Parallel Differential Evolution Meta-Heuristics and Modeling for Network Slicing in 5G Scenarios

Rayner Gomes
rayner@ufpi.edu.br

Rayner Gomes
rayner@ufpi.edu.br
UFPI

Dario Vieira
dario.vieira@efrei.fr
EFREI Paris
France

Miguel F. de Castro
miguel@ufc.br
UFC

Leonel Feitosa
leonelfeitosa@ufpi.edu.br
UFPI

Francisco Silva
faps@ufpi.edu.br
UFPI

UFC: Federal University of Ceará - Brazil
UFPI: Federal University of Piauí - Brazil

The Eighteenth International Conference on Networking and Services - ICNS 2022

May 22, 2022 to May 26, 2022 - Venice, Italy

Parallel Differential Evolution Meta-Heuristics and Modeling for Network Slicing in 5G Scenarios

Rayner Gomes holds a master's degree in Computer Science in 2002 from the Federal University of Uberlândia (UFU - Brazil). He is a professor at the Federal University of Piauí (UFPI-Brasil) and has taught for over 20 years on computer networks and distributed computing. He is currently a Ph.D. student at the Federal University of Ceará (UFC - Brazil) and works together with the research group of the EFREI network (France).

His research interest is at the intersection of the Internet of Things, resource allocation, network slicing services in 5G systems, and QoS.

Aims and contributions

1. **In our work, we aim to:**
    a. Create the three versions of the Differential Evolution Meta-Heuristic to the mapping problem considering the Network Slicing as Service in 5G System.
        i. Sequential
        ii. Vertical Parallelization
        iii. Horizontal parallelization
    b. We design a new fitness function to take into account *bandwidth, delay, reliability* and *geographic* node position.

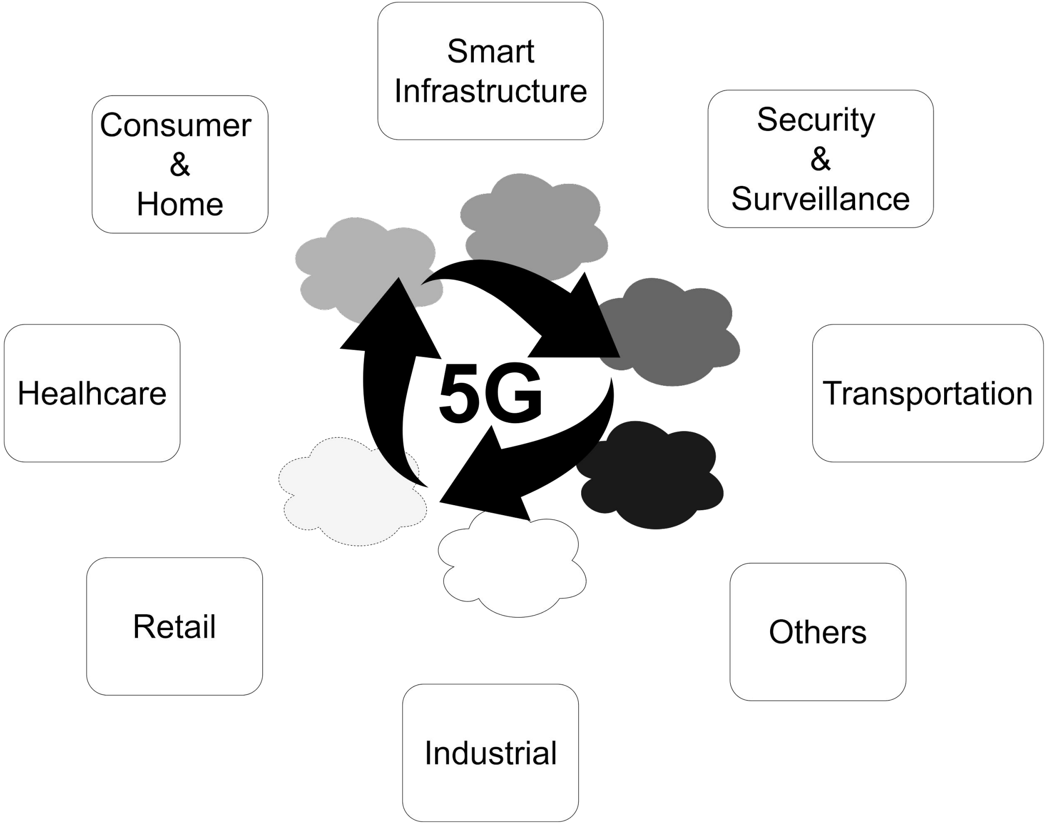
Aims and contributions

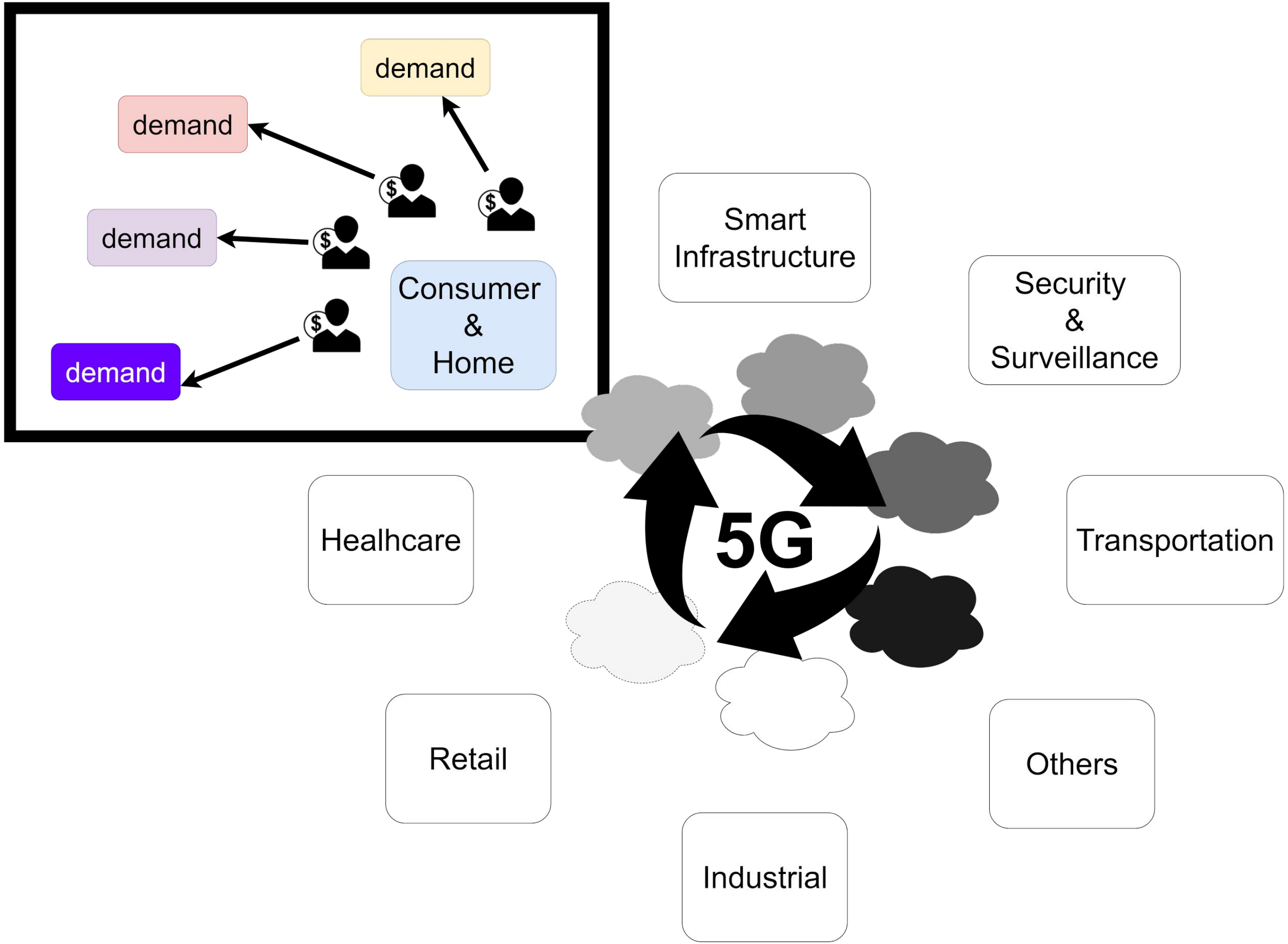
2. **The contributions of our study are:**
   a. We designed a Stochastic Petri Net (SPN) to model a Network Slicing as a Service component.
   b. We present some aspects in which parallel approaches have advantages and disadvantages.

Introduction

Fifth-generation mobile networks (5G) have the ambition to endure a wide range of services and applications!

