lock-in
- Inexpensive compared to HW

- Security Services in Amazon’s AWS Multitenant virtualized infrastructures
  - (2015) Firewall web application(WAF) ,Dec 2016 AWS Shield (DDoS protection services) , Nov 2017 GuardDuty (Intelligent threat detection)
  - Alert Logic , Armor, Cisco and Barracuda

Management of Software middleboxes

- Complex Problem
- NFV Network function Virtual
  - Software based NF
  - Efficient resource provisioning
  - Flexibility of Placement
- SDN Software Defined Network
  - Centralised control
  - Programmability
  - Global view of the network



Research Objectives

- Problem
  - The placement of Security Functions in Multi-tenant Data Centers.
- Research questions and Objectives
  - Are security functions have unique characteristic as VNF? Identify
  - Design of a placement framework
    - Consider the unique characteristics of security functions
    - Provide customised security services
    - in multi-tenant data centers.

Security Functions Equivalence Classes

Stateless

- Firewalls
- Signature-based (IDS)
- Deep Packet Inspection(DPI)
- Examples: ZoneAlarm, Snort, Suricata

Stateful

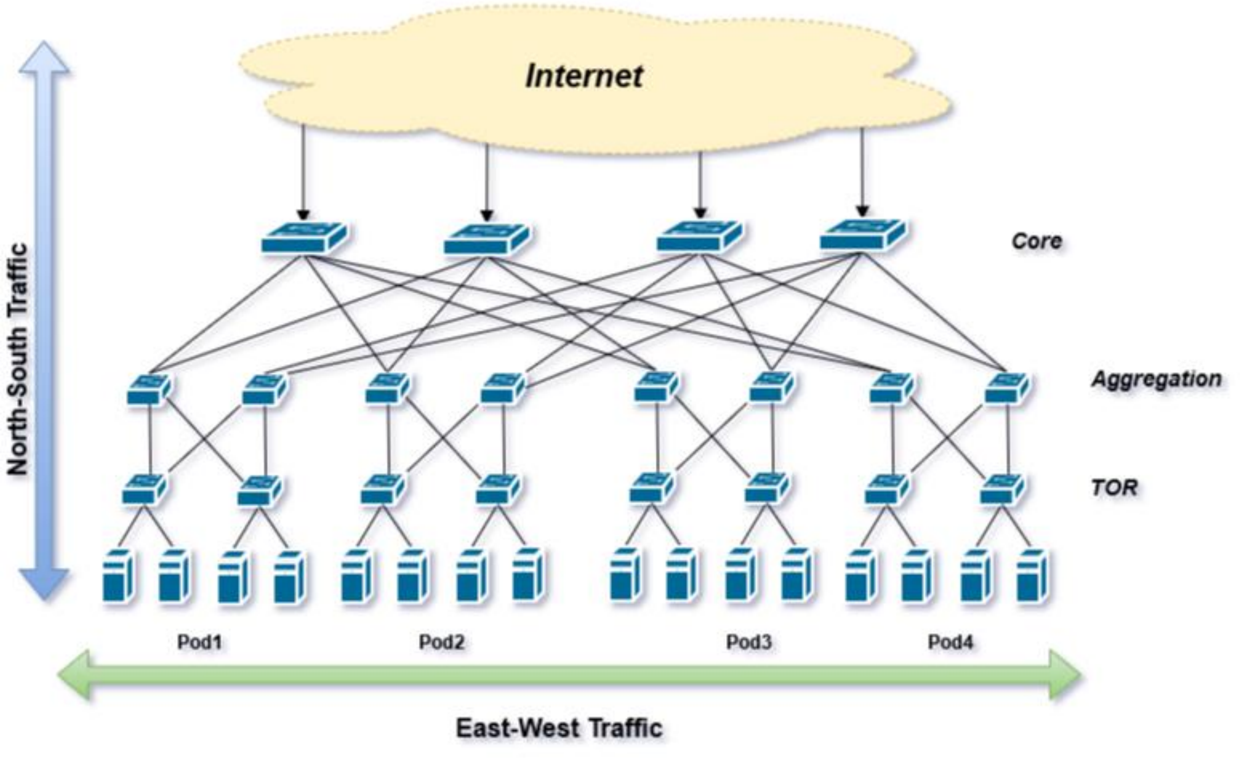
- Anomaly based IDS,IPS
- Examples: Changepoint Detection, Entropy and Classifiers

