IEEE CloudNet™

IEEE International Conference on Cloud Networking
27–29 November 2024 // Rio de Janeiro, Brazil

A Novel Method for SFC Segmentation in Edge-Cloud Environments

presenting author:

Anselmo Luiz Éden Battisti

Flávia C. Delicato

Débora Christina Muchaluat-Saade

Laboratório MídiaCom

Instituto de Computação uff

uff Universidade Federal Fluminense

Agenda

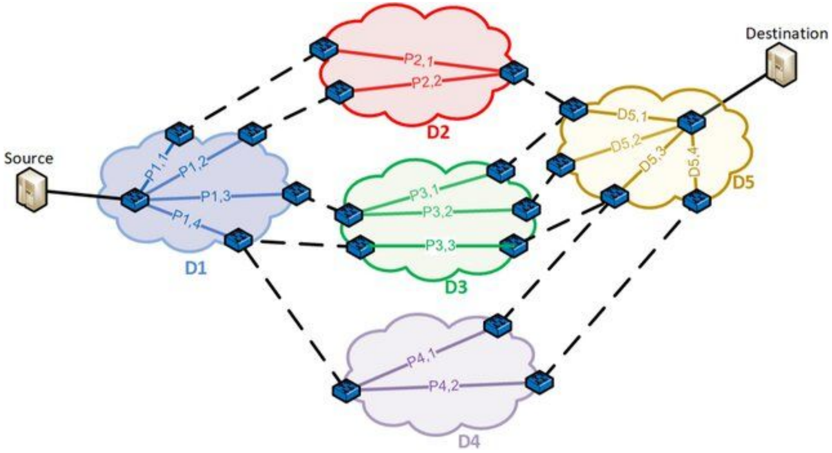
- Introduction
  - SFC Segmentation Problem
  - Contributions
- Distributed Segmentation Strategy (DSS)
- Performance Evaluation
- Final Remarks

Introduction

- **Virtual Network Functions (VNFs)** allow a software-based implementation of network functions that are traditionally executed in dedicated hardware.
- **Service Function Chains (SFCs)** are sequences of VNFs, arranged to process network traffic in a predefined order to provide complex network services.

Introduction

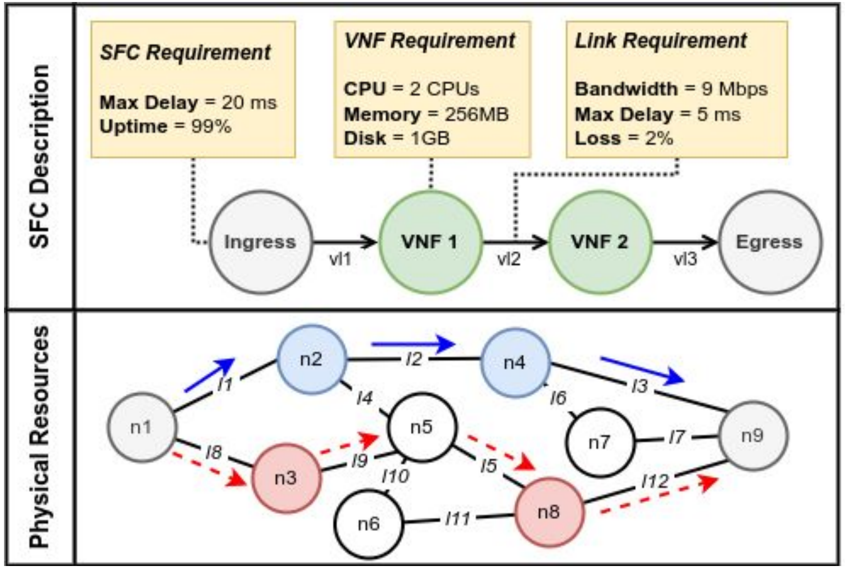
- A **domain** is a **set of** network and compute **resources**.
    - These resources can be used during the SFC execution.
- In a **multi-domain** environment, **resources** from **more than one domain** can be used during the **execution** of VNFs from an **SFC**.



Moufakir, T., Zhani, M.F., Gherbi, A. *et al.* Collaborative Multi-domain Routing in SDN Environments. *J Netw Syst Manage* 30, 23 (2022).

