

UDC 004.7:621.396

A Distributed Multi-Tier Orchestration Framework for the Rapid Recovery of Virtualized Network Functions

Sulima S. V. , Karashevych Ye. D. 

National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine

E-mail: itssulima@gmail.com

The rapid adoption of 5G network slicing over the virtualized infrastructure has led to the identification of a key gap in the existing recovery frameworks whereby centralized orchestration approaches are unable to enforce the sub-second restoration mandates of the Ultra-Reliable Low-Latency Communication (URLLC) slices, while simple heuristics cannot observe the Service Level Agreement (SLA) heterogeneity across slice types. This paper presents the DLR (Distributed Latency-aware Recovery) framework – a three-tier hierarchical orchestration architecture for the prompt recovery of VNFs (Virtualized Network Functions) in sliced 5G. The framework is based on a control decomposition into a Global Orchestrator, Regional Slice Managers (RSMs), and Local Monitoring Agents (LMAs), where decisions on the recovery are made locally based on the SLA violations without any central bottlenecks. RSM placement is formulated as a multi-objective optimization problem with minimum node-to-manager latency, load imbalance, and inter-manager coordination overhead with a genetic algorithm solver producing 11–33% better combined quality scores than baseline strategies. Dedicated recovery algorithms deal with both a catastrophic failure of nodes by a hop limited and prioritized order of reassignment of VNFs as well as dealing with progressive overloads by using scoring-based proactive migration with given cutoff to expand or shorten the migration period. A novel four-component VNF Migration Cost Function is formulated based on jointly accounting for compute, network, migration, and SLA penalty costs, resulting in up to 44% fewer migration steps for progressive overload instances. For cross-domain scenarios, a lightweight consensus protocol coordinates with neighboring RSMs without central arbitration. Simulation results on a synthetic 10-node topology show that DLR has a median recovery time of 0.7 s (5.4× faster than CGO), with URLLC SLA violations limited to 4% (compared to 14% for the Nearest Available Matching) and a 27% reduction in operational costs compared to latency-only approaches.

Keywords: Network Function Virtualization (NFV); network slicing; local reconfiguration; distributed recovery; SLA-driven optimization

DOI: [10.64915/RADAP.2026.105.22-30](https://doi.org/10.64915/RADAP.2026.105.22-30)

Introduction

Multiple logical, isolated, fractional networks running on shared physical infrastructure through network slicing is encouraging the transition from being focused on hardware to a fully virtualized software-defined network environment. 3GPP TS 23.501 identifies a network slice as being a logical network with its own features and functions; TR 28.801 provides the framework for managing and orchestrating a network slice. The ITU has defined three classes of services in 5G: Enhanced Mobile Broadband (eMBB) and Ultra-Reliable Low Latency Communications (URLLC) and Massive Machine Type Communications (mMTC) [1–5]. The extreme differences in service delivery requirements (e.g.; under 1 millisecond end-to-end latency with 99.9999% network availability for autonomous vehicles vs. significantly longer periods

of acceptable latency for IoT sensors) complicate optimal resource allocation and reservation.

Slicing relies on the use of Network Function Virtualization (NFV) in order to run telecom applications over virtualized resources that are common off-the-shelf equipment [6]. However, NFV is less reliable than traditional carrier-grade telecom app/server hardware; in addition, NFV is subject to a significantly higher level of exposure in the form of software vulnerabilities [7] due to the introduction of additional layers of virtualization by the hypervisor layer that are implemented in NFVs as compared to carrier-grade equipment [8, 9]. Therefore, Virtualized Network Function (VNF) placement in heterogeneous 5G environments for all types of networks, will be very important for meeting Service Level Agreement (SLA) commitments

for 5G services (consequently creating the basis for the Distributed Latency-aware Recovery (DLR) framework).

1 Related Work

Most studies formulate VNF placement as MILP (Mixed-Integer Linear Programming)/ILP/MINLP problems minimizing cost, latency, or energy subject to QoS/SLA constraints [1, 3, 4, 10, 11]. While mathematically precise, these methods suffer scalability issues with network size, and are based on batch training thus cannot be applied in real-time URLLC scenarios directly – motivating the presented distributed DLR approach in this work.

Then, to overcome solver complexity, heuristic methods have been widely adopted. Thien et al. [8] employed NSGA-III genetic algorithm for joint latency and VNF processing optimization; Hirayama et al. [9], Shinde et al. [5], and Nguyen et al. [12] tackled VNF placement with slice and bandwidth constraints. Genetic-algorithm heuristics are adapted to Regional Slice Managers (RSM) placement problem for node-to-manager latency, load balance, and inter-manager coordination optimization simultaneously.