5G World Integration

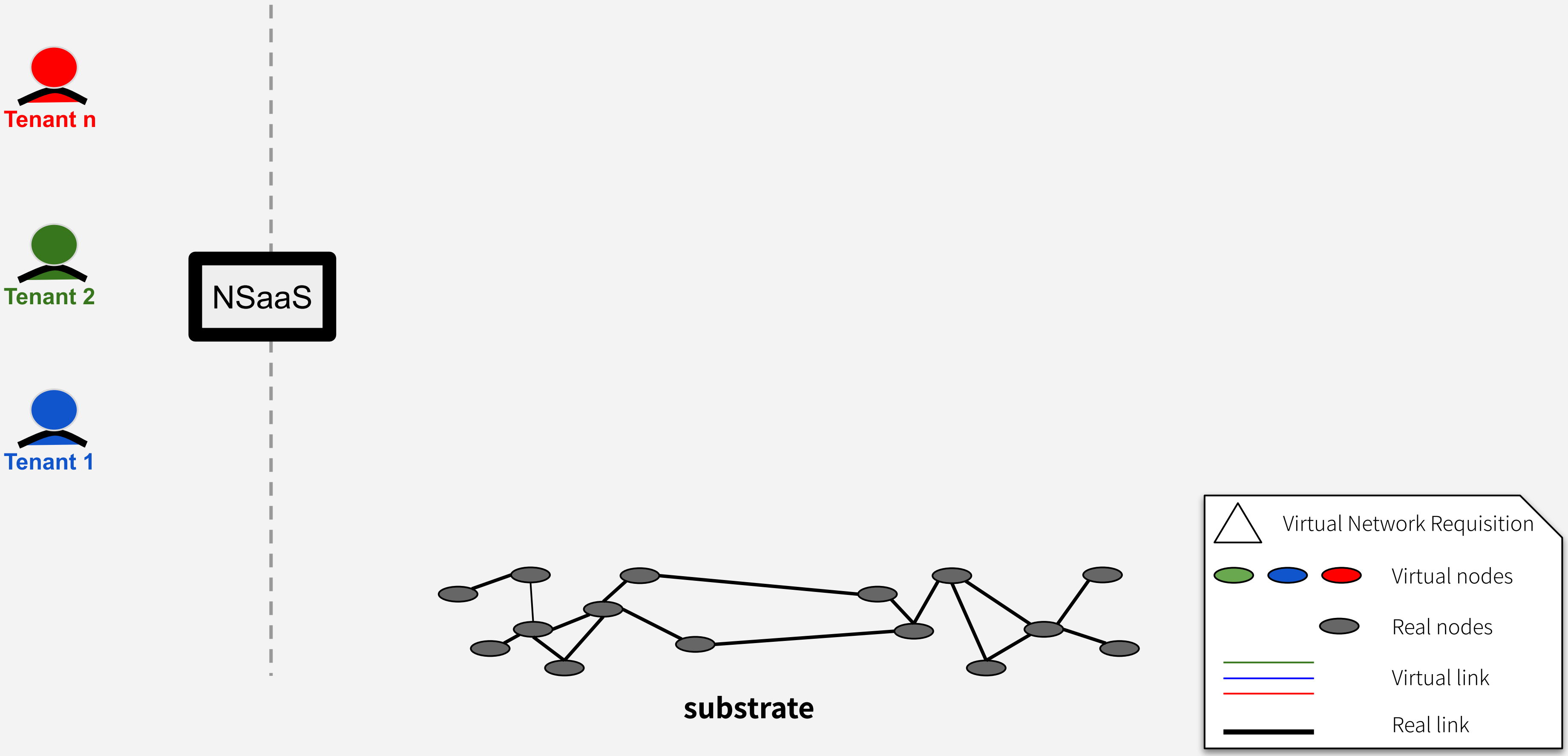



demand
demand
demand
demand
Consumer & Home
Smart Infrastructure
Security & Surveillance
5G
Healhcare
Transportation
Retail
Others
Industrial

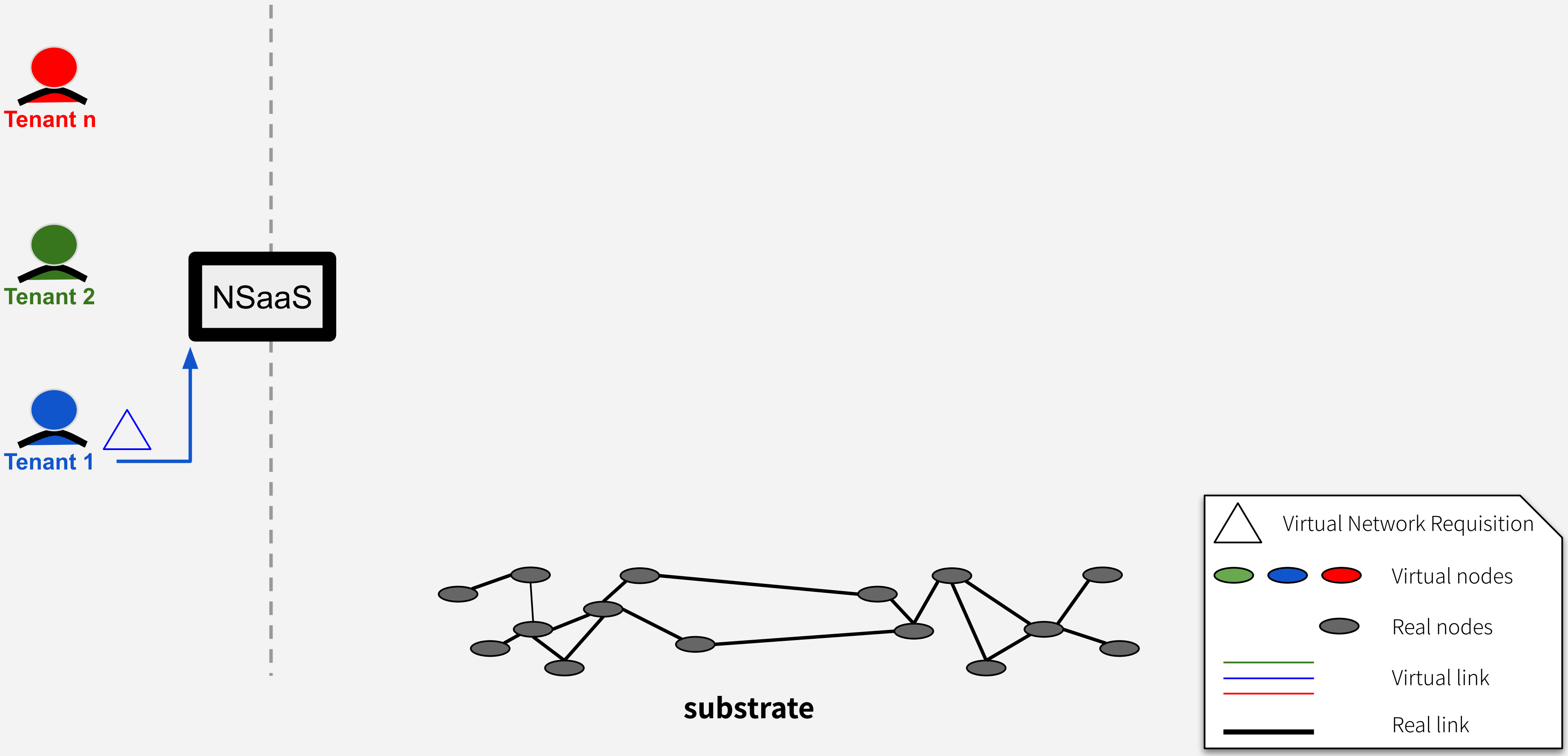
Network Slicing as a Service - NSaaS

NSaaS is a standard mechanism adopted in 5G to face the diversity of demands and distinct requirements of services and applications. NS has been designed as a key enabler to allow 5G to handle the Internet of Things and other vertical markets.