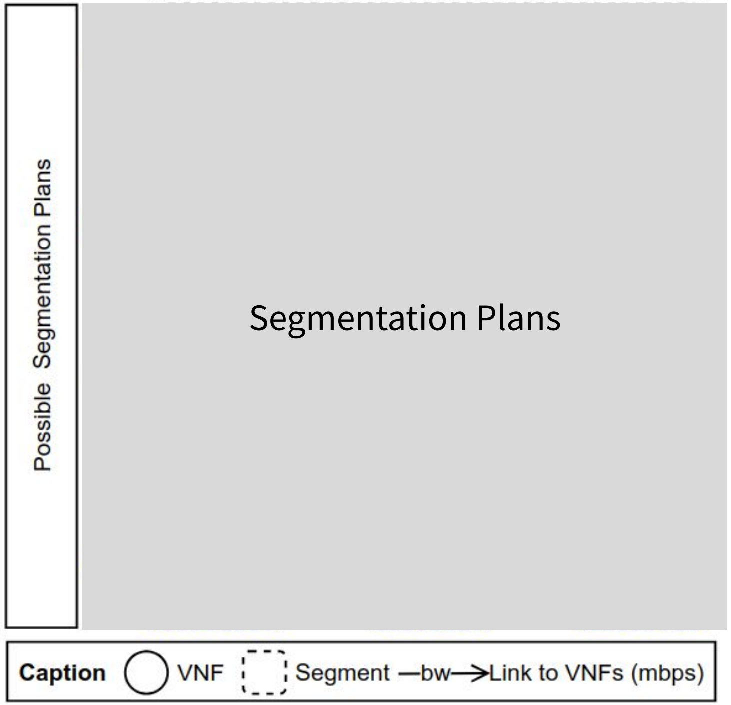
Introduction

- To **execute an SFC,** each **VNF** that makes up the chain must be **instantiated** on network **nodes** for execution.
- The process of selecting which nodes will execute each VNF is denoted **SFC placement**.



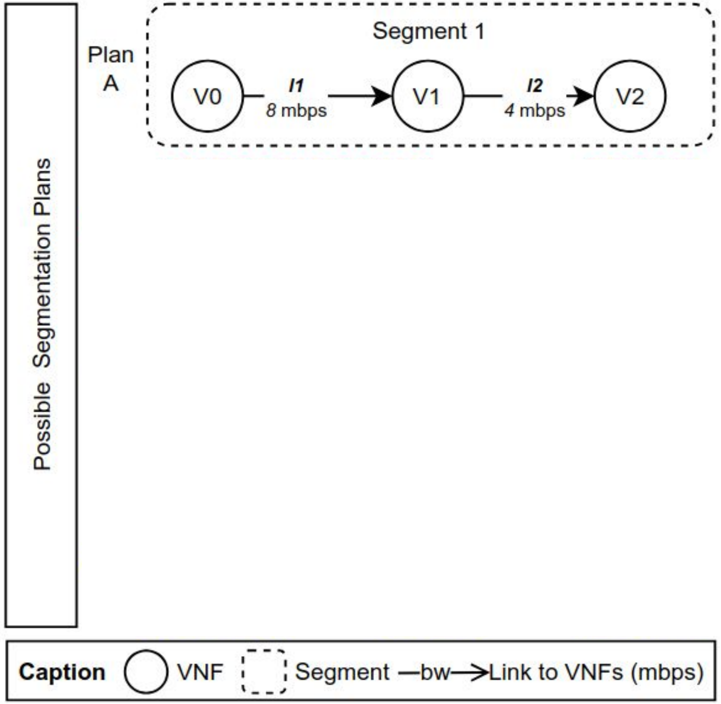
SFC Segmentation Problem

- **To execute an SFC in a multi-domain** environment, the SFC **must be segmented**.
  - Segments are **portions of an SFC** that are confined within a single domain.
- The **SFC segmentation** divides an SFC into **segments**.
  - SFC with 3 VNFs **{v0, v1, v2}**.



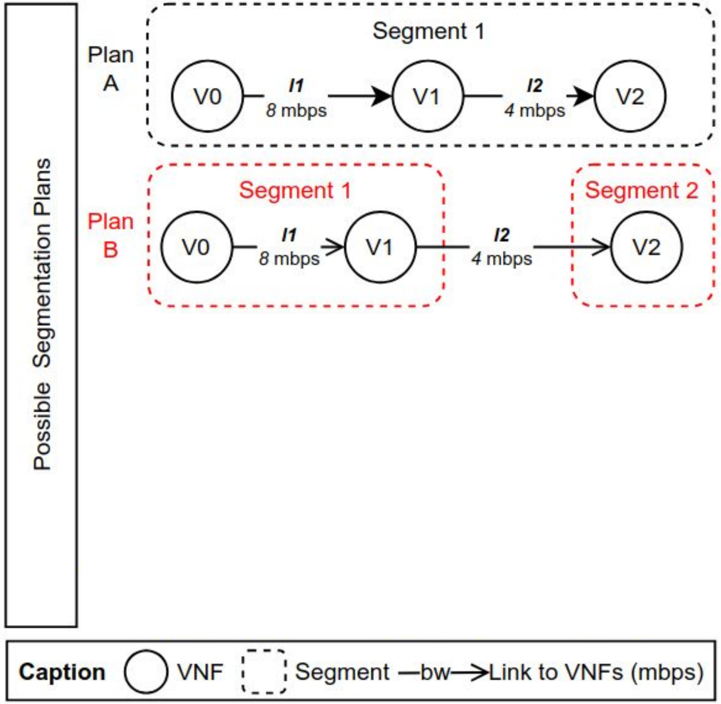
SFC Segmentation Problem

- **To execute an SFC in a multi-domain** environment, the SFC **must be segmented**.
  - Segments are **portions of an SFC** that are confined within a single domain.
- The **SFC segmentation** divides an SFC into **segments**.
  - SFC with 3 VNFs **{v0, v1, v2}**.



