

Modelagem Unificada para Posicionamento e Roteamento de VNFs e Funções de Acesso em Redes O-RAN Energicamente Eficientes

Miguel Santos, Matheus Pantoja, Carlos Costa, Rafael Vieira, Diego Cardoso.

Resumo—Este artigo propõe um modelo unificado para o posicionamento e roteamento de Funções de Rede Virtuais (VNFs) em arquiteturas O-RAN, com o objetivo de minimizar o consumo energético da infraestrutura. O modelo integra funções de acesso desagregadas (vO-DU e vO-CU) com VNFs de serviço em uma única formulação, respeitando restrições de latência de *fronthaul*, isolamento de hardware e capacidade de enlaces. Uma heurística gulosa do tipo *Nearest-Node First* é proposta e avaliada contra uma alocação aleatória em duas topologias de rede. Os resultados demonstram que a abordagem gulosa alcança maior taxa de aceitação e menor latência ao consolidar cargas em nós próximos, resultando em melhor eficiência energética por demanda atendida.

Palavras-Chave—O-RAN, VNF Placement, eficiência energética, *greedy*, NFV.

Abstract—This paper proposes a unified model for the placement and routing of Virtual Network Functions (VNFs) in O-RAN architectures, aiming to minimize the infrastructure energy consumption. The model integrates disaggregated access functions (vO-DU and vO-CU) with service VNFs into a single formulation while respecting fronthaul latency constraints, hardware isolation requirements, and link capacity limitations. A greedy heuristic based on the *Nearest-Node First* approach is proposed and evaluated against a random allocation strategy in two network topologies. The results demonstrate that the greedy approach achieves a higher acceptance rate and lower latency by consolidating workloads on nearby nodes, resulting in better energy efficiency per served demand.

Keywords—O-RAN, VNF Placement, energy efficiency, *greedy*, NFV.

I. INTRODUÇÃO

A evolução para as redes 5G e 6G impulsiona a Virtualização de Funções de Rede (NFV) para suportar serviços heterogêneos com requisitos rigorosos [1]. Nesse contexto, o tráfego deve atravessar sequências ordenadas de Funções Virtuais (VNFs), formando *Service Function Chains* (SFCs), cujo problema de alocação e roteamento, conhecido como VNFP-RP, é NP-difícil [1].

Paralelamente, a arquitetura *Open Radio Access Network* (O-RAN) introduz flexibilidade à rede de acesso por meio da desagregação da estação base em O-RU, O-DU e O-CU [2], [3]. Isso permite virtualizar funções vO-DU e vO-CU e alocá-las dinamicamente em ambientes de borda e nuvem regional [2].

Apesar dos avanços, a literatura normalmente trata o VNFP-RP e a alocação em O-RAN de forma separada, resultando em

uso ineficiente da infraestrutura e maior consumo energético [4]. Para enfrentar essa limitação e evitar a alta complexidade de modelos exatos (ILP) [5], este artigo propõe um modelo unificado de orquestração baseado em uma heurística gulosa (*greedy search*), visando otimizar o posicionamento e roteamento de VNFs em arquiteturas O-RAN, consolidando cargas em poucos nós para reduzir o consumo energético total [6] e respeitando restrições rigorosas de latência.

II. MODELO DO SISTEMA

A. Rede de Transporte

A infraestrutura de rede é modelada como um grafo não-direcionado $G = (\mathcal{N}, \mathcal{L})$, onde $\mathcal{N}$ é o conjunto de nós computacionais e $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ o conjunto de enlaces físicos. Os nós são classificados em três *tiers* hierárquicos conforme a arquitetura O-RAN [2]: *Far Edge*, com CPU limitada e forte restrição de latência de *fronthaul*; *Edge*, com recursos computacionais moderados; e *Core*, representando a nuvem regional com elevada capacidade. Cada nó $n \in \mathcal{N}$ possui capacidade de CPU C_n (em *cores*), potência máxima $P_n^{\max}$ (em Watts) e limite de instâncias vDU $V_n^{\max}$. Cada enlace $\ell \in \mathcal{L}$ possui latência de propagação d_ℓ (ms) e capacidade de banda B_ℓ (Gbps).

