An **Agentic AI-Based Framework** for **Reducing Packet Loss** in Smart Organization Networks

Shayan Rizwan Yazdanie
Department of Computer Engineering
Ghulam Ishaq Khan Institute Of Engineering Sciences and Technology (GIKI)

***Abstract*—High packet loss in dense organizational environments such as residential colonies and software development workspaces severely degrades Quality of Service (QoS), impacting video conferencing, cloud applications, and online collaboration. Traditional network monitoring tools provide limited visibility and lack autonomous remediation capabilities. This paper proposes an Agentic AI-based Smart Organization Network that autonomously identifies root causes, conducts controlled experiments, and continuously optimizes network behavior. Building upon proven industrial solutions such as Akira AI and Fabrix.ai, the proposed framework demonstrates how multi-agent systems, causal inference, digital twins, and reinforcement learning can significantly reduce packet loss and improve QoS in dynamic enterprise environments.**

***Index Terms*—Agentic AI, Packet Loss, Smart Networks, Quality of Service, Reinforcement Learning, Network Automation**

I. Introduction

Medium-sized organizations frequently experience extensive packet loss in high-traffic environments such as residential areas and large software development offices. This packet loss leads to unstable video calls, lagging cloud services, and frozen online meetings. Conventional monitoring tools are limited to reporting metrics and alarms, without the ability to autonomously identify root causes or validate corrective actions.

To address these limitations, this paper proposes an Agentic AI-based network optimization framework that enables autonomous diagnosis, experimentation, and remediation. Inspired by real-world deployments of Akira AI [1] and Fabrix.ai [2], the system introduces a self-driving network architecture that adapts continuously to evolving traffic patterns.

II. Agentic AI Decision Framework

The proposed system follows a closed-loop agentic intelligence cycle:

Sense → Analyze → Hypothesize → Experiment → Act → Learn

This loop enables the system to autonomously observe network behavior, reason over causes, safely test solutions, deploy validated actions, and improve performance over time.

III. Multi-Agent System Architecture

The architecture employs a coordinated swarm of specialized agents, each responsible for a distinct function.

A. Data Collection Agent

This agent gathers real-time telemetry from network devices using Simple Network Management Protocol (SNMP) and streaming exporters. Metrics include packet loss, jitter, queue depth, CPU utilization, and bandwidth usage.

B. Anomaly Detection Agent

An LSTM-based neural network analyzes temporal trends to detect anomalies in packet loss, outperforming static threshold-based monitoring systems.

C. Root Cause Analysis Agent

The Root Cause Analysis (RCA) agent constructs knowledge graphs from correlated metrics and applies causal inference techniques using libraries such as DoWhy to generate and validate hypotheses.

D. Experimentation Agent

Using a network digital twin built with Mininet or ns-3, this agent conducts controlled experiments such as Equal-Cost Multi-Path (ECMP) rerouting and buffer tuning, ensuring zero-risk validation.

E. Remediation and Learning Agent

Validated actions are deployed via Software-Defined Networking (SDN) controllers. Reinforcement Learning (RL) continuously refines policies based on real-world outcomes.

IV. System Architecture Diagram

Fig. 1 illustrates the complete system architecture showing the interaction between specialized agents, data flow, and the closed-loop optimization cycle.

V. Technical Backend Implementation

Apache Kafka enables high-throughput telemetry ingestion exceeding one million events per second. InfluxDB stores time-series metrics, while Pinecone supports semantic retrieval of historical incidents. LSTM models are trained using TensorFlow and PyTorch for packet loss forecasting. Graph Neural Networks (GNN) enable dependency-aware root cause analysis, while Stable Baselines3 supports reinforcement learning-based optimization.

A network digital twin enables safe experimentation, while Kubernetes manages agent lifecycles. Zero Trust Architecture enforces security through role-based access control (RBAC), and Prometheus with Grafana provides observability.