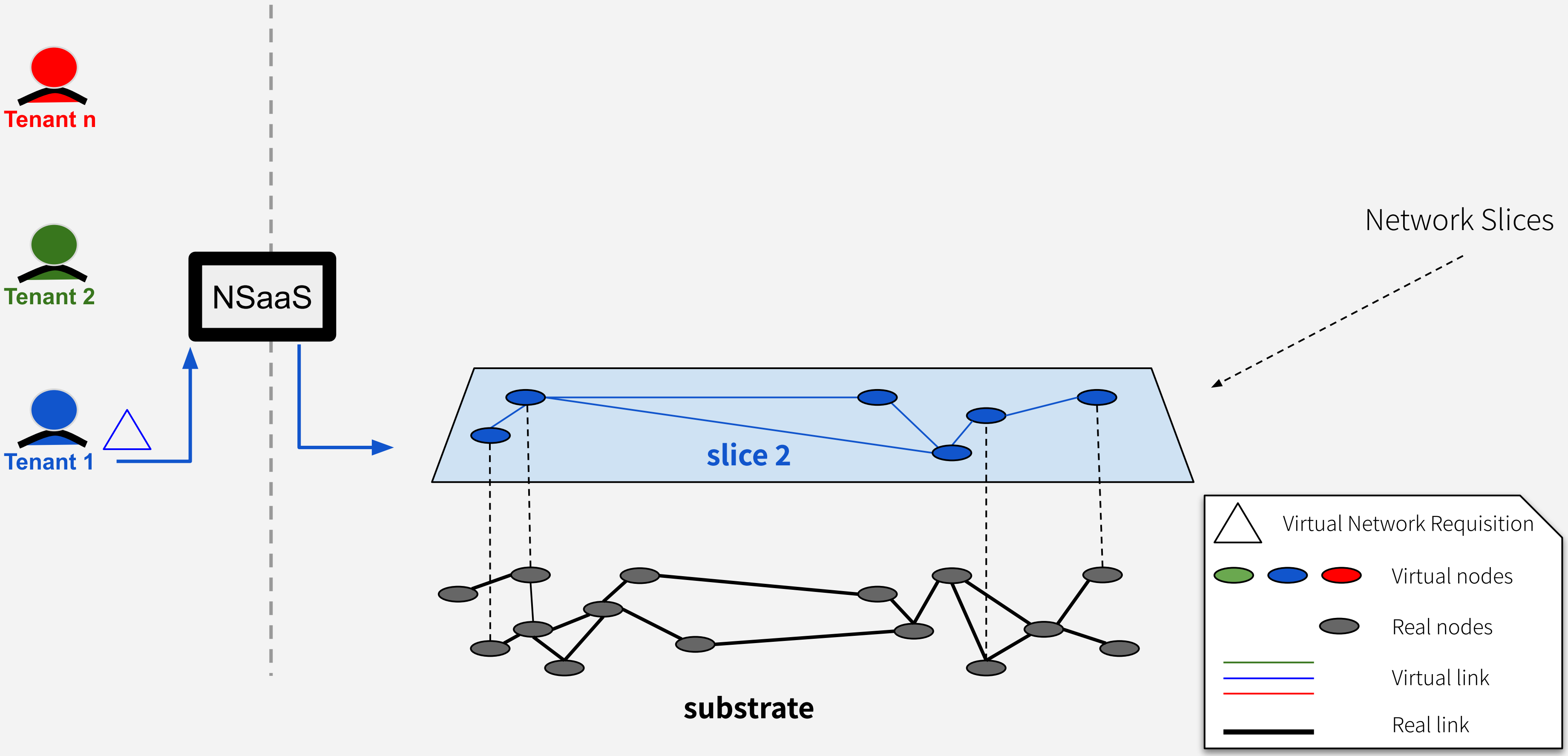
NSaaS: Network Slice as Service



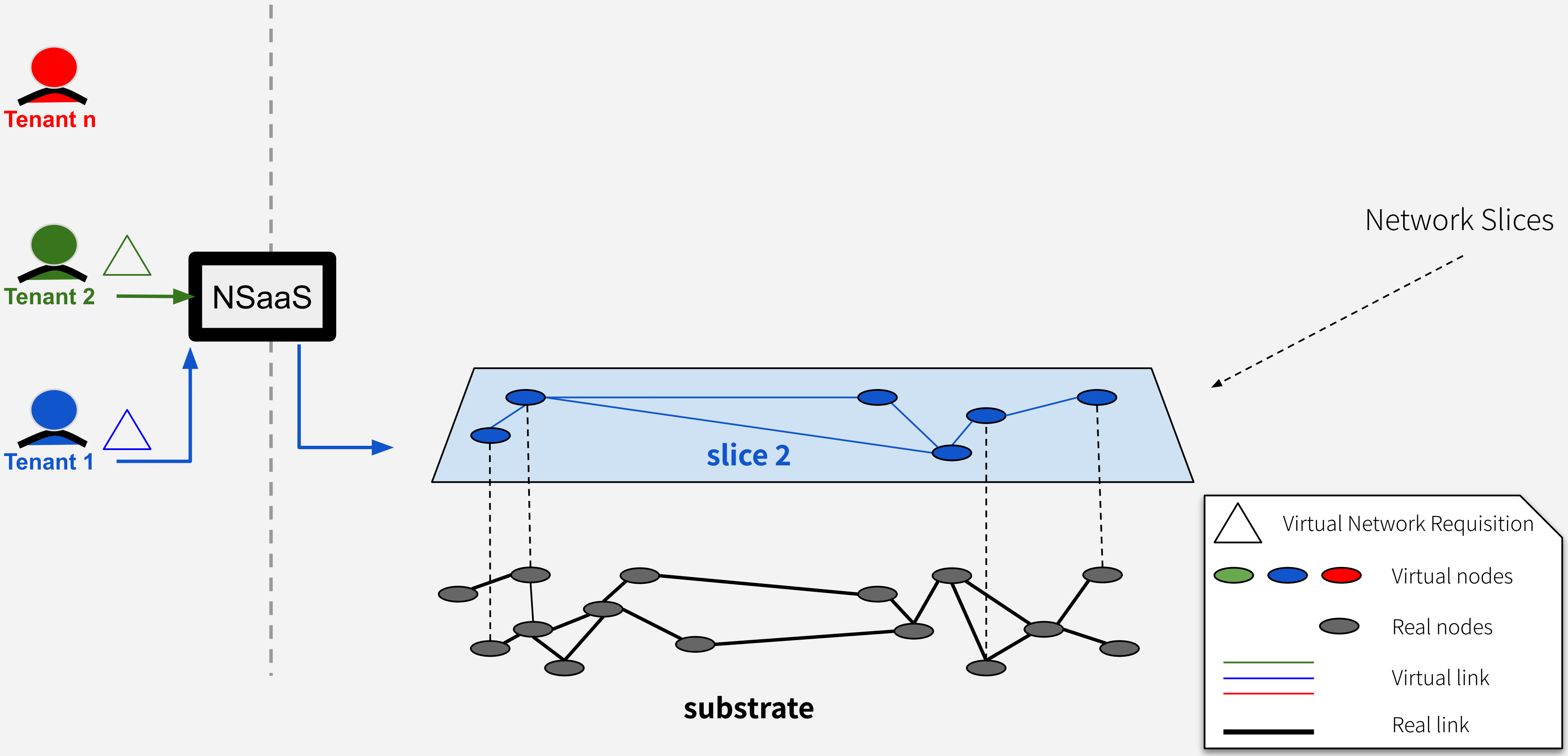
NSaaS: Network Slice as Service



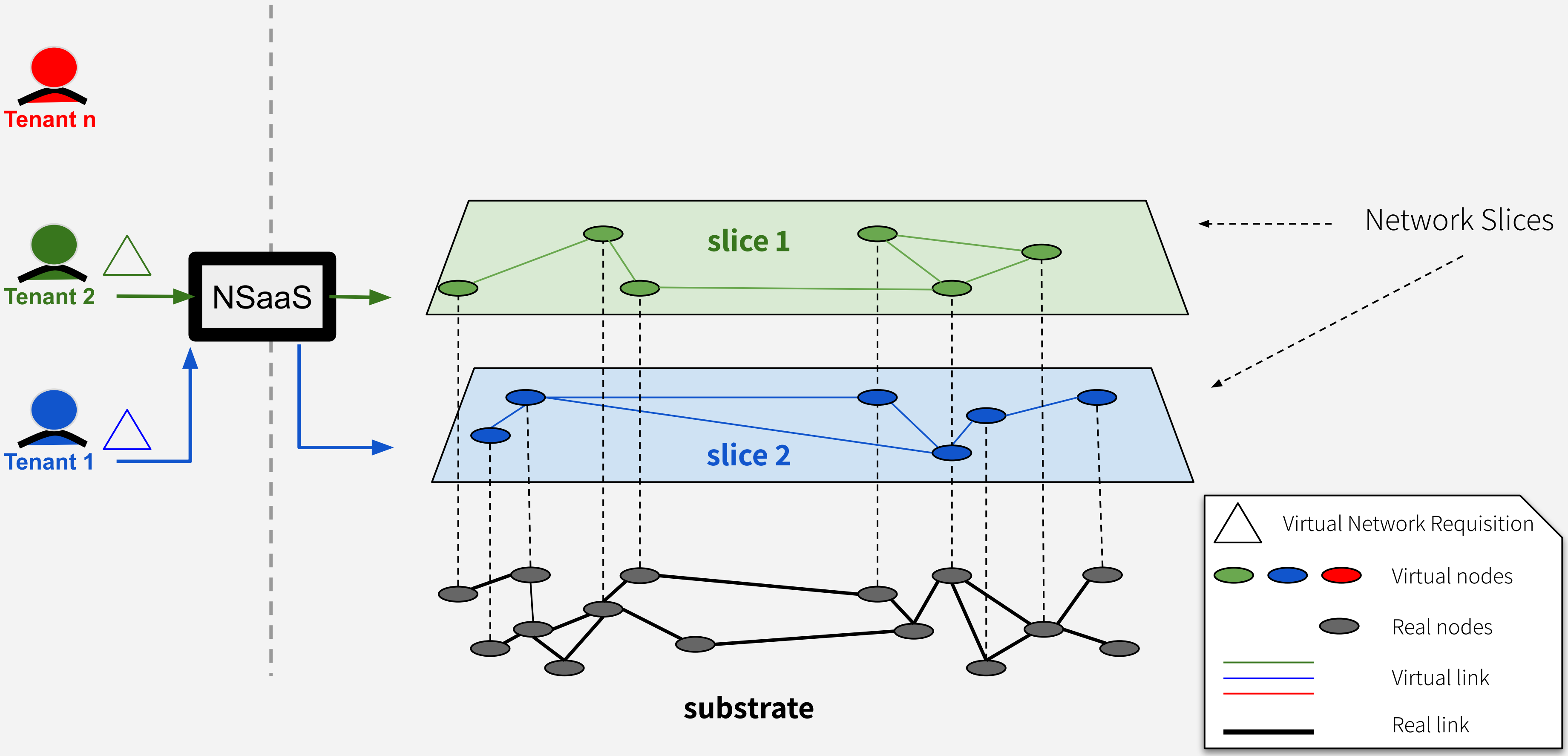
NSaaS: Network Slice as Service



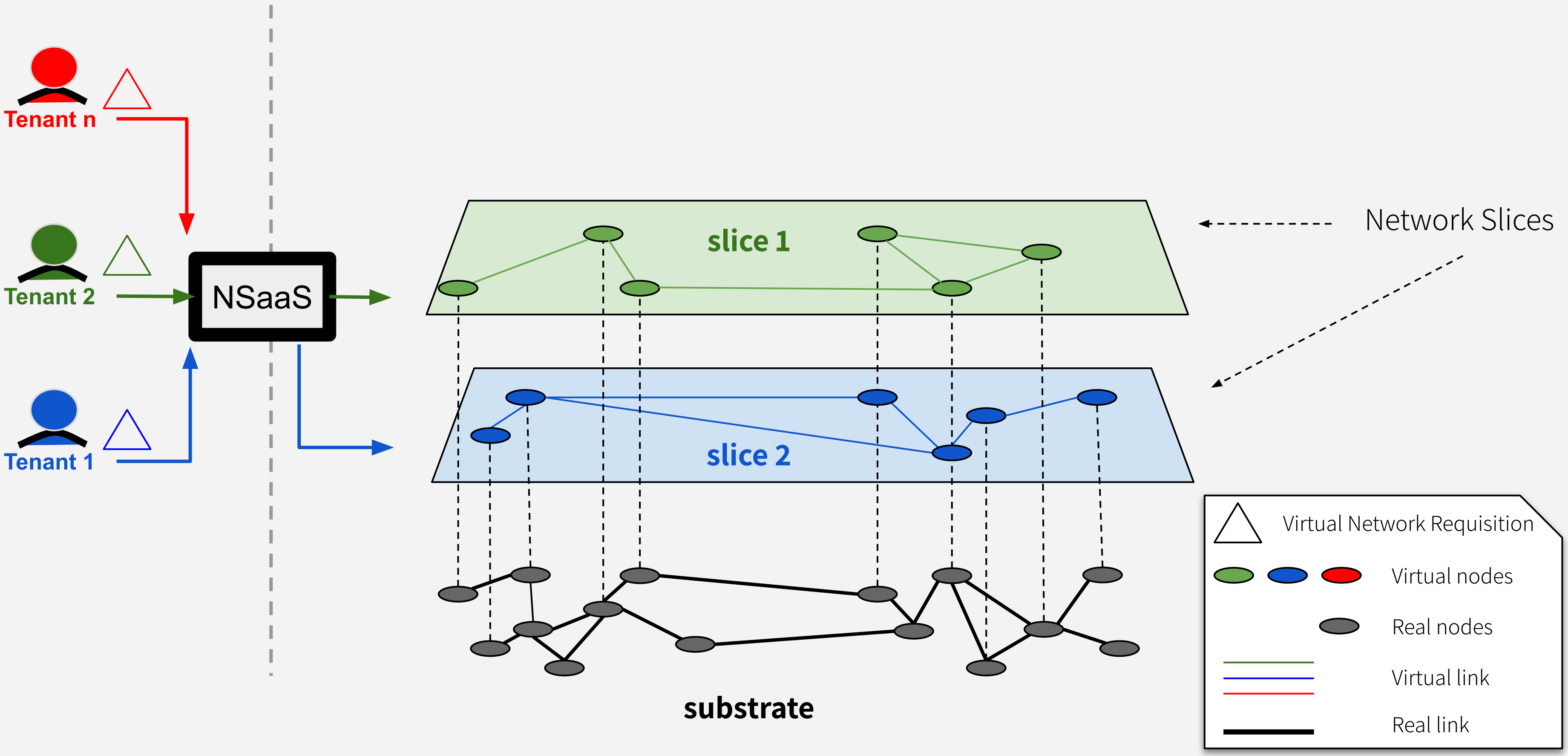
NSaaS: Network Slice as Service



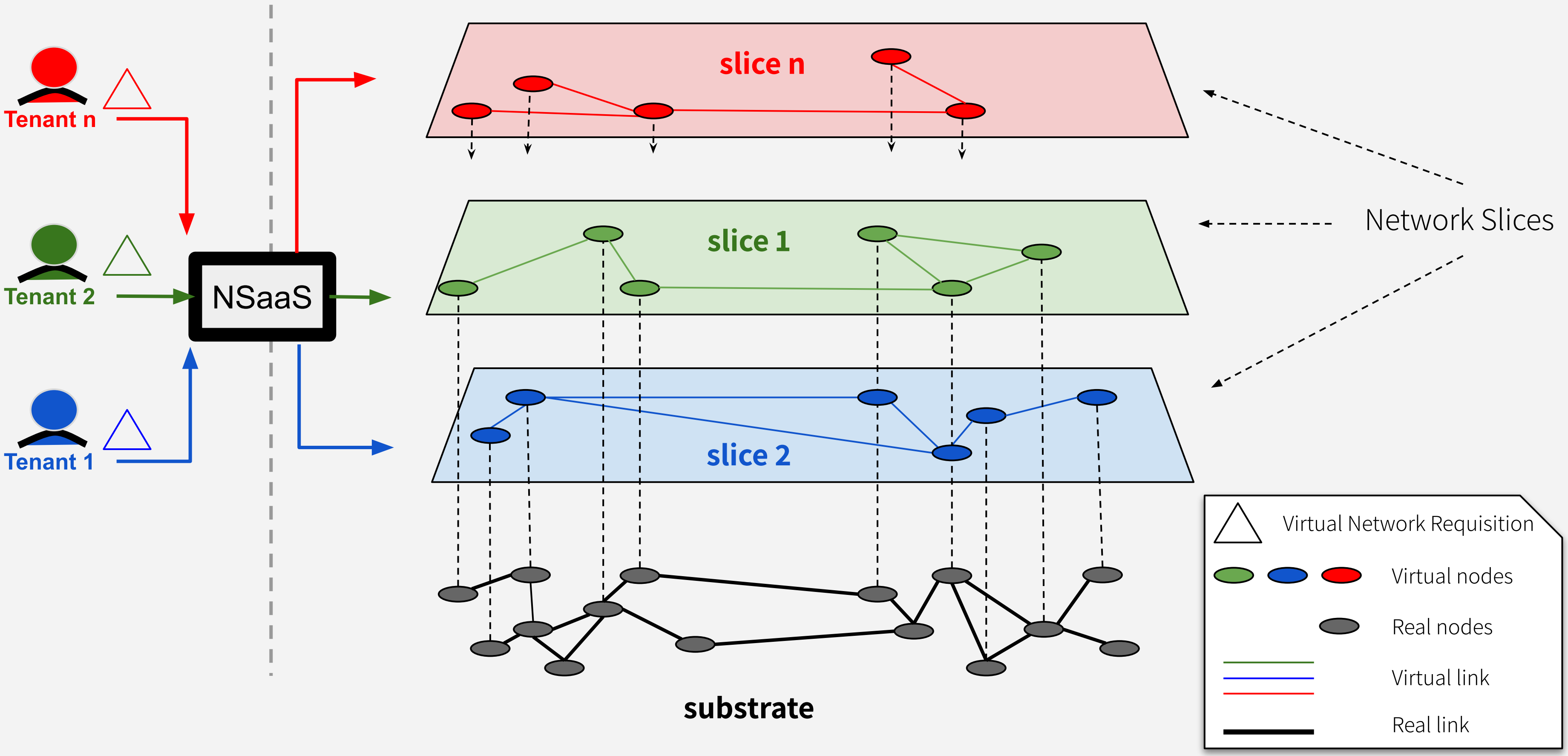
NSaaS: Network Slice as Service



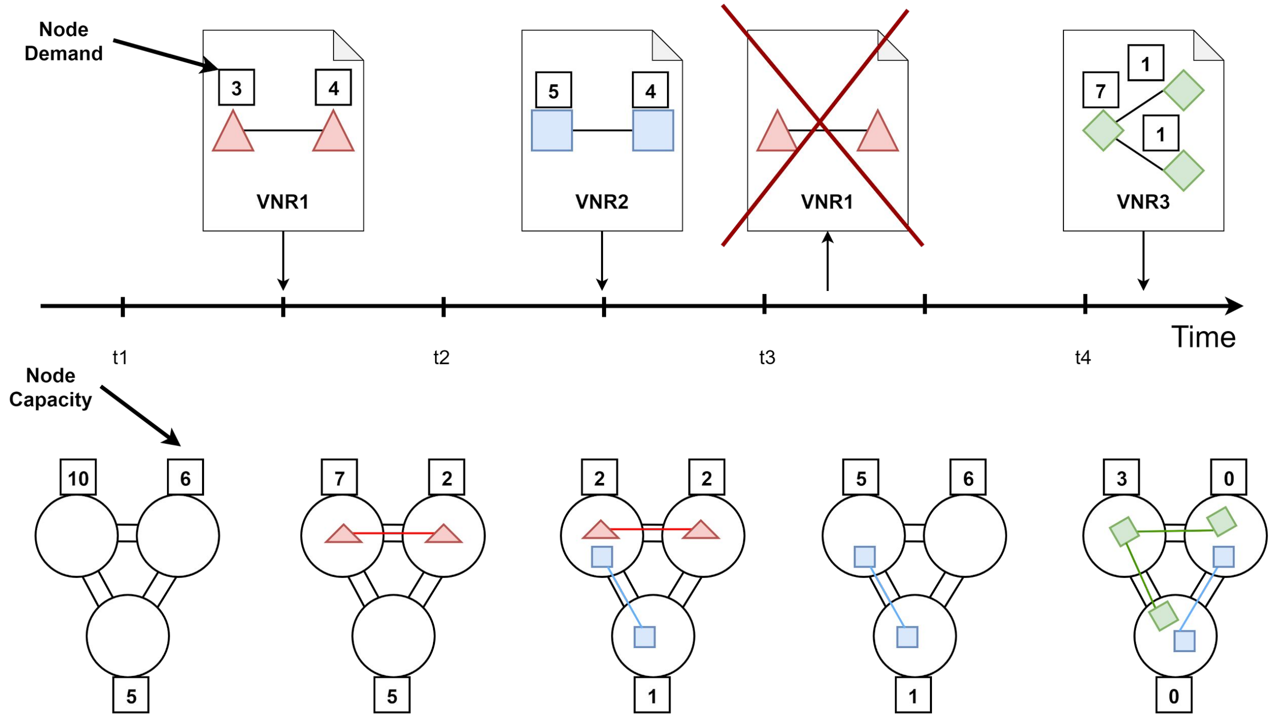
NSaaS: Network Slice as Service



NSaaS: Network Slice as Service



Online Approaches



Each Network Slice has a duration!

Common Types of Slices

The Next Generation Mobile Network Alliance (**NGMN**), the Third Generation Project Partnership (**3GPP**), and the International Telecommunication Union (**ITU**) have suggested three types of slices:

- Enhanced mobile broadband (eMBB)
- Ultra-Reliable and Low Latency Communications (URLLC)
- massive Machine Type Communications (mMTC)

These three and others can be defined and have different network behaviors based on their priorities and values for bandwidth, delay, and reliability!

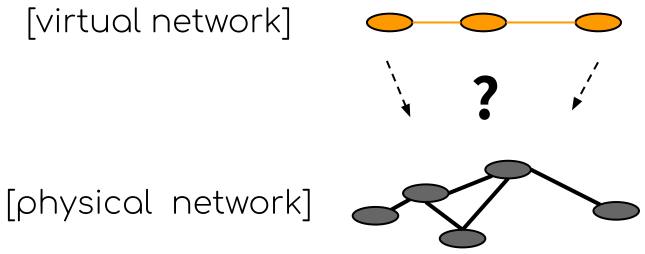
Network Slicing as a Service - NSaaS

Important points from the introduction:

- NSaaS is a service that offers slices as a service.