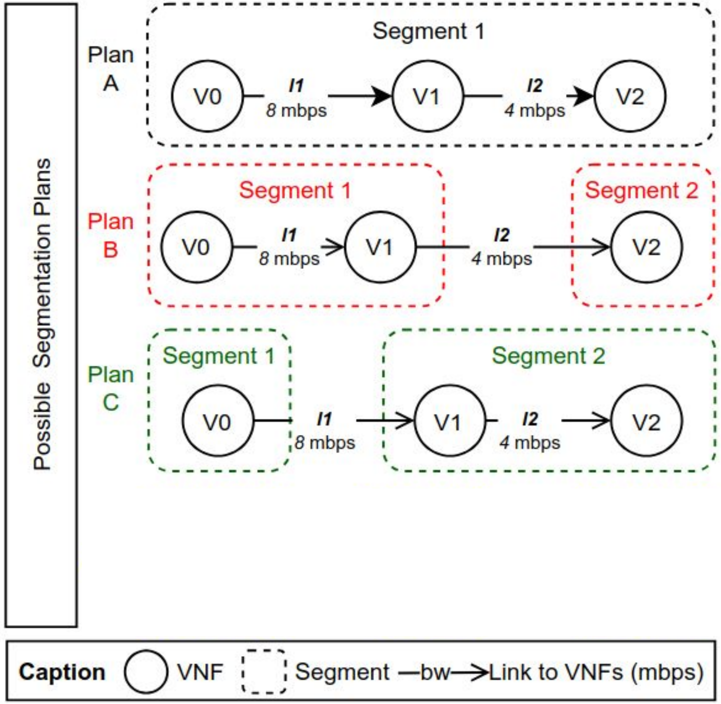
SFC Segmentation Problem

- **To execute an SFC in a multi-domain** environment, the SFC **must be segmented**.
  - Segments are **portions of an SFC** that are confined within a single domain.
- The **SFC segmentation** divides an SFC into **segments**.
  - SFC with 3 VNFs **{v0, v1, v2}**.



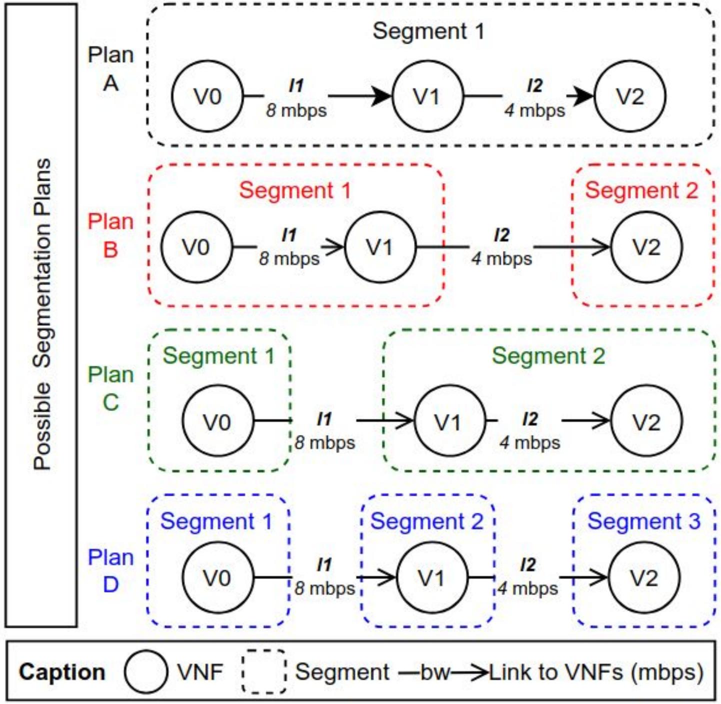
SFC Segmentation Problem

- **To execute an SFC in a multi-domain** environment, the SFC **must be segmented**.
  - Segments are **portions of an SFC** that are confined within a single domain.
- The **SFC segmentation** divides an SFC into **segments**.
  - SFC with 3 VNFs **{v0, v1, v2}**.



SFC Segmentation Problem

- **To execute an SFC in a multi-domain** environment, the SFC **must be segmented**.
  - Segments are **portions of an SFC** that are confined within a single domain.
- The **SFC segmentation** divides an SFC into **segments**.
  - SFC with 3 VNFs **{v0, v1, v2}**.



Contributions

- We propose a **new heuristic** named Dynamic Segmentation Strategy (DSS) that allows **a flexible segmentation of SFCs.**
    - Unlike traditional methods that often rely on static segmentation strategies, **DSS enables dynamic adjustments during the placement process.**

Contributions

- We developed **a new system model for the SFC segmentation problem**.
  - To the best of our knowledge, this is the *first model dedicated exclusively to this problem*.
- We adapted the combinatorial method **“stars and bars” to the SFC segmentation problem.**

DSS

- The **SFC segmentation problem** can be mapped to a **combinatorial** mathematics **problem**.
  - To identify the **number of partitions of a set**, it is possible to use the Bell number.
- However, it generates **invalid segmentation plans**.