Vidhya et al. [6] applied Multi-Agent Deep Reinforcement Learning (MADRL) for dynamic VNF migration while maintaining SLA compliance. Abdellatif et al. [13] proposed Intelligent-Slicing for joint optimized across multiple KPI. Panitsas et al. [14] introduced SlicePilot (2025), which delivers a $19\times$ speedup, with an average cost deviation of less than 8% in achieving near-optimal multi-cloud VNF placement. ML-based predictive migration for proactive URLLC management remains the critical future direction.

Recent orchestration attempts, based on reinforcement learning, show potential for structural adaptability with dynamical traffic flow. They typically require extensive offline training and large state-space exploration, which limit their applicability for strict sub-second URLLC recovery constraints.

ETSI NFV [15] and NFV-REL 001 [16] recommend redundancy-based protection. Gember-Jacobson et al. [17] suggested live stateful VNF migration with recovery time in hundreds of milliseconds but suffered limited scalability due to centralized control. Perhaps, a fundamental limitation that spans the state-of-the-art literature is uniformity in recovery requirements for all the services which would be inappropriate in 5G slicing where the URLLC service demands restoration in less than a second while the eMBB services can tolerate degradation for seconds. Despite the distributed SDN controller placement research [13, 14, 18], all NFV orchestration approaches remained centralized [19]. The DLR framework proposed in this paper addresses these limitations by achieving a median recovery time of 0.7s ($5.4\times$ better than the centralized optimization baseline),

a rate of SLA violations of 4% for URLLC, compared to 14% for NAM (Nearest Available Matching), and 27% cost savings in comparison to an optimization based solely on latency.

Most migration cost models account for only one or two components [17, 20, 21]. This paper is the first to derive an analytic unified four component VNF Migration Cost Function, with compute resources, network overhead, service disruption (cost incurred by the migration) and SLA penalties. The resulting model yields up to 44% fewer overall migration steps (compared to naive approaches) under progressive overload. A systematic analysis examines 4 critical gaps addressed by the DLR: 1) lack of slice-aware recovery [1, 2]; 2) simplistic migration cost models; 3) centralized orchestration bottleneck [19]; and 4) inability of a single platform to support both progressive overloads as well as catastrophic failures.

2 Problem Formulation

The recovery optimization problem is mixed-integer, large-scale, multi-objective, and subject to strict latency limits; this is especially true for URLLC slices, where any delays will be unacceptable. Therefore, centralized re-optimization each time there is a failure will not be feasible due to the computational requirements of the 5G environment.

DLR helps to overcome this limitation by distributing recovery tasks among the slice managers with some knowledge of the underlying topology of the slices. The slices are ordered by the importance of the SLA to determine which slice has the highest priority (i.e. which URLLC slice would be the first to be processed). Recovery decisions regarding placement, routing, and migration are organized hierarchically to be compliant with the ETSI NFV MANO and distributed SDN policies, and include cost for migration, failure semantics, and heterogeneous resilience requirements.

The layout has three layers:

- **Infrastructure Layer** = Substrate Network: $SN = (N, N_E)$;
- **Control Layer** = Distributed Slice Managers: $M \subseteq N$, each managing a domain D_m and performing failure detection, resource selection, and coordination;
- **Optimization Layer** = Hierarchical Decision Making using fast heuristics for initial decision making and SLA-based evaluation of migration costs.

There are two types of failures:

- **Catastrophic Node Failure** = immediate recovery required due to failure of one or more nodes;

- **Progressive Overload** = detectable through utilization trend analysis and can be mitigated proactively through resource migration.

The recovery workflow includes four stages (Fig.1):

1. **Slice identification:** affected slices $S_{affected}$ are prioritized by π_s ;
2. **Candidate filtering:** feasible nodes N_v^{cand} are selected based on resources, suitability, latency, and proximity;

3. **Cost-aware migration:** minimize

$$\Delta C = \Delta Cost_{compute} + \Delta Cost_{network} + \Delta Cost_{migration} + \Delta Penalty_{SLA} \quad (1)$$

using greedy or multi-criteria strategies;

4. **Inter-management coordination:** inter-managers reach a fast agreement about available local resources; when local resources are inadequate, coordination relies on lightweight consensus.



Fig. 1. Algorithm 1. Distributed Slice Recovery

3 Proposed Approach

Building on distributed SDN control [22,23] and ETSI NFV MANO principles [19], DLR implements a three-tier hierarchical architecture for low-latency, scalable recovery in network slicing environments:

- **Tier 1 – Global Orchestrator (GO):** maintains a global resource view; performs initial slice instantiation per ETSI NFV-IFA [24] and long-term capacity planning. GO does not participate in real-time recovery to avoid centralization bottlenecks.
- **Tier 2 – Regional Slice Managers (RSMs):** deployed on selected nodes via the placement optimization procedure; maintain detailed VNF and resource state within their domain; coordinate local

recovery and synchronize with neighboring RSMs for cross-domain scenarios.