- A slice is a virtual network to meet a specific tenant's desire.
- A tenant sends a requisition (VNR) to the NSaaS provider.
- A VNR has:
  - Topology
    - Nodes (*each one with its geographic position*)
    - Links
  - Duration
  - Demands (QoS parameters)
    - Nodes capacity
    - Links capacities
      - Delay
      - Bandwidth
      - Reliability

Virtual Network Embedding (VNE)

Complexity

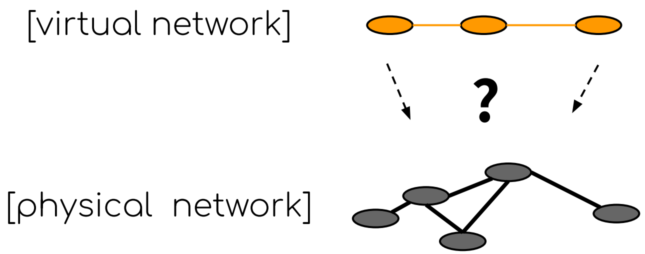


The process of mapping VNRs comprises a complex task to associate a set of virtual to a set of physical nodes and links, taking into account all network constraints and resource availability.

Mathematically this problem is known as Virtual Network Embedding (VNE)

Virtual Network Embedding (VNE)

Complexity



Network slicing:

- Mapping process (VNE)
    - Optimization Process
- Resource allocation
    - Over-provisioning
    - Under-provisioning

- The basic VNE problem is an integer linear problem (ILP), which is *NP-hard* as it can be proven by reduction to the multiway separator problem.
- Even with a given virtual to physical node mapping, the problem of optimally allocating a set of virtual links to single physical paths reduces to the unsplittable multicommodity flow problem and therefore is also ***NP-hard***.