DSS

To be a valid segmentation plan:

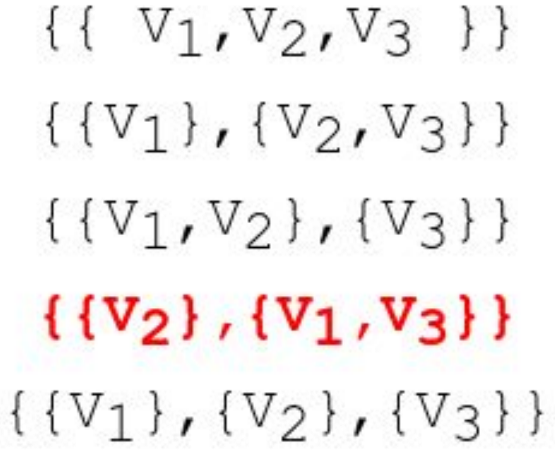
1) Each segment must contain at least one VNF.

2) Each VNF should be allocated in exactly one segment.

3) The VNF chain order in the SFC must be maintained inside each segment.

4) The order of segments must follow the VNF chain defined in the SFC.

Segmentation Plans



$\{\{V_1, V_2, V_3\}\}$

$\{\{V_1\}, \{V_2, V_3\}\}$

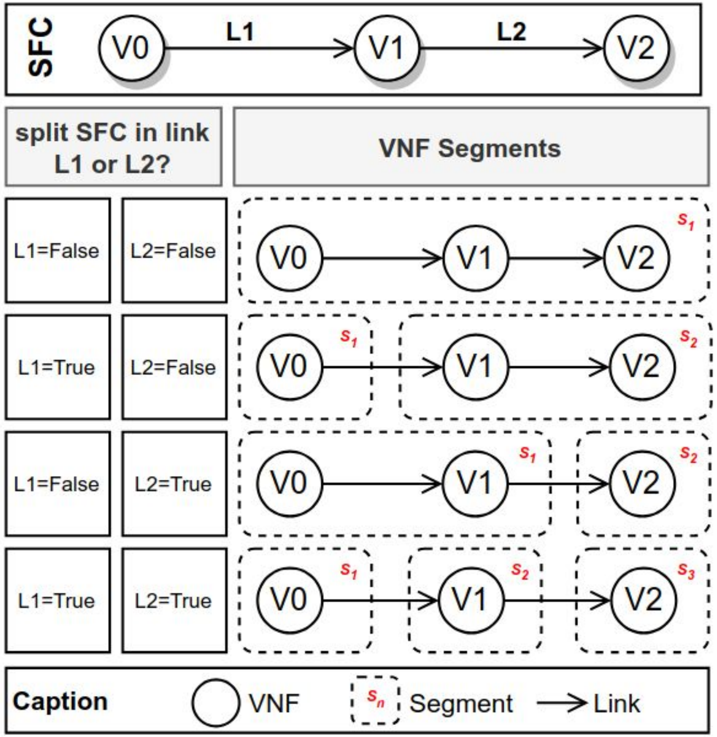
$\{\{V_1, V_2\}, \{V_3\}\}$

$\{\{V_2\}, \{V_1, V_3\}\}$

$\{\{V_1\}, \{V_2\}, \{V_3\}\}$

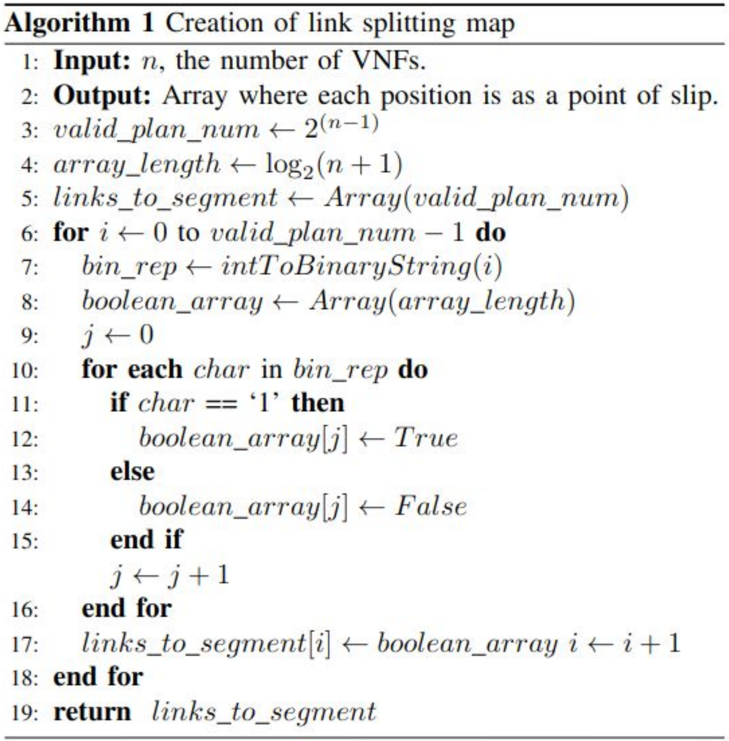
DSS

- Finding all **valid segment plans** for an SFC is related to the "**stars and bars**" theorem.
  - the stars represent the VNFs, and the bars represent the boundaries between the segments.



DSS

- Algorithm 1 generates **all possible split points** for the VNFs of an SFC.
- It creates an array where each element is a **boolean array** that indicates in each element if a **link must be used as the split point or not**.

