UDC 004.7:621.396 **doi: 10.26906/SUNZ.2026.2.247**

Svitlana Sulima

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

LOCAL RECONFIGURATION OF 5G NETWORK SLICES UNDER NODE FAILURES AND OVERLOADS

Abstract. Relevance. The rapid deployment of 5G networks and the widespread use of network slicing create new challenges for ensuring service reliability and resilience. Virtualized infrastructures based on Network Functions Virtualization increase flexibility but also introduce higher failure rates and performance variability. In such environments, centralized recovery mechanisms often fail to meet strict latency requirements, especially for ultra-reliable low-latency communication services. **Research object**: The research object is the process of failure recovery and resource reconfiguration in virtualized 5G network slicing environments under node failures and progressive overload conditions. **Purpose**: The purpose of the study is to develop an efficient distributed method for local reconfiguration of network slices that enables rapid recovery of virtual network functions while considering slice priorities, migration costs, and latency constraints. **Results**. A distributed local reconfiguration (DLR) framework is proposed, based on a hierarchical architecture consisting of a global orchestrator, regional slice managers, and local monitoring agents. The approach introduces a multi-objective optimization model for slice manager placement and a unified migration cost function that considers computational, network, disruption, and SLA penalty components. Localized recovery algorithms are developed to handle both catastrophic node failures and progressive overload scenarios while prioritizing slices according to their service requirements. **Conclusions**. The proposed distributed slice recovery framework enables fast and scalable reconfiguration of 5G network slices under failure conditions. By combining slice-aware prioritization, cost-aware migration decisions, and localized management, the approach improves recovery speed and operational efficiency while preserving service quality for latency-critical applications.

Keywords: 5G networks; network slicing; network function virtualization; failure recovery; distributed network management; virtual network function migration; slice-aware resource management; resilience; service level agreement; distributed local reconfiguration.

Introduction

Formulation of the problem. Visualize a single server residing in a solitary rack located somewhere in the city, currently balancing three unique existences. One existence is that of providing haptic feedback for a surgeon while he utilizes a robotics apparatus to execute a procedure in a faraway healthcare facility. Another existence is that of assisting a fleet of self-driving trucks to communicate to one another to execute a lane change on the freeway. And yet another existence is that of providing streaming HD movies to an endless number of people who are killing time with their phones while waiting for their next task.

Now picture that one of the three servers dies suddenly, and the network is suddenly placed into triage mode regarding life and death. If the network is unable to re-establish the connection to that server within a fraction of a millisecond, the surgeon cannot see in the real world. If the network does not reestablish the connection in that timeframe, the rest of the network will become unstable, and lives will be endangered because of it. On the other hand, individuals streaming movie content from their phones may be able to tolerate a 3- or 4-second delay before they begin to notice something is awry.

Thus, the core problem of managing a 5G network is how to address the same issue with three different outcome requirements, using three different methodologies, all occurring at the same time. Centralized, thinking systems will not be able to share the electronic capacity needed to meet the timing requirement of the robotics issue while maintaining an efficient operating cost to deliver cellular phone data.

The use of operational network slicing in 5G currently is real, and presents an operationally challenging issue yet to be resolved in today's market. The ITU specifies three different service categories within 5G: enhanced mobile broadband (eMBB), ultra reliable low latency communications (URLLC), and massive machine type communication (mMTC) [1]. Each of the three service categories can share physical infrastructure to operate independently while still supporting the requirements of the service categories within each slice. Network slicing accomplishes this through the creation of a logically isolated (virtualized) network for each family of services operating over the same physical infrastructure [2, 3]. The 3GPP Technical Specification 23.501 defines how network slices are logical networks comprised of defined capabilities or features [4] – an example of where there are significant differences in the end-to-end latency associated with services (e.g. autonomous driving requiring ≤1ms maximum end-to-end latency with reliability at 99.9999%, and IoT sensory networks possibly exceeding between 10s to hundreds of billions of users within the same service family) [5, 6].

Recently, mobile data usage has consistently exceeded expectations as new types of services are rolled out that change how data is being used. Network Functions Virtualization (NFV) is now the way in which network slicing will be executed by using general-purpose server hardware instead of specialized devices to implement telecom networks. Although NFV provides substantial savings, it also introduces weaknesses to the reliability of networks that do not exist with traditional carriers. For example, generic server hardware has a significantly greater failure rate than dedicated telecommunication hardware, and the virtual machine layers (hypervisors) create new vulnerabilities for security breaches while using a commodity hardware platform compared with a dedicated hardware platform. Furthermore, Virtual

Network Functions (VNFs) running on the same physical hardware may create unexpected interactions between the VNFs, complicating the service performance or reliability [7][8][9].

In this case, failure is considered a good frequency as it is guaranteed to happen as well as certain measures for mitigation are in place. A single physical node going down can affect dozens of virtual network functions belonging to slices with significantly varying recovery priorities. Relevant approaches to the issue exist in two groups, and neither may be considered satisfactory. Centralized orchestration can produce globally optimal recovery plans, but in doing so, it uses time resources that URLLC slices cannot spare, and it does not scale well with large networks [10]. Proactive redundancy — preplacing backup resources everywhere — keeps recovery fast but exactly wastes the kind of physical capacity, which network slicing exists to consolidate[11]. That is missing is an approach that is distributed enough to be manageable as networks expand, resource-efficient enough to be economically viable, and fast enough to satisfy the most demanding slices.

The paper presents a method to reconstruct data from a location that is close to where a failure occurs by using regional slice managers located throughout a network as part of a recovery process. Each slice manager is responsible for a specific geographical area, and can retrieve and migrate from a host resource to another within that geographical region without waiting for a view of the overall network. If a node fails, the regional slice manager immediately knows what slices are affected, can sort those slices based on the severity of the failure, is able to identify which hosts within the geographical region could be used for the new virtual functions, can calculate the costs associated with migrating, and then will begin to migrate the virtual functions all before the orchestrator is even aware of what has happened. The paper describes three concepts in detail; a hierarchical distributed architecture based on literature on the placement of SDN controllers [12, 13, 14]; a cost model that incorporates factors beyond latency into the determination of the cost of migrating [15, 16, 17]; and a recovery mechanism that treats sudden failures and progressive overload as distinct problems requiring distinct strategies [18, 19].

There are three different interlinked aspects present in the scientific novelty of the Distributed Local Reconfiguration (DLR) approach and not combined so far in any of the provided solutions:

1. New approach for placing managers (RSM placement). All previous works on SDN controller placement either minimized latency to nodes or load balancing. However, the author is first in NFV context to introduce a third criterion – latency between the managers themselves as a separate optimization goal. For failure recovery purposes, this is very important, because if the managers are located far away, the cross-domain recovery will be too long for URLLC.

2. Unified migration cost model. Existing works consider up to at most one or two componentsonly state transfer time, only resource constraints, only reliability constraints. The paper is the first to build a single 4-components cost function (compute + network + migration disruption + SLA penalty), which allows migration decisions to be made considering all these aspects together.