VNE Approaches

There are three approaches to dealing with VNE: exact, heuristic, and meta-heuristic:

1. **Exact**: Only indicated to small network, *not appropriated to 5G networks!* Its solution is fixated on the global optimum and can easily suffer from the problem of being stuck in a local optimum.
2. **Heuristic** and **Meta-heuristic**: aim at finding a suitable solution under realistic network scenarios. The meta-heuristic solution can improve the quality of results by escaping from the local optimum [4] [6].

Related Works

- In our previous work [5], based on a review of 125 articles studied in [6], we revealed that the Genetic Algorithm (GA) is one of the most used meta-heuristics to solve VNE problems.
- Additionally, our previous work [5] showed that with small population size and repetition, DE has a better acceptance rate than GA.
- However, with a longer execution time. Therefore, in this work, we developed a way through parallelization to reduce the execution time of DE.

Differential Evolution Meta-Algorithm

Step 1: Creating the initial population (Size of Population)

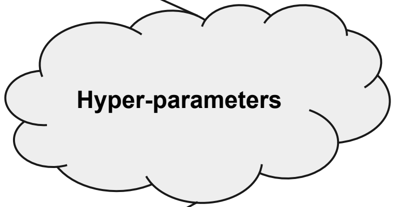
Step 2: **REPEAT**

Step 3: Mutation and regeneration

Step 4: Crossover

Step 5: Selection

Step 6: **UNTIL** (number of iterations = Maximum number of iterations)



Algo 01: Meta-Algorithm of Differential Evolution

Differential Evolution Meta-Algorithm

Step 1: Creating the initial population

Step 2: **REPEAT**

Step 3: Mutation and regeneration

Step 4: Crossover

Step 5: Selection

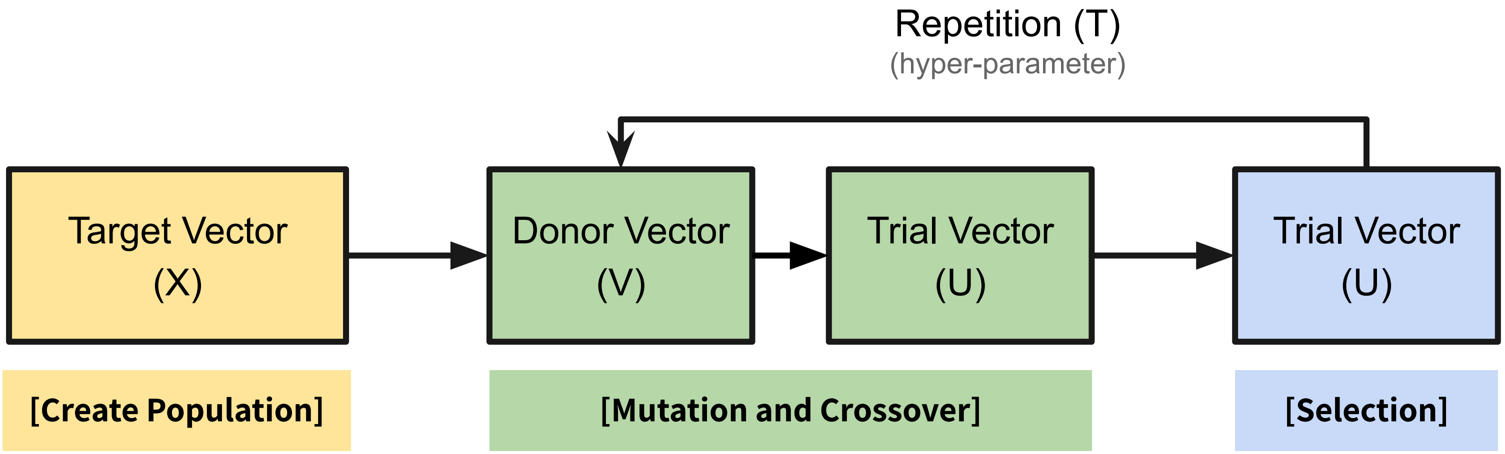
Step 6: **UNTIL** (number of iterations = Maximum [illegible]ations)

This red block is where the sequential code is parallelized!

Algo 01: Meta-Algorithm of Differential Evolution

Differential Evolution Meta-Algorithm

Meta-Algorithm of Differential Evolution Oriented by Vector (Data) Blocks

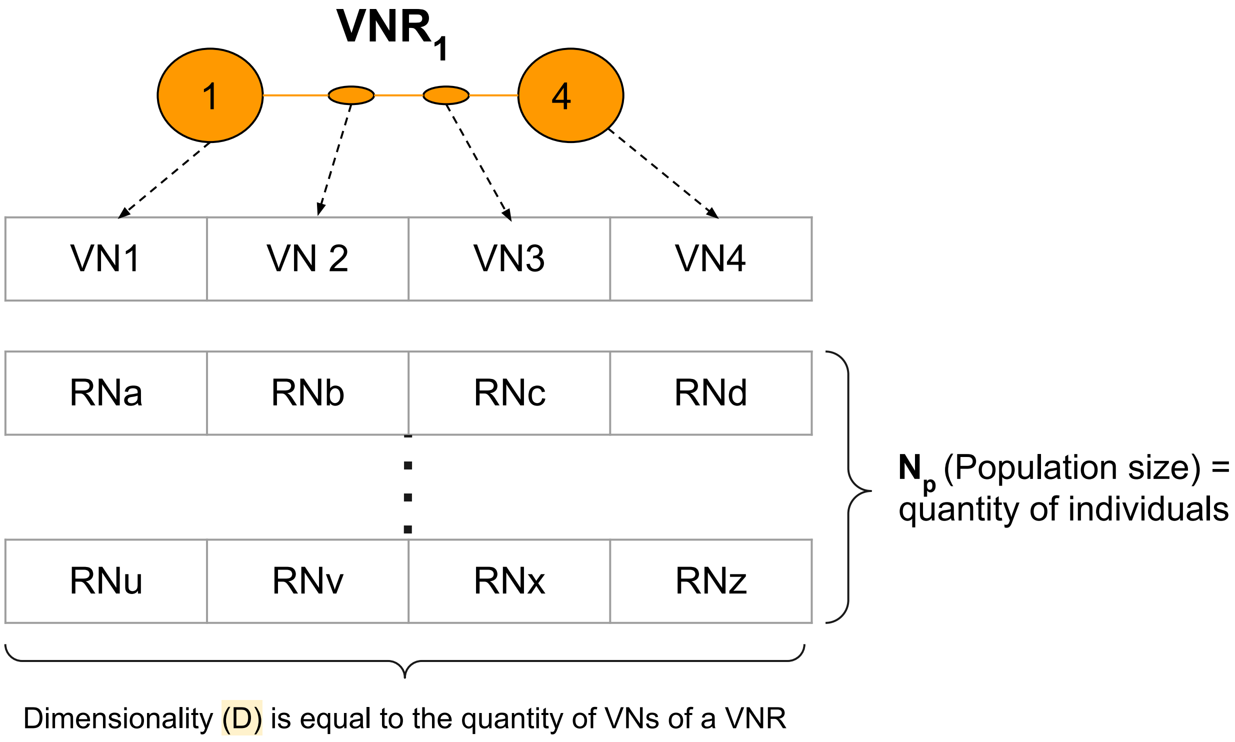


The mutation of each individual into the population can be performed individually, this feature allow different parallelization variations.

Differential Evolution Meta-Algorithm

Step 1: Creating the initial population

- The population is a set of individuals.
- An individual or chromosome is a random map of a virtual node to a real node.
- Each individual is a possible map
- The population size is a hyper-parameter.
- It is a stochastic population-based technique.
- The population is defined by Target Vector (X).



Differential Evolution Meta-Algorithm

Step 3: Mutation

- The mutation involves a creation of the Donor Vector (V).
- The mutation process is more complex than in the Genetic Algorithm (GA), and it is the point where in general the DE spend more time than the GA when running with same hyper-parameters.
- Each chromosome undergoes mutation followed by recombination.
- Donor Vector (V_i) of a chromosome (X_i) is created as:

$$\boxed{V_i = X_{r1} + F(X_{r2} - X_{r3})}$$

- F: Scaling factor, a constant between 0 and 2
- r_1,r_2,r_3: Random solutions in N_p

Differential Evolution Meta-Algorithm

Step 4: Crossover

- The DE's crossover involves a creation of the Trial Vector (U).
- Also, the DE's crossover process is more complex than in the Genetic Algorithm (GA), and it is the point where in general the DE spends more time than the GA when running with same hyper-parameters.
- Each individual in the vector U is created based on this rule:

$$
\boxed{u^j = \begin{cases} v^j, & \text{if } r \leq p_c \vee j = \delta \\ x^j, & \text{if } r > p_c \wedge j \neq \delta \end{cases}}
$$

Where:

- P_c: crossover probability
- □: randomly selected variable location ∈ {1,2,3,...,D}
- R random number between 0 and 1
- u^j : j^{th} variable of trial vector
- v^j : j^{th} variable of donor vector
- x^j : j^{th} variable of target vector

Differential Evolution Meta-Algorithm

Fitness Function

- Our fitness function is a special contribution which considers all the QoS parameters (bandwidth, delay and reliability) required in VNRs.
- Fitness takes the individual's properties as its parameter.
- It returns values between 0 and $+\infty$
  - if zero means that an individual does not meet all demands.
  - If the return is 1, it means that an individual fully meets a VNR.
  - Return values greater than 1 mean that an individual represents a larger set of resources than needed.
  - The higher the value, the more unnecessary resources are selected. Based only on the fitness, the DE can select more adapted individuals.