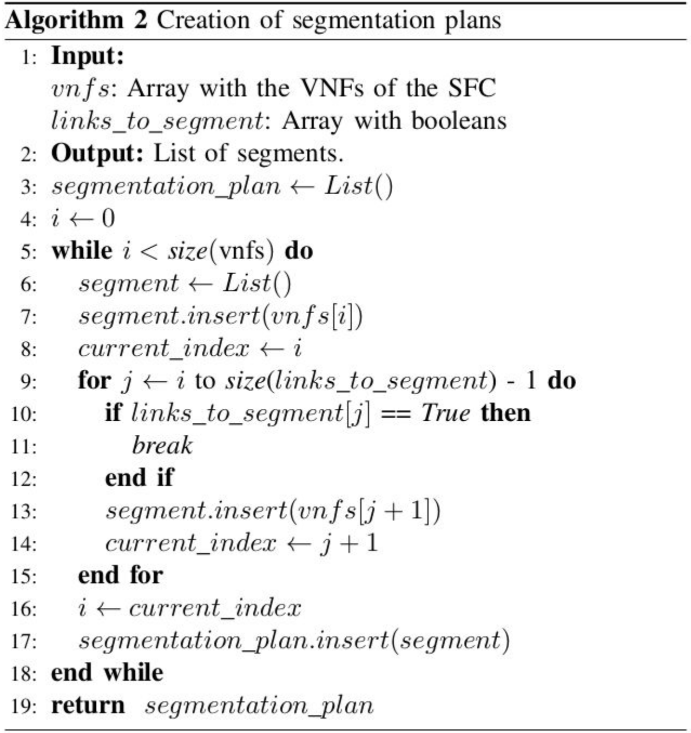


Algorithm 1 Creation of link splitting map

```
Input: $n$, the number of VNFs.
Output: Array where each position is as a point of slip.
$valid\_plan\_num \leftarrow 2^{(n-1)}$
$array\_length \leftarrow \log_2(n+1)$
$links\_to\_segment \leftarrow Array(valid\_plan\_num)$
for $i \leftarrow 0$ to $valid\_plan\_num - 1$ do
  $bin\_rep \leftarrow intToBinaryString(i)$
  $boolean\_array \leftarrow Array(array\_length)$
  $j \leftarrow 0$
  for each $char$ in $bin\_rep$ do
    if $char$ == '1' then
      $boolean\_array[j] \leftarrow True$
    else
      $boolean\_array[j] \leftarrow False$
    end if
    $j \leftarrow j + 1$
  end for
  $links\_to\_segment[i] \leftarrow boolean\_array$ $i \leftarrow i + 1$
end for
return $links\_to\_segment$
```

DSS

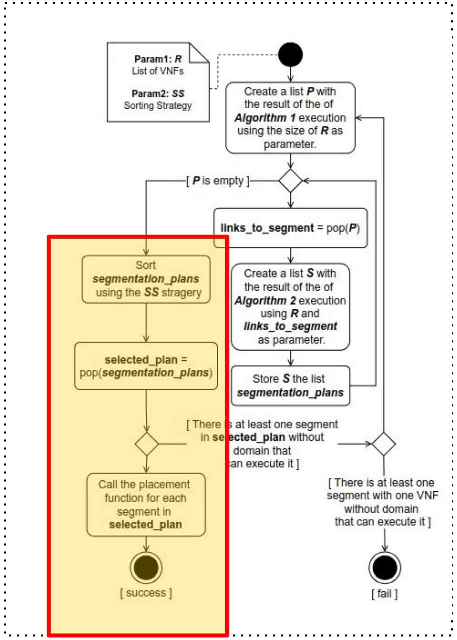
- The goal is to create a segmentation plan that divides the SFCs into segments.
- **Algorithm 2** groups the VNFs (from an SFC or a segment) into distinct segments.
- This segmentation is based on the result of the **Algorithm 1**.



Algorithm 2 Creation of segmentation plans

```
Input:
  $vnfs$: Array with the VNFs of the SFC
  $links\_to\_segment$: Array with booleans
Output: List of segments.
$segmentation\_plan \leftarrow List()$
$i \leftarrow 0$
while $i < size$(vnfs) do
    $segment \leftarrow List()$
    $segment.insert(vnfs[i])$
    $current\_index \leftarrow i$
    for $j \leftarrow i$ to $size(links\_to\_segment)$ - 1 do
        if $links\_to\_segment[j]$ == $True$ then
            $break$
        end if
        $segment.insert(vnfs[j+1])$
        $current\_index \leftarrow j+1$
    end for
    $i \leftarrow current\_index$
    $segmentation\_plan.insert(segment)$
end while
return $segmentation\_plan$
```

DSS



Param1: ***R***
List of VNFs

Param2: ***SS***
Sorting Strategy

Create a list ***P*** with the result of the of ***Algorithm 1*** execution using the size of ***R*** as parameter.

[***P*** is empty]

links_to_segment = pop(***P***)

Create a list ***S*** with the result of the of ***Algorithm 2*** execution using ***R*** and ***links_to_segment*** as parameter.

