# Orchestration of VNF Placement and Routing Using SDN in 5G Networks

By: Gelareh Hasel Mehri
Supervisor: Dr. Vahid Shah-Mansouri
September 2019

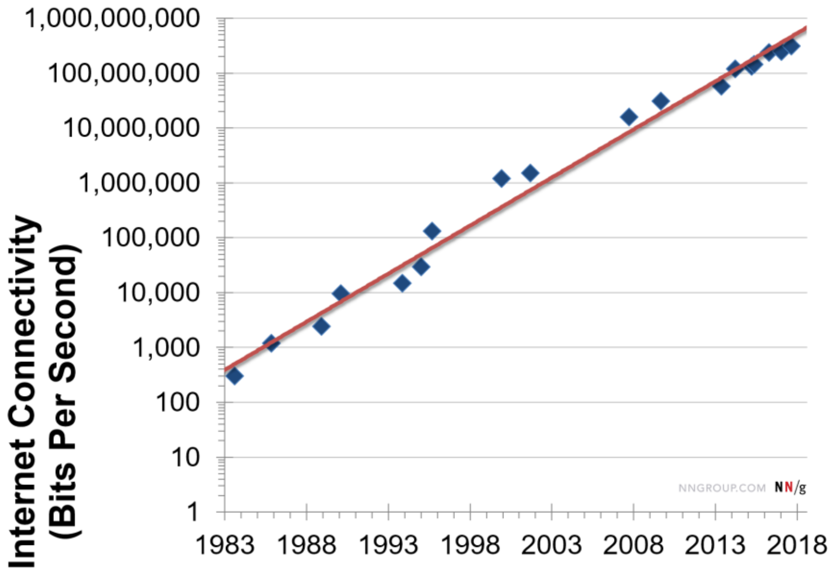
# Index

# Introduction

- Introduction
- Contributing Concepts
- Definitions
- System Model
- Simulations
- Conclusion & Future Works

Nielsen's Law of Internet Bandwidth:

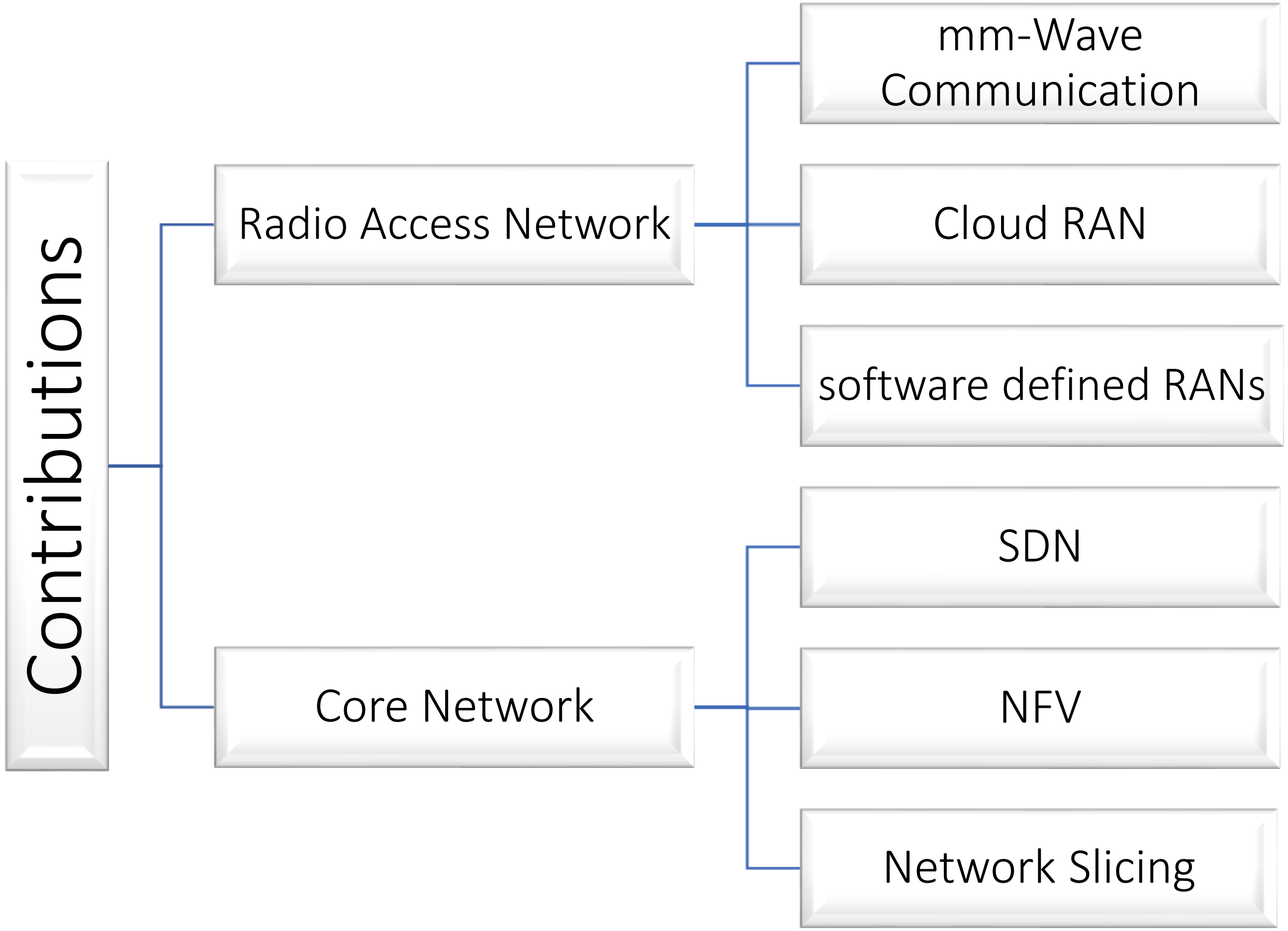
- A high-end user's connection speed grows by 50% per year




