Availability-Aware Backup Provisioning and Resource Allocation for Energy-Efficient vRANs

Taeyun Kim, Daeyoung Jung, Eunsok Lee, Daesik Kim, and Sangheon Pack
School of Electrical Engineering, Korea University, Seoul, Korea
{kimtyoun123, eodud1884, tinedge, dsk24, shpack}@korea.ac.kr

***Abstract*—Virtualized radio access networks (vRANs) enhance scalability and architectural flexibility, but failures of both servers and containerized network functions (cNFs) intensify availability requirements. Availability is typically ensured through redundancy mechanisms, with two representative strategies: dedicated and shared backups. Dedicated backups provide immediate recovery by replicating the state of original cNFs, but incur substantial energy consumption. In contrast, shared backups improve resource efficiency but expose the system to traffic loss or overload during failures. Thus, relying on a single backup strategy is inherently inefficient, and the choice of backup type must adapt to traffic dynamics and heterogeneous availability demands. In this paper, we formulate an availability-aware backup provisioning and resource allocation problem in vRAN as an integer linear programming (ILP) model, with the objective of minimizing total cost—including energy consumption and traffic loss penalties—under resource and availability constraints. To address the NP-hard complexity, we develop a threshold-based heuristic algorithm that adaptively selects backup types and allocates resources. Trace-driven evaluations demonstrate that the proposed approach reduces total cost by up to 52.9% and decreases failure-related penalties by about 94.1% compared to state-of-the-art methods.**

***Index Terms*—vRAN, O-RAN, availability, backup provisioning, resource scaling, containerized network function.**

I. Introduction

As mobile networks evolve into virtualized radio access networks (vRANs), deploying containerized network functions (cNFs) on commercial off-the-shelf (COTS) servers provides enhanced scalability and architectural flexibility [1]–[3]. However, virtualization introduces potential failures at both hardware and software levels, making availability assurance a critical objective for service operation. In general, service availability is maintained through redundancy mechanisms such as backup provisioning, and the choice of backup strategy directly affects resource utilization and overall operational costs.

In the literature, two representative backup strategies are available: 1) dedicated backup and 2) shared backup. Dedicated backup replicates both the state and incoming flows of the original cNF, enabling immediate recovery in the event of a failure [10], [11]. Nevertheless, maintaining such replication consumes considerable resources, leading to energy overhead. In contrast, shared backup does not replicate the state of a specific cNF but reserves an empty cNF instance with predefined capacity [4], [5]. This approach achieves better resource efficiency and lower energy consumption compared to dedicated backup. However, recovery is slower because state transfer is required after a failure, resulting in temporary traffic loss. Furthermore, shared backup is vulnerable to overload when multiple cNFs fail simultaneously.

Therefore, relying on a single backup strategy is inherently inefficient, as it often leads to excessive resource consumption or significant performance degradation due to frequent failures. It is thus essential to determine the appropriate backup type according to the traffic volume and availability requirements of each flow. In vRAN environments, traffic volume fluctuates significantly over time, and the required availability level also varies with service characteristics. Consequently, frequent decisions must be made between dedicated and shared backups, and the capacity of each backup resource should be dynamically adjusted. However, existing availability-aware VNF management frameworks are designed for data center environments with fixed traffic patterns and static availability requirements, making them unsuitable for highly dynamic vRAN environments with heterogeneous traffic and availability demands [4]–[6].

In this paper, we formulate an availability-aware backup provisioning and resource allocation problem in vRAN as an integer linear programming (ILP) model. The objective is to minimize the total cost, which consists of energy consumption from operating and scaling PMs and cNFs, as well as penalty costs from traffic loss during failures, while satisfying both resource and availability constraints. Given the NP-hard nature of the problem, we develop a threshold-based greedy heuristic to provide scalable and adaptive solutions. Trace-driven evaluations show that the proposed approach reduces total cost by up to 52.9% and decreases failure-related penalty costs by about 94.1% compared to state-of-the-art schemes.