Store ***S*** the list ***segmentation_plans***

Sort ***segmentation_plans*** using the ***SS*** stragery

selected_plan = pop(***segmentation_plans***)

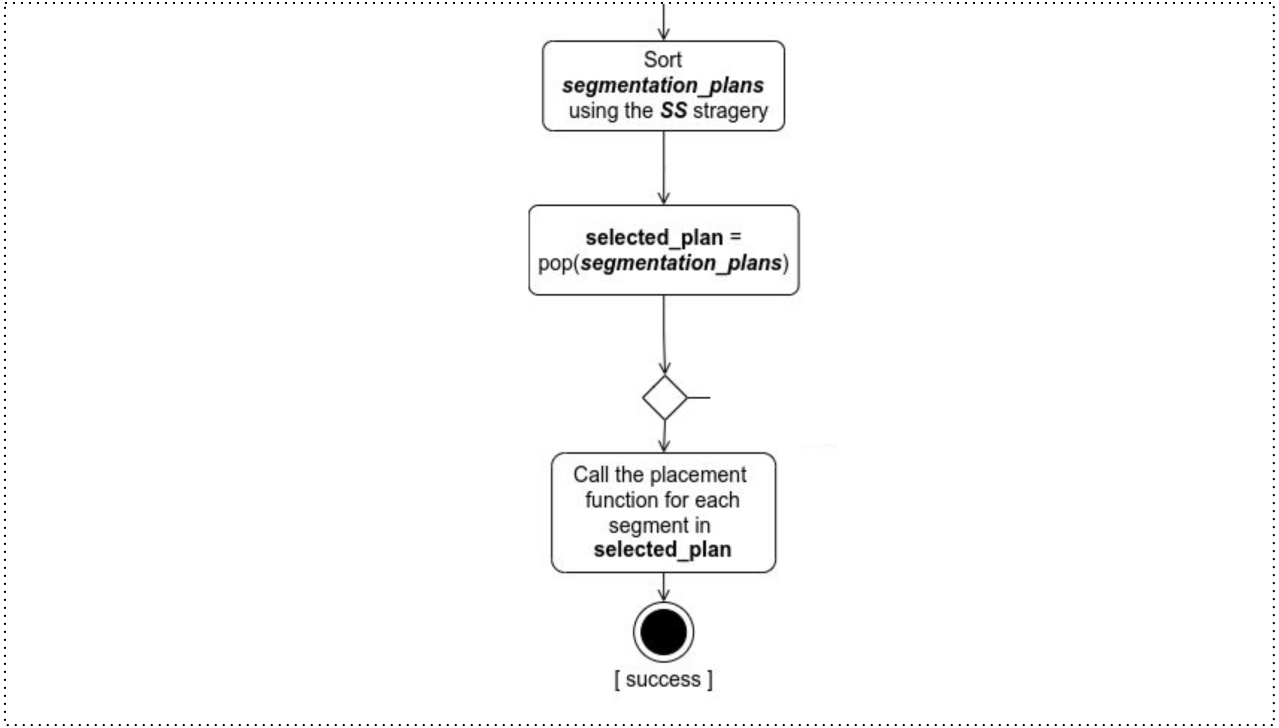
[There is at least one segment in **selected_plan** without domain that can execute it]

Call the placement function for each segment in **selected_plan**

[success]

[There is at least one segment with one VNF without domain that can execute it]

[fail]



Sort ***segmentation_plans*** using the ***SS*** stragery

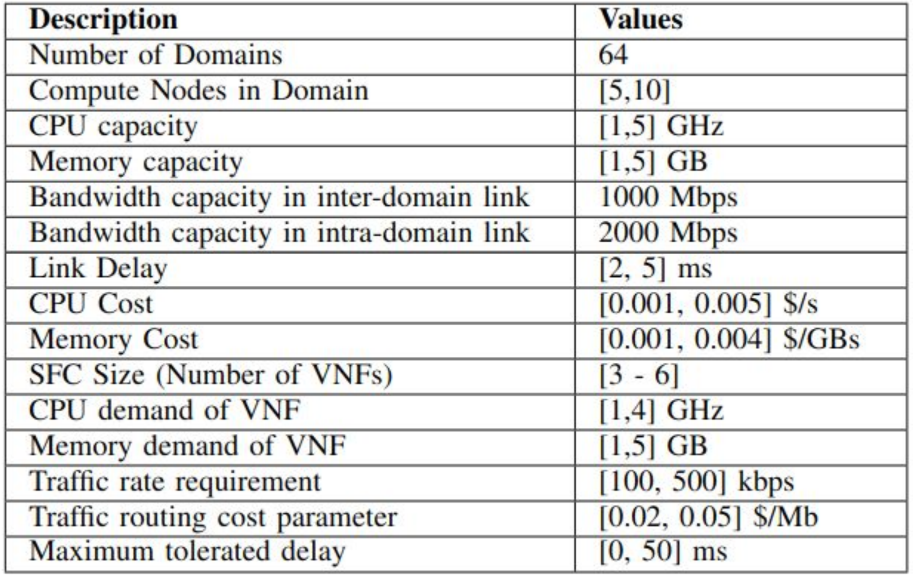
selected_plan = pop(***segmentation_plans***)