Requirements:

- Maximum reusability of network infrastructure capabilities
- flexibility and adaptation to diverse and changing requirements of applications
- efficient but open integration between application

# 5G Technologies



# Software Defined Networking

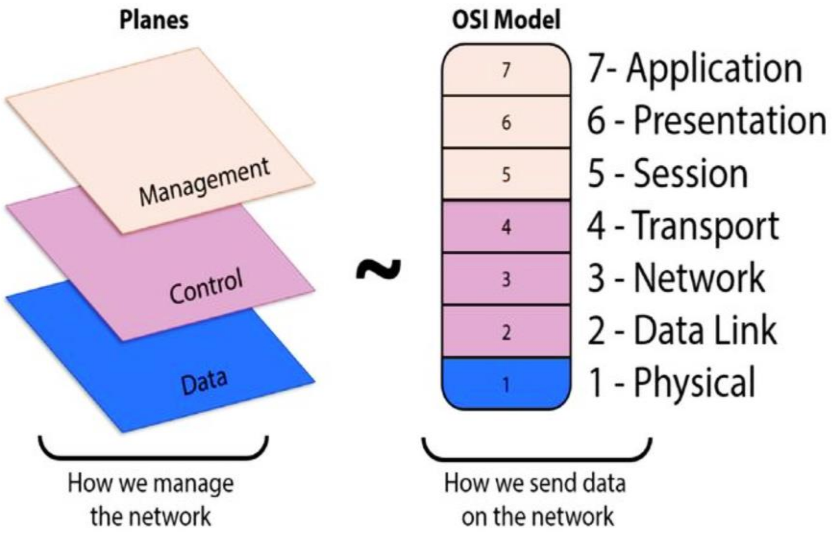
## Definition

- Centralizes command and control in the network
- Delegates the network flow control decision making to a device with network omniscience
- Separates the control plane from the data plane

## Features

- Robust network security
- Lower operating costs

## Architecture



# Network Function Virtualization



## Definition

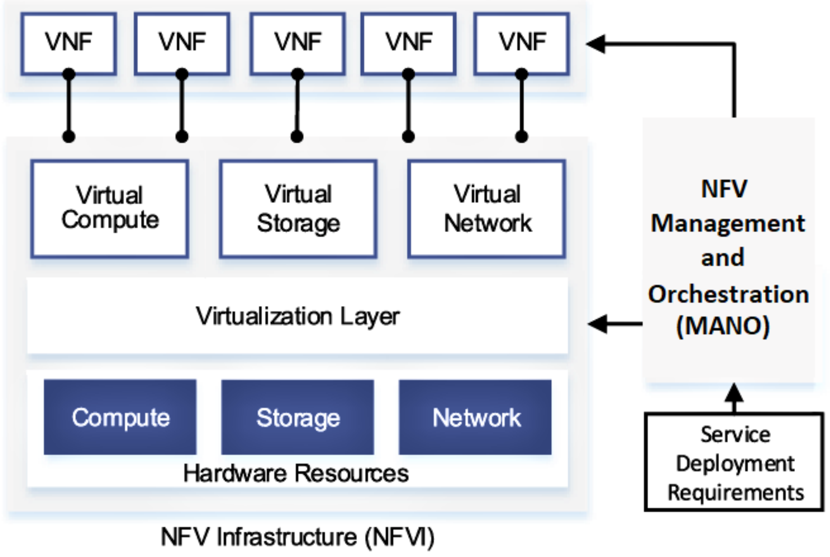
- Decoupling functions like a firewall or encryption from dedicated hardware and moving them to virtual servers
- Covering a wide range of network applications including evolved packet core, network slicing, network monitoring, internet of things and 5G

## Features

- Reduce costs in purchasing network equipment via migration to software on standard servers
- Efficiencies in space, power, and cooling
- Faster time to deployment
- Flexibility – elastic scale up and scale down of capacity
- Access to a broad independent software community, including open source