3. Dealing with two failure types in one slice-aware framework. The vast majority of NFV studies consider only catastrophic node failures. As a result, progressive overload is either neglected or handled by unassociated solutions. Also, there are no known works that distinguish the recovery priorities based on the slice type (URLLC vs eMBB vs mMTC). The paper, for the first time ever, combines both scenarios into a single algorithm where the recovery depends explicitly on slice priority as per 3GPP specification.

Current State of Web Accessibility.

1. Network Slicing in 5G

The concept of network slicing has moved from being a theoretical idea in 5G into a concrete part of present-day mobility networks. This has been accomplished primarily through many years of effort from the 3rd Generation Partnership Project (3GPP), which provided the basis for both the system architecture (as outlined in TS 23.501) [4] and the management and orchestration framework (as detailed in TR 28.801) [20]. As a result, operators can now develop end-to-end “slices” that span the last two, three or four layers of the 5G ecosystem: from the Radio Access Network (RAN) to the transport layers (T-Layers) to the core network.

Researchers like Ordonez-Lucena et al. [21] have offered a broad view of how network slicing works in practice. They note that while technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV) serve as essential foundations, real-world deployment depends on intricate orchestration systems. These systems must handle every stage of a slice’s life—from creating and configuring it to continuously monitoring and eventually shutting it down. The key challenges, they argue, often revolve around maintaining strong isolation between slices, managing resources effectively, and ensuring that different slices can work in harmony without competing for resources.

Rost et al. [22] approached slicing from a scalability and flexibility standpoint. Their research focused on how 5G networks can dynamically allocate resources among multiple slices without causing interference. Using simulations, they demonstrated that with proper architecture, it’s possible to meet the needs of very different services—say, a high-speed video stream and a low-latency industrial application—at the same time.

Meanwhile, Foukas et al. [23] placed special emphasis on RAN slicing. They observed that while slicing in the core network has matured thanks to NFV, RAN slicing remains more difficult. That’s largely because the radio environment requires fine-grained control and real-time coordination, making efficient resource distribution a tougher technical challenge.

One noticeable gap in much of the literature is how slicing behaves when things go wrong. Most studies focus on how to set up slices and make them run efficiently, but they rarely address what happens during faults or network failures. For instance, Ksentini and Nikaein [24] discuss RAN slicing with resource abstraction but stop short of failure management, while Zhang et al. [25] explore reinforcement learning methods for dynamic

resource allocation under the assumption that everything operates smoothly. That leaves a clear opportunity for research into recovery and resilience mechanisms within 5G network slicing.

2. NFV Resilience and Failure Recovery

NFV resilience has been examined primarily through the prism of cloud-based deployments, with a strong emphasis on how to maintain service continuity under frequent infrastructure failures. The ETSI NFV architectural framework [26] outlines high-level principles for achieving resilience but intentionally leaves concrete realization choices to individual operators and vendors, reflecting the diversity of deployment environments. In parallel, ETSI NFV-REL 001 [27] refines this view by formalizing reliability and availability requirements and by explicitly recommending redundancy-based protection mechanisms as the baseline approach for sustaining target service levels.

Han et al. [8] provide a detailed discussion of NFV challenges and opportunities, identifying reliability as a central concern rather than a secondary design objective. They argue that while virtualization greatly simplifies rapid deployment, scaling, and flexible placement of network functions, it also introduces novel failure modes compared to traditional, purpose-built telecom appliances. Empirical observations reported in their work indicate that commodity cloud servers experience failures roughly one to two orders of magnitude more frequently than dedicated carrier-grade hardware, which fundamentally shifts the reliability engineering problem toward software- and platform-level mitigation.

This tension is articulated clearly in the foundational NFV white paper by Chiosi et al. [11], where the authors highlight the gap between carrier-grade availability targets (on the order of 99.999%) and the typical enterprise-grade availability (around 99.9%) delivered by commodity hardware platforms. To bridge this gap, they advocate multi-layer redundancy and fast failover mechanisms spanning the NFV infrastructure, VNFs, and management layers, while noting that such designs inevitably increase resource footprint and operational complexity.

Subsequent research proposes more specialized failure recovery mechanisms. Cohen et al. [15] formulate VNF placement under reliability constraints as an optimization problem that aims to minimize expected service disruption in the presence of node failures. Their approach relies on computing backup VNF placements offline so that failover can be executed quickly at runtime, but this strategy requires reserving substantial standby capacity, which may be costly in resource-constrained environments. Gember-Jacobson et al. [28] introduce OpenNF, a control framework that enables fine-grained manipulation of network function state, including live migration of stateful VNFs with disruption on the order of hundreds of milliseconds for typical state sizes, though the reliance on centralized control inherently limits scalability in very large deployments.

Rajagopalan et al. [29] address elasticity through Split/Merge, a system that allows VNFs to be dynamically partitioned across multiple servers or consolidated onto fewer servers to adapt to load variations, focusing primarily on performance and resource efficiency rather than explicit fault tolerance. Their results show that elastic execution of virtual middleboxes can effectively track fluctuating demand, suggesting that similar mechanisms could be extended to support resilience-aware scaling policies.

A common limitation across much of the NFV resilience literature is the implicit assumption that services share homogeneous reliability and recovery requirements. With the advent of network slicing in 5G, this assumption becomes problematic, as slices support heterogeneous service classes with distinct resilience profiles: for example, URLLC slices require sub-second recovery, whereas eMBB slices can tolerate several seconds of degraded performance. This heterogeneity motivates the need for slice-aware recovery mechanisms that explicitly account for per-slice resilience objectives—a gap that the present work is designed to address.

3. Distributed Management in Virtualized Networks

Although theoretically optimal, centralized orchestration suffers in scalability and suffers from latency in large-scale networks. The controller placement problem in SDN has been studied thoroughly on this basis.

Heller et al. [12] initiated research on controller placement in SDN, with the identification of a k-center optimization problem minimizing the maximum node-to-controller latency. On real network topologies, their results show that the communication latency among controllers remains high in large networks even with minimized maximum node-to-controller latency.

Hock et al. [13] extend this work with multi-objective optimization, considering both latency and resilience. They formulate controller placement as a Pareto optimization with three conflicting objectives: minimizing average latency, maximizing resilience from failures, and balancing the load on controllers. Their results show that single objective optimization leads to brittle solutions with poor performance when multiple criteria are important.

Lange et al. [14] propose heuristic algorithms for controller placement in large-scale networks, demonstrating that genetic algorithms and simulated annealing approaches can yield near-optimal solutions in orders of magnitude less time compared to exact optimization methods. The heuristic algorithms provide controller placement solutions within 5% of optimal solutions for networks with a couple of hundreds of nodes in a matter of seconds rather than hours required by exact algorithms.

