anima
Internet-Draft
Intended status: Standards Track
Expires: 5 February 2026

Y. Yue, Ed.
X. Zhang, Ed.
China Unicom

4 August 2025

Task-Oriented Multi-Agent Recovery Framework for High-Reliability in Converged Mobile Networks
draft-yue-anima-agent-recovery-networks-00

Abstract

This document defines a task-oriented, agent-based method for fault recovery in converged public-private mobile networks. The proposed method introduces a multi-agent collaboration framework that enables autonomous failure detection, scoped diagnosis, inter-domain coordination, and intent-driven policy reconfiguration. It is particularly applicable in complex 5G/6G network deployments, such as Multi-Operator Core Networks (MOCN) and Standalone Non-Public Networks (SNPN), where traditional centralized management is insufficient for ensuring high service reliability and dynamic recovery. The document also specifies protocol requirements for inter-agent communication, state consistency, and secure coordination, aiming to support interoperability and resilience across heterogeneous network domains.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 5 February 2026.

Copyright Notice

Table of Contents

1. Introduction 3
2. Conventions and Terminology 3
3. Use Cases 4
3.1. Dynamic Fault Recovery in Shared 5G MOCN Infrastructure 4
3.2. Autonomous Recovery in Enterprise SNPN 4
3.3. Cross-Domain Policy Conflict Resolution 4
3.4. SLA-Aware Remediation in AI-Driven RAN 5
4. Problem Statement 5
5. Protocol Requirements 6
5.1. Agent Communications Interface 6
5.2. Message Semantics and Encoding 6
5.3. Reliability, Ordering, and Timeout Handling 7
5.4. Security and Trust Requirements 7
5.5. Behavior and State Consistency 7
5.6. Interoperability Considerations 7
6. Task-Oriented Agent-Based Recovery Method for High-Reliability Assurance 8
6.1. Objectives 8
6.2. Agent Roles and Responsibilities 8
6.3. Recovery Workflow 9
6.3.1. Scoped Fault Correlation 9
6.3.2. Intent-Driven Recovery Evaluation 9
6.3.3. Inter-Domain Coordination 9
6.3.4. Execution and Safety Enforcement 10
6.3.5. Feedback Loop and Adaptive Monitoring 10
7. Security Considerations 10
8. IANA Considerations 10
9. Normative References 10
Authors' Addresses 10

1. Introduction

As mobile networks evolve toward 5G and 6G architectures, new deployment paradigms such as Multi-Operator Core Networks (MOCN), Shared RAN, and Standalone Non-Public Networks (SNPN) have emerged to support both public and enterprise services. These converged deployments introduce unprecedented complexity in terms of topology, administrative boundaries, resource sharing, and dynamic service intent management.

Ensuring high reliability in such networks is increasingly difficult using traditional centralized network management systems, which often suffer from limited scalability, slow responsiveness, and single points of failure. These limitations are particularly critical in enterprise and industrial environments, where service-level agreements (SLAs) mandate deterministic latency, availability, and adaptability.

This document introduces a task-oriented, agent-based recovery method that enables distributed fault detection, context-aware correlation, inter-agent negotiation, and closed-loop policy execution. Agents operate at various roles — including telemetry monitoring, domain coordination, policy interpretation, and action enforcement — and communicate through a structured Agent Communication Interface (ACI). The method is designed to autonomously localize faults, assess recovery strategies based on service intents, and coordinate recovery actions across administrative domains, with minimal human intervention.

In addition to describing the recovery workflow and agent roles, this document outlines the associated protocol requirements to ensure secure, consistent, and interoperable interactions among agents. These requirements cover communication semantics, message formats, transport assumptions, and behavioral guarantees. The goal is to enable standards-compliant, intent-aware, and autonomous fault management in future mobile network infrastructures.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC2119 [RFC8174] when, and only when, they appear in all capitals, as shown here.

Abbreviations and definitions used in this document: *ACI: Agent Communication Interface. *DCA: Domain Coordination Agent. *EA: Execution Agent. *FDA: Fault Detection Agent. *FSM: Finite State

Machine. *LLM: Large Language Model. *MOCN: Multi-Operator Core Network. *MTTR: Mean Time to Recovery. *PIA: Policy Interpretation Agent. *SLA: Service-Level Agreement. *SNPN: Standalone Non-Public Network. *URI: Uniform Resource Identifier.