Differential Evolution Meta-Algorithm

Fitness Function

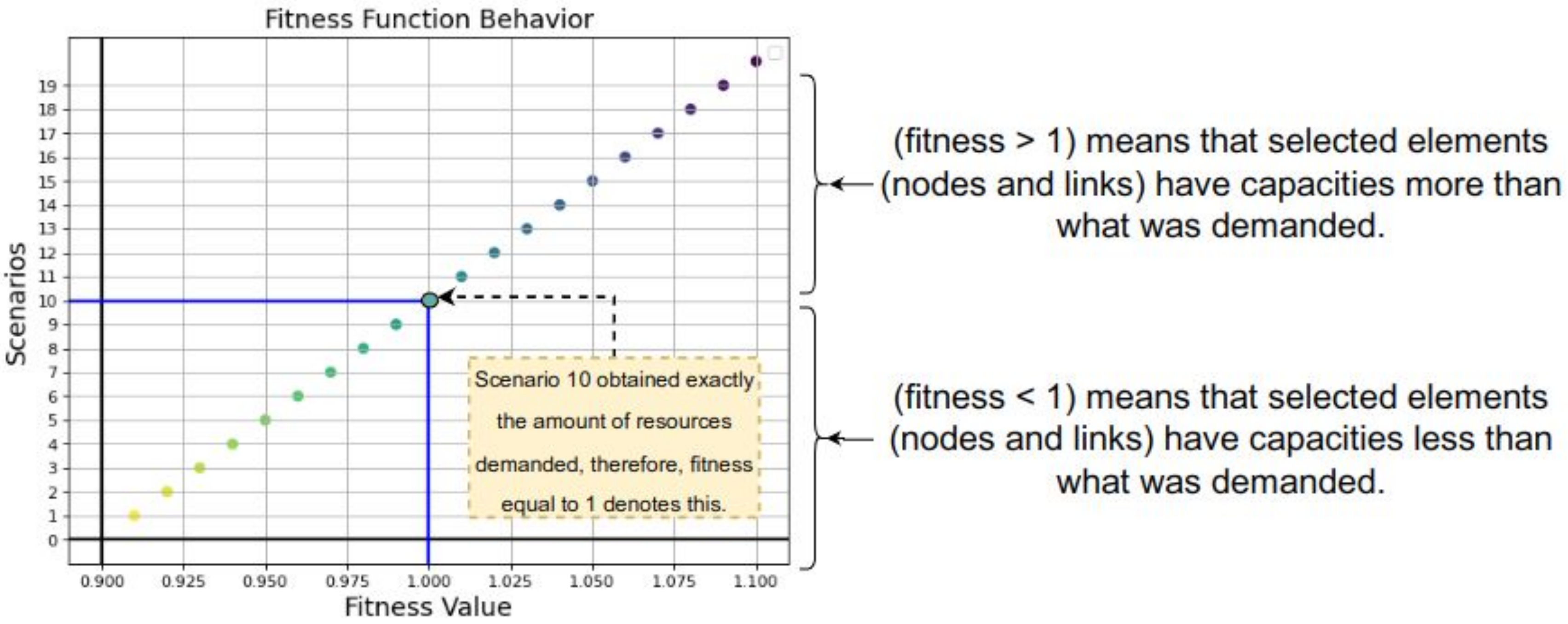


Figure 4. Fitness function behavior.

Differential Evolution Meta-Algorithm

Fitness Function: Mathematical Description

- Equation 3 denotes the final score of an individual.
- The individual's fitness is the average of all scores plus the result of the score logarithm based on the number of hops (Equation 3).
- The function ***c(r)*** is the coefficient defined for each QoS parameter in a VNR
- The variable ***P*** denotes a set of parameters from each type of slice, ranging from 0 to 1, and the greater the number, the higher the relevance. Thought the variable P, it is possible to create countless types of slices using these coefficients.

$$total = \frac{\sum_{i}^{|P|} c(P[i]) * \underline{score(i)}}{|P|} + log(hops, \underline{score(i)}) \quad (3)$$

Differential Evolution Meta-Algorithm

Fitness Function: Mathematical Description

- Each individual ***i*** is associated to a VNR.
- The score is a value that expresses how efficient an individual is, and its equation is defined in Equation (2).
- The ***vMax(r)*** denotes the maximum capacity of a specific-resource in the whole infrastructure
- ***R*** is a set of resources, such as ***r ∈ R***.
- The ***vGot(r)*** returns a value that denotes the maximum capacity of resource ***r***.
- The ***vDesired(i,r)*** returns a demanded capacity of resource ***r*** which is required by individual ***i***.
- The variable ***r*** can assume three specific-resources in this work: *(i) bandwidth; (ii) delay; and (iii) reliability*.
- The score is a normalized measure, thus we can carry out different resources at once.
- Each QoS parameter present in the VNR has its own score.
- The symbol diamond (◊) is the addition operator when the variable ***r*** represents the bandwidth or reliability and it is the subtraction operator when it represents the delay parameter.

$$score(i) = \frac{vMax(r) \diamond (vGot(i) - vDesired(i,r))}{vMax(r)} \qquad (2)$$

Differential Evolution Meta-Algorithm

Step 5: Selection

- After the mutation and crossover all results are stored in the Trial Vector (U).
- Evaluate the fitness of all elements in the Trial Vector (offspring) (f_u).
- Population is update using greedy selection: [minimization problem]

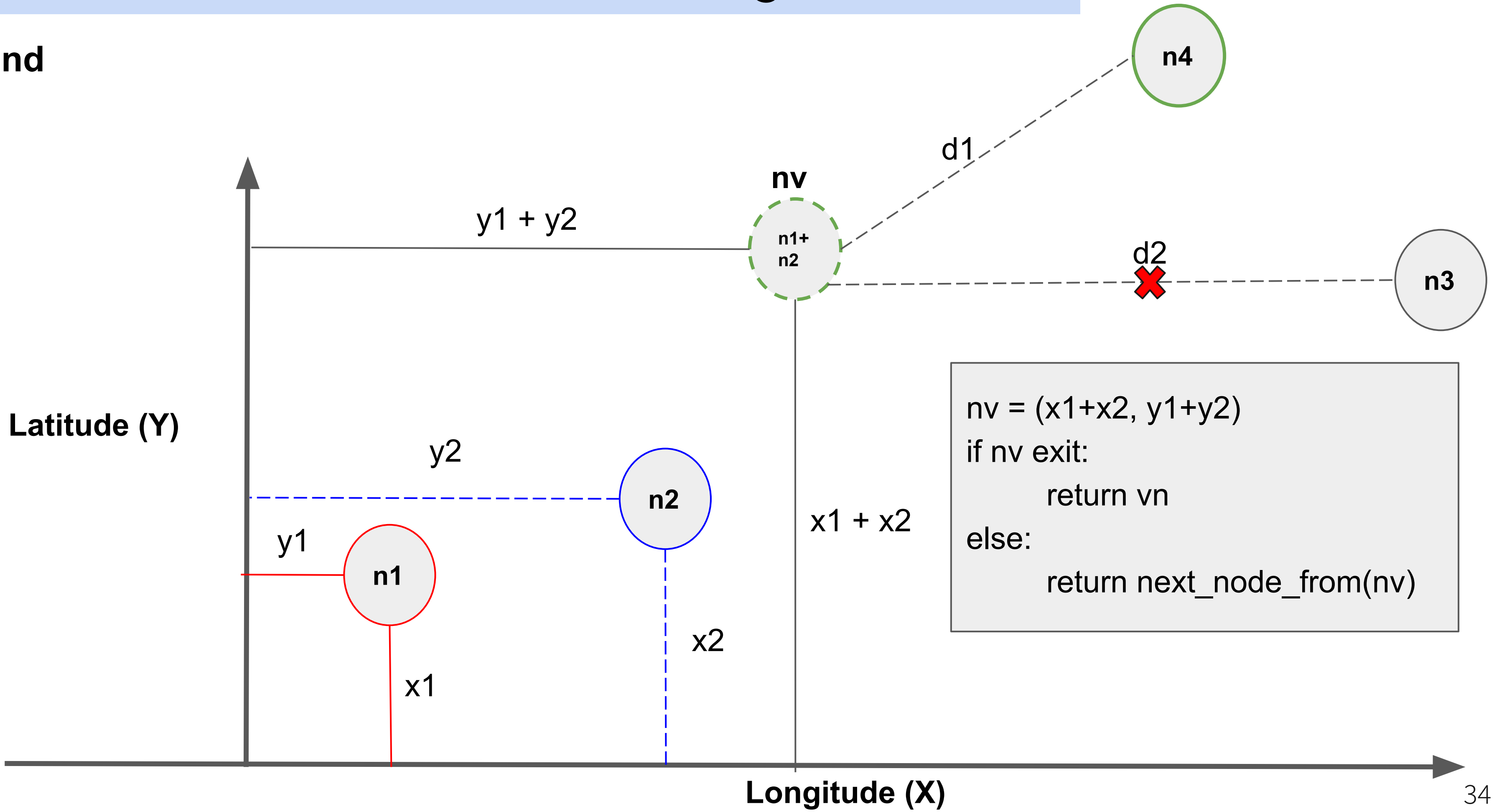
$$\left.\begin{matrix} X_i = U_i \\ f_i \end{matrix}\right\} \text{if } f_{ui} < f_i \qquad = \qquad f_{ui}$$

X and f remains the same if $f_{ui} > f_I$

The goal is to return the better individuals considering the Trial Vector and Target Vector.

Differential Evolution Meta-Algorithm

Bound



Differential Evolution Approach

Derived from Meta-Algorithm.

