VNF-PR in B5G Networks: A Heuristic Approach to Minimize Latency and Energy Consumption

Matheus Pantoja, Albert Santos, Rafael Vieira, Carlos Natalino, Diego Cardoso

***Abstract*— This paper proposes the Efficient Node Activation Heuristic (ENAH) to solve the Virtual Network Function Placement and Routing Problem (VNF-PR) in B5G networks. ENAH effectively manages the On/Off state of servers, activating only the necessary servers before allocating Virtual Network Functions (VNFs). This strategic activation minimizes both latency and energy consumption while maintaining high Quality of Service (QoS). Extensive simulations in high-demand scenarios demonstrated that ENAH significantly reduces latency by up to 29% and energy consumption by up to 37%. These improvements contribute substantially to the efficiency, sustainability and performance of modern telecommunication networks.**

***Keywords*— VNF, NFV, Service Function Chain, VNF Placement and Routing, B5G**

I. Introduction

In the current telecommunications scenario, driven by advances in cloud computing and virtualization technology, network service providers have increasingly adopted the implementation of Virtual Network Functions (VNFs) [1]. VNFs are software components that can be optimally incorporated into general computing platforms. This approach, by avoiding the need for dedicated and specialized hardware, not only reduces operational costs but also increases flexibility and agility in deploying new network services [2]. However, the diversity of available services, each with its own requirements in terms of resources represents a significant challenge. These services are provided as an ordered sequence of VNFs, forming a service function chaining (SFC) [3]. Thus, the effective deployment and resource allocation for these VNF chains represent a complex yet crucial challenge that needs to be overcome [4].

Network Function Virtualization (NFV) is a well-established concept proposed by the European Telecommunications Standards Institute (ETSI) to simplify the deployment and management of network services using virtualization technologies [4]. Compared to traditional hardware appliances, NFV allows service deployment on virtual machines to enhance flexibility and scalability. The basic idea behind NFV is to decouple network functions, such as firewall, proxy, or intrusion detection systems from dedicated hardware, allowing them to run as software [3]. This approach enables network functions to be implemented as dynamic applications, activated according to specific demands and locations. This technology facilitates efficient traffic management, allowing computational scalability and high-precision connection management [2].

The energy consumed in data centers is a growing concern, representing a significant portion of operational costs, with forecasts that it may exceed capital expenditures (CAPEX) in the near future [1]. In the context of NFV and VNF, energy consumption impacts the overall efficiency and sustainability of network operations. High energy consumption not only increases operational costs but can also affect the Quality of Service (QoS) by limiting the resources available for network functions. In NFV-enabled networks, computing capacity is designed for traffic peaks, making it unnecessary to activate all servers during low traffic periods [5]. Controlling the number of active nodes is crucial for efficient resource utilization and reducing operational costs while ensuring low latency to guarantee QoS and end-user satisfaction [6]. In NFV-enabled networks, latency can be influenced by the location of servers, and an efficient solution for VNF placement on servers, considering the selection of the most suitable servers to be activated, can significantly minimize latency and improve service quality.

Considering network resource constraints, a critical factor for NFV success is the efficient placement of VNFs and the cost-effective routing of service demands [3]. This study focuses on the VNF Placement and Routing Problem (VNF-PR), which aims to optimize the activation of the necessary servers and subsequently the placement of VNFs to use resources effectively. The problem consists of finding (i) the optimal placement of VNFs on nodes and (ii) feasible routing paths, respecting node latency and capacity constraints [2].

The main motivation of this work is to develop an efficient approach for controlling the activation of network nodes that minimizes latency and energy consumption while maintaining QoS and network availability, especially in B5G networks. The primary contribution is the proposal of a heuristic that improves resource utilization by selectively activating network nodes based on demand, thereby reducing operational costs and service demand latency. As a consequence, this selective activation leads to an optimized VNF allocation. To validate the approach, we conducted tests and analyzed two classic objectives: energy consumption and QoS (minimization of total latency). The remainder of the paper is structured as follows: Section II presents a review of related work. Section III defines the VNF-PR. The heuristic is introduced in Section IV, followed by the discussion of results in Section V. Finally, Section VI concludes this work.

II. RELATED WORK

The strategy of turning servers on and off to optimize energy efficiency is widely accepted as an effective solution. The

Matheus Pantoja, UFPA, Belém-PA, e-mail: matheus.pantoja@itec.ufpa.br; Albert Santos, UFPA, Belém-PA, e-mail: albert.santos@itec.ufpa.br; Rafael Vieira, UEAP, Macapá-AP, e-mail: rafael.vieira@ueap.edu.br; Carlos Natalino, Chalmersa, Suécia, e-mail: carlos.natalino@chalmers.se; Diego Cardoso, UFPA, Belém-PA, e-mail: diego@ufpa.br

main difference between approaches lies in how to determine which servers should be activated or deactivated, which can significantly impact energy savings and network efficiency.

The solution proposed in [7] analyzed scheduling policies for NFV in B5G networks, considering server reliability and boot time. A Markov chain model was proposed to account for the on/off states of servers and their impact on system reliability, along with an algorithm to set activation/deactivation thresholds aimed at optimizing energy consumption while ensuring the desired reliability. The results showed that the proposed approach can significantly improve energy efficiency while maintaining system reliability.

The architecture proposed in [6] for B5G networks integrates holistic network virtualization and pervasive network intelligence. This architecture includes network slicing management and the control of on/off states of nodes and links in software-defined networks (SDN) to save energy. Resource virtualization and network intelligence are addressed together to optimize network performance, including energy efficiency and resource allocation. The results indicate that the integrated approach can lead to significant energy savings and better utilization of network resources.

The approach in [5] for energy efficiency in different modes of SDN/NFV networks, including partial and hybrid networks for B5G, proposes a general system model and a multi-stage graph-based algorithm to optimize energy consumption. The algorithm considers the routing and placement of network functions as well as the on/off states of nodes and links to maximize energy efficiency during the transition to fully virtualized networks. Simulation results showed a significant reduction in energy consumption, highlighting the effectiveness of the proposed approach for transitional networks.

The solution proposed in [8] optimizes the embedding and scheduling of SFC in B5G networks with NFV. The approach selects physical nodes with abundant resource capacities to accommodate the SFCs, linking the necessary nodes and optimizing the total scheduling time of the SFC. This process involves the dynamic activation and deactivation of servers to ensure efficient refsource use. Experimental results show an SFC acceptance rate at least 10% higher than typical heuristics, highlighting its efficiency in dynamic server management and resource optimization.

As can be observed, most existing approaches do not consider the joint optimization of latency and energy consumption, nor the activation of nodes based on demand. Additionally, these works propose specific algorithms that may not be easily scalable for large-scale networks due to their complexity. This work aims to fill these gaps by proposing an efficient and scalable heuristic for VNF-PR in B5G networks, which will be detailed in the following sections.

III. PROBLEM DEFINITION

The VNF-PR involves finding the placement of virtual network functions on nodes and the routing paths so that the sum of the network function installation costs and node activation costs is minimized. Solutions must consider node and network function capacity constraints, as well as flow precedence and routing [2]. A solution to the problem is considered feasible if it satisfies all required constraints and is called optimal if the associated cost is minimized. The associated cost is defined according to each specific objective, which can be the sum of node activation costs, energy consumption, or even the total system latency. The problem is NP-Hard, even for a single flow and without additional constraints [3].

A. System Model

The telecommunications network is modeled as a bidirectional graph $G(N, L)$, where N represents the set of nodes (servers or switches) and L is the set of links interconnecting these nodes. Each node $i \in N$ can be equipped with computational resources, such as CPU, memory, and storage, necessary to host VNFs. The links L are characterized by bandwidth capacity and latency.

Client demands are modeled as a tuple $d(d_o, d_d, T_d, B_d, c_d)$, where d_o is the demand's origin node, d_d is the destination node, T_d is the allowable end-to-end latency, B_d is the required bandwidth, and c_d is the ordered sequence of VNFs that the demand must traverse. Each VNF $v \in c_d$ must be allocated on a network node, and the links between nodes must be selected to minimize total latency.

B. Energy Consumption Model

To model energy consumption in the network, we consider switches and servers (PMs). The energy consumption of switches has a constant component and a variable component related to the number of active ports [9]. Thus, the total energy consumed by a switch is:

$$P_{sw} = P_{const} + P_{port} \cdot N_{port} \tag{1}$$

where P_{const} is the idle state energy consumption and P_{port} is the energy consumed by each active port. N_{port} represents the total number of active ports. If a physical link $l_{i,j} \in L$ between nodes i and j is active, it requires two ports. Thus, the total network energy consumption is given by:

$$P^{T}_{network} = P_{const} \sum_{i \in N} y_i + 2P_{port} \sum_{(i,j) \in L} l_{i,j} \tag{2}$$

Here, $l_{i,j}$ is 1 if the link between i and j is active, and y_i indicates if switch i is active.

In data centers, the energy consumed by computing nodes mainly depends on the CPU, which is the largest energy consumer. Therefore, the energy consumption model for PMs considers CPU utilization. The energy consumption of a PM is proportional to CPU load, varying from a minimum consumption in idle state to a maximum at full load [10]:

$$P_{pm} = P_{idle} + (P_{max} - P_{idle}) \times \theta_{cpu} \tag{3}$$

where P_{idle} is the energy consumed when the CPU is at 0

$$\theta_{cpu} = \frac{C_{util}}{C_{total}} \tag{4}$$

Considering C_{util} as the utilized resource, the total energy consumption of a PM, P^{T}_{pm}, is described as:

$$P_{pm}^{T} = \sum_{i \in N} \Bigg(P_{idle}\, x_i + (P_{max} - P_{idle}) \sum_{f \in F} \sum_{C_{util} \in R} \frac{C_{f,C_{util}}}{C_{i,C_{util}}^{T}}\, z_{i,f} \Bigg) \tag{5}$$

where x_i is a binary variable that indicates if the PM is active, and $z_{i,f}$ is an integer variable representing the number of instances of function f allocated to the PM at node i.

C. System Latency

Eq. 6 below calculates the total system latency through two components: (i) the total delay introduced to all services by VNF processing, and (ii) the total delay due to data transmission between network nodes. The overall system latency is defined as:

$$T_{total} = T_{proc} + T_{trans} \tag{6}$$

T_{proc} represents the total processing latency of the system and is expressed as:

$$T_{proc} = \sum_{v \in V} \sum_{i \in N} \sum_{d \in D} u_{i,v,d} \cdot T_v \tag{7}$$

where $u_{i,v,d}$ is a binary variable that indicates if demand d is being processed by VNF v allocated at node i, and T_v represents the processing latency of VNF v.

T_{trans} represents the total transmission latency of the system and is expressed as:

$$T_{trans} = \sum_{(i,j) \in L} \sum_{(k,l) \in F} \sum_{d \in D} w_{i,j,k,l,d} \cdot T_{i,j} \tag{8}$$

where $w_{i,j,k,l,d}$ is a binary variable that indicates if virtual link (k,l) of demand d uses physical link (i,j), and $T_{i,j}$ represents the transmission latency of physical link (i,j).

IV. HEURISTIC FORMULATION

We consider that all servers have the same processing capacity. In a scenario where all servers are turned on, demand allocation becomes less complex, as any available server can meet the requests.

The Efficient Node Activation Heuristic (ENAH) aims to minimize latency by activating servers based on network topology and current demand. By iteratively applying these steps, ENAH efficiently activates servers based on evolving network demands, ensuring efficient resource use and minimizing latency. This results in more efficient demand allocation, directing traffic to servers based on the shortest total distance. ENAH operates in two distinct scenarios:

1. All Nodes Off: When all servers are initially turned off, the algorithm calculates the sum of distances from each node to all other nodes in the network. If a link between two nodes is already active, the Euclidean distance is used. If the link is inactive, the distance is multiplied by a penalty factor (in this case, 2) to discourage the use of links that would require activation. The node with the shortest distance, which in this scenario equals the smallest sum of distances, is then activated. The idea in this scenario is to identify and activate the most central node in the network, which has the shortest accumulated distance to all other nodes, thereby minimizing the average network latency.

Algorithm 1 Efficient Node Activation Heuristic (ENAH)

```
Input: Graph $graph$, Set of Nodes $N$, Set of Links $l$, Distance Vector $D$ (initialized to zeros and of size equal to the number of nodes in the graph)
Output: Index of the node to be activated
if all nodes are off then
    for each node $n \in N$ do
        for each node $i \in N$ do
            if the link between $n$ and $i$ is on then
                $D[n] \leftarrow D[n] +$ euclidean distance$(n, i)$
            else
                $D[n] \leftarrow D[n] +$euclidean distance$(n, i) \times 2$ ▷ Assign a higher weight to unconnected links
            end if
        end for
    end for
    Node $n$ with the smallest sum is turned on
else    ▷ Calculate the distances of nodes relative to the connected nodes
    Initialize the list of connected nodes: ConnectedNodes ← []
    for each node $n \in N$ do
        if $n$.on = True then
            Add $n$ to the list ConnectedNodes
        end if
    end for
    for each node $n \in N$ do
        for each node $i \in$ ConnectedNodes do
            if the link between $n$ and $i$ is on then
                $D[n] \leftarrow D[n] +$ euclidean distance$(n, i)$
            else
                $D[n] \leftarrow D[n] +$euclidean distance$(n, i) \times 2$ ▷ Assign a higher weight to unconnected links
            end if
        end for
    end for
    Node with the smallest distance is turned on
end if
```

2. Partially Connected Nodes: When some servers are already on, ENAH focuses on the nodes that are not yet active. It calculates the sum of distances from each inactive node to the nodes that are already active. If a link between an inactive node and an active node is already active, the Euclidean distance is used. If the link is inactive, the distance is multiplied by a penalty factor (in this case, 2) to discourage the use of links that would require activation. The inactive node with the smallest sum of distances to the active nodes is chosen to be activated. In this scenario, the idea is to prioritize nodes that are already connected or closer to connected nodes, taking into account the state of the links to minimize the total path length. Here, the distance is influenced by the state of

Service Type	VNF Chain	Bandwidth	Delay	Traffic %
Web Service	NAT-FW-TM-WOC-IDPS	100 kbps	500 ms	18.2%
VoIP	NAT-FW-TM-FW-NAT	64 kbps	100 ms	11.8%
Video Streaming	NAT-FW-TM-VOC-IDPS	4 Mbps	100 ms	69.9%
Online Games	NAT-FW-VOC-WOC-IDPS	50 kbps	60 ms	0.1%

TABLE I

CONSIDERED SERVICES [11]

the node, not just by its smallest sum.

After selecting the server node, the routing of the service flow is carried out efficiently, activating the links between the demand's origin node and the chosen server node by the heuristic. This ensures that the activated nodes minimize latency, considering both the distances between the nodes and the current state of the network links.

V. RESULTS

In this section, we analyze and compare the proposed heuristic with two different approaches: the Random approach, where nodes are activated randomly, and the All-On approach, where all nodes are always on. The comparison will be made in terms of latency and energy consumption. The heuristic was implemented in Python 3.8.17, and the simulations were run on a machine equipped with an Intel® Core(TM) i7-7700 @ 3.60GHz, 16GB of RAM, running Linux Mint. The experiments were repeated 30 times for different workloads, and the average results are presented.

We define service flows as video streaming (VS), web services (WS), voice over IP (VoIP), and online gaming (OG), as shown in Table I. Each service requires a specific set of functions and bandwidth [11].

For the tests, we considered six types of VNFs: Network Address Translation (NAT), Firewall (FW), Traffic Monitor (TM), WAN Optimization Controller (WOC), Video Optimization Controller (VOC), and Intrusion Detection System (IDS). Different sets of demands were generated and distributed between origin and destination nodes. Each demand was assigned to a service type based on the traffic percentage presented in Table I. We assumed that all VNFs have a processing capacity of 200 Mb/s (Megabits per second) and a processing time of 10 ms [11].

The tests were performed using the DFN topology from the SNDLib with 10 nodes and 45 links [12]. The capacity of each physical link was set to 1 GB/s, and the propagation delay of optical fiber is approximately 10ms/km [11]. We tested five different scenarios, varying the number of demands per node from 8 to 40, allowing a detailed analysis of the heuristic's behavior under different workloads.

Figure 1 shows the system's energy consumption profile under different server demand scenarios. The ENAH and Random approaches demonstrate almost identical energy consumption at all demand levels, as evidenced by the overlapping shaded areas representing standard deviation. This proximity is due to both approaches using the same number of servers and different amounts of active links. Most energy is consumed by the servers, making the demand distribution almost irrelevant in terms of energy consumed.

On the other hand, the All-On approach, where all servers are kept active, shows significantly higher consumption, highlighting the inefficiency of this approach and the importance of selectively activating servers to optimize energy consumption. For the scenario with 8 demands per server, ENAH shows approximately 37.2% savings compared to the All-On approach. For 16 demands per server, ENAH shows a reduction of about 29.72% in energy consumption compared to the All-On approach. For 24 demands per server, ENAH reduces consumption by about 19.3% compared to the All-On approach. Therefore, the ENAH heuristic proves efficient in optimizing energy consumption, providing significant reductions compared to the All-On approach. The results indicate that ENAH is robust and effective, maintaining consistent energy consumption even with varying demands. This reduction in energy consumption aligns with one of the key objectives of the VNF-PR, which is to minimize energy usage while ensuring network performance. Such improvements are crucial for data centers and other environments where energy efficiency is essential, resulting in lower operational costs.

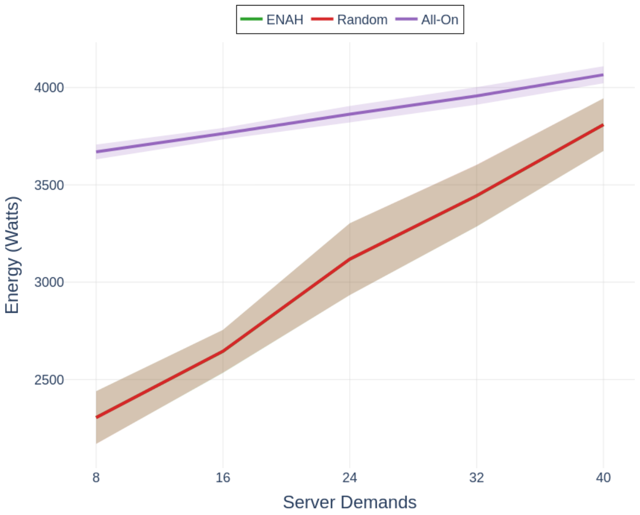

Fig. 1. Energy

Figure 2 illustrates the system's latency under different server demand scenarios. The results demonstrate the need for a solution to determine which nodes to activate, as the "Random" approach, which activates servers randomly, shows higher latencies, especially in low to medium demand scenarios. For the scenario with 8 demands per node, ENAH presents a reduction of approximately 29.38% compared to the Random approach. For the scenario with 16 demands per node, ENAH presents a reduction of approximately 25.62% compared to the Random approach. For 32 demands per node, ENAH's latency is reduced by about 23.8% compared to the Random approach. The All-On approach shows slightly lower latencies than ENAH because it keeps all servers active, allowing allocation to occur on servers closer to the demand origin. Analyzing the shaded areas in the graph, which represent the standard deviation of measurements, shows that the ENAH heuristic not only reduces average latency but also presents lower variability in results, suggesting greater consistency. This reduction in latency aligns with one of the key objectives of the VNF-PR, which is to minimize system latency while ensuring network performance. This is crucial for time-sensitive applications such as online gaming and VoIP, where perceptible delays can significantly compromise service quality.

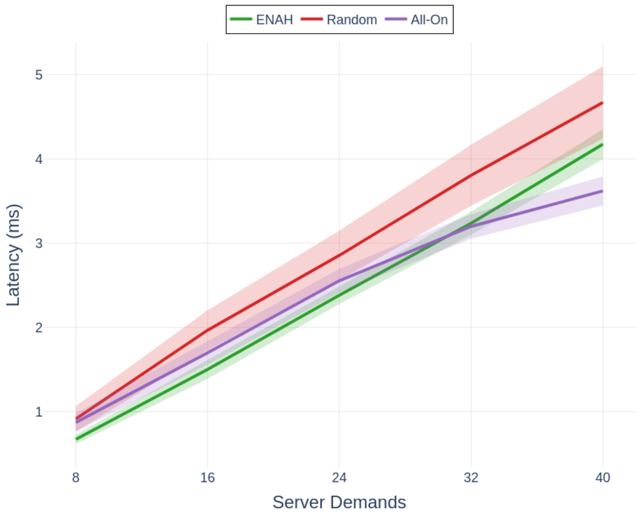

Fig. 2. Latency

VI. Conclusion

In this work, we addressed the VNF-PR problem, focusing on its specific objectives of minimizing energy consumption and system latency while maintaining QoS. We proposed the ENAH heuristic to optimize server activation in virtualized network environments. ENAH proved to be an effective and robust solution, excelling in both latency reduction and energy consumption. Compared to alternative approaches, the heuristic reduced average latency by up to 29% and energy consumption by up to 37% in high-demand scenarios. The main results indicate that ENAH manages the state of server activation, turning on only the necessary servers, resulting in an optimal allocation of VNFs as a consequence. This balance between energy efficiency and performance is crucial for B5G networks, especially for latency-sensitive applications such as online gaming and VoIP.

For future work, we suggest investigating the implementation of scenarios with servers of different processing capacities and the inclusion of multiple servers per node. Another promising direction is to combine the ENAH heuristic with other solutions specifically focused on demand routing. Additionally, exploring reinforcement learning techniques could enable the creation of a model that learns from network behavior, orchestrating resources more automatically and efficiently.

Acknowledgements

This work was partially supported by Fapespa and iSACI.

References

[1] C. R. De Mendoza, B. Bakhshi, E. Zeydan, and J. Mangues-Bafalluy, "Near optimal vnf placement in edge-enabled 6g networks," in *2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*. IEEE, 2022, pp. 136–140.

[2] A. Mouaci, É. Gourdin, I. LjubiĆ, and N. Perrot, "Virtual network functions placement and routing problem: Path formulation," in *2020 IFIP Networking Conference (Networking)*. IEEE, 2020, pp. 55–63.

[3] J. Li, X. Qi, J. Li, Z. Su, Y. Su, and L. Liu, "Fault diagnosis in the network function virtualization: A survey, taxonomy and future directions," *IEEE Internet of Things Journal*, 2024.

[4] A. Mouaci, É. Gourdin, I. Ljubić, and N. Perrot, "Two extended formulations for the virtual network function placement and routing problem," *Networks*, 2023.

[5] R. Moosavi, S. Parsaeefard, M. A. Maddah-Ali, V. Shah-Mansouri, B. H. Khalaj, and M. Bennis, "Energy efficiency through joint routing and function placement in different modes of sdn/nfv networks," *Computer Networks*, vol. 200, p. 108492, 2021.

[6] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6g," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 1–30, 2021.

[7] J. Ortin, P. Serrano, J. Garcia-Reinoso, and A. Banchs, "Analysis of scaling policies for nfv providing 5g/6g reliability levels with fallible servers," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1287–1305, 2022.

[8] H. Cao, J. Du, H. Zhao, D. X. Luo, N. Kumar, L. Yang, and F. R. Yu, "Resource-ability assisted service function chain embedding and scheduling for 6g networks with virtualization," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3846–3859, 2021.

[9] W. Wu, Y. Li, L. Chen, B. Zhang, W. Wang, Y. Zhao, and J. Zhang, "Service function chain mapping based on joint load balancing in computing power network," in *2023 Opto-Electronics and Communications Conference (OECC)*. IEEE, 2023, pp. 1–4.

[10] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[11] A. Varasteh, M. De Andrade, C. M. Machuca, L. Wosinska, and W. Kellerer, "Power-aware virtual network function placement and routing using an abstraction technique," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[12] Zuse Institute Berlin, "Sndlib: The sndlib network optimization library," http://sndlib.zib.de/home.action, 2024, accessed: 2024-05-24.