B. Service Function Chains (SFCs)

Cada demanda $d \in \mathcal{D}$ origina-se em uma O-RU e solicita um serviço associado a uma SFC — sequência ordenada de VNFs que o tráfego deve atravessar [1]. Neste modelo, toda SFC é prefixada pelas funções de acesso O-RAN:

$$\text{SFC}(d) = [\underbrace{\text{vDU}, \text{vCU}}_{\text{acesso O-RAN}}, \underbrace{f_1, f_2, \dots, f_K}_{\text{VNFs de serviço}}], \quad (1)$$

onde cada VNF f é caracterizada por um requisito de CPU c_f (*cores*), taxa de dados r_f (kbps) e atraso de processamento δ_f (ms). Cada demanda d possui latência máxima fim-a-fim $L_d^{\max}$ e taxa de dados r_d , determinados pelo serviço solicitado.

III. FORMULAÇÃO DO PROBLEMA

A. Variáveis e Parâmetros

A decisão de alocação é representada pela variável binária $x_{d,f,n} \in \{0, 1\}$, onde $x_{d,f,n} = 1$ indica que a VNF f da demanda d é alocada no nó $n \in \mathcal{N}$. Define-se $\mathcal{F}_n = \{f \mid \exists d : x_{d,f,n} = 1\}$ como o conjunto de VNFs alocadas no nó n , e $\mathcal{D}_\ell$ como o conjunto de demandas cujo caminho de roteamento utiliza o enlace ℓ .

Miguel Santos, UFPA, Belém-PA, e-mail: miguel.santos@itec.ufpa.br; Matheus Pantoja, UFPA, Belém-PA, e-mail: matheus.pantoja@itec.ufpa.br; Carlos Costa, UFPA, Belém-PA, e-mail: costa.carlos@itec.ufpa.br; Rafael Vieira, UEAP, Macapá-AP, e-mail: rafael.vieira@ueap.edu.br; Diego Cardoso, UFPA, e-mail: diego@ufpa.br.

B. Modelo de Latência

A latência total de uma SFC alocada para a demanda d é a soma das latências de propagação e processamento ao longo da cadeia:

$$L(d) = \sum_{k=1}^{|SFC(d)|} [d_{n_{k-1}, n_k} + \delta_{f_k}], \quad (2)$$

onde n_k é o nó que hospeda a k -ésima VNF, n_0 é o nó da O-RU de origem, d_{n_{k-1}, n_k} é a latência de propagação do caminho mais curto entre nós consecutivos, e δ_{f_k} é o atraso de processamento da VNF f_k .

C. Modelo de Energia

O consumo energético de cada nó segue o modelo linear de potência de [6]:

$$P(n) = \begin{cases} 0, & \text{se } n \text{ está inativo} \\ P_n^{\text{idle}} + (P_n^{\text{max}} - P_n^{\text{idle}}) \cdot u(n), & \text{se } n \text{ está ativo,} \end{cases} \quad (3)$$

onde $P_n^{\text{idle}} = \alpha \cdot P_n^{\text{max}}$ é a potência em estado ocioso ($\alpha = 0,6$) [6] e $u(n) = \sum_{f \in \mathcal{F}_n} c_f / C_n$ é a utilização de CPU do nó. A consolidação de VNFs em poucos nós, promovida pela heurística proposta, faz com que nós sem carga alocada permaneçam inativos ($P(n) = 0$), reduzindo o consumo total — efeito equivalente ao desligamento de nós ociosos. O consumo energético total da rede é $E_{\text{total}} = \sum_{n \in \mathcal{N}} P(n)$, e o objetivo é minimizá-lo:

$$\min E_{\text{total}} = \sum_{n \in \mathcal{N}} P(n). \quad (4)$$

D. Restrições

A Tabela I formaliza as restrições do modelo.

TABELA I: Restrições do modelo de otimização.