- **Tier 3 – Local Monitoring Agents (LMAs):** deployed on every physical node; continuously monitor for failures and overload; report to responsible RSMs and execute migration commands.

This decomposition enables the sub-second recovery required for URLLC slices [2, 25] while avoiding the scalability and reliability limitations of centralized NFV management [26, 27].

3.1 RSM Placement Optimization

Given $SN = (N, N_E)$ and a desired RSM count $|M|$, binary variables $p_n \in \{0, 1\}$ and $\pi_{n,m} \in \{0, 1\}$

denote placement and node-to-manager assignment. Three objectives are minimized jointly:

$$U_{latency} = \max_{n \in N} \min_{m \in M} L_{n,m} \pi_{n,m},$$

$$U_{imbalance} = \max_{m \in M} load_m - \min_{m \in M} \{load_m : load_m > 0\},$$

where $load_m = \sum_{n \in N} \pi_{n,m}$, and

$$U_{inter-latency} = \max_{\substack{m_1, m_2 \in M \\ m_1 \neq m_2}} L_{m_1, m_2}.$$

The combined objective is:

$$\min_{p_n, \pi_{n,m}} U_{total} = w_{lat} U_{latency} + w_{imbal} U_{imbalance} + w_{inter} U_{inter-latency} \quad (2)$$

subject to an exact number of RSMs:

$$\sum_{n \in N} p_n = |M|, \quad (3)$$

each node assigned to at least one RSM:

$$\sum_{m \in M} \pi_{n,m} \geq 1, \quad \forall n \in N, \quad (4)$$

and assignment only to active RSMs:

$$\pi_{n,m} \leq p_m, \quad \forall n \in N, m \in M. \quad (5)$$

This NP-hard problem is solved via a genetic algorithm [8, 20] with binary encoding, tournament selection ($k = 3$), uniform crossover ($p_c = 0.8$) with constraint repair, and random manager-node swap mutations ($p_m = 0.05$). The algorithm runs for 200 generations with a population size of 50 individuals.

3.2 Failure Detection

LMAs apply threshold-based monitoring. A catastrophic failure is declared if an LMA is unresponsive for $T_{timeout}$ (e.g., 100 ms) [28]. Progressive overload triggers when $\rho_n^{CPU} > \theta_{high}^{CPU}$ or $\rho_n^{mem} > \theta_{high}^{mem}$, with stricter thresholds for URLLC slices. RSMs classify overload as a transient spike, sustained overload, or imminent failure based on duration relative to $T_{transient}$, $T_{sustained}$, and $\theta_{critical}$.

3.3 Catastrophic Failure Recovery

Upon detecting failure at n_{fail} , the RSM executes five steps:

1. identify $V_{fail} = \{(s, v) \mid x_{n_{fail}, v}^s = 1\}$;
2. sort by slice priorities π_s and VNF criticality $criticality_v$;

3. find candidates within hop limit $r_{max} = 2$ for URLLC, $r_{max} = 4$ for eMBB, satisfying capacity and suitability constraints;
4. compute recovery cost:

$$Cost_{s,v}(n') = c_{comp} \cdot D_v^{comp} + c_{net} \cdot \Delta BW(n', s, v) + Penalty_{latency}(n', s, v);$$

5. greedily assign each VNF to the minimum-cost feasible node:

$$(s, v) \rightarrow n^*, \quad n^* = \arg \min_{n'} Cost_{s,v}(n').$$

Hop-limited localized recovery significantly reduces decision latency versus global optimization [17, 29], meeting URLLC recovery bounds [30, 31].

3.4 Progressive Overload Recovery

Upon detecting overload at node n_{over} , all hosted VNFs are ranked using:

$$score_{s,v} = \alpha_1(1 - \pi_s) + \alpha_2 U_{s,v}^{actual} + \alpha_3(1 - statefulness_v), \quad (6)$$

where π_s is the slice priority, $U_{s,v}^{actual}$ is the current resource utilization, and $statefulness_v \in [0, 1]$ reflects migration complexity.

VNFs are then migrated in descending order of $score_{s,v}$ to feasible nodes until the overload condition is resolved:

$$\rho_{n_{over}}^{CPU} \leq \theta_{target}^{CPU}, \quad \theta_{target}^{CPU} = 0.7. \quad (7)$$

An adaptive cutoff mechanism terminates the migration process when either marginal load decrease becomes negligible or migration cost overcomes its benefit [32].

3.5 Cross-Domain Coordination

When no local resources are available, RSMs use an express round protocol: (a) broadcast a recovery request with resource, latency, and priority constraints; (b) request neighboring RSM nodes to search for the requested resource in parallel; (c) select the minimum-cost option and coordinate migration.