While these works address control plane placement, they do not explicitly handle data plane VNF recovery or migration cost optimization. More relevant to our work, Baumgartner et al. [30] investigate mobile core network virtualization with combined VNF placement and topology optimization. They formulate a joint optimization problem with initial placement and reconfiguration cases, showing by simulations that considering reconfiguration in advance is more resource-efficient than treating it as a reactive case.

Moens and De Turck [16] propose VNF-P, a model for efficient VNF placement with respect to the resource

constraints and the topology of the service chain. The model includes latency constraints and validates that intelligent placement reduces resource consumption by 20-30% compared to naive placement. However, they focus on the initial placement only and not on the dynamic reconfiguration.

Mehraghdam et al. [17] propose a method to specify and place service chains, which are sequences of VNFs. For this purpose, they propose formal models of service chain specification and algorithms for optimal chain placement. However, this work does not consider failures and migrations at runtime.

4. Virtual Network Reconfiguration

Virtual network reconfiguration is studied in virtual network embedding (VNE). Based on a comprehensive survey of network virtualization, Chowdhury and Boutaba [31] identify reconfiguration as a core challenge which has been minimally addressed relative to embedding.

Fajjari et al. [18] propose a greedy algorithm for virtual network reconfiguration, where the initial embedding can be changed if some nodes and links run out of resources or require rearrangement. Their algorithm greedily migrates virtual nodes and links repeatedly and improves the results in terms of resource utilization, but no failure recovery scenarios are considered.

Beloglazov and Buyya [32] dynamically investigate consolidation of virtual machines in cloud data centers with energy efficiency objectives. Their adaptive heuristics are designed to continuously monitor the physical resource utilization levels, triggering virtual machine migration actions aimed at consolidating the virtual machine workload on a minimum number of physical servers in the data center with a view to reducing energy consumption. Their VM migration techniques are applicable to our work, even though their focus is not on failure recovery.

Qu et al. [33] formulate reliability-aware network service chain provisioning in NFV-enabled enterprise datacenters. They propose algorithms that place service chains proactively considering possible failures, and ensure that backup resources can be used for a quick recovery. While providing a high level of reliability, the proposed approach incurs high resource over-provisioning, ranging between 30-50% overhead.

Some of the major gaps in knowledge still exist despite much research that has been conducted to date:

• Lack of Slice Aware Recovery: All VNFs are treated the same and therefore, do not take into consideration different slice resiliency requirements and recovery priorities as specified by 3GPP standards [4,20].

• Limited Migration Cost Models: Most of the current research has an overly simplified view of migration costs. Gember-Jacobson et al. [28] looked only at state moving times, but did not develop a complete view of the computational overhead and cascading effects associated with those movements.

• Centralized Bottlenecking: Although there are several studies which examine distributed controller placement [12,13,14], all of the NFV orchestration approaches proposed to date are central in nature as noted by the ETSI MANO specification [34]. Therefore, the latencies associated with these centralized solutions would not meet URLLC requirements.

• Inability to Handle Overloads: Most of the research that has been published about failure recovery has focused on catastrophic failures. Failure recovery to date has ignored progressive overload scenarios and as more data centers and shared infrastructures become common, progressive overload recovery will become even more relevant to the state of the art.

This paper fills these gaps with a framework of distributed, cost-aware, slice-specific recovery in 5G network slicing environments.

Key difference from closest competitors are summarized in Table 1.

Table 1 – **Key difference**

Feature	CGO	NAM	DLR (proposed)
No central orchestrator required	✗	partially	✓
Slice-aware prioritization	✗	✗	✓
Full migration cost model	✓	✗	✓
Sub-second URLLC recovery	✗	partially	✓
Progressive overload handling	✗	✗	✓
Inter-manager coordination	—	—	✓ (1 round)

Research Objectives.

Our contributions include:

• Formulation of the slice manager placement problem as multi-objective optimization considering latency, load distribution, and coordination between the managers, considering SDN controller placement work [12,13,14] in the NFV contexts.

• Development of localized reconfiguration algorithms that can work for both forms of reconfiguration: catastrophic failures and progressive overload, based on the lessons from virtual network reconfiguration [17,18] and live migration [28,29].

• Migration cost models, which take into account the computational resources used during migration, the bandwidth costs caused by migration, and any service disruption caused by migration, are introduced on top of VNF placement optimization [15,16,17].

• Experimental validation over different failure scenarios with different slices shows better performance compared to the centralized and benchmarks.

Main material

The optimization problem we consider is inherently complex: it is large-scale, mixed-integer, and multi-objective, and it has to be solved under very tight recovery-time constraints, especially for URLLC slices where delays are unacceptable. In practice, trying to solve the full problem centrally every time a failure occurs is not realistic, because the computation would take too long and would directly conflict with the low-latency guarantees expected in 5G systems.

To cope with this, we introduce a distributed, slice-aware recovery framework that restructures how the network reacts to failures. Instead of relying on a single, global decision point, we assign responsibilities to slice

managers that each have only partial knowledge of the overall topology, allowing them to localize and speed up the failure response. Within this framework, slices are not treated equally: they are prioritized according to the criticality of their SLAs, so more demanding services are handled first. We further separate placement, routing, and migration decisions and organize them in a hierarchical optimization process, which reduces complexity while still enabling coordinated recovery.

This design is intentionally consistent with existing ETSI NFV MANO concepts and with architectures that use multiple distributed SDN controllers. At the same time, it makes explicit room for practical aspects that are often overlooked, such as the cost of VNF migrations, the specific semantics of different failure types, and the heterogeneous resilience requirements of different slices.

Fig. 1 illustrates the proposed architecture, which consists of three logical layers:

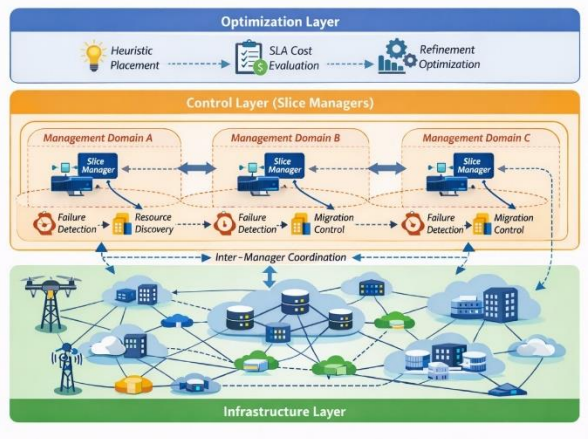

Fig. 1. Architecture for the solution

1. Infrastructure Layer

Physical nodes and links as defined by the substrate network $SN = (N, NE)$.

2. Control Layer (Slice Managers)

A set of distributed slice managers $M \subseteq N$, each responsible for a management domain D_m. Managers perform:

- Local failure detection,
- Candidate resource discovery,
- Migration orchestration,
- Inter-manager coordination for cross-domain recovery.

3. Optimization Layer

A hierarchical decision process combining:

- Local heuristic decisions for fast reaction,
- Lightweight optimization for placement refinement,
- SLA-aware cost evaluation.

Each slice manager operates autonomously for failures within its domain while coordinating with neighboring managers when migration targets lie outside D_m.

Upon detecting an anomaly, the responsible slice manager classifies the event as one of the following:

1. Catastrophic Node Failure

Immediate service disruption for all VNFs placed on n^{fail}.

Recovery must be reactive and fast.

2. Progressive Node Overload

Detected via monitoring of resource utilization trends:

$$\sum_{s,v} x_{n,v}^{s} \cdot D_v^{comp} \to C_n^{comp}.$$

This scenario enables proactive migration, reducing SLA penalties.

The failure type determines the urgency, optimization depth, and migration scope.

The recovery process follows four stages:

Stage 1: Affected Slice Identification.

For a failed or overloaded node n, the manager identifies all impacted slices:

$$S^{aff} = \{s \in S \mid \exists v \in V_s : x_{n,v}^{s} = 1\}.$$

Slices are sorted by priority level π_s, ensuring URLLC slices are handled first.

Stage 2: Candidate Node Filtering.

For each disrupted VNF v, a candidate node set $N_v^{cand} \subseteq N$ is constructed based on:

- Resource feasibility,
- Administrative suitability $\text{suit}_{n,v}^{s}$,
- Latency feasibility with respect to L_s^{max},
- Manager proximity (prefer nodes within D_m).

This pruning step dramatically reduces the solution space.

Stage 3: Cost-Aware Migration Decision.

For each candidate migration $n \to n'$, the manager computes a local reconfiguration cost:

$$\Delta C = \Delta\text{Cost}_{compute} + \Delta\text{Cost}_{network} + \Delta\text{Cost}_{migration} + \Delta\text{Penalty}_{SLA}.$$

Migration decisions are selected using:

- Greedy minimization for URLLC slices,
- Multi-criteria ranking for eMBB and mMTC slices.

For progressive overload, migration is triggered before capacity violation, minimizing disruption time.

Stage 4: Inter-Manager Coordination.

If no feasible candidate exists within D_m, the manager:

1. Requests candidate resources from neighboring managers,

2. Exchanges summarized state information (capacity, latency bounds),

3. Negotiates placement using a lightweight consensus protocol.

This avoids global state synchronization while ensuring feasibility.

To balance optimality and responsiveness, we adopt a two-level optimization approach:

Local Optimization (Fast Reaction)

- Scope: Single failure event
- Variables: Subset of $x_{n,v}^{s}$, $f_{(n_1,n_2),e}^{s}$
- Method: Heuristic + constrained local search
- Time scale: milliseconds to seconds

Global Refinement (Optional)

- Triggered during low-load periods
- Re-optimizes placements to reduce fragmentation
- Improves long-term cost efficiency

This separation ensures SLA compliance without sacrificing overall efficiency.

Algorithm 1: Distributed Slice Recovery

1. Detect failure at node n
2. Classify failure type
3. Identify affected slices S^{aff}
4. Sort S^{aff} by priority π_s
5. For each slice $s \in S^{aff}$:
   - Identify disrupted VNFs
   - Generate candidate nodes
   - Evaluate migration costs
   - Select minimal-cost feasible migration
6. Coordinate with neighboring managers if required
7. Enforce updated placement and routing
8. Monitor post-recovery SLA compliance

The proposed approach is designed to work in realistic, large-scale 5G environments, where many slices and network functions coexist. It does so while explicitly taking into account both the overhead introduced by VNF migrations and the fact that different slices have different performance and resilience requirements. As a result, it can trigger fast, localized recovery actions when failures occur, without violating the SLAs associated with each slice. In addition, the same framework can naturally handle both proactive strategies (anticipating problems before they occur) and reactive strategies (responding after a failure has been detected).

Motivated by distributed SDN control architectures [12,35] and ETSI NFV MANO principles [34], we propose a three-tier hierarchical management architecture (Fig. 2) designed for low-latency and scalable failure recovery in network slicing environments.

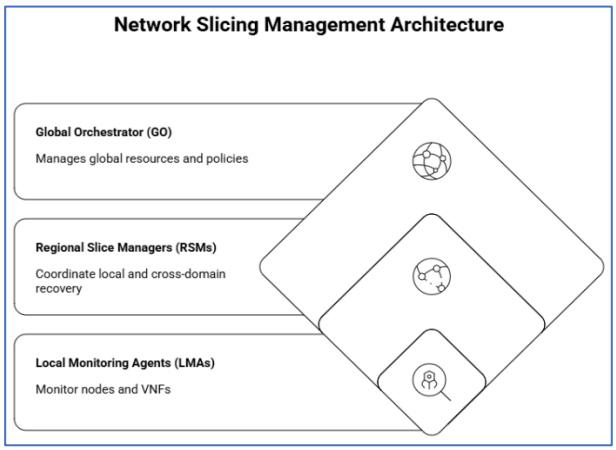

Fig. 2. Control architecture

Tier 1: Global Orchestrator (GO)

The GO maintains a coarse-grained, global view of network resources and performs: Initial network slice instantiation in accordance with ETSI NFV-IFA specifications [36]; Long-term capacity planning and policy enforcement; Therefore, to prevent centralization bottlenecks, the GO has no involvement in the real-time failure recovery process.

Tier 2: Regional Slice Managers (RSMs)

RSMs are deployed at selected nodes with the help of an optimization procedure. Every RSM; It has developed a very precise state of VNFs and physical resources within its domain; Coordinating local failure recovery; It also synchronizes the state with remote peers, i.e., the adjacent RSMs, for cross-domain failure recovery.

Tier 3: Local Monitoring Agents (LMAs)

LMAs are deployed on every physical node and; Continuously track nodes and VNFs for any faults; Perform failure and overload detection; Notify the responsible RSMs and actuation on migration commands.

This hierarchical decomposition makes it possible to perform low-latency recovery required by the URLLC slices [1,21] while alleviating the scalability and reliability challenges of the centralized management of NFV instances [10,8].

Building on SDN controller placement studies [12–14], we formulate RSM placement as a multi-objective optimization problem.

Let the physical network be $SN = (N, N_E)$, and let $|M|$ denote the desired number of RSMs. Binary variable $p_n \in \{0,1\}$ indicates whether node $n \in N$ hosts an RSM, and $\pi_{n,m} \in \{0,1\}$ indicates whether node n is assigned to manager m.