## Architecture

1. NFVI
2. VNFs
3. MANO



# SDN vs. NFV



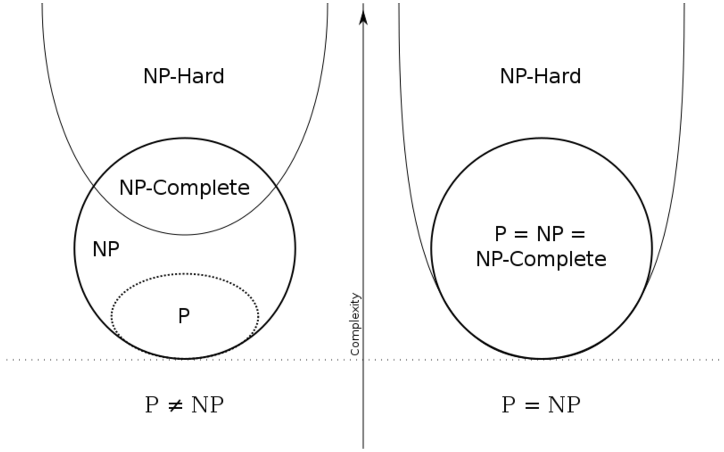
| NFV | SDN |
|---|---|
| A **concept** of implementing network functions in software manner | A **concept** of achieving centrally controlled and programmable network architecture to provide better connectivity |
| **Aim**s at reducing CapEx, OpEx, and space and power consumption | **Aim**s at providing network abstractions to enable flexible network control, configuration and fast innovation |
| **Decouple**s the network functions from the proprietary hardware to achieve agile provisioning and deployment | **Decouple**s the network control plane from the data plane forwarding to provide a centralized controller via enabling programmability |

# Heuristic Algorithms



- Optimization problems can be categorized according to the computational effort required to solve them.

- Euler diagram for P, NP, NP-complete and NP-hard set of problems:




- The consequences of NP-hardness of a problem, makes heuristic methods the only acceptable option for the variety of complex optimization problems that must be frequently solved in real-world applications.

# Definitions

- **VNF**: Virtualized network functions running on open computing platforms formerly carried out by proprietary, dedicated hardware technology


IMS VNF | Firewall VNF | Router VNF

- **Service Chain:** An E2E path, showing the traversing VNFs in order to expose a service

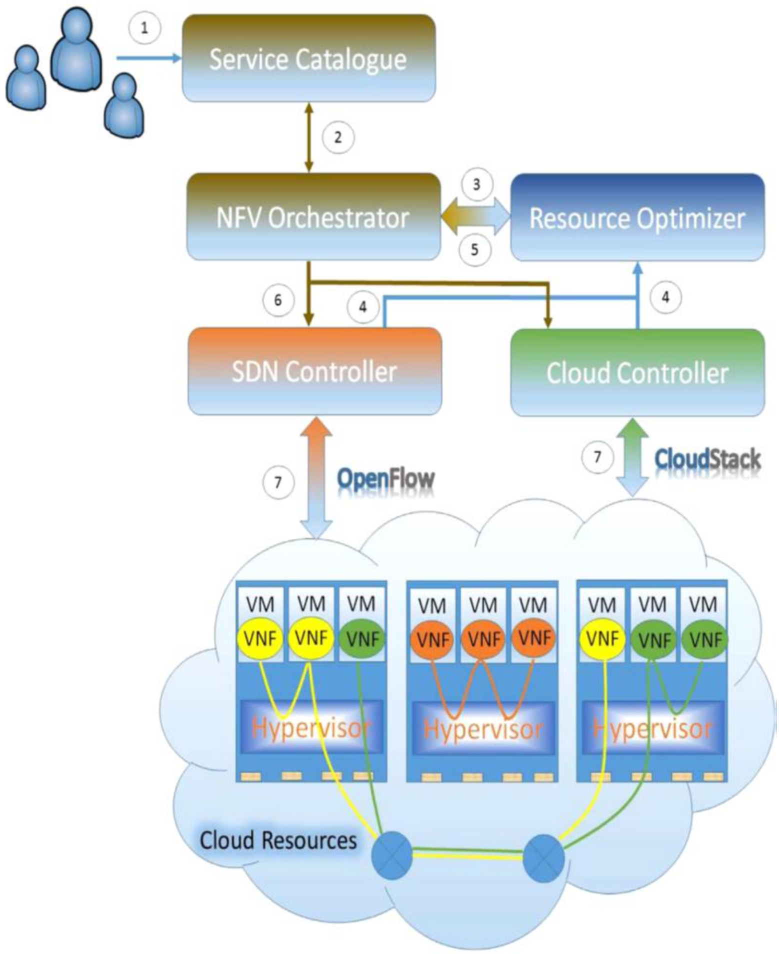
PDN Access Service Chain: Source → CPE → BRAS → Firewall → NAT → Destination

- **Placement:** Choosing the set of optimal locations for a chain of VNFs according to the service request and the current characteristics of available computing resources and network links
- **Routing:** Delivering the packets by choosing an optimal path from one host to another