ID	Formulação Matemática
C1	$\sum_{f \in \mathcal{F}_n} c_f \leq C_n, \forall n \in \mathcal{N}$
C2	$ \{f \in \mathcal{F}_n : f = \text{vDU}\} \leq V_n^{\text{max}}, \forall n \in \mathcal{N}$
C3	$\sum_{d \in \mathcal{D}_\ell} r_d \leq B_\ell, \forall \ell \in \mathcal{L}$
C4	$d_{\text{RU}, n_{\text{vDU}}(d)} \leq D_{\text{FH}}^{\text{max}}, \forall d \in \mathcal{D}$
C5	$L(d) \leq L_d^{\text{max}}, \forall d \in \mathcal{D}$
C6	$\mathcal{F}_n^{\text{acesso}} \neq \emptyset \Rightarrow \mathcal{F}_n^{\text{serviço}} = \emptyset$

As restrições (C1)–(C3) limitam, respectivamente, a CPU agregada por nó, o número de vDUs por *tier* e a banda por enlace. A restrição (C4) impõe o limite de *fronthaul* do Split 7.2x [6] e (C5) o SLA de latência fim-a-fim conforme (2). Por fim, (C6) proíbe a coexistência de VNFs de acesso e de serviço no mesmo nó [2].

IV. HEURÍSTICA GULOSA

Como o VNFP-RP é NP-difícil [1], propõe-se a heurística gulosa *Nearest-Node First* (NNF), apresentada no Algoritmo 1. A abordagem percorre sequencialmente as VNFs da SFC e aloca cada função no nó válido mais próximo em termos de latência, respeitando as restrições (C1)–(C6). Para evitar consumo desnecessário de recursos, utiliza-se uma política *all-or-nothing*, na qual as reservas só são confirmadas caso toda a SFC seja alocada com sucesso.

A complexidade por demanda é $\mathcal{O}(|SFC| \cdot |\mathcal{N}| \cdot \log |\mathcal{N}|)$, dominada pelo cálculo de caminhos mínimos via Dijkstra.

Algoritmo 1 Heurística *Nearest-Node First* (NNF)

Entrada: Demanda d com $SFC(d)$, grafo G , estados dos nós e enlaces

Saída: Alocação $\mathcal{A}$ ou rejeição da demanda

```

1:  $n_{\text{atual}} \leftarrow$  nó O-RU de origem de  $d$ 
2:  $L_{\text{acum}} \leftarrow 0$ ; criar cópia temporária dos estados
3: para cada VNF  $f_k$  em  $SFC(d)$  faça
4:   Calcular  $\{d(n_{\text{atual}}, n)\}_{n \in \mathcal{N}}$  usando o algoritmo de Dijkstra
5:   Ordenar os candidatos por distância crescente
6:   para cada candidato  $n$  faça
7:     se  $f_k = \text{vDU} \wedge d(n_{\text{atual}}, n) > D_{\text{FH}}^{\text{max}}$  então
8:       continue
9:     fim se
10:    se  $L_{\text{acum}} + d(n_{\text{atual}}, n) + \delta_{f_k} > L_d^{\text{max}}$  então
11:      continue
12:    fim se
13:    se as restrições (C1)–(C3), (C6) forem satisfeitas então
14:      Reservar  $f_k$  em  $n$  (estado temporário)
15:       $L_{\text{acum}} \leftarrow L_{\text{acum}} + d(n_{\text{atual}}, n) + \delta_{f_k}$ 
16:       $n_{\text{atual}} \leftarrow n$ ; break
17:    fim se
18:  fim para
19:  se nenhum candidato válido for encontrado então
20:    retornar Rejeitar demanda  $d$ 
21:  fim se
22: fim para
23: Confirmar as alocações temporárias no estado real
24: retornar Alocação  $\mathcal{A}$ , latência  $L_{\text{acum}}$ 

```

V. CENÁRIO EXPERIMENTAL

A. Topologias

Duas topologias são utilizadas: **Polska** (12 nós, 18 enlaces; 2 Core, 5 Edge, 5 Far Edge), baseada na instância SNDlib [8] — amplamente adotada como *benchmark* em NFV [5], [7] — escalada 0,05 $\times$; e **RNP POPs** (27 nós, 31 enlaces; 2 Core, 7 Edge, 18 Far Edge), baseada na topologia de *backbone* da RNP [9]. Todos os enlaces possuem $B_{ij} = 400$ Gbps e latências calculadas via Haversine com fator de tortuosidade 1,3 $\times$ [7] e penalidade de +2 ms para nós Core [6]. A Tabela II detalha os perfis de hardware de cada *tier*.