Objective 1: Minimize Maximum Node-to-Manager Latency

$$U^{\text{latency}} = \max_{n \in N} \min_{m \in M} \{L_{n,m} \cdot \pi_{n,m}\}.$$

Objective 2: Balance Manager Load

Following [14], load imbalance is defined as:

$$U^{\text{imbalance}} = \max_{m \in M} \text{load}_m - \min_{m \in M} \{\text{load}_m : \text{load}_m > 0\},$$

where:

$$\text{load}_m = \sum_{n \in N} \pi_{n,m}.$$

Objective 3: Minimize Inter-Manager Communication Latency

To support coordinated failure recovery, we introduce:

$$U^{\text{inter-latency}} = \max_{\substack{m_1, m_2 \in M \\ m_1 \neq m_2}} L_{m_1,m_2}$$

Combined Objective

$$\min_{p_n, \pi_{n,m}} U_{\text{total}} = w^{\text{lat}} U^{\text{latency}} + w^{\text{imbal}} U^{\text{imbalance}} + w^{\text{inter}} U^{\text{inter-latency}}$$

Subject to:

$$\sum_{n \in N} p_n = |M|$$

$$\sum_{m \in M} \pi_{n,m} \geq 1 \forall n \in N$$

$$\pi_{n,m} \leq p_m \forall n \in N, m \in M.$$

The NP-hard problem is solved using a genetic algorithm, following [14]:

- Encoding: Binary vector p of length $|N|$;
- Fitness: U_{total};
- Selection: Tournament selection;
- Crossover: Uniform crossover with constraint-preserving repair;
- Mutation: Random swaps between manager and non-manager nodes.

Unlike prior controller placement formulations [12–14], our approach explicitly optimizes inter-manager latency, which is critical for coordinated, low-latency slice recovery.

Each LMA performs continuous monitoring using threshold-based detection inspired by dynamic VM consolidation systems [32].

Catastrophic Failure:

A node is declared failed if its LMA becomes unresponsive for T^{timeout}(e.g., 100 ms), consistent with high-availability practices [7].

Progressive Overload:

An alert is triggered if:

$$\rho_n^{\text{CPU}} > \theta_{\text{high}}^{\text{CPU}} \text{or } \rho_n^{\text{mem}} > \theta_{\text{high}}^{\text{mem}}.$$

Thresholds are slice-specific, with stricter limits for URLLC slices [4].

The RSM classifies overloads as:

- Transient spike: utilization falls below θ_{low} within $T^{\text{transient}}$;
- Sustained overload: utilization exceeds θ_{high} for $T^{\text{sustained}}$;
- Imminent failure: utilization exceeds θ_{critical}.

Upon detecting failure of node n^{fail}, the responsible RSM executes the following steps.

Step 1: Identify Affected VNFs

$$V^{\text{fail}} = \{(s, v): x^s_{n^{\text{fail}},v} = 1\}.$$

Step 2: Slice-Aware Prioritization

$$V^{\text{fail}}_{\text{sorted}} = \text{sort}(V^{\text{fail}}, (\pi_s, \text{criticality}_v)).$$

Step 3: Localized Candidate Discovery

$$N^{\text{cand}}_{s,v} = \{n: d(n, n^{\text{fail}}) \le r^{\max}, \text{suit}^s_{n,v} = 1, \text{has\_capacity}(n, D_v^{\text{comp}})\}.$$

with $r^{\max} = 2$ hops for URLLC and $r^{\max} = 4$ for eMBB slices.

Step 4: Recovery Cost Computation

$$\text{Cost}_{s,v}(n') = c^{\text{comp}} D_v^{\text{comp}} + c^{\text{net}} \Delta_{\text{BW}}(n', s, v) + \text{Penalty}_{\text{latency}}(n', s, v).$$

Step 5: Greedy Assignment

VNFs are greedily placed on the minimum-cost feasible node; failure triggers regional reconfiguration.

Localized, hop-constrained recovery dramatically reduces decision latency compared to global optimization [15,16], enabling compliance with URLLC recovery bounds [4,20].

For sustained overload on node n^{over}, the RSM performs incremental migration.

VNF Scoring

$$\text{score}_{s,v} = \alpha_1(1 - \pi_s) + \alpha_2 U^{\text{actual}}_{s,v} + \alpha_3(1 - \text{statefulness}_v).$$

Incremental Migration

VNFs are migrated in descending score order until:

$$\rho_{n^{\text{over}}}^{\text{CPU}} \le \theta_{\text{target}}^{\text{CPU}} (\theta_{\text{target}}^{\text{CPU}} = 0.7).$$

Adaptive halting prevents excessive migrations, a known issue in reactive resource management [32].

If local resources are exhausted, RSMs engage in lightweight coordination:

1. Broadcast recovery request (resource, latency, and priority constraints);

2. Parallel candidate search by neighboring RSMs;

3. Minimum-cost selection and coordinated migration.

This single-round protocol enables fast cross-domain recovery without centralized computation.

For complex scenarios, recovery is formulated as an optimization problem.

Decision Variables

- $x'^s_{n,v}$: post-recovery placement;
- $\delta^s_{v,n\to n'}$: migration indicator.

Objective

$$\min \sum_{s,v,n,n'} \delta^s_{v,n\to n'} C^{\text{mig}}_{v,n\to n'} + \lambda \mid \text{Cost}^{\text{new}}_{\text{oper}} - \text{Cost}^{\text{old}}_{\text{oper}} \mid$$

Subject to migration continuity, concurrency limits, and slice-specific recovery deadlines [4].

Solution: MILP for small instances (≤ 20 VNFs); greedy heuristics [14,18] for large-scale deployments.

To evaluate the effectiveness of the proposed distributed local reconfiguration (DLR) method, a synthetic network topology representing a 5G deployment in a metropolitan area was used. The network consists of 10 physical nodes (servers/data centers), including mixed edge and regional nodes. Channel delays vary from 1 ms (intra-regional) to 20 ms (inter-regional). Node capacities are heterogeneous: edge nodes have 8-16 processor cores and 32-64 GB of RAM, while regional nodes have 32-64 cores and 128-256 GB of RAM.

In accordance with 3GPP standards, three types of slices with different resource requirements and SLAs were created:

1. URLLC (Ultra-Reliable Low-Latency Communication): 3 VNFs (MME, SGW, PGW), requirement — 15 cores, maximum latency (Lmax) — 5 ms, priority (π) — 1 (highest).

2. eMBB (Enhanced Mobile Broadband): 5 VNF (including PCRF, HSS), requirement — 30 cores, Lmax — 50 ms, priority — 2.

3. mMTC (Massive Machine-Type Communications): 3 VNFs, requirement — 10 cores, Lmax — 1000 ms, priority — 3 (lowest).

In total, 33 VNFs were deployed in the system (11 VNFs per 3 slice instances). The experiments simulated Single Node Failure (SNF) (sudden catastrophic failure affecting 2-4 VNFs) and Progressive Overload (PO) (gradual depletion of resources) scenarios.

Comparison methods and metrics the proposed approach (DLR) was compared with three baseline methods:

• CGO (Centralized Global Optimization): Complete re-solving of the placement problem using MILP (theoretical optimum).

• NAM (Nearest Available Migration): Greedy migration to the nearest node with available capacity.

• LOO (Latency-Only Optimization): Minimization of end-to-end latency without considering migration costs.

The main performance metrics were: recovery time, resource cost, SLA violations, and migration overhead.