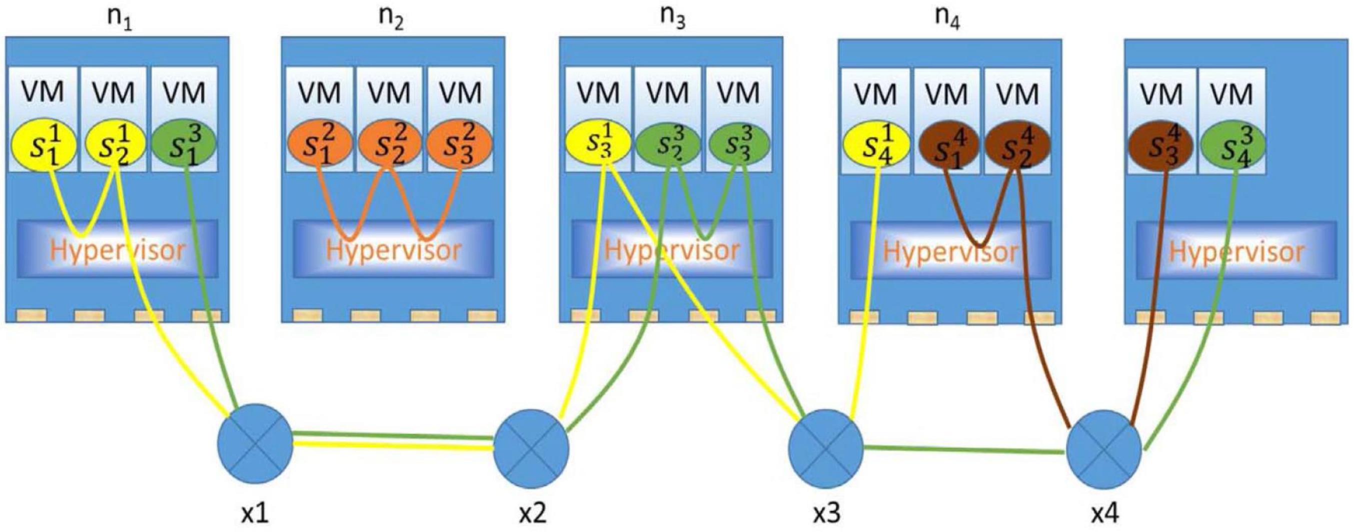
# System Model



- Service Catalogue
  - Offering a large range of NFs to users to form the service chain
- NFV Orchestrator
  - The main core of the NFV architecture, responsible for the management of the VNFs
- Resource optimizer
  - Providing optimal placement of the VNFs in the service chain
- SDN controller
  - A centralized network monitor, providing the status of the networking resources to the resource optimizer
- Cloud Controller
  - Configuration of the computational resources of the cloud infrastructure



# Routing and Placement




| Server | C=[cpu, mem, sto] |
|---|---|
| $n_1$ | [6, 16 GB, 1 TB] |
| $n_2$ | [4, 8 GB, 1 TB] |
| $n_3$ | [6, 16 GB, 1 TB] |
| $n_4$ | [4, 8 GB, 1 TB] |
| $n_5$ | [4, 8 GB, 1 TB] |

| Switch | C = [VLANs, Flow entries] |
|---|---|
| $x_1$ | [2048, 384] |
| $x_2$ | [2048, 384] |
| $x_3$ | [2048, 384] |
| $x_4$ | [2048, 384] |

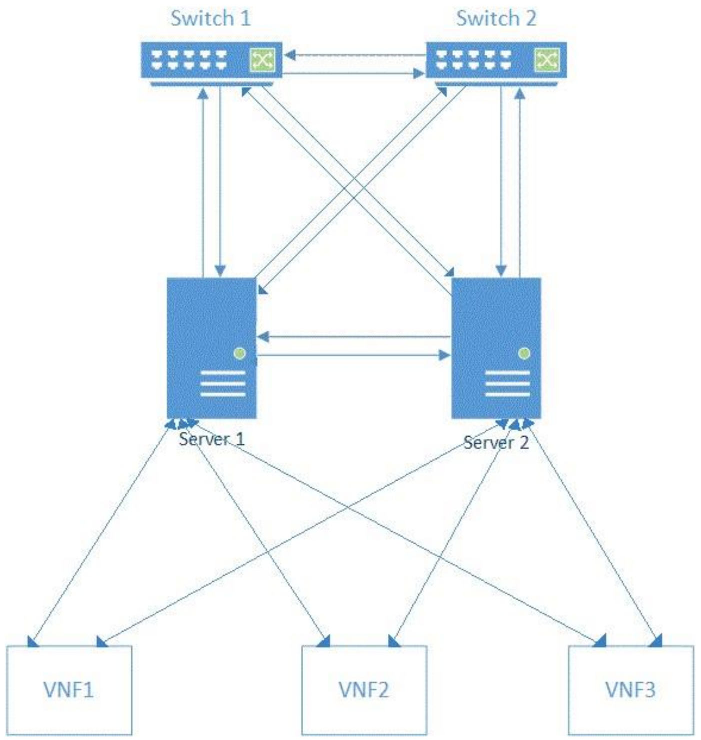
| Service Chain | VNFs | D = [cpu , mem, sto] | $bw(s_k^i, s_{k+1}^i)$ |
|---|---|---|---|
| $S^1$ | $(s_1^1, s_2^1, s_3^1, s_4^1)$ | [ 2 , 4 GB, 200 GB] | 200 MB |
| $S^2$ | $(s_1^2, s_2^2, s_3^2)$ | [ 1, 1 GB, 50 GB] | 100 MB |
| $S^3$ | $(s_1^3, s_2^3, s_3^3, s_4^3)$ | [ 2, 2 GB, 100 GB] | 200 MB |
| $S^4$ | $(s_1^4, s_2^4, s_3^4)$ | [ 1, 2GB, 100 GB] | 100 MB |

| Link | Bandwidth |
|---|---|
| $\left\{\begin{matrix}(n_1,x_1),(n_2,x_1),(n_2,x_2),(n_3,x_2),\\(n_3,x_3),(n_4,x_3),(n_4,x_4),(n_5,x_5)\end{matrix}\right\}$ | 1 Gbps |
| $\{(x_1,x_2),(x_2,x_3),(x_3,x_4)\}$ | 10 Gbps |