The remainder of this letter is organized as follows. Section II presents the system model. Section III formulates the optimization problem. Section IV describes the proposed algorithm to solve the formulated problem. Section V provides the simulation results. Finally, Section VI concludes the letter.

This work was supported in part by Samsung, in part by the National Research Foundation (NRF) of Korea Grant funded by the Korea Government (MSIT) (No. RS-2024-00341965), and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. RS-2024-00405128).

II. System model

Figure 1 illustrates our system model, which consists of a set of radio units (RUs) and a pool of physical machines (PMs) hosting virtualized central units (CUs) and distributed units (DUs). These CUs/DUs are implemented as cNFs running on PM. The set of PMs is denoted by $P = \{p_1, p_2, \ldots, p_n, \ldots, p_N\}$. For each time slot t, the state of PM p_n is represented as $p_n^t = \{o_n^t, h_n^t\}$. The binary variable $o_n^t \in \{0, 1\}$ indicates the operational status of p_n at time t, where $o_n^t = 1$ means p_n is powered on and $o_n^t = 0$ means it is turned off. The binary variable $h_n^t \in \{0, 1\}$ indicates whether PM p_n has experienced a failure at time t, with $h_n^t = 0$ representing a failure and $h_n^t = 1$ indicating normal operation.

Each flow from an RU is processed by cNF deployed on an active PM. The set of cNFs is denoted by $E = \{e_1, e_2, \ldots, e_i, \ldots, e_I\}$, and the state of each cNF e_i at time t is given by $e_i^t = \{u_i^t, l_i^t, v_i^t\}$. Here, $u_i^t \in \{0, 1, \ldots, U\}$ denotes the number of CPU cores allocated to e_i at time t, where U is the maximum number of cores available on a single PM. The variable l_i^t indicates the index of PM to which e_i is assigned. The binary variable $v_i^t \in \{0, 1\}$ represents whether e_i has failed at time t, with $v_i^t = 0$ indicating failure and $v_i^t = 1$ indicating normal operation. The availability of each PM and cNF is denoted by A_{P} and A_{E}, respectively.

Traffic flows are requested through RUs, and the set of flows at time slot t is defined as $F^t = \{f_1^t, f_2^t, \ldots, f_k^t, \ldots, f_K^t\}$, where t denotes the current time slot and K is the total number of flows. Each flow f_k^t is represented as a tuple $f_k^t = \{m_k^t, r_k^t, s_k^t, y_k^t, z_k^t\}$. The parameter $m_k^t \in \{1, \ldots, M\}$ indicates the number of CPU cores required to process the flow, where M is the maximum core request. The parameter r_k^t denotes the availability requirement of the flow and takes a value within the range $[R_{\min}, R_{\max}]$ (e.g., $R_{\min} = 0.99$ and $R_{\max} = 0.9999999$). The variable s_k^t indicates the index of cNF to which the flow is assigned. The binary variable y_k^t represents the backup type: if $y_k^t = 1$, the flow is protected with a dedicated backup; if $y_k^t = 0$, the flow is protected using a shared backup. Finally, z_k^t specifies the index of cNF designated to serve as the backup for the flow.

III. Problem formulation

A. Objective function

At each time slot t, the total cost is composed of two components: 1) energy cost and 2) penalty cost.

The energy cost reflects the energy consumed for scaling and operating PMs or cNFs, and is defined as

$$\mathbf{C}_{\text{energy}}^t = C_{sc}^t + C_{op}^t, \tag{1}$$

where C_{sc}^t denotes the scaling cost and C_{op}^t denotes the operational cost.