Allocation

- Stateless: Independent duplication
- Stateful: Single instance or depended duplication

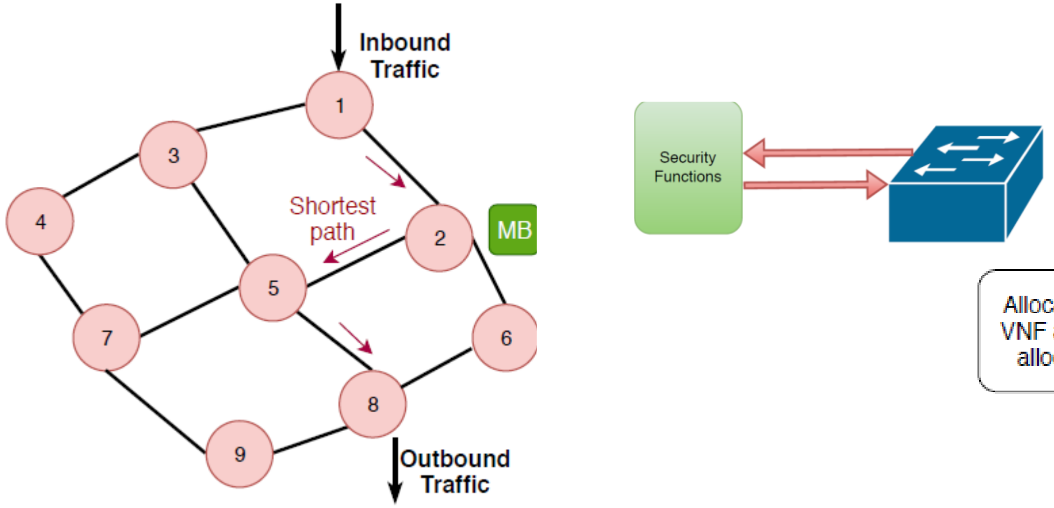
Literature Review

- Management of Softwarized middleboxes
- VNF placement
- Issues and limitation for security function in Multitenant infrastructure
  - Traffic direction (North-South , East-West)
  - Traffic constraints (Stateless ,Stateful)
  - Duplicating security functions
  - Shared security