# Network Topology



- A service chain consisting of a certain number of VNFs, (as in the diagram, VNF1-VNF2-VNF3) is considered as the input of the network, whose VNFs need to be placed and optimally routed through.
- The number and order of VNFs for a service chain are prespecified and fixed.
- Servers and switches of the infrastructure are assumed to be interconnected.
- It is assumed that every VNF is connected with every server.
- Switches, servers and VNFs of the service chain are considered as graph's nodes in the graphical representation of the network.
- All links in the network are two-sided, creating a undirected graph for the network.
- Each server has a specified amount of CPU capacity and each link has a specified bandwidth capacity.



# Parameters

**Input Parameters:**

Network Graph $G = (V, E)$
Switches $X$
Servers $N$
$(N \cup X = V)$
Service Chain $S$
Link Bandwidth $(u, v)$
Server CPU Capacity $(n)$
Bandwidth Demand $(vnf)$
CPU Demand $(vnf)$
Server Activation Cost $R(n)$
Link Activation Cost $R(u, v)$

**Output Parameters:**

Placement Binary Variables $h_{uv}^{s_k s_{k+1}}$
Routing Variables $f_{uv}^{s_k s_{k+1}}$

# Cost Functions



## Service Provider View

| | |
|---|---|
| Minimize the summation of the involved activation costs | $min \sum_{n \in N} R_n + \sum_{(u,v) \in E} R_{(u,v)}$ |

## + Cloud Provider View

| | |
|---|---|
| Allowing the cloud provider to efficiently utilize the available resources (Introducing a load balancing scheme) | $min \sum_{n \in N} A_n R_n + \sum_{(u,v) \in E} B_{(u,v)} R_{(u,v)}$ <br> $(A_n = \frac{1}{C(n)}, B_{(u,v)} = \frac{1}{bw(u,v)})$ |

# Constraints

Resource Constraints:

- CPU resource limitation

$$\sum_{s_p \in S} D(s_p).\, h_{s_p n}^{s_k s_{k+1}} \leq C(n) \quad \forall n \in N, s_k, s_{k+1} \in S$$

- Link bandwidth limitation

$$\sum_{k=0 \ldots K-1} (f_{uv}^{s_k s_{k+1}} + f_{vu}^{s_k s_{k+1}}) \leq bw(u,v) \quad \forall (u,v) \in E'$$

## Constraints



Flow conservation constraints:

- Middle nodes

$$\sum_{\omega \in V'} f_{u\omega}^{s_k s_{k+1}} - \sum_{\omega \in V'} f_{\omega u}^{s_k s_{k+1}} = 0 \quad \forall k \in (0 \ldots K-1), \qquad \forall \omega \in V' \backslash s_k, s_{k+1}$$

- Destination nodes

$$\sum_{\omega \in V'} f_{s_k \omega}^{s_k s_{k+1}} - \sum_{\omega \in V'} f_{\omega s_k}^{s_k s_{k+1}} = bw(s_k, s_{k+1}) \quad \forall k \in (0 \ldots k-1)$$

- Source nodes

$$\sum_{\omega \in V'} f_{s_{k+1} \omega}^{s_k s_{k+1}} - \sum_{\omega \in V'} f_{\omega s_{k+1}}^{s_k s_{k+1}} = -bw(s_k, s_{k+1}) \quad \forall k \in (0 \ldots k-1)$$

# Constraints



- Ensuring that the flows are not split-able

$$f_{uv}^{s_k s_{k+1}} + f_{uv}^{s_k s_{k+1}} = bw(s_k, s_{k+1}).\, h_{uv}^{s_k s_{k+1}} \qquad \forall k \in (0 \ldots K-1), (u, v) \in E'$$

- Every VNF is allocated on a single server

$$\sum_{n \in N} h_{s_p n}^{s_k s_{k+1}} = 1 \qquad \forall s_p \in S, k \in (0 \ldots K-1)$$

- The placement of the VNFs composes a connected graph

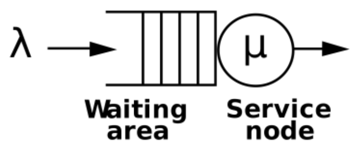
$$h_{uv}^{s_k s_{k+1}} = h_{uv}^{s_{k+1} s_{k+2}} \qquad k \in (0 \ldots K-2), \qquad (u, v) \in E'$$

# Constraints



- Congestion of a link in an M/M/1 queue:

$$\frac{\lambda}{\mu-\lambda},$$




- Congestion of each link in our problem:

$$\frac{\sum_k (f_{uv}^{s_k s_{k+1}} + f_{vu}^{s_k s_{k+1}})}{bw(u,v) - \sum_k (f_{uv}^{s_k s_{k+1}} + f_{vu}^{s_k s_{k+1}})}$$