Step 1: Creating the initial population

Step 2: Evaluate fitness function in all individuals of Population (X)

Step 3: **for t in 1 to T:**

Step 4:   **for i in i to Np:**

Step 5:   Create Donor-Vector (V)

Step 6:   Perform Crossover (U)

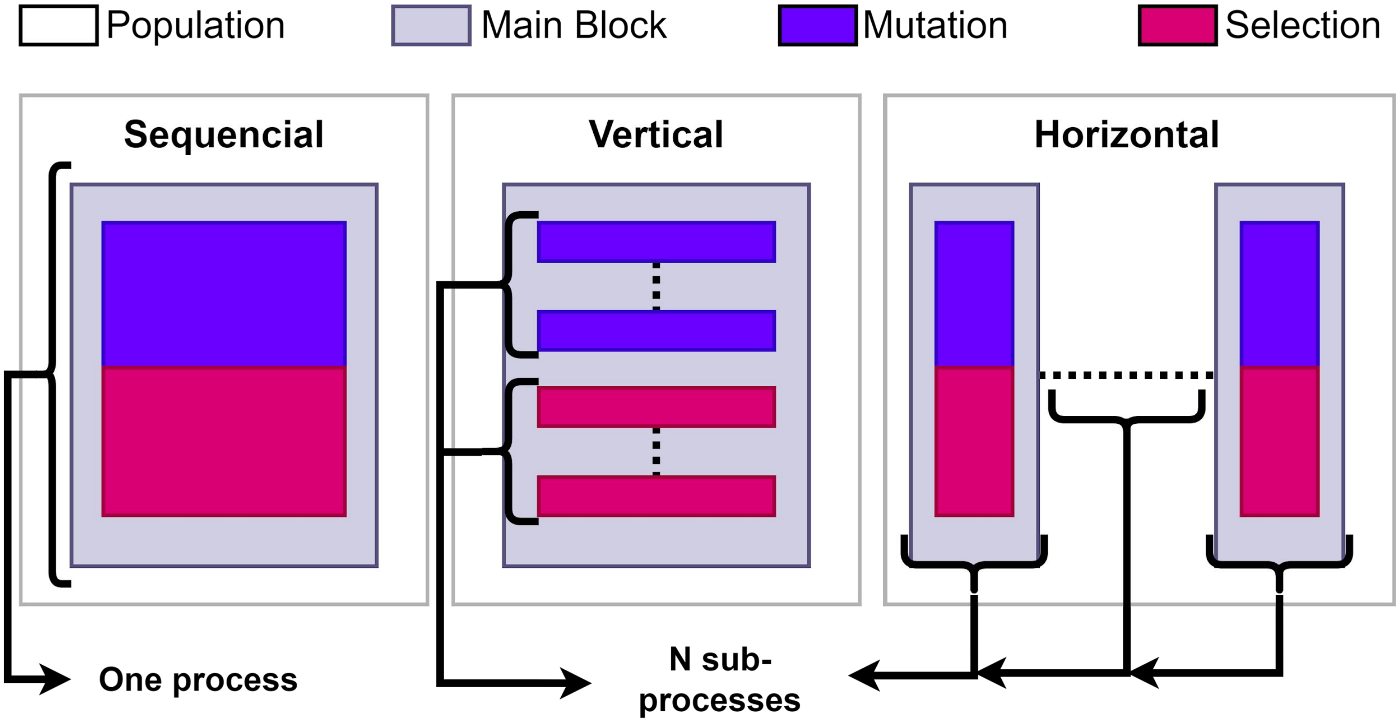
Step 7:   **for i in i to Np:**

Step 8:   Bound(U)

Step 9:   Evaluate fitness function in all individuals of Population (X)

Step 10:   Greedy Selection (U_i, X_i)

Differential Evolution Approach



Stochastic Petri Network

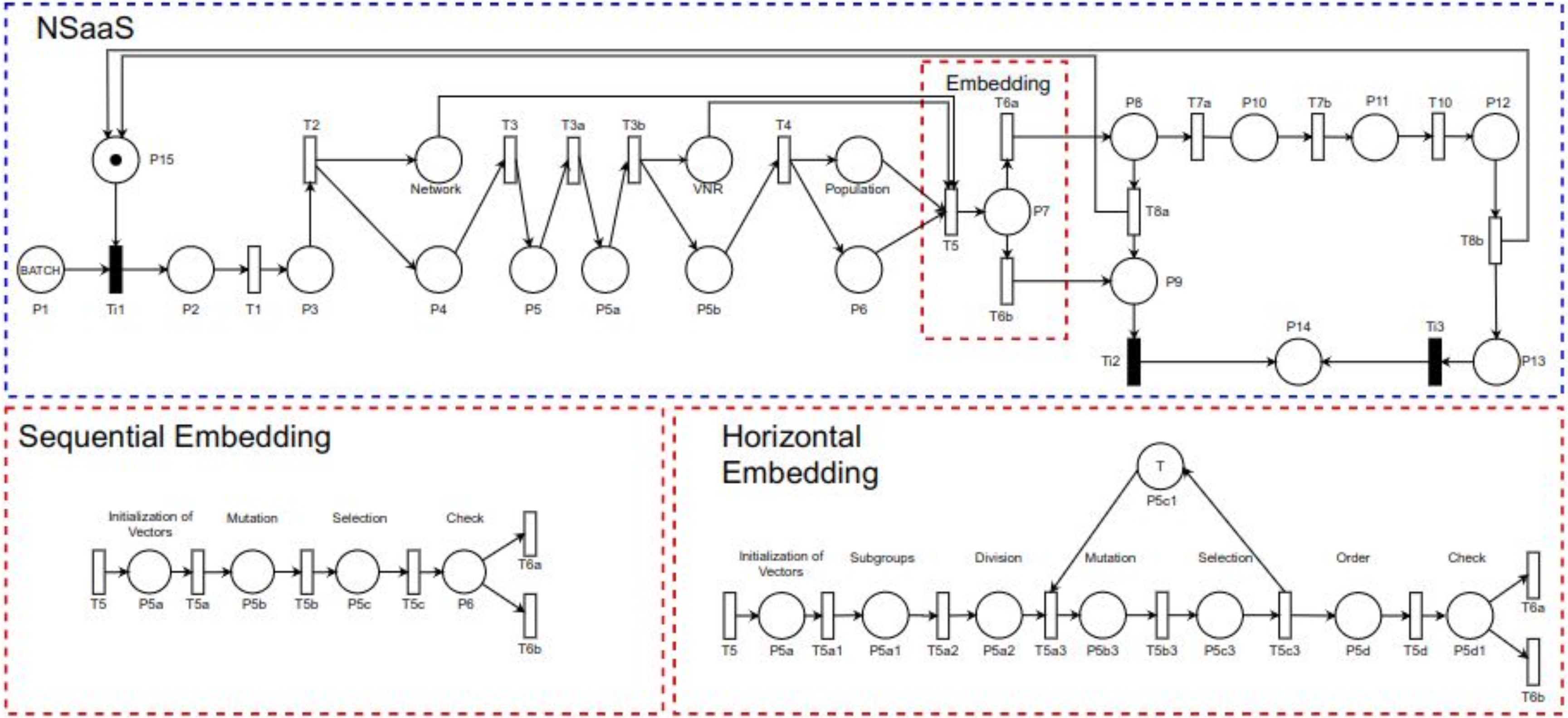


Figure 2: NSaaS (Base), Sequential, Vertical and Horizontal Embedding models.

Stochastic Petri Network

TABLE I: Places and transitions denotations

Name	Denotation
P1	The repository of tokens, each token means a VNR.
Ti1	The immediate transition represents the entry of the request into the system. All transitions are immediate, and there is no delay associated with its execution.
T1	The preparation of the system to cope with the requisition.
T2	System obtaining infrastructure status.
Network	This place means the network substrate.
VNR	The repository of VNRs.
T3	Decoding the VNR.
T4	Creating the first population used in the VNE approaches.
Population	The place means the population element used in the approaches.
T5	Fitness calculation of each token.
P5[a,b,b,d]	There are specific places present in each approach. These places represent positions or steps used in each DE version (Sequential, Vertical e Horizontal).
T7a	They represent the flow of message processing after going through the mapping process. In this flow, the resources are allocated to the mapped VNR.
T7b	Similar to T7a, but in this flow, the resources are not allocated to the mapped VNR.
T8	Resources release to each VNR expired.
P13	Synchronization position, token means the system is ready to deal with another VNR.
T5a	Initialization of DE structures, trial, donor and target vectors.
T5b	Mutation of chromosomes.
T5c	Selection of individuals.
T6a	Checking if VNR can be meet.
T6b	Checking if VNR cannot be meet.
T5a1	Population division.
T5b[1-5]	Parallel mutation in each individual.
T5c[1-5]	Parallel selection in each individual.
T5d	Reordering individuals in the population when Vertical Embedding case.
T5d	Grouping sub-population in a new population when Horizontal Embedding case.
T5a2	Creation of population sub-groups.
T5a3	Creation of processes to deal with each sub-group of population.

Evaluations: Simulation Properties

Artifacts and Scenarios for Evaluation

- Two datasets were created, following the same process described in [5].
- **Dataset 1**: Aggregation of Europe datasets from http://www.topology-zoo.org/
- **Dataset 2**: Aggregation of US datasets from http://www.topology-zoo.org/

TABLE II: Dataset properties used in the simulation.

Properties	Dataset 1	Dataset 2
Nodes	112	2,138
Links	125	2,395
Degree	12	50
Bandwidth	[103;1,640;9,881;2,807]	[100;1,215;9,988;2,338]
Delay	[1;121;294;103]	[1;113;300;98]
Reliability	[90;95.7;99;3]	[90;95;99;3]
Values = [minimal; average; maximum; standard deviation]		

Evaluations: Simulation Properties

Simulation Execution

- We created four sets of VNRs and the number of requisitions is: (i) **set 1** has 20; (ii) **set 2** has 50; (iii) **set 3** has 100; and (iv) **set 4** has 150.
- Each algorithm was carried out 10 times with 9 tuples of [repetition, population size]: (5, 25), (50, 25), (100, 25), (5, 50), (50, 50), (100, 50), (5, 100), (50, 100), (100, 100).
- Each set is kept the same for each different mapping algorithm.
- One VNR is composed of:
  - *VNR identification,*
  - *Virtual nodes demands (vnd),*
  - *Links*
  - *Type of slice*
  - *Bandwidth demand*
  - *Delay demand*
  - *Reliability demand*