TABELA II: Perfis de hardware dos nós computacionais.

Tier	CPU (cores)	P^{max} (W)	V^{max} (vDUs)	Servidor
Far Edge	40	300	8	Dell XR4000
Edge	128	800	20	Dell R760
Core	512	1500	0	Dell R7615

B. Serviços e SFCs

A Tabela III apresenta os quatro serviços [5], [7], com a composição de cada SFC e o requisito de CPU (sobrescrito). As funções de acesso vDU e vCU requerem 4 e 1 *cores* ($\delta_f = 0,5$ e 0,25 ms); as VNFs de serviço possuem atraso uniforme de $\delta_f = 10$ ms.

C. Configuração dos Experimentos

Para cada topologia, são gerados 30 cenários com $D \in \{25, 50, 75, 100, 150, 200\}$ demandas, originadas em nós Far

TABELA III: Serviços, SFCs e parâmetros das VNFs ($\text{CPU}^{\text{cores}}$, $\delta_f=10\text{ ms}$). NAT: Network Address Translation; FW: Firewall; TM: Traffic Monitor; WOC: WAN Optimization Controller; VOC: Video Optimization Controller; IDS: Intrusion Detection System.

Serviço	SFC de Serviço (VNF ^{CPU})	L_d^{max} (ms)	r_d (kbps)
Web Service	NAT ² →FW ⁸ →TM ¹ →WOC ² →IDS ⁸	500	1000
VoIP	NAT ² →FW ⁸ →TM ¹ →FW ⁸ →NAT ²	100	640
Video Str.	NAT ² →FW ⁸ →TM ¹ →VOC ² →IDS ⁸	100	40 000
Online Gaming	NAT ² →FW ⁸ →VOC ² →WOC ² →IDS ⁸	60	500

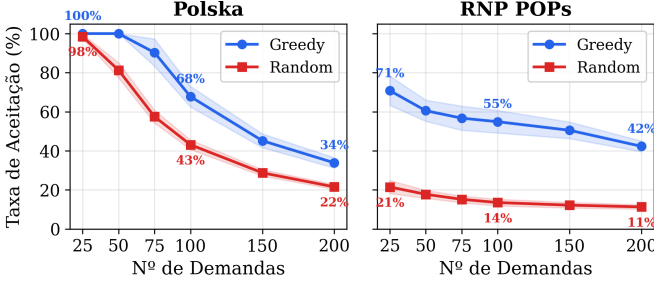


Fig. 1: Taxa de aceitação em função do número de demandas para as topologias Polska e RNP POPs.

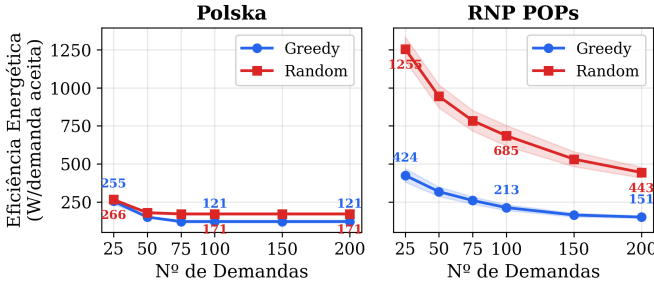


Fig. 2: Eficiência energética (W/demanda aceita) em função do número de demandas.

Edge ou Edge. Como *baseline*, utiliza-se uma alocação aleatória (Random) que seleciona uniformemente entre os nós válidos conforme (C1)–(C6), executada 30 vezes por cenário. Os experimentos foram implementados em Python 3.12 e executados em um processador Intel Core I5-13420H com 16 GB de RAM. Avaliam-se taxa de aceitação, eficiência energética por demanda atendida e latência média.

VI. RESULTADOS E DISCUSSÃO

A Figura 1 mostra que o Greedy apresenta maior taxa de aceitação que o Random em ambas as topologias. Para $D=100$, o Greedy alcança 67,7% na Polska e 54,9% na RNP POPs, enquanto o Random obtém 43,1% e 13,5%, evidenciando a vantagem da consolidação por proximidade em redes mais dispersas.