- Upper bound of tolerated congestion of each link:

$$Buffer\ size$$

- **Congestion constraint:**

$$\sum_{k=0\ldots K-1} (f_{uv}^{s_k s_{k+1}} + f_{vu}^{s_k s_{k+1}}) \leq \frac{bfr(u,v)}{1+bfr(u,v)} bw(u,v)$$

# Problem Restatement



minimize $\sum_{k=0\ldots K-1} \sum_{n \in N} \sum_{s_p \in S} A_n . R_n . h_{s_p n}^{s_k s_{k+1}} + \sum_{k=0\ldots K-1} \sum_{(u,v) \in E} B_{(u,v)} . R_{(u,v)} . h_{uv}^{s_k, s_{k+1}}$

subject to:

$$\sum_{s_p \in S} D(s_p) . h_{s_p n}^{s_k s_{k+1}} \leq C(n) \quad \forall n \in N, s_k, s_{k+1} \in S$$

$$\sum_{k=0\ldots K-1} (f_{uv}^{s_k s_{k+1}} + f_{vu}^{s_k s_{k+1}}) \leq bw(u,v) \quad \forall (u,v) \in E'$$

$$\sum_{\omega \in V'} f_{u\omega}^{s_k s_{k+1}} - \sum_{\omega \in V'} f_{\omega u}^{s_k s_{k+1}} = 0 \quad \forall k \in (0 \ldots K-1), \forall \omega \in V' \backslash s_k, s_{k+1}$$

$$\sum_{\omega \in V'} f_{s_k \omega}^{s_k s_{k+1}} - \sum_{\omega \in V'} f_{\omega s_k}^{s_k s_{k+1}} = bw(s_k, s_{k+1}) \quad \forall k \in (0 \ldots k-1)$$

$$\sum_{\omega \in V'} f_{s_{k+1} \omega}^{s_k s_{k+1}} - \sum_{\omega \in V'} f_{\omega s_{k+1}}^{s_k s_{k+1}} = -bw(s_k, s_{k+1}) \quad \forall k \in (0 \ldots k-1)$$

$$f_{uv}^{s_k s_{k+1}} + f_{uv}^{s_k s_{k+1}} = bw(s_k, s_{k+1}) . h_{uv}^{s_k s_{k+1}} \quad \forall k \in (0 \ldots K-1), (u,v) \in E'$$

$$\sum_{n \in N} h_{s_p n}^{s_k s_{k+1}} = 1 \quad \forall s_p \in S, k \in (0 \ldots K-1)$$

$$h_{uv}^{s_k s_{k+1}} = h_{uv}^{s_{k+1} s_{k+2}} \quad k \in (0 \ldots K-2), (u,v) \in E'$$

$$\sum_{k=0\ldots K-1} (f_{uv}^{s_k s_{k+1}} + f_{vu}^{s_k s_{k+1}}) \leq \frac{bfr(u,v)}{1 + bfr(u,v)} bw(u,v) \quad \forall (u,v) \in E'$$

# Optimization Considerations



- Simulations are executed in a computer running win10 and CORE i7-8th Gen intel processor.
- The proposed Mixed Integer Linear Problem (MILP) is simulated for a small scale network.
- The MOSEK toolbox for MATLAB is used to solve the MILP.
- Due to the NP-hardness of the problem, the placement sub-problem is mapped to a specific type of well known optimization problems, called variable sized bin packing problem (VSBPP).
- Two specific heuristic algorithms are then presented and simulated: Adaptive Best Fit Decreasing (A-BFD) and Subset Sum Problem based (SSP-based) heuristics
- Besides MILP simulations, one of these two algorithms is simulated, too: A-BFD
- To deal with the routing NP-hard sub-problem, again, the problem is mapped to a well known optimization problem, called Shortest Path Problem (SPP).
- The Dijkstra algorithm is then used for the routing part of the problem.

# Adaptive-BFD Heuristic Algorithm



**Algorithm** Adapted BFD

**Input** $S$ : Set of VNFs to be accommodated into the servers
**Input** $N$ : Set of Servers available to load the VNFs
**Input** $G$ : Topology graph of network

Sort the VNFs in $S$ according to non-increasing order of their volumes
Sort the servers in $N$ according to non-decreasing order of ratio $A_n R_n / C_n$
$\bar{S}$ : Set of unpacked VNFs
$K$ : Set of selected servers
$\bar{S}$ = sorted $S$
$K = \{\emptyset\}$

**while** $\bar{S} \neq \emptyset$ **do**
  **if** the first VNF in $\bar{S}$, naming $s$, can be accommodated into a server in $K$ **then**
    Accommodate $s$ into the best server of the set $K$ , naming $n$
  **else**
    Accommodate $s$ into $n'$, where $n'$ is the first server in the ordered list $N \backslash K$
    $K := K \cup n'$
    $\bar{S} := \bar{S} \backslash s$
  **end if**
**end while**

# Adaptive-BFD Heuristic Algorithm



**for all** $k \in K$ **do**
  **for all** $m \in N \backslash K$ **do**
    $U_k = \sum_{s \text{ loaded in } k} D_s$
    **if** $C_m > U_k$ and $A_m R_m < A_k R_k$ **then**
      Move all the items from $k$ to $m$
      $K = K \backslash k \cup m$
    **end if**
  **end for**
**end for**
**if** $\bar{S} = \emptyset$ **then**
  **for all** $s_k \in S$ **do**
    **if** selected servers of $s_k$ and $s_{k+1}$ are different **then**
      $dijkstra(G, s_k, s_{k+1})$
    **end if**
  **end for**
**end if**

# SSP-Based Heuristic Algorithm



**Algorithm** SSP Based

**Input** $S$ : Set of VNFs to be accommodated into the servers
**Input** $N$ : Set of available servers
**Input** $G$ : Topology graph of network

Sort the servers in $N$ according to non-increasing order of their capacities
$\bar{S}$ : Set of unpacked VNFs
Set $\bar{S} = S$

**while** $\bar{S} \neq \emptyset$ **do**
  Solve a SSP to load a subset of $\bar{S}$ into the first server of the set $N$, naming $n$
  Define $S_n$ as the set of selected VNFs by solving SSP
  $N := N \backslash n$
  $\bar{S} := \bar{S} \backslash S_n$
**end while**
**if** $\bar{S} = \emptyset$ **then**
  **for all** $s_k \in S$ **do**
    **if** selected servers of $s_k$ and $s_{k+1}$ are different **then**
      $dijkstra(G, s_k, s_{k+1})$
    **end if**
  **end for**
**end if**

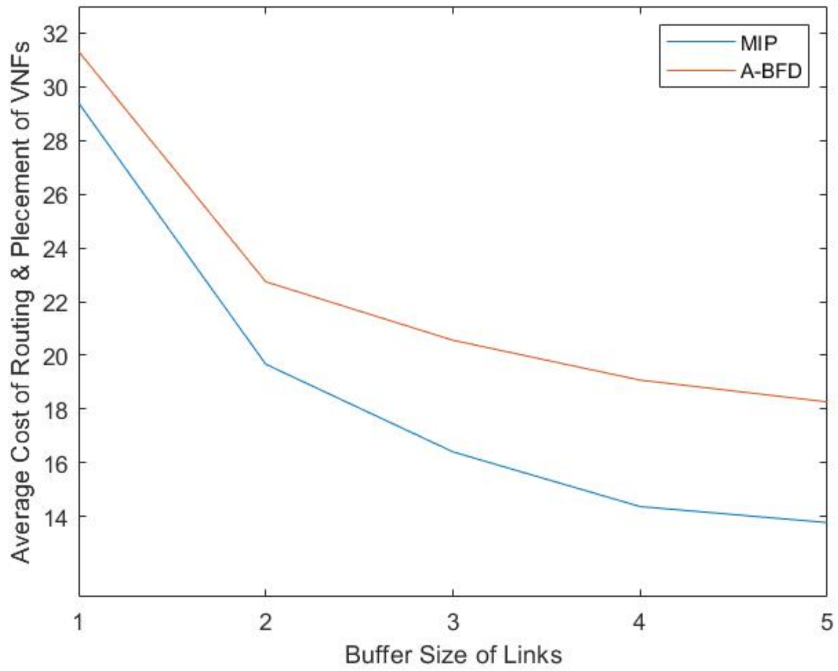
# Dijkstra Heuristic Algorithm

**Algorithm** Dijkstra

**Input** $G$ : Graph of the network
**Input** $S$ : Source node
$Q$ : the set of all nodes in $G$
**for each** vertex v in Graph: **do**
  dist[v] := infinity
  previous[v] := undefined
**end for**
dist[S] := 0
**while** $Q$ **is not** empty: **do**
  u := node in $Q$ with smallest dist[]
  remove u from $Q$
  **for each** neighbor v of u **do**
    alt := dist[u] + dist-between(u,v)
    **if** alt $<$ dist[v] **then**
      dist[v] := alt
      previous[v] := u
    **end if**
  **end for**
**end while**
**return** previous[]

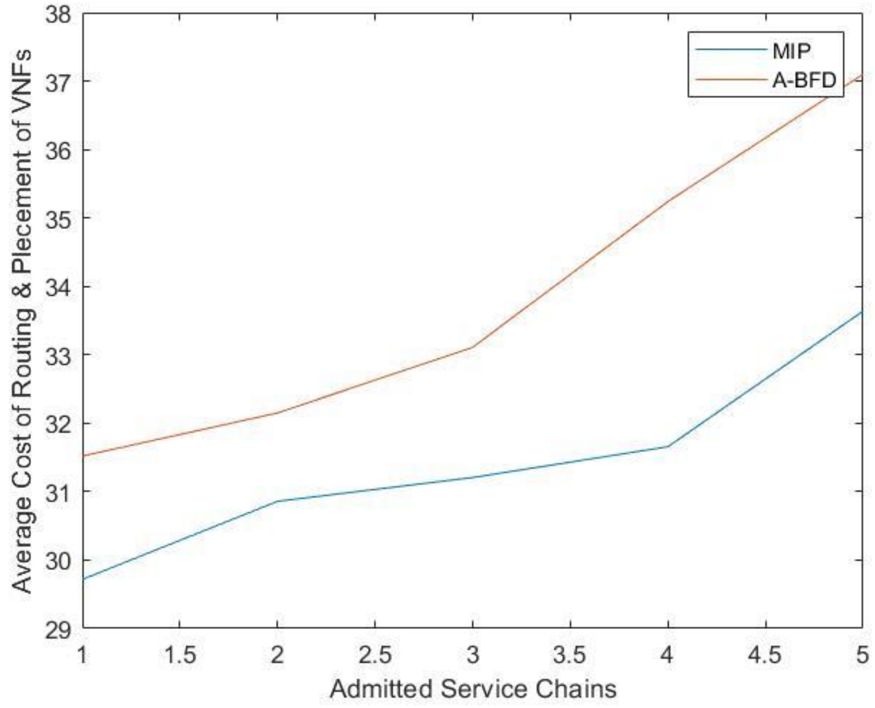
# Simulations

- The effects of changing the links' buffers sizes on the total cost, in both MILP and A-BFD scenarios, are illustrated in the following diagrams.
- It can be seen that for a certain buffer size, the average cost of routing and placement derived from MIP algorithm is lower than the A-BFD heuristic algorithm.
- Buffer size of each link determines its congestion tolerability. The larger the buffer size, the less congestion experienced on the link. This results in using fewer number of links in the network and decreasing the total cost.
- P.S: The term "Average" is because of repeating the procedure over 100 iterations.



# Simulations

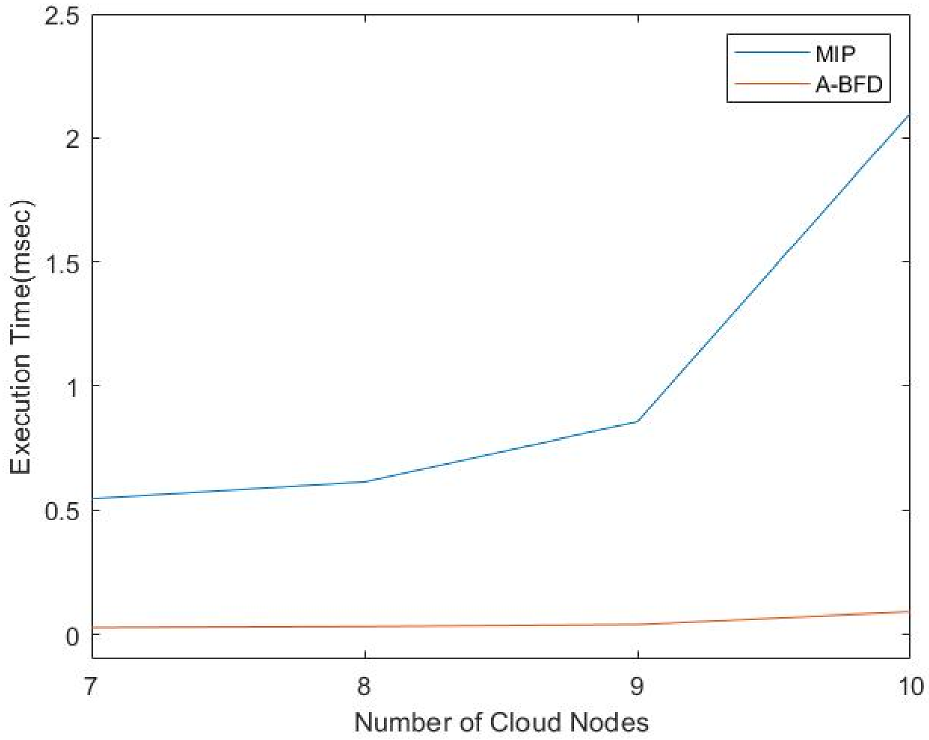
- The effects of changing the number of admitted service chains on the total cost, in both MILP and A-BFD scenarios, are illustrated in the following diagrams.
- Again, as expected, for a certain number of admitted chains to the network, the optimal algorithm can perform better, leading to a lower average cost.
- As the number of admitted service chains to the network increases, the average cost of routing and placement of VNFs increases, too. This is because of the increasing need for CPU and bandwidth in the network.



# Simulations



- The effects of changing the network size on the total execution time, in both MILP and A-BFD scenarios, are illustrated in the following diagrams.
- It can be seen that increasing the number of nodes in the network will lead to an increase in both algorithms' execution times. Specifically, this increase is obvious in MIP algorithm, as expected, leading to its unscalability.



# Conclusions



- The problem of placement and routing of a service chain in network was proposed as a **MILP**, and investigated.
- A QoS constraint, **congestion** constraint, was the additional constraint which was taken into account besides topological and computational constraints.
- The proposed problem was the simulated in MATLAB, using **MOSEK** toolbox.
- Due to the NP-hard nature of the problem, some **heuristic** algorithms were then proposed and simulated.
- The **performance** of the optimal and a heuristic algorithm was the compared, considering different criteria.

Introduction

Contributing Concepts

Definitions

System Model

Simulations

Conclusion & Future Works

# Future Works

- Considering other QoS constraints in order to achieve High-quality services

**QoS:**

Reliability

Latency

Providers' Revenue

Fairness

Energy Consumption

Quality of service

- Instead of considering one service chain, a set of chains and their arrival statistical characteristics can be taken into consideration.

# THANKS