Call the placement function for each segment in **selected_plan**

[success]

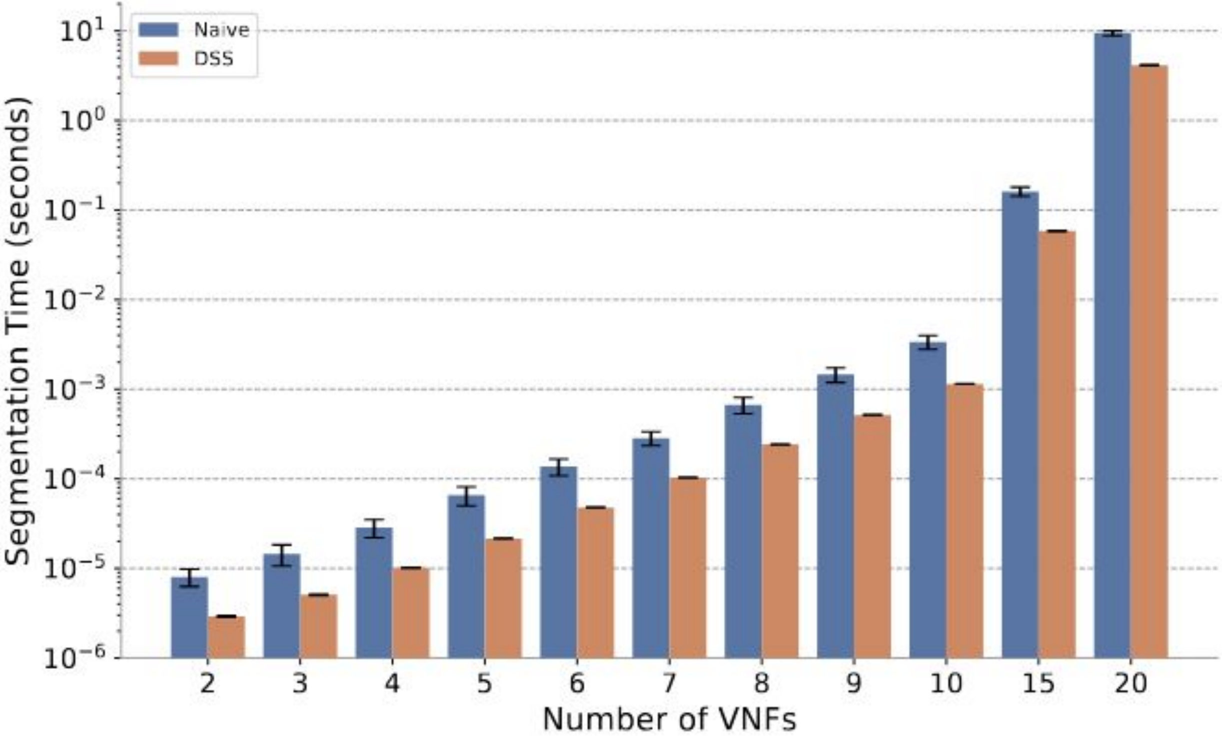
Performance Evaluation

- **Simulation** was implemented using **SimPy**.
- Scenarios with 5, 10, 20, 50 and 100 SFC requests.
- We use our heuristic with **different SFC placement methods (Greedy and Singleton Congestion Game)**.



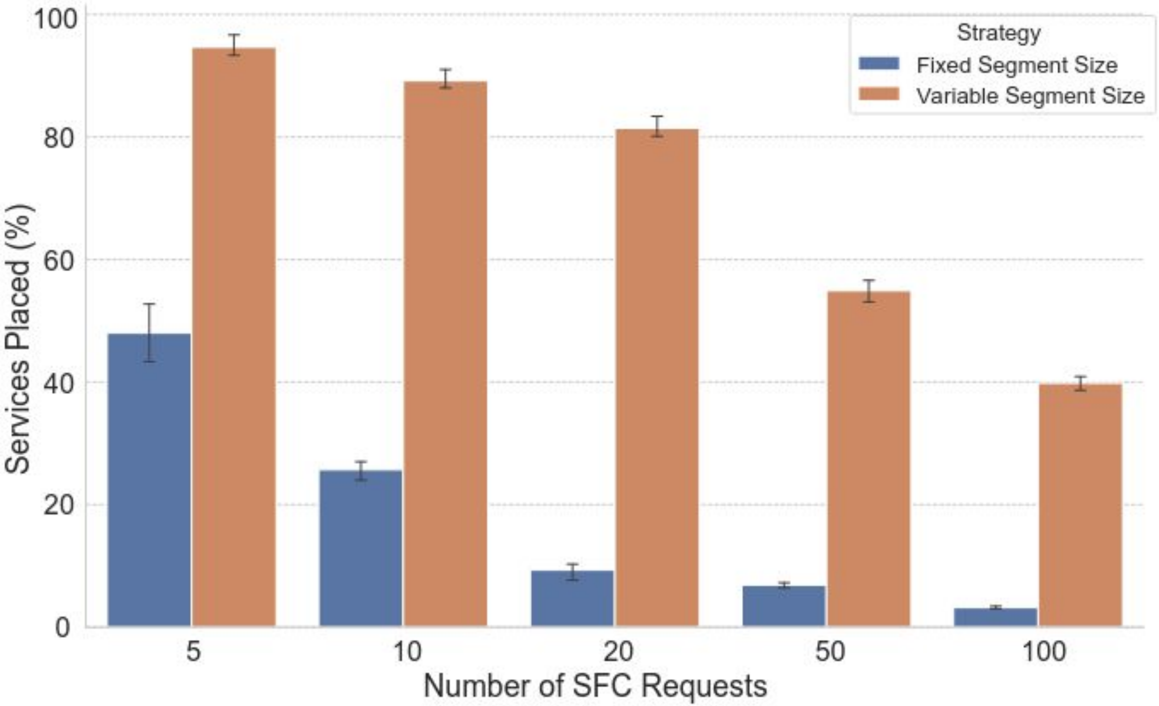
Description	**Values**
Number of Domains	64
Compute Nodes in Domain	[5,10]
CPU capacity	[1,5] GHz
Memory capacity	[1,5] GB
Bandwidth capacity in inter-domain link	1000 Mbps
Bandwidth capacity in intra-domain link	2000 Mbps
Link Delay	[2, 5] ms
CPU Cost	[0.001, 0.005] $/s
Memory Cost	[0.001, 0.004] $/GBs
SFC Size (Number of VNFs)	[3 - 6]
CPU demand of VNF	[1,4] GHz
Memory demand of VNF	[1,5] GB
Traffic rate requirement	[100, 500] kbps
Traffic routing cost parameter	[0.02, 0.05] $/Mb
Maximum tolerated delay	[0, 50] ms