Manager placement results First, the effectiveness of the regional slice manager (RSM) placement algorithm was evaluated. For a topology with 10 nodes, the number of managers is |M|=3. The proposed genetic algorithm showed an improvement in the composite placement quality score of 11-33% compared to the baseline approaches (Random, Latency-Only, K-Center). In particular, it was possible to achieve a 15% reduction in latency between managers compared to the K-Center

method, confirming the importance of explicit modeling of manager-manager coordination.

The results of modeling a single node failure scenario are shown below:

• CGO: Median recovery time – 3.8 s, SLA violation – 0%, cost – 1.00 (baseline).

• NAM: Median recovery time – 0.6 s, SLA violation – 14%, cost – 1.38.

• DLR (Proposed method): Median recovery time – 0.7 s, SLA violation – 4%, cost – 1.12.

Our approach demonstrated the ability to recover 90% of failures within 1.2 s, confirming the possibility of sub-second recovery for URLLC slices. In progressive overload scenarios, DLR performed 44% fewer migrations compared to naive approaches thanks to dynamic threshold management (Fig. 3).

The results show that the proposed DLR achieves a well-balanced tradeoff between the recovery speed, cost, and SLA violations, all essential in latency-critical 5G applications.

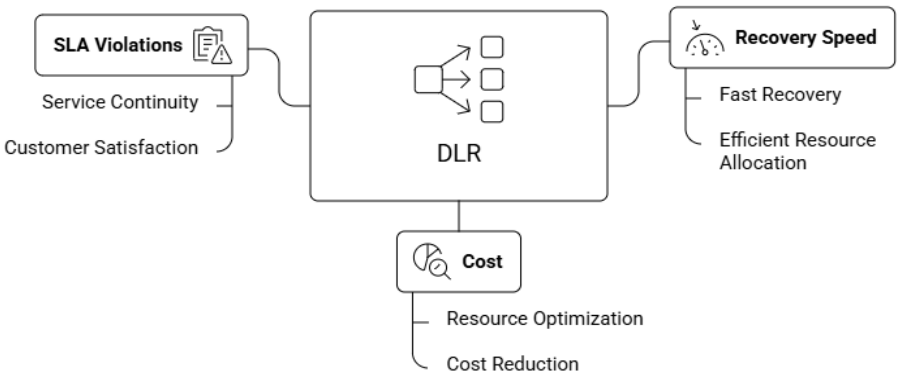

Fig. 3. DLR: Balancing Recovery Speed, Cost, and SLA Violations

Regarding recovery speed, DLR provides a median recovery time of 0.7 seconds, which is about 5.4× faster than the Centralized Global Optimization (CGO). This gain stems directly from confining the search space to a 2–3 hop neighborhood, avoiding the overhead of full network re-optimization and permitting near real-time reconfiguration under dynamic conditions (Fig. 4).

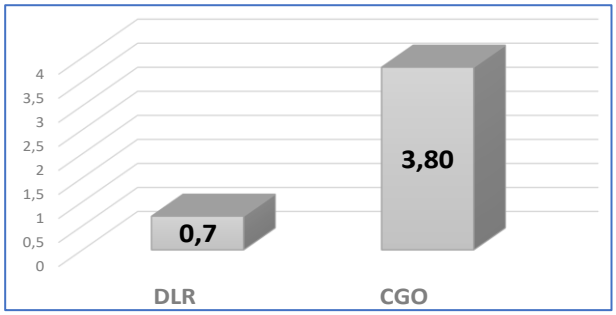

Fig. 4. DLR: median recovery time

In terms of cost, DLR is only 12% more than CGO's optimal solution, which is reasonable considering the speedup, and is 19% less than NAM and 10% less than LOO heuristics.

This is mainly attributed to the candidate selection process that explicitly takes migration and resources costs into consideration to avoid unnecessary or extremely expensive reconfigurations (Fig. 5).

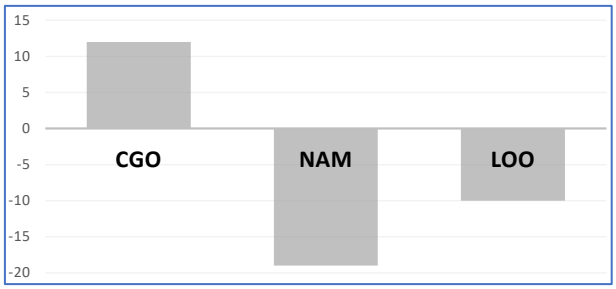

Fig. 5. DLR cost comparison

On SLA compliance, DLR keeps the URLLCs latency violation rate at approximately 4%, with violations arising only during short transients during service migration. (Fig. 6).

Fig. 6. SLA compliance and recovery delay for URLLC solutions

This is compared to the 14% violation rate shown by NAM. In contrast, although CGO produces zero SLA violations, it experiences a recovery delay of 3.8 s, making the solution untenable for URLLC downtime

Finally, towards progressive overload, DLR's use of dynamic thresholds and proactive migration yields effective prevention of cascading failure to contain the extent of cascading failure propagation to at most two rounds of reconfiguration. This body of results combine to show that distributed, cost-aware local reconfiguration does not only stand on theoretical grounds, but is indeed a practically feasible approach towards resilient network slicing in 5G.

The drawbacks of the approach can be summarized as follows.

The evaluation relies solely on synthetic results from a 10-node, handcrafted network. The network characteristics of an actual metropolitan deployment will exhibit far more variance: link latencies will vary widely, traffic patterns will display bursty behavior, and failure correlations will occur among physically co-located nodes.

Because there have not been any validations of the results in either a real testbed or benchmark topologies of record (like GÉANT or Internet2), it is impossible to determine how much the results would differ in practice.

The cost estimation model for VNF (Virtual Network Function) migration assumes that the migration (transfer of state) is a linear function of the state size. This assumption is an oversimplification of the cost of live migration of a stateful function such as an SGW (Serving Gateway) or PGW (PDN Gateway), which includes hypervisor-level checkpointing, dirty-page tracking for memory, and possible sessions involving tunnelling must be re-established; there may be many others as well. All of these introduce non-linear, workload-dependent latency that is not captured by the cost estimates in the model. Therefore, the cost estimates provided in the model and subsequently used to make migration decisions are likely to be systematically optimistic.

The framework will respond to failures and excess loads only after they have exceeded pre-defined thresholds. This means that by the time a recovery process commences, the damage sustained to the SLA has already taken place. A predictive layer could have assisted in pre-migrating workloads before the threshold for the resource is breached. An example of such a predictive layer would be an anomaly detection algorithm on a time-series based on resource usage trends; this would be especially beneficial for URLLC Slices where even a small disruption would constitute a breach of the SLA.

Conclusions

This paper provides an extensive framework for local reconfiguration of 5G network slices due to node failures and overload conditions. The proposed approach solves scalability and performance issues that come with dynamic and large-scale networks.

The proposed architecture has a three-tier hierarchical distribution based upon global orchestrators, regional slice managers, and local management agents to quickly and efficiently recover through the distribution of control to facilitate coordinated decisions throughout the network.