For complex scenarios, recovery is formulated as:

$$\min \sum \delta_{v,n \rightarrow n'}^s \cdot C_{v,n \rightarrow n'}^{mig} + \lambda |Cost_{oper}^{new} - Cost_{oper}^{old}|, \quad (8)$$

where $\delta_{v,n \rightarrow n'}^s \in \{0, 1\}$ is the migration decision variable, $C_{v,n \rightarrow n'}^{mig}$ is the migration cost, λ is the trade-off parameter, and $Cost_{oper}^{new}$, $Cost_{oper}^{old}$ are the operational costs after and before recovery. MILP is used for small instances (≤ 20 VNFs); greedy heuristics for large-scale deployments.

3.6 DLR Algorithm

Algorithm 1 summarizes the complete DLR procedure.

Algorithm 1 Distributed Low-Latency Recovery (DLR)

Require: $SN = (N, N_E)$, M , LMA reports, thresholds $(T_{timeout}, \theta^{CPU}, \theta^{mem})$, parameters $(r_{max}, \alpha_i, c_{comp}, c_{net})$

Ensure: Updated VNF placement

```

1: while TRUE do
2:   for each node  $n \in N$  (monitored by LMA) do
3:     if no response for  $T_{timeout}$  then
4:       CATASTROPHICFAILURE( $n$ )
5:     else if  $\rho_{CPU}(n) > \theta_{high}^{CPU}$  or  $\rho_{mem}(n) > \theta_{high}^{mem}$  then
6:       PROGRESSIVEOVERLOAD( $n$ )
7:     end if
8:   end for
9: end while

10: procedure CATASTROPHICFAILURE( $n_{fail}$ )
11:    $V_{fail} \leftarrow \{(s, v) \mid x_{n_{fail}, v}^s = 1\}$ 
12:   Sort  $V_{fail}$  by priority  $(\pi_s, criticality_v)$ 
13:   for each  $(s, v) \in V_{fail}$  do
14:      $CandidateNodes \leftarrow$  nodes within  $r_{max}$  hops satisfying capacity, slice constraints, latency bounds
15:     for each  $n' \in CandidateNodes$  do
16:        $Cost(n') \leftarrow c_{comp} \cdot D_v^{comp} + c_{net} \cdot \Delta BW(n', s, v) + Penalty_{latency}(n', s, v)$ 
17:     end for
18:     Select  $n^* = \arg \min Cost(n')$ ; migrate VNF  $(s, v) \rightarrow n^*$ 
19:   end for
20: end procedure

21: procedure PROGRESSIVEOVERLOAD( $n_{over}$ )
22:    $V_{over} \leftarrow$  VNFs hosted on  $n_{over}$ 
23:   for each  $(s, v) \in V_{over}$  do
24:      $score(s, v) \leftarrow \alpha_1(1 - \pi_s) + \alpha_2 \cdot U_{actual}(s, v) + \alpha_3(1 - statefulness_v)$ 
25:   end for
26:   Sort  $V_{over}$  in descending order of score
27:   for each  $(s, v) \in V_{over}$  do
28:     Select feasible target  $n'$ ; migrate VNF  $(s, v) \rightarrow n'$ 
29:     if  $\rho_{CPU}(n_{over}) \leq \theta_{target}^{CPU}$  then break
30:   end if
31: end for
32: end procedure

33: procedure CROSSDOMAINRECOVERY(request)
34:   Broadcast request to neighboring RSMs
35:   for each neighbor  $m \in M$  in parallel do
36:     Search feasible nodes; compute local best cost
37:   end for
38:   Collect responses; select global best  $n^*$ ; coordinate migration to  $n^*$ 
39: end procedure

```

3.7 Computational Complexity

The DLR algorithm operates on three main processes. Detection complexity is $O(1)$ per node per monitoring interval, yielding $O(|N|)$ for the full network.

Catastrophic Failure Recovery: identifying V_{fail} costs $O(|V|)$; sorting $O(|V_{fail}| \log |V_{fail}|)$; candidate selection $O(|N|)$ per VNF; cost computation $O(|N|)$ per VNF; greedy assignment $O(|V_{fail}|)$. Total worst-case: $O(|V_{fail}| \cdot |N|)$, significantly less than global MILP at $O(|V| \cdot |N|^2)$.

Progressive Overload Recovery: scoring $O(|V_{nover}|)$, sorting $O(|V_{nover}| \log |V_{nover}|)$, greedy migration $O(|V_{nover}|)$. Total: $O(|V_{nover}| \log |V_{nover}|)$.

Cross-Domain Coordination: broadcasting $O(|M|)$; parallel search $O(|N|)$ per RSM; selection $O(|M|)$. Total: $O(|M| \cdot |N|)$.

Overall, assuming a single event: $O(|V_{fail}| \cdot |N| + |V_{nover}| \log |V_{nover}| + |M| \cdot |N|)$.