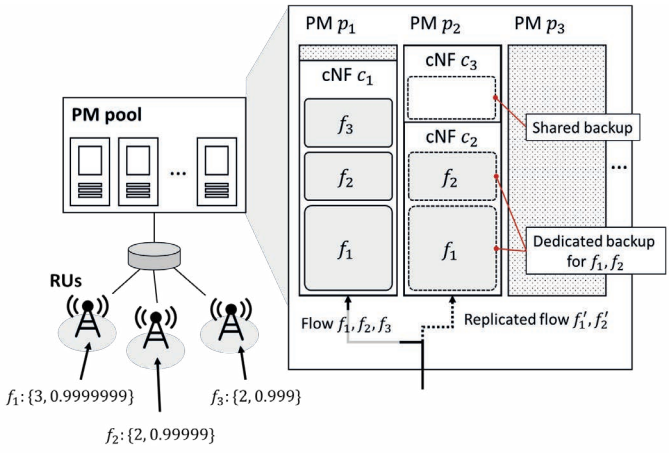

Fig. 1. System model.

The scaling cost C_{sc}^t represents the energy consumed when turning PMs on or off, adjusting the resources allocated to a cNF, or migrating a cNF to another PM, and is defined as

$$C_{sc}^t = c_\alpha \cdot \sum_{n=1}^{N} \mathbf{1}(o_n^{t-1} \neq o_n^t) + \sum_{i=1}^{I} c_\delta \cdot \mathbf{1}(u_i^{t-1} \neq u_i^t \vee l_i^{t-1} \neq l_i^t), \tag{2}$$

where c_α denotes the constant energy cost associated with turning a PM on or off, and c_δ denotes the constant energy cost associated with modifying the state of a cNF, such as adjusting its allocated CPU resources or migrating it to another PM. The indicator function $\mathbf{1}(\cdot)$ equals 1 if the condition inside the parentheses is satisfied and 0 otherwise.

The operational cost C_{op}^t represents the energy consumed in operating PMs and cNFs to process flows and is defined as

$$C_{op}^t = c_\gamma \cdot \sum_{n=1}^{N} o_n^t + c_\kappa \cdot \sum_{i=1}^{I} u_i^t + c_\omega \cdot \sum_{k=1}^{K} (1 + y_k^t) \cdot d_k^t, \tag{3}$$

where c_γ denotes the base energy constant for keeping a PM active, c_κ denotes the energy constant for operating a cNF (increasing proportionally with the number of allocated CPU cores), and c_ω denotes the base processing energy constant per flow, which also applies to replicated flows handled by dedicated backups.

The penalty cost reflects the traffic loss caused by service disruption during failures, and is defined as

$$\mathbf{C}_{\text{penalty}}^t = C_{fi}^t + C_{bk}^t, \tag{4}$$

where C_{fi}^t denotes the failure cost and C_{bk}^t denotes the backup cost.

The failure cost C_{fi}^t represents the penalty incurred when both the original cNF and its backup fail, or when the original cNF fails but the shared backup is overloaded and unable to recover the flow. In such cases, complete traffic loss for the affected flow occurs. It is defined as

$$C_{fi}^t = c_\zeta \cdot \sum_{k=1}^{K} \frac{1}{1 - r_k^t} \cdot m_k^t \cdot L_k^t \cdot y_k^t \cdot D_k^t + (1 - y_k^t) \cdot S_k^t, \tag{5}$$

where c_ζ denotes the penalty constant for traffic loss. The penalty reflects the availability requirement, such that flows with higher availability demands incur proportionally higher costs when dropped, modeled as $\frac{1}{1-r_k^t}$. The term $L_k^t = 1 - v^t_{s_k^t} \cdot h^t_{\mathrm{PM}(s_k^t)}$ indicates whether the primary cNF for flow f_k^t has failed and equals 1 in case of failure. Similarly, $D_k^t = 1 - v^t_{z_k^t} \cdot h^t_{\mathrm{PM}(z_k^t)}$ denotes whether the designated dedicated backup has failed. Here, $\mathrm{PM}(s_k^t)$ refers to the PM hosting cNF s_k^t, i.e., if $s_k^t = i$, then $\mathrm{PM}(s_k^t) = l_i^t$. Finally, $S_k^t = 1 - v^t_{z_k^t} \cdot h^t_{\mathrm{PM}(z_k^t)} \cdot \mathbf{1}\left(u^t_{z_k^t} \geq \sum_{k'=1}^{K} L^t_{k'} \cdot \mathbf{1}(z^t_{k'} = z^t_k) \cdot m^t_k\right)$ becomes 1 when the shared backup either fails or lacks sufficient capacity to support all assigned flows during simultaneous failures.