The work also created a multi-objective manager placement problem that considers latency, load balancing, and inter-manager cooperation in one approach. Furthermore, this harbored the concepts of controller placement in a network function virtualization environment to consider inter-manager latency as part of the optimization problem.

Localized recovery algorithms have been established for recovering from two categories of disruptions (catastrophic failures and progressive overload), and also account for slice-specific priorities and for migration costs to provide the basis for adaptive and service-aware decisions.

A comprehensive migration cost framework was proposed that includes the costs of computation, bandwidth consumption, and service disruption. This framework was formalized with a mixed-integer linear programming (MILP) model; an efficient heuristic was developed to make the framework usable in practice.

A light-weight distributed coordination protocol was designed for inter-manager communication and cross-domain recovery. This protocol enables effective collaboration between all management entities, while avoiding the scalability constraints imposed by central control.

Experimental validation showed that: recovery times of up to 6× faster than global optimization; cost savings of 27% in comparison to cost minimization based upon latency; linear scalability (to 200+ nodes); and recovery within less than 1 second for 89% of single-node failures, with less than 5% SLA violations.

The Proposed Framework addresses multiple important operational challenges:

• Lower OPEX - 27% decrease in migration costs provides a great deal of savings on an operational basis to those networks that have numerous daily failures to perform.

• Better User Experience - Sub-second URLLC recovery enables ITU-R M.2083-0 compliant mission critical applications.

• Infrastructure Flexibility - Support for heterogeneous nodes will allow incremental deployment of 5G networks following 3GPP architecture.

• Easier Operations - Automation of failure remediation reduces manual efforts and potential for human error.

Despite its effectiveness, the proposed framework has several limitations that can be investigated in the future. They include more realistic VNF state migration overheads, strong security and isolation mechanisms during migration, and explicitly considering inter-slice dependencies due to shared physical resources. The current recovery model may also be preempted using predictive failure models based on machine learning models and preemptive migration. The extension of the framework to multi-domain and multi-operator settings also remains non-trivial, particularly with respect to cross-domain coordination and enforcement of SLAs.

This work highlights the importance of distributed, and localized intelligence for a large scale network

management; At a higher level, this work is significant for valuing the distributed, localized intelligence for managing large network infrastructures operational overhead and recovery latency.

The more the network improves to 6G, and the more reliable it becomes with stringent demand for latency and scalability, the more critical localized intelligence, slice-aware differentiation discussed, and cost-aware optimization become in realizing fully autonomous mobile networks.

Conflict of interest

The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship, or other, that could affect the study and its results presented in this article.

Using artificial intelligence tools

The authors confirm that they did not use artificial intelligence technologies when creating the submitted work.

REFERENCES

1. ITU-R (2023) IMT-2030 Framework – Framework and overall objectives of the future development of IMT for 2030 and beyond. Recommendation ITU-R M.2160-0. Geneva: International Telecommunication Union. https://www.itu.int/rec/R-REC-M.2160-0-202311-I/en
2. NGMN Alliance (2020) 5G White Paper 2. Frankfurt am Main: NGMN Alliance. https://www.ngmn.org/publications/5g-white-paper-2.html
3. NGMN Alliance (2021) NGMN 6G Drivers and Vision. Frankfurt am Main: NGMN Alliance. https://www.ngmn.org/work-programme/ngmn-6g-drivers-and-vision.html
4. 3GPP (2022) System architecture for the 5G System (5GS). TS 23.501, Release 17. https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/
5. Singh, D. and Singh, J.P. (2024) 'A review on evolution, expectations and key enabling techniques of 5G', i-Manager's Journal on Communication Engineering and Systems, 13(1), pp. 38–48. https://doi.org/10.26634/jcs.13.1.20859
6. Shafi, M. et al. (2017) '5G: A tutorial overview of standards, trials, challenges, deployment, and practice', IEEE Journal on Selected Areas in Communications, 35(6), pp. 1201–1221. https://doi.org/10.1109/JSAC.2017.2692307
7. 3GPP (2018) Service requirements for the 5G system. TS 22.261, Rel. 16. https://www.3gpp.org/ftp/Specs/archive/22_series/22.261/
8. 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. https://doi.org/10.48550/arXiv.2503.22034
9. Ammar, S., Lau, C.P. and Shihada, B. (2023) 'An in-depth survey on virtualization technologies in 6G integrated terrestrial and non-terrestrial networks', arXiv preprint, arXiv:2312.01895. https://doi.org/10.48550/arXiv.2312.01895
10. Herrera, J.G. and Botero, J.F. (2016) 'Resource allocation in NFV: A comprehensive survey', IEEE Transactions on Network and Service Management, 13(3), pp. 518–532. https://doi.org/10.1109/TNSM.2016.2598460
11. Chiosi, M. et al. (2012) Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action. ETSI White Paper. https://portal.etsi.org/NFV/NFV_White_Paper.pdf
12. Heller, B., Sherwood, R. and McKeown, N. (2012) 'The controller placement problem', in Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN). Helsinki, Finland, pp. 7–12. https://doi.org/10.1145/2342441.2342444
13. Hock, D. et al. (2013) 'Pareto-optimal resilient controller placement in SDN-based core networks', in Proceedings of the 25th International Teletraffic Congress (ITC). Shanghai, China, pp. 1–9. https://doi.org/10.1109/ITC.2013.6662939
14. Lange, S. et al. (2015) 'Heuristic approaches to the controller placement problem in large-scale SDN networks', IEEE Transactions on Network and Service Management, 12(1), pp. 4–17. https://doi.org/10.1109/TNSM.2015.2400758
15. Cohen, R., Lewin-Eytan, L., Naor, J.S. and Raz, D. (2015) 'Near optimal placement of virtual network functions', in Proceedings of IEEE INFOCOM. Hong Kong, China, pp. 1346–1354. https://doi.org/10.1109/INFOCOM.2015.7218483
16. Moens, H. and De Turck, F. (2014) 'VNF-P: A model for efficient placement of virtualized network functions', in Proceedings of the 10th International Conference on Network and Service Management (CNSM). Rio de Janeiro, Brazil, pp. 418–423. https://doi.org/10.1109/CNSM.2014.7014205
17. Mehraghdam, S., Keller, M. and Karl, H. (2014) 'Specifying and placing chains of virtual network functions', in Proceedings of IEEE CloudNet. Luxembourg, pp. 7–13. https://doi.org/10.48550/arXiv.1406.1058
18. Islam, M.S. and Chowdhury, S.A.H. (2024) 'Mobility management in next generation wireless networks', American Journal of Networks and Communications, 13(1), pp. 75–83. https://doi.org/10.11648/j.ajnc.20241301.16
19. Beck, M.T. and Botero, J.F. (2017) 'Scalable and coordinated allocation of service function chains', Computer Communications, 102, pp. 78–88. https://doi.org/10.1016/j.comcom.2016.12.003
20. 3GPP (2018) Telecommunication management; Study on management and orchestration of network slicing for next generation network. TR 28.801, Release 15. https://www.3gpp.org/ftp/Specs/archive/28_series/28.801/
21. Ordonez-Lucena, J. et al. (2017) 'Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges', IEEE Communications Magazine, 55(5), pp. 80–87. https://doi.org/10.1109/MCOM.2017.1600935
22. Rost, P. et al. (2017) 'Network slicing to enable scalability and flexibility in 5G mobile networks', IEEE Communications Magazine, 55(5), pp. 72–79. https://doi.org/10.1109/MCOM.2017.1600920
23. Foukas, X. et al. (2017) 'Network slicing in 5G: Survey and challenges', IEEE Communications Magazine, 55(5), pp. 94–100. https://doi.org/10.1109/MCOM.2017.1600951
24. Ksentini, A. and Nikaein, N. (2017) 'Toward enforcing network slicing on RAN: Flexibility and resource abstraction', IEEE Communications Magazine, 55(6), pp. 102–108. https://doi.org/10.1109/MCOM.2017.1600934
25. Zhang, H. et al. (2017) 'Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges', IEEE Communications Magazine, 55(8), pp. 138–145. https://doi.org/10.1109/MCOM.2017.1600940