4 Experiments

The experimental evaluation was conducted using a custom discrete-event simulation environment implemented in Python 3.12. Network topology process-

ing used the NetworkX library; optimization used NumPy and SciPy packages.

DLR was evaluated on a synthetic 10-node metropolitan 5G topology with heterogeneous edge (8–16 cores, 32–64 GB RAM) and regional nodes (32–64 cores, 128–256 GB RAM), and link latencies of 1–20 ms.

Three 3GPP slice types were instantiated: URLLC (3 VNFs, 15 cores, $L_{max} = 5$ ms, $\pi = 1$); eMBB (5 VNFs, 30 cores, $L_{max} = 50$ ms, $\pi = 2$); and mMTC (3 VNFs, 10 cores, $L_{max} = 1000$ ms, $\pi = 3$), totaling 33 VNFs across three slices.

Figure 2 illustrates the simulated topology structure.

Comparative performance results under single node failure are presented in Table 1.

Three failure scenarios were tested: (1) Single Node Failure; (2) Progressive Overload (gradual CPU utilization increase beyond 0.85); and (3) Multi-Node Failure (simultaneous failure of 2 nodes). DLR was compared with CGO (MILP-based optimum), NAM (greedy nearest migration), and LOO (Latency-Only Optimization).

The genetic algorithm for RSM placement produced a combined quality score 11–33% better than random, latency-only, and K-center baselines, with an additional 15% reduction in inter-manager latency compared to the K-center method. DLR achieved 90% restoration in 1.2 s and 44% fewer migrations than standard methods due to adaptive threshold management.

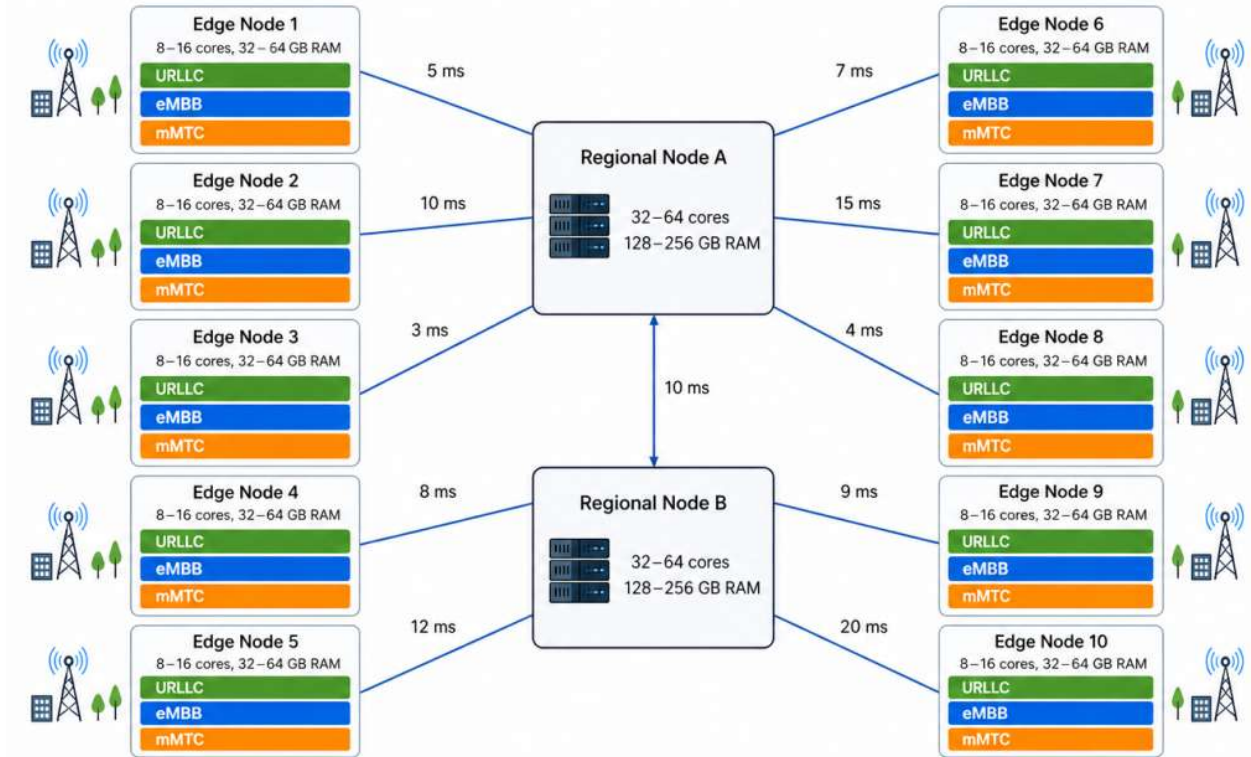


Fig. 2. Network topology

Table 1 Comparative results under single node failure

Method	Median Recovery Time (s)	SLA Viol. (%)	Rel. Cost
CGO	3.8	0	1.00
NAM	0.6	14	1.38
DLR	0.7	4	1.12

Additional experiments on 25-, 50-, and 100-node topologies confirmed scalability: median recovery time increased moderately from 0.7 s to 1.4 s, while centralized CGO exceeded 8.9 s. URLLC SLA violation rates remained below 6% for all evaluated topologies. Compared to an RL-based recovery strategy (RL-Orch), DLR showed 18–24% lower recovery latency due to elimination of inference and policy synchronization overhead.