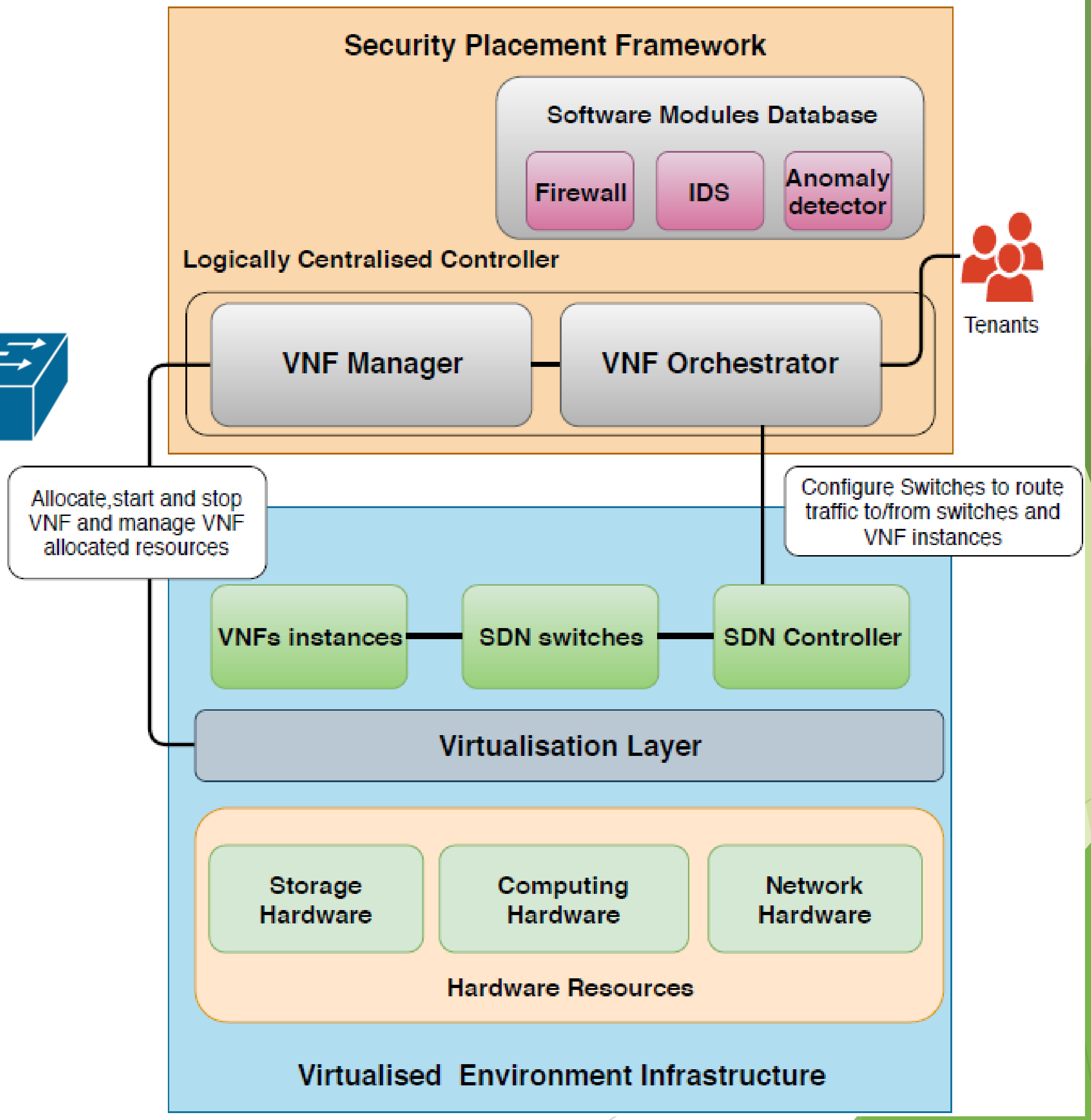


Resource-Aware Security Placement Framework

- On-path deployment
  - Deployment locations is collected with the network switches