26. ETSI (2014a) Network Functions Virtualisation (NFV); Architectural Framework. GS NFV 002 v1.2.1. Sophia Antipolis: ETSI. https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_nfv002v010201p.pdf
27. ETSI (2015) Network Functions Virtualisation (NFV); Resiliency Requirements. GS NFV-REL 001 v1.1.1. Sophia Antipolis: ETSI. https://www.etsi.org/deliver/etsi_gs/NFV-REL/001_099/001/01.01.01_60/gs_nfv-rel001v010101p.pdf
28. Gember-Jacobson, A. et al. (2014) 'OpenNF: Enabling innovation in network function control', in Proceedings of ACM SIGCOMM. Chicago, IL, USA, pp. 163–174. https://doi.org/10.1145/2619239.2626313
29. Rajagopalan, S. et al. (2013) 'Split/Merge: System support for elastic execution in virtual middleboxes', in Proceedings of USENIX NSDI. Lombard, IL, USA, pp. 227–240. https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/rajagopalan
30. Baumgartner, A., Reddy, V.S. and Bauschert, T. (2015) 'Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization', in Proceedings of IEEE NetSoft. London, UK, pp. 1–9. https://doi.org/10.1109/NETSOFT.2015.7116162
31. Bera, A. et al. (2024) 'Network function virtualization and service function chaining frameworks: A comprehensive review', Electronics, 13(4), Article 748. https://doi.org/10.3390/electronics13040748
32. Sudhamani, C. et al. (2023) 'A survey on 5G coverage improvement techniques', Sensors, 23(4), Article 2356. https://doi.org/10.3390/s23042356
33. Qu, L. et al. (2017) 'A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks', IEEE Transactions on Network and Service Management, 14(3), pp. 554–568. https://doi.org/10.1109/TNSM.2017.2732343
34. ETSI (2014b) Network Functions Virtualisation (NFV); Management and Orchestration. GS NFV-MAN 001 v1.1.1. Sophia Antipolis: ETSI. https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf
35. Sahu,V., Sahu, N. and Sahu, R. (2024) 'Challenges and opportunities of 5G network: A review of research and development', American Journal of Electrical and Computer Engineering, 8(1), pp. 11–20. https://doi.org/10.11648/j.ajece.20240801.12
36. ETSI (2016) Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Os-Ma-nfvo reference point – Interface and Information Model Specification. GS NFV-IFA 013 v2.1.1. Sophia Antipolis: ETSI. https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/013/02.01.01_60/gs_nfv-ifa013v020101p.pdf

Received (Надійшла) 12.01.2026
Accepted for publication (Прийнята до друку) 15.04.2026
Publication date (Дата публікації) 22.05.2026

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Суліма Світлана Валеріївна – доктор філософії, доцент, доцент кафедри інформаційних технологій в телекомунікаціях, Національний технічний університет України «Київський політехнічний інститут імені І. Сікорського», Київ, Україна;
Svitlana Sulima – PhD, Associate Professor, Associate Professor of Department of Information technologies in telecommunications, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine;
e-mail: itssulima@gmail.com; ORCID Author ID: https://orcid.org/0000-0002-6333-7693;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=55226282100.

Локальна реконфігурація слайсів мережі 5G у разі виходу з ладу або перевантаження вузлів

Світлана Суліма

Анотація. Актуальність. Швидке розгортання мереж 5G та широке застосування технології мережевого сегментування створюють нові виклики для забезпечення надійності та відмовостійкості послуг. Віртуалізовані інфраструктури, засновані на віртуалізації мережевих функцій, підвищують гнучкість, але водночас призводять до збільшення частоти відмов та коливань продуктивності. У таких середовищах централізовані механізми відновлення часто не відповідають суворим вимогам до затримки, особливо для наднадійних послуг зв'язку з низькою затримкою. **Об'єкт дослідження**: Об'єктом дослідження є процес відновлення після збою та реконфігурації ресурсів у віртуалізованих середовищах сегментації мереж 5G в умовах виходу з ладу вузлів та поступового перевантаження. **Мета**: Метою дослідження є розробка ефективного розподіленого методу локальної реконфігурації мережевих сегментів, який забезпечує швидке відновлення віртуальних мережевих функцій з урахуванням пріоритетів сегментів, витрат на міграцію та обмежень щодо затримки. **Результати**. Запропоновано архітектуру розподіленої локальної реконфігурації (DLR), що базується на ієрархічній структурі, яка складається з глобального оркестратора, регіональних менеджерів сегментів та локальних агентів моніторингу. Цей підхід передбачає використання багатокритеріальної моделі оптимізації для розміщення менеджерів сегментів та уніфікованої функції вартості міграції, яка враховує компоненти обчислювальних ресурсів, мережі, перебоїв у роботі та штрафних санкцій за порушення SLA. Розроблено алгоритми локалізованого відновлення для обробки як катастрофічних відмов вузлів, так і сценаріїв прогресивного перевантаження, при цьому пріоритетність сегментів визначається відповідно до їхніх вимог до послуг. **Висновки**. Запропонована архітектура розподіленого відновлення сегментів забезпечує швидку та масштабовану реконфігурацію сегментів мережі 5G в умовах збою. Завдяки поєднанню пріоритетності з урахуванням сегментів, рішень щодо міграції з урахуванням витрат та локалізованого управління цей підхід підвищує швидкість відновлення та операційну ефективність, зберігаючи при цьому якість обслуговування для додатків, для яких критично важлива низька затримка.

Ключові слова: мережі 5G; сегментація мережі; віртуалізація мережевих функцій; відновлення після збою; розподілене управління мережею; міграція віртуальних мережевих функцій; управління ресурсами з урахуванням сегментів; відмовостійкість; угода про рівень обслуговування; розподілена локальна реконфігурація.