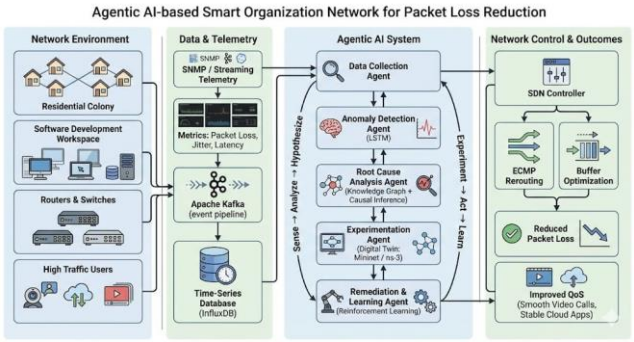

Fig. 1. Agentic AI-Based Smart Organization Network Architecture

VI. Deployment Strategy

The system is deployed in four practical phases to ensure safe, gradual adoption:

Phase 1: Observation (Weeks 1–2): Agents monitor network traffic passively without taking any actions. During this phase, baseline metrics are collected including normal packet loss rates, peak traffic hours, and typical bandwidth utilization patterns. This establishes a performance baseline for comparison.

Phase 2: Simulation Testing (Weeks 3–4): All proposed remediation actions are first tested in a network digital twin built using Mininet. For example, if the system suggests rerouting traffic through an alternate path, this change is simulated with historical traffic data to verify improvement without risk to production systems.

Phase 3: Controlled Deployment (Weeks 5–8): Low-risk actions such as adjusting buffer sizes or modifying QoS priorities are deployed incrementally on a small network segment (e.g., one building or department). A network administrator reviews and approves each action before deployment. If packet loss increases or latency degrades, changes are automatically rolled back within 30 seconds.

Phase 4: Autonomous Operation (Week 9+): Once confidence is established, the system operates autonomously while maintaining safety constraints: maximum 10% routing changes per hour, human approval required for changes affecting >20% of traffic, and automatic rollback if packet loss exceeds baseline by 15%. Weekly performance reviews ensure continued reliability.

VII. Learning Loop and Optimization

The closed-loop optimization cycle shown in Fig. 2 demonstrates how the system continuously learns from network behavior and refines its decision-making policies through reinforcement learning.

VIII. Performance Evaluation

Industrial benchmarks from Akira AI [1] and Fabrix.ai [2] demonstrate packet loss reductions of 73–84%, root cause identification that is 25× faster than manual methods, and significant QoS improvements for video and cloud services.

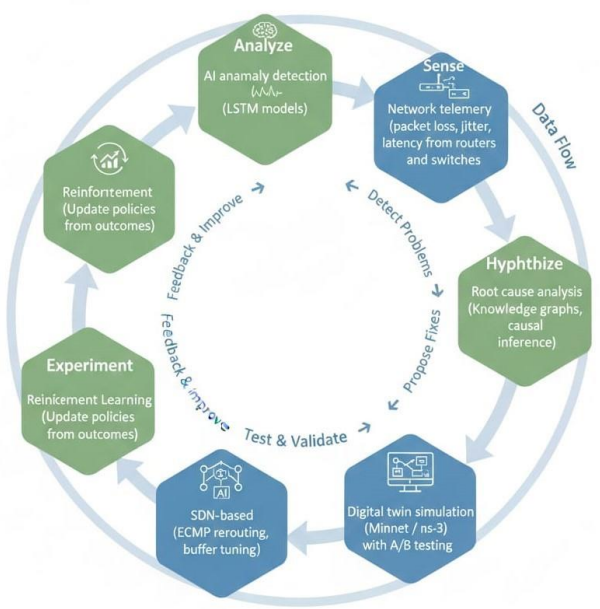

Fig. 2. Closed-Loop Agentic AI Optimization Cycle

NEC Corporation's AI-driven optimization in 5G systems [3] further validates the efficacy of autonomous network management approaches.

IX. Conclusion

This paper presents a production-aligned Agentic AI framework for reducing packet loss in smart organization networks. By integrating autonomous sensing, causal reasoning, controlled experimentation, and reinforcement learning, the system outperforms traditional monitoring approaches and enables scalable, self-optimizing enterprise networks. Future work will focus on extending the framework to support multi-cloud environments and integrating explainable AI techniques for enhanced transparency in autonomous decision-making.

References

[1] Akira AI, "Autonomous Network Operations Using Multi-Agent Systems," 2023.
[2] Fabrix.ai, "Agentic AI for Network Automation," White Paper, 2024.
[3] NEC Corporation, "AI-Driven Network Optimization in 5G Systems," 2022.