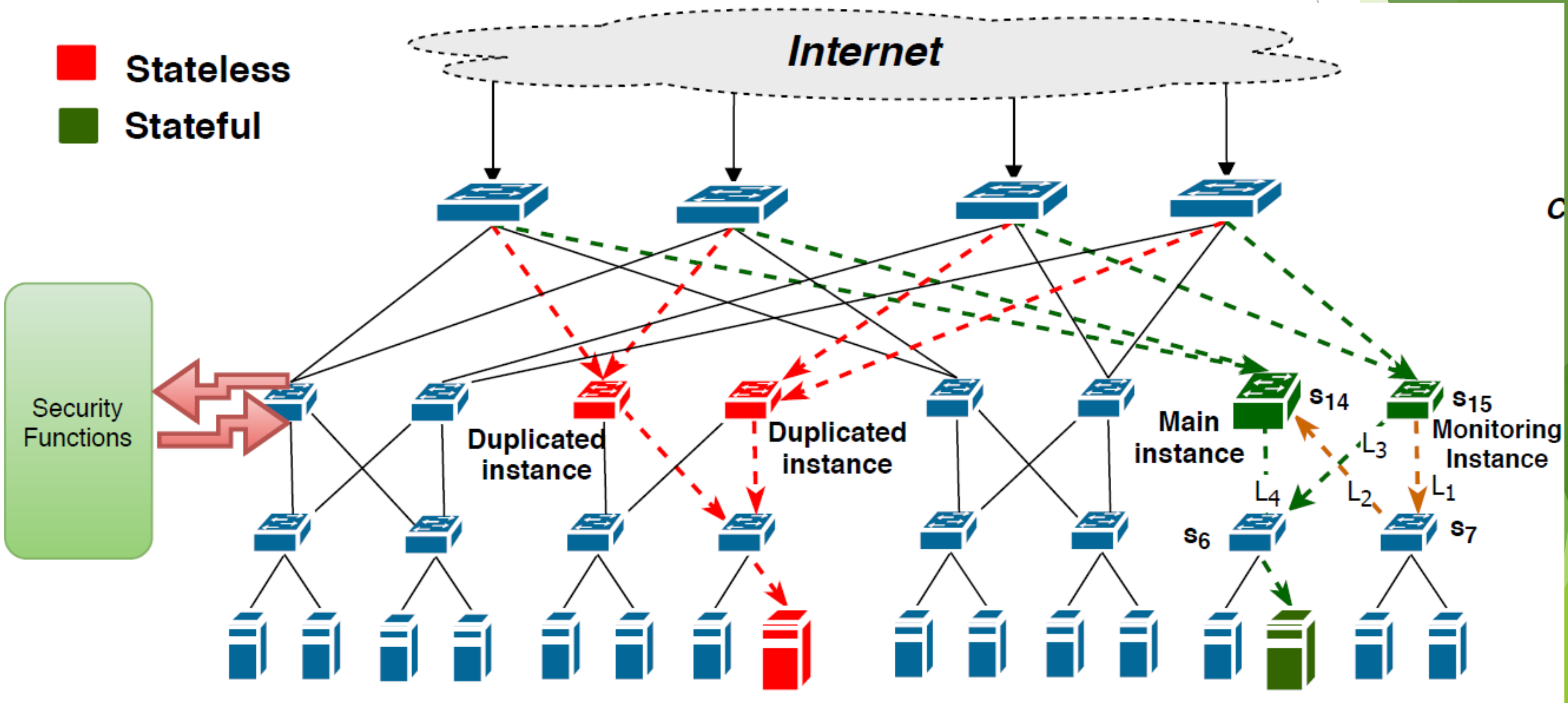
- North-South traffic
- Non sharing strategy
- Elastic security provisioning
- Service-based model
- Resource-aware placement



Security Placement Framework

Resources-Aware Placement implementation of Fat-Tree architecture

- Target efficient management of resources , minimums overhead and consider ECMP
  - Independed duplication (stateless class)
  - Depended duplication (stateful class)
  - Single instance
  - Ingress control
- Constraints
  - Traffic
  - Resources
  - Security
- Two Models
  - LP
  - CP



Fat-tree k=4 data center

Mathematical Models

CP Model		LP Model
$\max.\left[\sum_{\forall s\in S} s.c - \sum_{\forall r\in Q}\sum_{\forall p\in P}\sum_{\forall s\in S} \mathrm{x}_{r,p}\cdot u_{p,r,s}\right]$ $\min.\left[\sum_{\forall r\in Q}\sum_{\forall p\in P}\sum_{\forall l\in L} \mathrm{x}_{r,p}\cdot v_{p,r,l}\cdot l.w\right]$	Max Residual Resources RS Min Communication Overhead CO	$\max.\left[\sum_{\forall s\in S} s.c - \sum_{\forall r\in Q}\sum_{\forall p\in P}\sum_{\forall s\in S} \mathrm{x}_{r,p}\cdot u_{p,r,s}\right]$
s.t. $\sum_{\forall r\in Q}\sum_{\forall p\in P} \mathrm{x}_{r,p}\cdot u_{p,r,s} \le s.c \quad \forall s\in S$	Switches Capacity	s.t. $\sum_{\forall r\in Q} \mathrm{x}_{r,p}\cdot u_{p,r} \le p.c \quad \forall p\in P$
$\sum_{\forall r\in Q}\sum_{\forall p\in P} \mathrm{x}_{r,p}\cdot v_{p,r,l} \le l.b \quad \forall l\in L$	Links Capacity	
$\mathrm{x}_{r,p}=0, \quad \forall r\in Q, \forall p\in P \qquad if\ w_{p,r}=0$	Location Validity	$\mathrm{x}_{r,p}=0, \quad \forall r\in Q, \forall p\in P \qquad if\ w_{p,r}=0$
$\sum_{\forall p\in P} \mathrm{x}_{r,p}=1, \quad \forall r\in Q$	One allocation	$\sum_{\forall p\in P} \mathrm{x}_{r,p}=1, \quad \forall r\in Q$