The backup cost C_{bk}^t represents the penalty incurred when shared backups are used. Specifically, it accounts for traffic loss caused by delays during service migration, where the state and flow of the failed cNF must be transferred to the shared backup. It is defined as

$$C_{bk}^t = c_\tau \cdot \sum_{k=1}^{K} \frac{1}{1-r_k^t} \cdot (1 - y_k^t) \cdot L_k^t \cdot (1 - S_k^t) \cdot m_k^t, \tag{6}$$

where c_τ denotes the penalty constant associated with service migration. The backup cost arises only when a shared backup is used (i.e., $y_k^t = 0$), the original cNF fails ($L_k^t = 1$), and the shared backup remains available to recover the flow ($S_k^t = 0$). The penalty is proportional to the number of CPU cores requested by the flow, represented by m_k^t.

The objective function, which aims to minimize the total cost across all timeslots, is defined as

$$\min_{\Xi} f(\Xi) = \sum_{t=1}^{T} \left(\lambda_e \mathbf{C}^t_{\text{energy}} + \lambda_p \mathbf{C}^t_{\text{penalty}}\right), \tag{7}$$

where Ξ denotes the set of all scaling decisions, and T is the total number of timeslots. The decision variables at time slot t are defined as $\xi^t = \{s_k^t, y_k^t, z_k^t, o_n^t, u_i^t, l_i^t\}$. The weighting parameters λ_e and λ_p are introduced to normalize the relative impact of the energy cost and the penalty cost on the total cost.

B. Constraints

We have the following constraints:

$$\forall i, \quad u_i^t \geq \sum_{k=1}^{K} \mathbf{1}(s_k^t = i) \cdot m_k^t \tag{8}$$

$$\forall i, \quad u_i^t \geq \sum_{k=1}^{K} y_k^t \cdot \mathbf{1}(z_k^t = i) \cdot m_k^t \tag{9}$$

$$\forall l_i^t = n, \quad o_n^t = 1 \tag{10}$$

$$\forall s_k^t = i, \quad \sum_{k=1}^{K} \mathbf{1}(z_k^t = i) = 0 \tag{11}$$

$$\forall z_k^t = i, \quad \sum_{k=1}^{K} \mathbf{1}(s_k^t = i) = 0 \tag{12}$$

Constraint (8) ensures that each cNF has at least as many CPU cores as the sum of resource demands from the flows assigned to it. Constraint (9) guarantees that if a cNF is designated as a dedicated backup, it must also have sufficient resources to accommodate all its assigned replicated flows. Constraint (10) enforces that a PM hosting a cNF must be powered on. Constraints (11) and (12) ensure role separation: if a cNF is assigned as a primary processor for any flow, it cannot simultaneously serve as a backup, and vice versa.

IV. Heuristic algorithm

Solving the formulated problem via exhaustive search incurs a computational complexity of $O(I^{2KT} 2^{2(I+N)} U^{IT} N^{IT})$, which is NP-hard. Owing to the real-time decision-making requirements in vRAN environments, we propose a threshold-based heuristic algorithm that achieves fast processing with significantly lower complexity.