3. Use Cases

The method defined in this document applies to several real-world use cases in future mobile network environments:

3.1. Dynamic Fault Recovery in Shared 5G MOCN Infrastructure

In Multi-Operator Core Network (MOCN) deployments, multiple mobile network operators (MNOs) share the same RAN and transport infrastructure. A node failure or link degradation in the shared segment can affect multiple tenant slices simultaneously. With agent-based coordination, local agents at affected nodes can detect the fault, and domain-level agents from each operator can negotiate temporary recovery strategies (e.g., re-routing or resource reallocation) without requiring centralized orchestration or full-stack configuration reloading.

3.2. Autonomous Recovery in Enterprise SNPN

Standalone Non-Public Networks (SNPN) are often deployed by enterprises to support on-site applications such as industrial automation, AGV coordination, or safety monitoring. In these environments, recovery must be both low-latency and intent-aware. For example, if a compute node hosting a real-time controller fails, the agent system can trigger service migration to a backup node based on the intent to maintain <10ms latency for URLLC traffic, without requiring manual administrator intervention.

3.3. Cross-Domain Policy Conflict Resolution

In hybrid deployments where a public network operator provides managed service slices to enterprises, misaligned policies across administrative domains may cause service disruptions (e.g., route loops, priority mismatches). With inter-domain agent negotiation, agents can exchange scoped views of current state and intent, evaluate compatibility, and agree on a temporary policy contract to preserve service continuity until a global policy reconciliation occurs.

3.4. SLA-Aware Remediation in AI-Driven RAN

With the rise of AI-native RAN optimization, agents embedded within distributed units (DU/CU) or edge compute nodes may detect performance anomalies (e.g., increased jitter, burst loss). Rather than waiting for offline model retraining, the system can dynamically adapt configuration (e.g., buffer allocation, scheduler adjustment) using the agent-based recovery workflow to preserve SLA requirements in real time.

4. Problem Statement

In converged public-private mobile networks, ensuring service continuity and network reliability in the event of failures is a fundamental requirement, particularly for enterprise and critical infrastructure scenarios. Traditional centralized network management systems often suffer from single points of failure and delayed recovery, which are unacceptable in contexts where deterministic availability and ultra-low downtime are essential. Multi-agent systems enable fault-tolerant operation through distributed intelligence and redundancy. When a failure occurs—such as link disconnection, node crash, or policy conflict—a well-coordinated group of agents can dynamically detect, localize, and mitigate the issue through real-time communication and cooperative decision-making. This distributed resilience mechanism reduces mean time to recovery (MTTR) and minimizes the impact radius of failures. Moreover, in cross-domain environments (e.g., MOCN with multiple operators or SNPN with enterprise-hosted infrastructure), fault management becomes more complex due to administrative isolation and heterogeneous control planes. Intelligent agents deployed at domain boundaries can negotiate fallback strategies, synchronize state across domains, and maintain policy consistency during partial outages. For example, upon detecting performance degradation in a tenant slice, the agents can proactively rebalance traffic, reassign resources, or trigger intent re-interpretation without waiting for centralized orchestration. Without agent-based failure collaboration, the system risks becoming fragmented, with isolated components unable to respond effectively to cascading failures. Therefore, enabling resilient, autonomous coordination among agents in failure scenarios is essential to support high-availability SLAs, enhance robustness against dynamic network threats, and reduce operational overhead in complex network environments.

5. Protocol Requirements

To support the efficient and intelligent transmission of sensing data in 6G environments, enhancements to the MoQ protocol are proposed. These enhancements aim to enrich MoQ metadata or header extensions to include key information required for intelligent routing, data classification, service mapping, and QoS-aware scheduling in sensing-centric applications.

5.1. Agent Communications Interface

This section specifies the protocol-level requirements to support the agent-based recovery method defined in Section 5. These requirements cover message formats, communication interfaces, timing constraints, behavioral consistency, and inter-domain negotiation semantics. The goal is to ensure interoperability, reliability, and intent-aware execution of fault recovery workflows across diverse network domains and agent implementations. REQ-1: The system SHOULD define a structured Agent Communication Interface (ACI) to support asynchronous and event-driven communication among agents. REQ-2: ACI SHOULD support the following core message types: FAULT_EVENT: Sent from FDA to DCA; conveys detected fault condition. SCOPE_CORRELATION_QUERY/REPLY: Between DCAs; used for inter-domain fault localization. INTENT_REQUEST/RESPONSE: Between DCA and PIA; conveys service-level intent and policy goals. RECOVERY_PROPOSAL: Sent from initiating DCA to peer DCA(s); contains proposed joint recovery actions. RECOVERY_CONTRACT: Formalizes agreement among domains on resource reallocation and rollback.

EXECUTION_COMMAND: Sent from DCA to EA to enact recovery actions. EXECUTION_STATUS: Sent from EA to DCA to report outcome and validation results. REQ-3: All ACI messages SHOULD include: Agent identity and role Timestamp Message type and version Unique transaction/session ID Integrity protection (e.g., signature or HMAC) REQ-4: The ACI protocol SHOULD support both push and pull modes for event dissemination and agent querying.

5.2. Message Semantics and Encoding

REQ-5: Protocol messages SHOULD be encoded using a format that is both human-readable and machine-processable. JSON and CBOR are RECOMMENDED; protocol buffers MAY be used in constrained environments. REQ-6: Each message type SHOULD conform to a pre-defined schema, including required and optional fields. REQ-7: Message payloads involving intent retrieval or policy proposals SHOULD include a service identifier that maps to a known SLA or intent profile.

5.3. Reliability, Ordering, and Timeout Handling

REQ-8: Protocol exchanges involving recovery workflows MUST support acknowledgment and retry mechanisms. REQ-9: Agents participating in a recovery transaction MUST support: Timers for detecting negotiation or execution timeout Fallback strategies upon failure to reach consensus or apply action REQ-10: ACI message transport MUST guarantee in-order delivery of messages within a session context, particularly for multi-step negotiation sequences.

5.4. Security and Trust Requirements

REQ-11: All ACI communications MUST be secured using mutually authenticated channels. REQ-12: Agents MUST maintain a local trust registry of peer agents and their associated roles, identities, and access policies. REQ-13: Inter-domain messages MUST be cryptographically signed and include domain-level identifiers to prevent spoofing or replay. REQ-14: Sensitive data in intent evaluation MUST be protected during transit and only exposed to authorized agents.

5.5. Behavior and State Consistency

REQ-15: Agents MUST implement finite state machines (FSMs) to ensure correct handling of message sequences and recovery states. REQ-16: In case of multi-agent execution, agents MUST agree on task status codes to track workflow progress consistently. REQ-17: Feedback and learning data SHOULD be stored in a common, queryable knowledge base accessible to policy training agents.

5.6. Interoperability Considerations

REQ-18: Implementations MUST support version negotiation for ACI messages to ensure forward compatibility. REQ-19: Domain-specific extensions (e.g., for 5G MOCN, SNPN) MUST be encapsulated using an optional extension field, and MUST NOT interfere with baseline schema validation. REQ-20: Recovery workflows MUST be idempotent where possible, allowing repeated execution without unintended side effects in failure or retry scenarios.

6. Task-Oriented Agent-Based Recovery Method for High-Reliability Assurance

This part defines a distributed, agent-based recovery method that supports high-reliability service assurance in converged public-private mobile networks. The method enables autonomous failure detection, scoped diagnosis, and intent-driven policy adaptation through coordination among multiple intelligent agents. It is designed to address both intra-domain and inter-domain failure scenarios while maintaining SLA compliance.

6.1. Objectives

The method is designed to fulfill the following objectives: (1) Resilience through distribution: Eliminate single points of failure by decentralizing failure detection and recovery logic across agents. (2) Scoped collaboration: Allow agents to reason over localized context while supporting inter-agent negotiation for broader fault scenarios. (3) Intent consistency: Ensure that all recovery decisions align with user or service-level intents registered in the system. (4) Closed-loop adaptability: Continuously monitor recovery outcomes and feed them into learning or policy refinement processes. (5) The method is applicable in deployment environments such as 5G MOCN, SNPN, or 6G hybrid infrastructures involving multiple tenants and administrative domains.