Solutions

- Heuristic (BFD, FFD and RANDOM)
- Metaheuristic (Tabu search)
- Near-optimal (Subset-Sum knapsack)
- Optimal CP
- Optimal LP
- Legacy single-instance strategy

Algorithm 2 BFD Placement for Fat-tree

Input: Set of requests Q, set of locations P

Output: Set of requests allocated to locations A

$A \leftarrow \emptyset$ ▷ initialisation
$Q^* \leftarrow Sort(Q)$ ▷ sort request w.r.t. resources
for all $r \in Q^*$ **do**
  for all $level \in levels\_list$ **do**
    $P' \leftarrow GetLocations(level)$
    $P* \leftarrow Sort(P')$ ▷ sort locations w.r.t. available resources
    for all $p \in P*$ **do**
      if $(capacity(A, r, p)=\texttt{TRUE}) \bigwedge (validation(r, p)=\texttt{TRUE})$ **then**
        $p^* = p$
        break
    if $(p^* \neq 0)$ **then**
      $A \leftarrow A \cup \{(r, p^*)\}$ ▷ allocate request r to location p^*
      break
return Set of allocated requests A

BFD for fat-tree

Evaluation

- On simulated Network (Python + Cplex)
- RS and CO performance metrics
  - Different workloads (Modules sizes ,Traffic demand)
- Optimality Gap and execution time.
- Class type Distribution
- Scalability

1. Heuristic (BFD, FFD and RANDOM)
2. Metaheuristic (Tabu search)
3. Near-optimal (Subset-Sum knapsack)
4. Optimal CP
5. Optimal LP
6. Legacy single-instance strategy

Optimality GAP

μ	Heuristic			Meta-heuristic (TABU)			NEAR_OPTIMAL
	BFD	FFD	RANDOM	LOWER	SWAP	LOWER+SWAP	
0.1	0.00	5.96	5.77	0.00	0.00	0.00	0.00
0.3	0.00	2.66	2.73	0.00	0.00	0.00	0.00
0.5	0.08	1.29	1.23	0.08	0.07	0.07	0.04
0.7	0.06	0.70	0.75	0.06	0.05	0.05	0.00
0.9	0.07	0.46	0.45	0.07	0.06	0.06	0.03