A Figura 2 mostra que o Greedy apresenta melhor eficiência energética por demanda aceita em ambas as topologias, pois consolida cargas em nós próximos. Na Polska, o Greedy consome 120,6 W/demanda contra 170,7 W/demanda do Random. A dispersão do Random ativa mais servidores, elevando o consumo ocioso P_{idle} sem ganho proporcional de aceitação.

A Figura 3 mostra que o Greedy apresenta menor latência média que o Random, especialmente na RNP POPs (60,7 ms contra 118,6 ms para $D=100$), ao evitar saltos desnecessários. Serviços com SLA mais restritivo, como Online Gaming e VoIP, sofrem maior rejeição devido ao limite de *fronthaul*. A heurística processa 100 demandas em menos de 0,2 s,

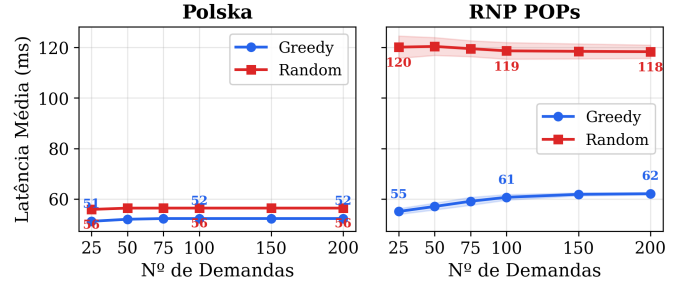


Fig. 3: Latência média das demandas aceitas em função do número de demandas.

viabilizando orquestração em tempo real frente a formulações ILP [5].

A restrição C6 (isolamento acesso/serviço) reduz o espaço de soluções e pode contribuir para a rejeição em cenários com poucos nós; em contrapartida, garante confiabilidade do plano de acesso O-RAN [2].

VII. CONCLUSÕES

Este artigo propôs um modelo unificado para o posicionamento e roteamento de VNFs em arquiteturas O-RAN, integrando funções desagregadas de acesso (vDU e vCU) com VNFs de serviço sob restrições de latência de *fronthaul* (Split 7.2x), isolamento de hardware, CPU e banda. A avaliação em duas topologias demonstrou que a heurística gulosa NNF é eficaz ao consolidar funções em nós próximos, proporcionando maior taxa de aceitação, menor latência média e melhor eficiência energética por demanda. Trabalhos futuros incluem a investigação de meta-heurísticas, a relaxação parcial da restrição C6 para permitir coalocação em nós com capacidade residual, e a avaliação em topologias de maior escala.

REFERÊNCIAS

- [1] R. F. Vieira, M. G. G. Pantoja, C. Natalino, and D. L. Cardoso, “Virtual Network Function Placement and Routing: Formulations and solutions,” *IEEE Access*, 2023.
- [2] M. Polese, L. Bonati, S. D’Oro, S. Basagni, and T. Melodia, “Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.
- [3] O-RAN Alliance, “O-RAN Use Cases and Deployment Scenarios Whitepaper,” Feb. 2020. [Online]. Available: www.o-ran.org
- [4] J. dos Santos, F. C. Fernandes, M. A. de Sousa, and F. G. C. Rocha, “Recursos na Borda: Otimização para o Open RAN usando Programação Inteira e Algoritmos Genéticos,” in *XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrt)*, Natal, RN, 2025.
- [5] 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)*, pp. 1–7, 2018.
- [6] W. T. Pires-Jr et al., “Optimizing Energy Consumption for vRAN Placement in O-RAN Systems,” *IEEE Transactions on Network and Service Management*, 2025.
- [7] A. Varasteh et al., “Holu: Power-aware and delay-constrained VNF placement and chaining,” *IEEE Trans. Network and Service Management*, vol. 18, no. 2, pp. 1524–1539, 2021.
- [8] S. Orłowski, R. Wessäly, M. Pióro, and A. Tomaszewski, “SNDlib 1.0 — Survivable Network Design Library,” *Networks*, vol. 55, no. 3, pp. 276–286, 2010.
- [9] Rede Nacional de Ensino e Pesquisa (RNP), “Backbone RNP — Mapa da Rede,” 2024. [Online]. Disponível em: <https://www.rnp.br/sistema-rnp/evolucao-da-rede-ipe/>.