5 Discussion

DLR achieves a balanced trade-off across three dimensions critical for 5G: recovery speed (0.7 s, 5.4× faster than CGO, enabled by limiting search to 2–3 hops), cost (only 12% above the CGO optimum; 19% and 10% below NAM and LOO respectively), and SLA compliance (4% URLLC violation rate vs. 14% for NAM). CGO’s 0% violation rate comes at the cost of 3.8 s recovery, which is unacceptable for URLLC.

A detailed comparison of DLR and CGO is provided in Table 2, and cost comparisons across all methods are illustrated in Fig. 3. The overall performance trade-offs are visualized in Fig. 4.

Extended experiments on larger topologies confirm that the distributed recovery strategy scales significantly better than centralized approaches (Table 2).

Table 2 DLR vs. CGO comparison

DLR	CGO
Local search (2–3 hops)	Global optimization
Reduced overhead costs	High overhead costs
Fast reconfiguration	Slower reconfiguration
Recovery time = 0.7 s	Recovery time = 3.8 s

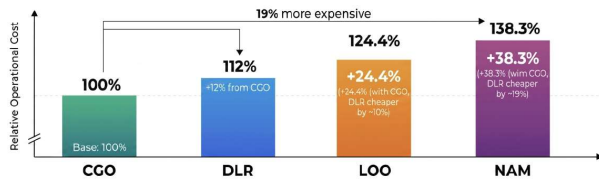


Fig. 3. Cost Comparison of Optimization: DLR and Other Solutions

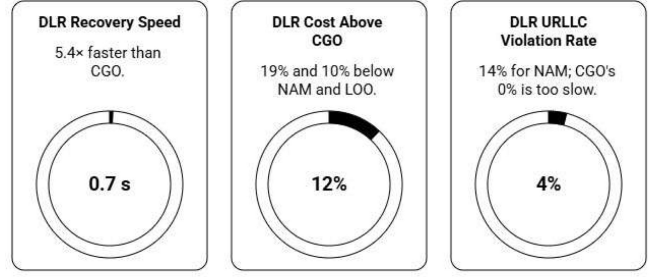


Fig. 4. DLR Performance vs. Alternatives

Three limitations are worth mentioning. First, evaluation is based entirely on a synthetic topology; real metropolitan deployments have more varied link latency, bursty traffic, and correlated node failures. Second, the migration cost model assumes linear state-transfer overhead, underestimating the cost of stateful VNFs (SGW, PGW) where live migration requires hypervisor checkpointing and session re-establishment. Third, the detection architecture is reactive, whereas proactive prediction using ML anomaly detection could trigger migration before thresholds are reached.

Conclusions

This paper proposes a holistic concept of local 5G network slice reconfiguration applicable to node failure and congestion scenarios. The three-level hierarchical architecture forms global orchestrators, regional managers, and local management agents that enable distributing control for fast and efficient recovery.

The paper formulates a multi-objective manager placement problem considering latency, load balancing, and inter-manager interaction simultaneously. Local recovery algorithms have been developed for both disruption categories (catastrophic failure and progressive overload), with considerations for slice-specific priorities and migration costs.

For cross-domain recovery, a lightweight distributed coordination protocol has been developed that coordinates all management entities without the scalability limitations of centralized management.

The proposed system achieves a 27% reduction in migration costs, URLLC restoration in under one second per ITU-R M.2083-0, support for heterogeneous nodes, and fault recovery automation.

Future work will focus on validating the framework in emulated cloud-native 5G environments based on Kubernetes and OpenStack, as well as integrating reinforcement learning mechanisms for predictive fault prevention and proactive migration planning.