Algorithm 1 Availability-aware backup provisioning and resource allocation algorithm

for each flow f_k^t **do**
  Assign f_k^t to $\mathcal{H}$ if $r_k^t \geq \theta$, else to $\mathcal{L}$
  Set $y_k^t \leftarrow 1$ if $f_k^t \in \mathcal{H}$, else $y_k^t \leftarrow 0$
for each group $g \in \{\mathcal{H}, \mathcal{L}\}$ **do**
  for each $f_k^t \in g$ in descending m_k^t **do**
    $s_k^t \leftarrow i$ ($e_i \in \mathcal{C}_g$), if $u_i^t \geq m_k^t + \sum_{k'=1}^{K} \mathbf{1}(s_{k'}^t = i) \cdot m_{k'}^t$
    if no such $e_i \in \mathcal{C}_g$ exists **then**
      Turn on new p_n and launch $e_{i'} \in \mathcal{C}_g$
      ($o_n^t \leftarrow 1$, $u_{i'}^t \leftarrow U$, $l_{i'}^t \leftarrow n$, $s_k^t \leftarrow i'$)
for each $f_k^t \in \mathcal{H}$ in descending m_k^t **do**
  $z_k^t \leftarrow i$ ($e_i \in \mathcal{B}_\mathcal{H}$), if $u_i^t \geq m_k^t + \sum_{k'=1}^{K} \mathbf{1}(z_{k'}^t = i) \cdot m_{k'}^t$
  if no such $e_i \in \mathcal{B}_\mathcal{H}$ exists **then**
    Turn on new p_n and launch $e_{i'} \in \mathcal{B}_\mathcal{H}$
    ($o_n^t \leftarrow 1$, $u_{i'}^t \leftarrow U$, $l_{i'}^t \leftarrow n$, $z_k^t \leftarrow i'$)
if $|\mathcal{B}_\mathcal{L}| < \lceil |\mathcal{L}|/W \rceil$ **then**
  $\psi^{\mathrm{up}} \leftarrow \psi^{\mathrm{up}} + 1$, $\psi^{\mathrm{dw}} \leftarrow 0$
  if $\psi^{\mathrm{up}} \geq \mathcal{T}$ **then**
    Turn on new p_n and launch $e_{i'} \in \mathcal{B}_\mathcal{L}$
    ($o_n^t \leftarrow 1$, $u_{i'}^t \leftarrow U$, $l_{i'}^t \leftarrow n$)
else if $|\mathcal{B}_\mathcal{L}| > \lceil |\mathcal{L}|/W \rceil$ **then**
  $\psi^{\mathrm{dw}} \leftarrow \psi^{\mathrm{dw}} + 1$, $\psi^{\mathrm{up}} \leftarrow 0$
  if $\psi^{\mathrm{dw}} \geq \mathcal{T}$ **then**
    Turn off one $e_i \in \mathcal{B}_\mathcal{L}$ and its PM ($u_i^t \leftarrow 0$, $o_{l_i^t}^t \leftarrow 0$)
Assign $f_k^t \in \mathcal{L}$ to $e_i \in \mathcal{B}_\mathcal{L}$ ($z_k^t \leftarrow i$) in round-robin
for each $e_i \in \mathcal{C}_\mathcal{H} \cup \mathcal{C}_\mathcal{L} \cup \mathcal{B}_\mathcal{H}$ **do**
  if e_i is idle **then**
    $\psi_i \leftarrow \psi_i + 1$; $u_i^t \leftarrow 0$ and $o_{l_i^t}^t \leftarrow 0$ if $\psi_i \geq \mathcal{T}$
  else
    $\psi_i \leftarrow 0$

Algorithm 1 outlines the detailed operation of the proposed availability-aware backup provisioning and resource allocation procedure. The algorithm first partitions flows into two groups based on their availability requirement r_k^t with respect to a predefined threshold θ. Flows with $r_k^t \geq \theta$ are assigned to

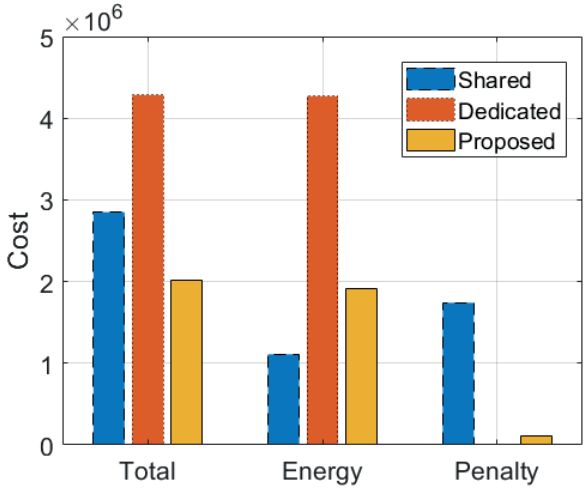

Fig. 2. Comparison with with state-of-the-art methods.