Table 5.2: Optimality Gap, when k=6

Results

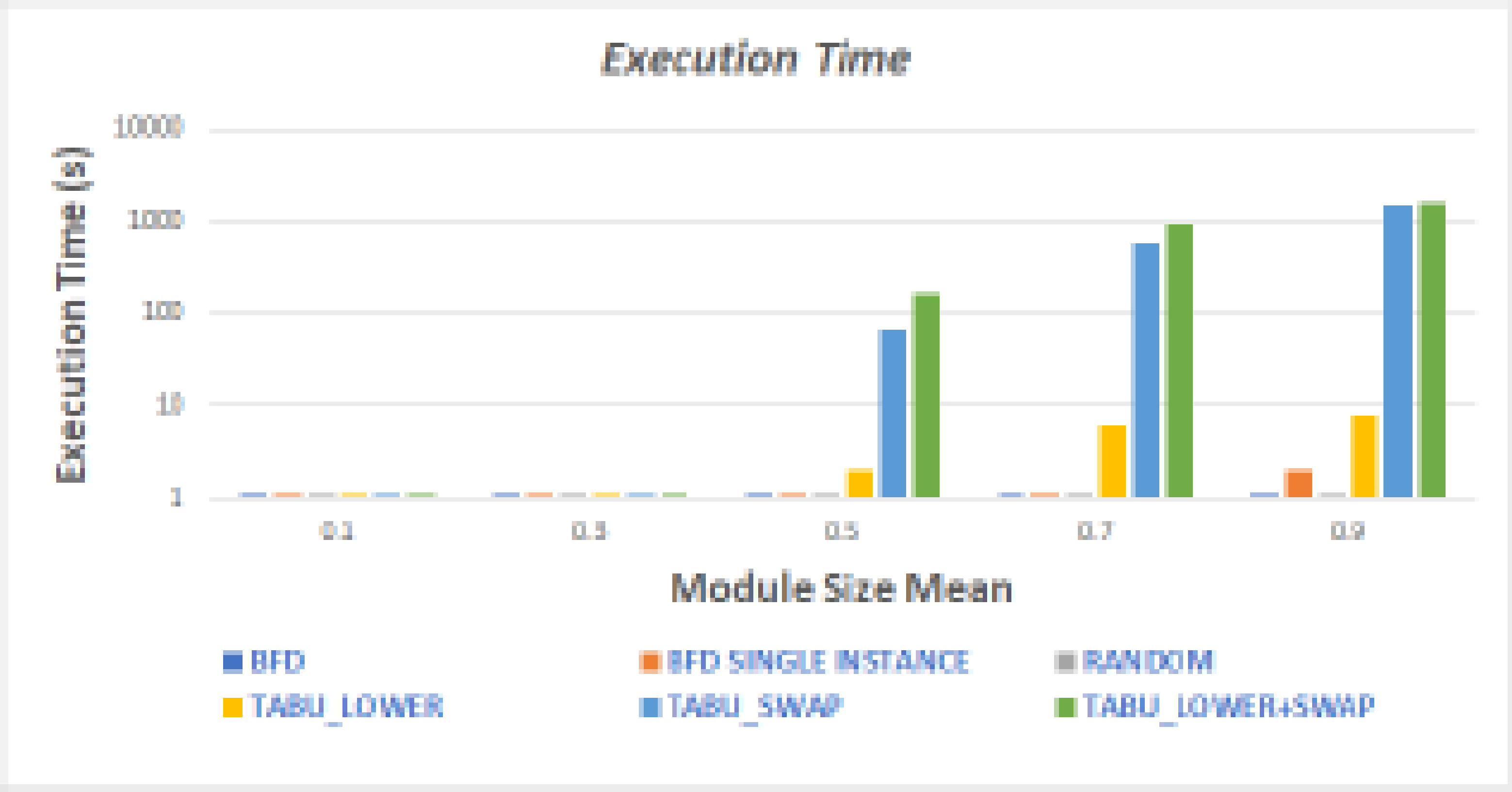


Figure 5.15: Execution Time for Modules Sizes Workload, when k=6

Final Results

- BFD algorithms has shown higher performance compared to other heuristic and meta-heuristic algorithms while balancing between utilising computing and communication resources.
- It showed less RS than Legacy single-instance strategy but less CO by .
- It showed near optimal performance compared to optimal CP and subset-sum solutions
- It showed optimised time compared to CP and subset-sum solutions and high success rate
- It showed scalability for network size and number of modules

Published Papers

1. Ali, Abeer, Christos Anagnostopoulos, and Dimitrios P. Pezaros. ”On the Optimality of Virtualized Security Function Placement in Multi-Tenant Data Centers.” In 2018 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE, 2018.
2. Ali, Abeer, Christos Anagnostopoulos, and Dimitrios P. Pezaros. ”Resource-aware placement of softwarised security services in cloud data centers.” In 2017 13th International Conference on Network and Service Management (CNSM), pp. 1-5. IEEE, 2017.
3. Ali, Abeer, Richard Cziva, Simon Jouet, and Dimitrios P. Pezaros. ”SDNFV-based DDoS detection and remediation in multi-tenant, virtualised infrastructures.” In Guide to Security in SDN and NFV, pp. 171-196. Springer, Cham, 2017.
4. *Ali, Abeer, Richard Cziva, Simon Jouet, and Dimitrios P. Pezaros. ” In-Network Placement of Security VNFs in Multi-Tenant Data Centers” In 2020 IEEE Symposium on Computers and Communications (ISCC).*

Future Work

- Supporting Placement of Security VNF Chains
- Dynamic Placement
- Exploring Real Data Center Architectures
- QoS Constraints

Thanks you

Questions