References

- [1] Li W., et al. (2019). Latency-Optimal VNF Resource Allocation for 5G Backhaul Transport Network Slicing. *Applied Sciences*, Vol. 9, Iss. 4, 701; DOI: 10.3390/app9040701.
- [2] Ordóñez-Lucena J., et al. (2017). Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. *IEEE Communications Magazine*, Vol. 55, No. 5, pp. 80–87, DOI: 10.1109/MCOM.2017.1600935.
- [3] Dobreff G., Báder A. and Pašić A. (2025). A QoE and Availability-Aware Framework for Network Slice Placement and Resource Allocation. *IEEE Access*, Vol. 13, pp. 1481–1495, DOI: 10.1109/access.2024.3523900.
- [4] Gatzianas M., et al. (2021). Offline Joint Network and Computational Resource Allocation for Energy-Efficient 5G. *Applied Sciences*, Vol. 11, Iss. 22, 10547, DOI: 10.3390/app112210547.
- [5] Shinde S., Marabissi D. and Tarchi D. (2021). A network operator-biased approach for multi-service NF placement in 5G slicing. *Computer Networks*, Vol. 201, DOI: 10.1016/j.comnet.2021.108598.
- [6] Vidhya P., et al. (2025). Dynamic network slicing based resource management and VNF migration in 5G networks. *Computer Networks*, Vol. 259, DOI: 10.1016/j.comnet.2025.111064.
- [7] Di Mauro M., et al. (2025). Reliability and availability in virtualized networks: A survey. *arXiv*, DOI: 10.48550/arXiv.2503.22034.
- [8] Thien P., et al. (2024). Optimizing Placement and Scheduling for VNF by a Multi-objective Optimization Genetic Algorithm. *International Journal of Computational Intelligence Systems*, Vol. 17, DOI: 10.1007/s44196-024-00430-x.
- [9] Hirayama T., Jibiki M. and Kaffle V. (2023). Salience-based VNF placement method for evolving multiple network slices. *Computer Networks*, Vol. 236, DOI: 10.1016/j.comnet.2023.110002.
- [10] Benkacem I., Taleb T., Bagaa M. and Flinck H. (2018). Optimal VNFs Placement in CDN Slicing Over Multi-Cloud Environment. *IEEE Journal on Selected Areas in Communications*, Vol. 36, Iss. 3, pp. 616–627, DOI: 10.1109/jsac.2018.2815441.
- [11] Addad R., et al. (2020). Optimization Model for Cross-Domain Network Slices in 5G Networks. *IEEE Transactions on Mobile Computing*, Vol. 19, Iss. 5, pp. 1156–1169, DOI: 10.1109/tmc.2019.2905599.
- [12] Nguyen T., et al. (2023). Optimizing Resource Allocation and VNF Embedding in RAN Slicing. *IEEE Transactions on Network and Service Management*, Vol. 21, Iss. 2, pp. 2187–2199, DOI: 10.1109/tnsm.2023.3319309.
- [13] Abdellatif A., et al. (2023). Intelligent-Slicing: An AI-Assisted Network Slicing Framework for 5G-and-Beyond. *IEEE Transactions on Network and Service Management*, Vol. 20, Iss. 2, pp. 1024–1039, DOI: 10.1109/tnsm.2023.3274236.
- [14] Panitsas I., et al. (2025). SlicePilot: Demystifying Network Slice Placement in Heterogeneous Cloud Infrastructures. *arXiv*, DOI: 10.48550/arXiv.2509.18545.
- [15] Chiosi M., et al. (2012). *Network Functions Virtualisation: An Introduction*. *ETSI White Paper*.
- [16] Gember-Jacobson A., et al. (2014). OpenNF: Enabling innovation in network function control. *ACM SIGCOMM*.
- [17] Cohen R., Lewin-Eytan L., Naor J.S. and Raz D. (2015). Near optimal placement of virtual network functions. *Proc. IEEE INFOCOM*, pp. 1346–1354, DOI: 10.1109/INFOCOM.2015.7218511.
- [18] Venkatapathy S., et al. (2024). Slice-aware 5G network orchestration framework based on dual-slice isolation (D-SIMS). *Scientific Reports*, Vol. 14, Article number: 18623, DOI: 10.1038/s41598-024-68892-9.
- [19] ETSI (2014). *Network Functions Virtualisation (NFV); Management and Orchestration*. GS NFV-MAN 001 v1.1.1. Sophia Antipolis.
- [20] Halabian H. (2019). Distributed Resource Allocation Optimization in 5G Virtualized Networks. *IEEE Journal on Selected Areas in Communications*, Vol. 37, Iss. 3, pp. 627–642, DOI: 10.1109/jsac.2019.2894305.
- [21] Motalleb M., et al. (2023). Resource Allocation in an Open RAN System Using Network Slicing. *IEEE Transactions on Network and Service Management*, Vol. 20, Iss 1, pp. 627–642, DOI: 10.1109/tnsm.2022.3205415.
- [22] Heller B., Sherwood R. and McKeown N. (2012). The controller placement problem. *Proc. ACM SIGCOMM HotSDN*, pp. 7–12, DOI: 10.1145/2342441.2342444.
- [23] Sahu V., Sahu N. and Sahu R. (2024). Challenges and opportunities of 5G network: A review. *American Journal of Electrical and Computer Engineering*, Vol. 8, No. 1, pp. 11–20. DOI: 10.11648/j.ajece.20240801.12.
- [24] ETSI (2016). *Network Functions Virtualisation (NFV); Os-Ma-nfvo Interface and Information Model Specification*. GS NFV-IFA 013 v2.1.1. Sophia Antipolis.
- [25] ITU-R (2023). *IMT-2030 Framework*. Recommendation ITU-R M.2160-0. Geneva.
- [26] Herrera J. G. and Botero J. F. (2016). Resource allocation in NFV: A comprehensive survey. *IEEE Transactions on Network and Service Management*, Vol. 13, No. 3, pp. 518–532, DOI: 10.1109/TNSM.2016.2598420.
- [27] Di Mauro M., et al. (2025). Reliability and availability in virtualized networks: A survey on standards, modeling approaches, and research challenges. *arXiv preprint arXiv:2503.22034*, DOI: 10.48550/arXiv.2503.22034.
- [28] 3GPP (2021). *Service requirements for the 5G system*. TS 22.261, Release 16.
- [29] Moens H. and De Turck F. (2014). VNF-P: A model for efficient placement of virtualized network functions. *Proc. CNSM*, Rio de Janeiro, pp. 418–423. DOI: 10.1109/CNSM.2014.7014205.
- [30] 3GPP (2023). *System architecture for the 5G System (5GS)*. TS 23.501, Release 17.
- [31] 3GPP (2018). *Telecommunication management; Study on management and orchestration of network slicing*. TR 28.801, Release 15.
- [32] Sudhamani C., et al. (2023). A survey on 5G coverage improvement techniques. *Sensors*, Vol. 23, No. 4, Article 2356, DOI: 10.3390/s23042356.