6.2. Agent Roles and Responsibilities

The method introduces four distinct roles for intelligent agents, each fulfilling a key functional responsibility in the recovery workflow: (1) Fault Detection Agent (FDA): Resides at network or compute nodes; performs real-time telemetry monitoring. Upon threshold violation, constructs a structured fault event including metadata such as event ID, node ID, timestamp, metric type, and severity. (2) Domain Coordination Agent (DCA): Aggregates events from multiple FDAs to determine failure scope and severity. Responsible for intra-domain coordination and inter-domain negotiation when needed. (3) Policy Interpretation Agent (PIA): Retrieves and parses registered service intents. Evaluates recovery options and generates adaptive policy updates based on current state and available resources. (4) Execution Agent (EA): Applies the reconfiguration actions (e.g., rerouting, resource migration, parameter adjustment) and performs post-configuration checks to ensure compliance and stability. All agents communicate over an Agent Communication Interface (ACI), which provides structured messaging primitives for event reporting, status querying, negotiation, and command dispatch.

6.3. Recovery Workflow

The recovery method consists of the following task-oriented workflow: ### Fault Detection and Event Generation FDA continuously monitors key performance metrics (e.g., latency, packet loss, CPU utilization). On violation, FDA emits a structured fault event: +-------------------+----------------------------+ | Field | Value | +-------------------+----------------------------+ | event_id | e12345 | | node_id | node-A | | timestamp | 2025-07-21T08:00:00Z | | metric | link_loss | | value | 15.2 | | threshold | 10.0 | | severity | major | +-------------------+----------------------------+ This event is transmitted to the local DCA via ACI.

6.3.1. Scoped Fault Correlation

DCA aggregates fault reports from FDAs and analyzes temporal-spatial correlations. If patterns emerge indicating a localized or distributed failure domain, DCA maps the affected logical services (e.g., slices, functions, access nodes). If the impact likely crosses domain boundaries (e.g., MOCN core or shared RAN), the DCA initiates inter-domain state queries.

6.3.2. Intent-Driven Recovery Evaluation

DCA invokes PIA with a fault-context descriptor. PIA queries the intent registry and retrieves the affected service’s constraints and goals, such as: +--------------------+---------------------------+ | Field | Value | +--------------------+---------------------------+ | intent_id | tenant-001-intent | | sla.latency | < 20ms | | sla.availability | 99.99% | | fallback_policy | [reroute, degrade_qos] | | priority | critical | +--------------------+---------------------------+ PIA evaluates multiple recovery strategies (e.g., traffic shift, resource migration, service downgrade) and scores them against SLA compliance and resource availability.

6.3.3. Inter-Domain Coordination

When faults span across domains, the DCA of the initiating domain sends a Recovery Proposal Message to peer DCAs. Each DCA evaluates local resource availability and responds with either: Acceptance of shared recovery effort (with constraints), or Negotiation of a fallback agreement (with time limits and rollback conditions). Upon consensus, a Recovery Execution Contract is established, which includes scope, roles, time windows, and validation checkpoints.

6.3.4. Execution and Safety Enforcement

DCA dispatches a recovery command to EA, which applies configurations (e.g., policy updates, slice rerouting, traffic prioritization). EA performs pre- and post-checks to verify: Policy consistency Compliance with intent System stability post-update

6.3.5. Feedback Loop and Adaptive Monitoring

After execution, FDA switches to enhanced monitoring mode in affected areas (e.g., higher-frequency sampling, link probing). DCA collects performance data and sends summary logs to a shared knowledge base for: Post-mortem analysis Learning model refinement (e.g., reinforcement learning agent tuning) If instability persists, PIA may auto-trigger policy reevaluation or escalate to supervisory agent layer.

7. Security Considerations

TBD

8. IANA Considerations

TBD

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Authors' Addresses

Yi Yue (editor)
China Unicom
Beijing
China
Email: yuey80@chinaunicom.cn

Xuebei Zhang (editor)
China Unicom
Beijing
China
Email: zhangxb170@chinaunicom.cn