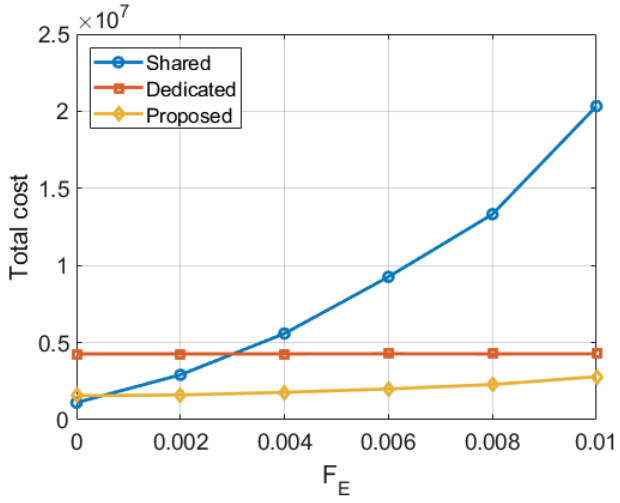

Fig. 3. Effect of the cNF failure probability.

the high-availability group ($\mathcal{H}$) and configured with dedicated backups ($y_k^t = 1$), while the rest are grouped into the low-availability group ($\mathcal{L}$) and assigned shared backups ($y_k^t = 0$) (lines 1–3). Flows are allocated to existing cNFs in descending order of demand m_k^t, and if no suitable cNF is available, a new PM is turned on and a cNF with full capacity U is launched (lines 4–8). For each flow in $\mathcal{H}$, a dedicated backup cNF is selected similarly; if no appropriate backup exists, a new one is provisioned (lines 9–12).

Next, the number of shared backup cNFs is dynamically adjusted. The target number is calculated as $\lceil |\mathcal{L}|/W \rceil$ based on the shared backup ratio W. If the current number is insufficient or excessive, a corresponding counter (ψ^{up} or ψ^{dw}) is incremented. When the counter exceeds a threshold $\mathcal{T}$, a shared backup is added or removed accordingly, and the opposing counter is reset (lines 13–20). Flows in $\mathcal{L}$ are assigned to available shared backups in a round-robin manner (line 21). Finally, for all primary and backup cNFs, if a cNF remains idle, its local counter ψ_i is incremented; when it reaches $\mathcal{T}$, cNF and its hosting PM are powered off. If cNF becomes active again, the counter is reset to zero (lines 23–26).

V. Simulation Results

To evaluate the performance of the proposed algorithm (**Proposed**), we compare it with two state-of-the-art schemes: **Shared** and **Dedicated**. **Shared** allocates sufficient shared backup resources to satisfy the availability requirements of all flows [4], [5], while **Dedicated** provisions each flow with a fully dedicated backup [10], [11]. The simulation spans 24 hours, divided into $T = 144$ timeslots of 10 minutes each. We consider $K = 100$ flows, where each flow's CPU demand is normalized within $[0, 8]$ based on traffic variations from the Milan Internet Traffic Dataset [12]. The parameters used in the simulation are summarized as follows [4], [6], [9]. The availability requirement of each flow is randomly selected between $R_{\min} = 0.99$ and $R_{\max} = 0.9999999$. The physical infrastructure consists of $N = 30$ COTS servers, each equipped with 64 cores and supporting up to $I = 100$ cNFs. The availability of servers and cNFs is set to $A_P = 0.999$ and $A_E = 0.999$, respectively. Accordingly, servers and cNFs independently fail in each timeslot with probabilities $F_P = 1 - A_P = 0.001$ and $F_E = 1 - A_E = 0.001$. The cost parameters are defined as follows: $c_\alpha = 10$ for server on/off, $c_\delta = 10$ for cNF scaling or migration, $c_\gamma = 10$ for server operation, $c_\kappa = 1$ for cNF maintenance per unit capacity, $c_\omega = 1$ for flow processing, $c_\zeta = 1$ for failure penalty, and $c_\tau = 0.001$ for shared backup overhead. The weighting factors for the objective function are set to $\lambda_e = 1$ and $\lambda_p = 1$.

Figure 2 presents the simulation results. The proposed scheme demonstrates significant improvements compared to the state-of-the-art methods. Specifically, the total cost is reduced by 29.3% compared to **Shared** and 52.9% compared to **Dedicated**. In terms of energy cost, the proposed scheme achieves a 55.2% reduction relative to **Dedicated**, while it is 72.3% higher than **Shared**. This is because **Shared** employs shared backups for all flows, thereby incurring relatively low energy costs for operating and scaling backup resources. However, when failures occur, **Shared** suffers from high penalty costs due to limited recovery capability, such as delays in state migration to backups or overload of shared backups that result in traffic loss. Consequently, the penalty cost of **Proposed** is about 94.1% lower than that of **Shared**. In contrast, **Dedicated** assigns a dedicated backup for all flows, and thus penalty costs are incurred only in very rare cases when both the original cNF and its dedicated backup fail simultaneously. As a result, **Dedicated** achieves 85.9% lower penalty cost compared to **Proposed**.

Figure 3 illustrates the impact of the cNF failure probability F_E. We vary F_E from 0 to 0.01. Except for the case of $F_E = 0$, **Proposed** achieves up to 86.2% lower total cost compared to the other methods. When $F_E = 0$, failures do not occur, and thus maintaining backups becomes unnecessary, resulting in wasted energy costs. In this case,

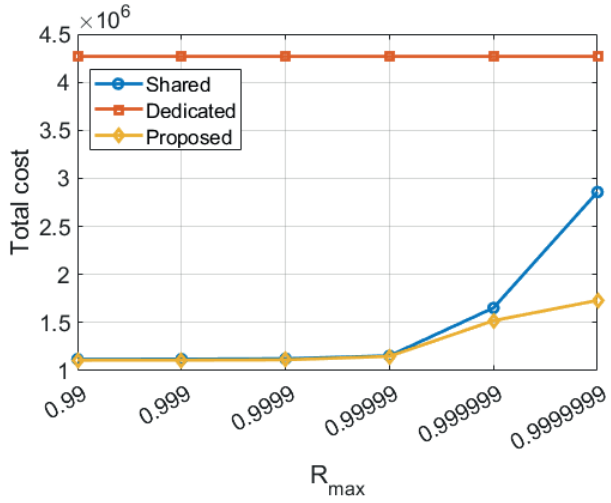

Fig. 4. Effect of the maximum availability requirements.

Shared outperforms **Proposed**. Conversely, when $F_E = 0.01$, i.e., under a high failure probability, frequent failures result in large traffic loss for unrecovered flows, causing **Shared** to incur significantly higher penalty costs. In this regime, both **Dedicated** and **Proposed** mitigate failures effectively through the use of dedicated backups, thereby maintaining lower total costs compared to **Shared**. However, **Dedicated** assigns backups to all flows indiscriminately, incurring significant resource overhead and yielding a 53.2% higher total cost than **Proposed**. These results demonstrate that **Proposed** achieves resilient yet resource-efficient operation by adaptively provisioning dedicated backups only for flows with stringent availability demands while utilizing shared backups for others.

Figure 4 shows the impact of the maximum availability requirement $R_{\max}$ on the total cost, where $R_{\max}$ is varied from 0.99 to 0.9999999. **Proposed** consistently achieves the lowest total cost across all scenarios, with reductions of up to 74.1%. When $R_{\max}$ lies between 0.99 and 0.99999, **Shared** performs similarly to **Proposed**, as all flows are protected with shared backups in both cases. However, when $R_{\max} \geq 0.999999$, the total cost of **Shared** increases sharply due to large penalties from failures of high-availability flows. In the range $R_{\max} \in [0.99999, 0.9999999]$, this escalation becomes even steeper for **Shared**, whereas the slope for **Proposed** decreases. This is because **Proposed** employs dedicated backups for flows with stringent availability requirements, leading to additional energy costs but avoiding significant penalties from unrecovered failures.

VI. Conclusion

In this paper, we studied availability-aware backup provisioning and resource allocation for vRANs. We formulated an ILP that minimizes the total cost—combining energy consumed by operating/scaling PMs and cNFs with penalties from traffic loss during failures—under resource and availability requirements. We then designed a threshold-based heuristic that partitions flows by availability requirement, assigns dedicated or shared backups accordingly, and applies counter-based delayed scaling in line with traffic demand. Trace-driven evaluations against **Shared** and **Dedicated** baselines show substantial gains: our method reduces total cost by up to 52.9% relative to **Dedicated**, and under high cNF failure probability achieves up to 86.2% lower total cost than **Shared** with markedly reduced penalty costs. These results indicate that adaptively selecting backup types by availability demand and co-optimizing resource usage are key to cost-efficient, failure-resilient vRAN operation. As future work, we will investigate learning-based mechanisms that dynamically tune the threshold θ (and related weights) in response to traffic dynamics and flow-level availability requirements.

References

[1] P. Rost, I. Berberana, A. Maeder, H. Paul, V. Suryaprakash, M. Valenti, D. Wübben, A. Dekorsy, and G. Fettweis, "Benefits and challenges of virtualization in 5G radio access networks" *IEEE Communications Magazine*, vol. 53, no. 12, pp.75-82, December 2015.

[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, January 2023.

[3] S. Singhm, R. Singh, and B. Kumbhani, "The evolution of radio access network towards Open-RAN: Challenges and opportunities," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC) Workshops*, April 2020.

[4] D. Li, P. Hong, K. Xue, and J. Pei, "Availability aware VNF deployment in datacenter through shared redundancy and multi-tenancy," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1651-1664, December 2019.

[5] G. Li, H. Chen, L. Wu, X. Chi, J. Yao, and F. Xia, "Efficient Deployment and Scheduling of Shared VNF Instances in Mobile Edge Computing Networks," *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 32259–32271, October 2024.

[6] X. Liu, B. Cheng, and S. Wang, "Availability-aware and energy-efficient virtual cluster allocation based on multi-objective optimization in cloud datacenters," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 972-986, June 2020.

[7] F. W. Murti, S. Ali, G. Iosifidis, and M. Latva-aho, "Deep reinforcement learning for orchestrating cost-aware reconfigurations of vRANs," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 200-216, February 2024.

[8] E. Amiri, N. Wang, M. Shojafar, M. Hamdan, C. Foh, and R. Tafazolli, "Deep reinforcement learning for robust VNF reconfigurations in O-RAN," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 1115-1128, February 2024.

[9] T. Kim, D. Jung, Y. Kim, and S. Pack, "Cost-Aware Neural Adaptive Scaling for vRAN Resource Allocation," *IEEE Transactions on Mobile Computing*, to appear.

[10] J. Xing, J. Gong, X. Foukas, A. Kalia, D. Kim, and M. Kotaru, "Enabling resilience in virtualized RANs with Atlas," in *Proc. ACM MobiCom*, Madrid, Spain, October 2023.

[11] N. Lazarev, T. Ji, A. Kalia, D. Kim, I. Marinos, F. Y. Yan, C. Delimitrou, Z. Zhang, and A. Akella, "Resilient baseband processing in virtualized RANs with Slingshot," in *Proc. ACM SIGCOMM*, New York, NY, USA, September 2023.

[12] T. Italia, "Telecommunications - SMS, call, internet - MI," 2015. [Online]. Available: https://doi.org/10.7910/DVN/EGZHFV