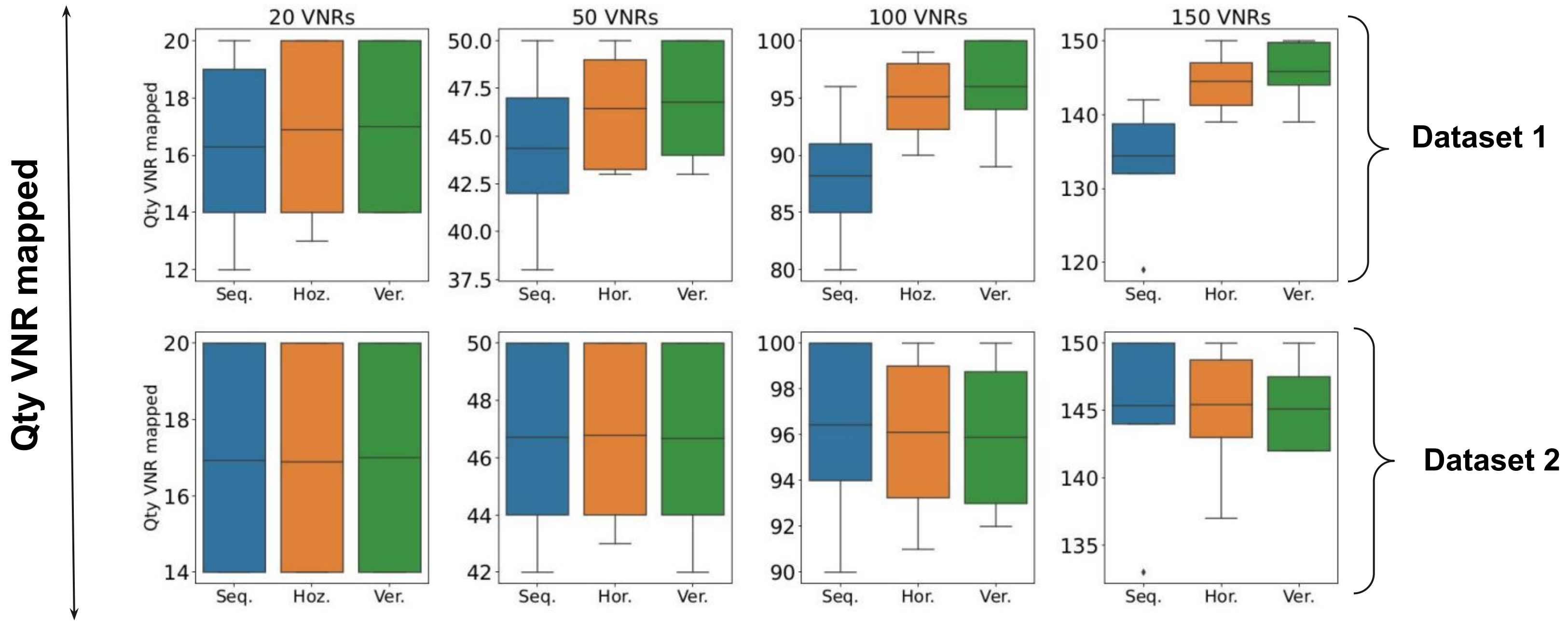
Evaluations: Simulation Properties

Sequential

Horizontal

Vertical

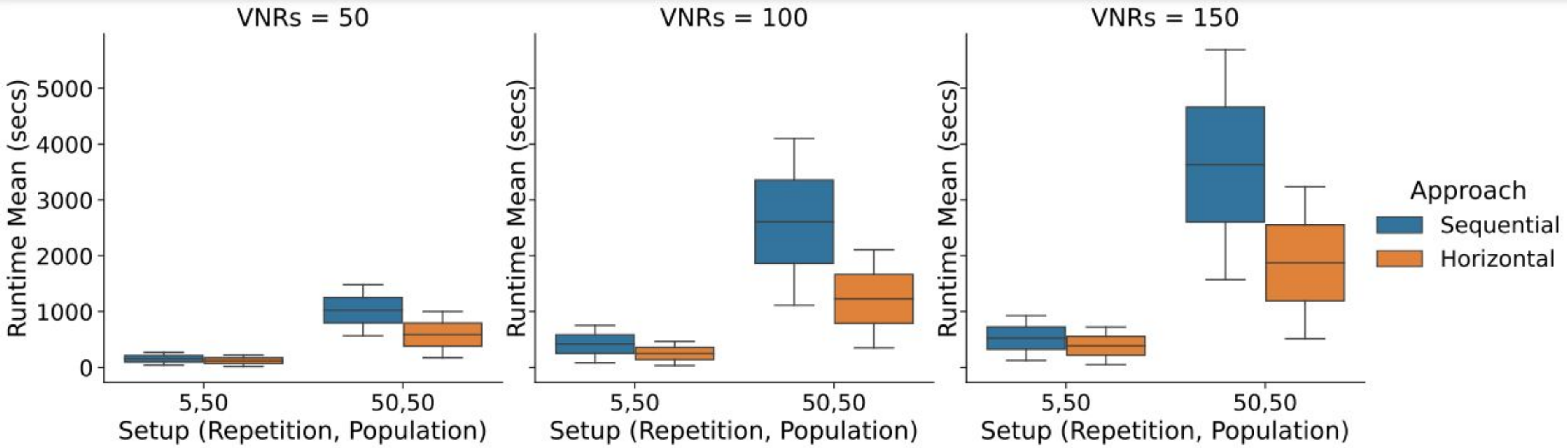
First Evaluation: Quantity of VNR mapped, that is, Acceptance Rate



Evaluations: Simulation Properties

First Evaluation: Quantity of VNR mapped, that is, Acceptance Rate

We are keeping the same population and changing the repetition.

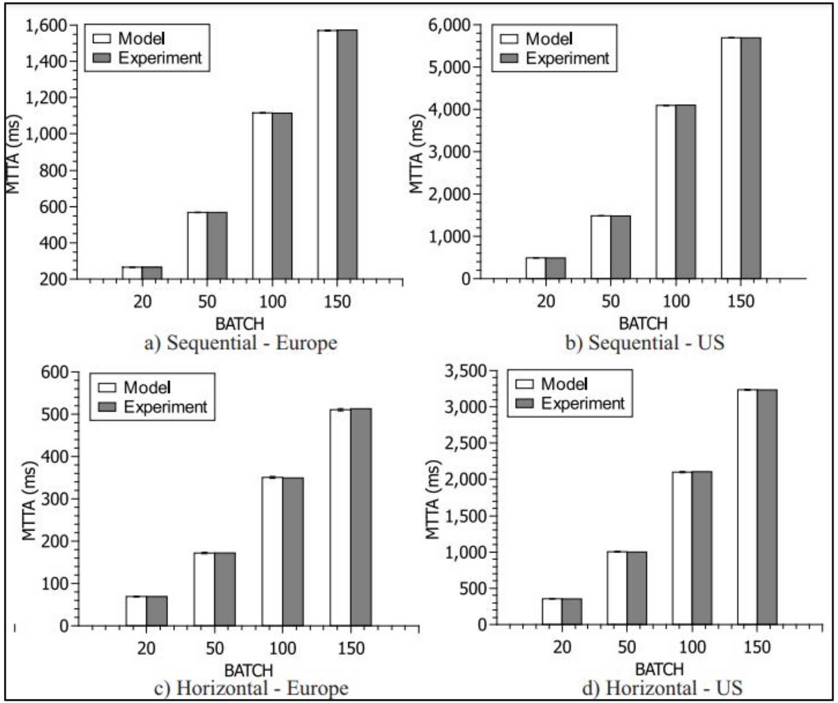


The Horizontal reaches 59% less time than the Sequential Version.

Difference between Model and Experiment runtime.

- The model is an essential artifact to compute the Mean Time to Absorption (MTTA) [11].
- We used the Mercury Tool, version 5.0.1 [12] to design the Petri Net model, and through it, it was possible to carry out the data analysis.
- For the validation process, the **T-Test** was performed for two samples where the MTTAs values found in the simulations and experiments were compared.

The P-Values are greater than 0.05, implying that there is no evidence that both means are statistically different.



Conclusion

- This work evaluated three versions of the Differential Evolution Meta-Heuristic (DE) applied to the VNE problem.
- The premise to adapt the DE in VNE is because of its efficiency and nature that favor its adaptation to parallel versions.
- The Horizontal version achieved the lower runtimes.
- Finally, the Stochastic Petri Net allows us to conduct evaluations with approximated results to validate a more extensible quantity of virtual network requests.

References

[1] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, “Network Slicing for 5G: Challenges and Opportunities,” IEEE Internet Comput., vol. 21, no. 5, pp. 20–27, 2017.

[2] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos,

“The Algorithmic Aspects of Network Slicing,” IEEE Communications Magazine, vol. 55, no. 8, pp. 112–119, 2017.

[3] H. Wu, F. Zhou, Y. Chen, and R. Zhang, “On Virtual Network Embedding: Paths and Cycles,” IEEE Transactions on Network and Service Management, vol. 4537, no. c, pp. 1–14, 2020.

[4] H. Cao, L. Yang, Z. Liu, and M. Wu, “Exact solutions of VNE: A survey,” China Communications, vol. 13, no. 6, pp. 48–62, 2016.

[5] R. Gomes, D. Vieira, and M. Franklin de Castro, “Differential evolution for vne-5g scenarios,” in 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2021, pp. 1–6.

[6] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, “Virtual network embedding: A survey,” IEEE Communications Surveys and Tutorials, vol. 15, no. 4, 2013.

[7] K. T. Nguyen and C. Huang, “An intelligent parallel algorithm for online virtual network embedding,” in 2019 International Conference on Computer, Information and Telecommunication Systems (CITS). IEEE, 2019, pp. 1–5.

[8] K. Nguyen and C. Huang, “Distributed parallel genetic algorithm for online virtual network embedding,” International Journal of Communication Systems, 12 2020.

References

[9] K. Nguyen, Q. Lu, and C. Huang, “Efficient virtual network embedding with node ranking and intelligent link mapping,” in 2020 IEEE 9th International Conference on Cloud Networking

(CloudNet). IEEE, 2020, pp. 1–5.

[10] K. Salimifard and S. Bigharaz, “The multicommodity network flow problem: state of the art classification, applications, and solution methods,” Operational Research, pp. 1–47, 2020.

[11] F. A. Silva, S. Kosta, M. Rodrigues, D. Oliveira, T. Maciel, A. Mei, and P. Maciel, “Mobile cloud performance evaluation using stochastic models,” IEEE Transactions on Mobile Computing, vol. 17, no. 5, pp. 1134–1147, 2017.

[12] A. Lobo, R. Matos, B. Silva, and P. Maciel, “Expolynomial modelling for supporting vanet infrastructure planning,” in 2017 IEEE 22nd Pacific rim international symposium on dependable computing (PRDC). IEEE, 2017, pp. 86–91.

[end]

Thank you

"Parallel Differential Evolution Meta-Heuristics and Modeling for Network Slicing in 5G Scenarios"

Rayner Gomes
rayner@ufpi.edu.br

... Special thanks ...

IARIA
ICNS 2022

UFPI

UFC

efrei PARIS