Розподілений фреймворк багаторівневої оркестрації для оперативного відновлення віртуалізованих мережевих функцій

Суліма С. В., Карашевич Є. Д.

Швидке впровадження слайсингу мережі 5G на віртуалізовану інфраструктуру призвело до виявлення ключової прогалини в існуючих системах відновлення, через яку централізовані підходи до оркестрації не здатні забезпечити виконання вимог щодо відновлення за менш ніж секунду для слайсів наднадійного зв'язку з низькою затримкою (URLLC, *Ultra-Reliable Low Latency Communications*), тоді як прості евристики не можуть спостерігати неоднорідність угоди про рівень обслуговування (SLA, *Service Level Agreement*) між типами слайсів.

У статті розглянуто систему DLR (*Distributed Latency-aware Recovery*), розподілене відновлення з урахуванням затримки) – трирівневу ієрархічну архітектуру оркестрації для швидкого відновлення віртуалізованих мережевих функцій (VNF, *Virtualized Network Function*) у сегментованих мережах 5G. Система базується на декомпозиції керування на глобальний оркестратор, регіональні менеджери слайсів (RSM, *Regional Slice Managers*) та локальні агенти моніторингу (LMA, *Local Monitoring Agents*), де рішення щодо відновлення приймаються локально на основі порушень SLA без будь-яких центральних вузлів місць. Розміщення RSM сформульовано як багатоцільова оптимізаційна задача з мінімальною затримкою від вузла до менеджера, дисбалансом навантаження та накладними витратами на координацію між менеджерами, при цьому

розв'язувач генетичних алгоритмів забезпечує на 11–33% кращі комбіновані показники якості, ніж базові стратегії. Виділені алгоритми відновлення справляються як з катастрофічною відмовою вузлів через стрибок обмеження та пріоритетний порядок перепризначення VNF, так і з наростаючими перевантаженнями, використовуючи проактивну міграцію на основі оцінювання із заданим граничним значенням для розширення або скорочення періоду міграції.

Сформульовано нову чотирикомпонентну функцію вартості міграції VNF, засновану на спільному врахуванні витрат на обчислення, мережу, міграцію та штрафні витрати за SLA, що призводить до зменшення кількості кроків міграції до 44% для випадків наростаючого перевантаження. Для міждомених сценаріїв полегшений консенсусний протокол координує дії із сусідніми RSM без центрального арбітражу.

Результати моделювання на синтетичній топології з 10 вузлами показують, що DLR має середній час відновлення 0,7 с (у 5,4 раза швидше, ніж CGO), при цьому порушення SLA URLLC обмежені 4% (порівняно з 14% для найближчого доступного збігу) та знижують експлуатаційні витрати на 27% порівняно з підходами, що враховують лише затримку. Ці результати формують DLR як масштабоване та економічно обізнане рішення, засноване на SLA, для стійкого управління 5G NS з потенціалом для прогнозовної міграції на основі машинного навчання та розширення для кількох операторів для майбутнього розгортання 6G.

Ключові слова: віртуалізація мережевих функцій (NFV); мережевий слайсинг; локальна реконфігурація; розподілене відновлення; SLA-орієнтована оптимізація