Performance Evaluation



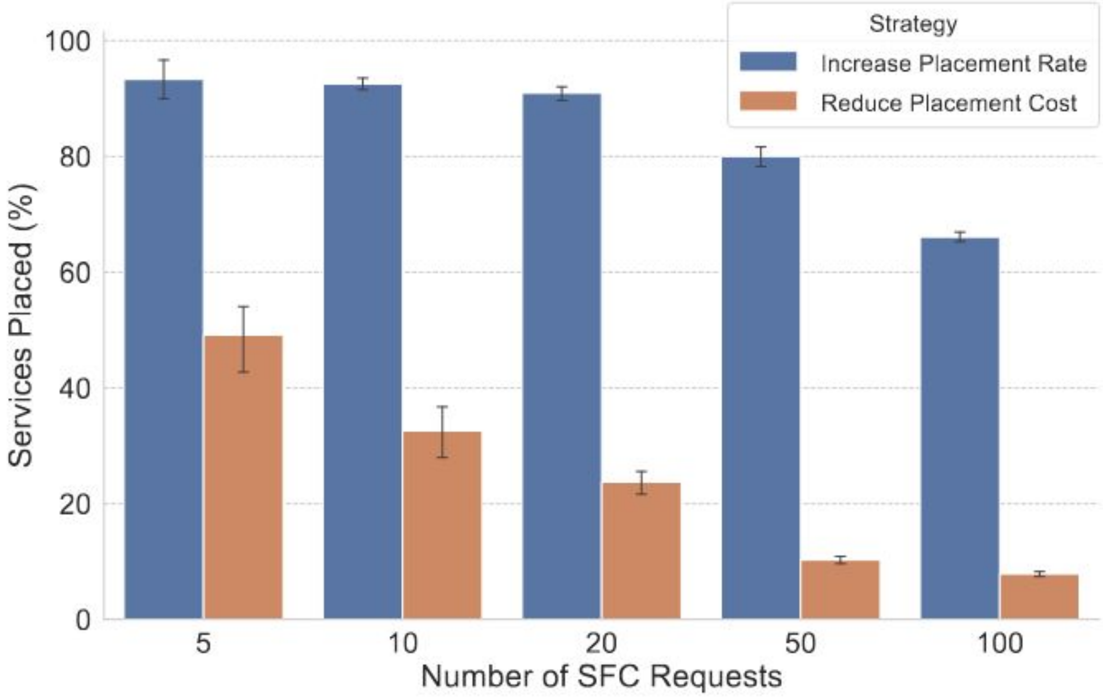
Performance Evaluation

Shows the **Success Rate** of SFC placement using a **placement method** based on **Singleton Congestion Game**



Performance Evaluation

Shows the **Success Rate** of SFC placement using a **placement method** based on **Greedy Approach**



Final Remarks

- We proposed a **new heuristic for solving the SFC segmentation problem**.
- The experimental results indicate that our **method is applicable to various SFC placement methods**.
- In future work, we will **implement** the proposed heuristic as a component within platforms such as **Open Source MANO (OSM) or the Edge Multi-Cluster Orchestrator (EMCO)**.

IEEE CloudNet
IEEE International Conference on Cloud Networking
27–29 November 2024 // Rio de Janeiro, Brazil

A Novel Method for SFC Segmentation in Edge-Cloud Environments

Anselmo Luiz Éden Battisti
anselmo@midiacom.uff.br

funding

Instituto de Computação uff
CAPES
openRAN BRASIL
FAPERJ Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro
CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico