Real-Time Observability of Peer-to-Peer Node Communication in Mesh Network: Towards Digital Sovereignty in Secure and Autonomous Networking

***Abstract*—In today's digitally interconnected world, achieving sovereign control over decentralized communication infrastructures has become a strategic imperative for national autonomy, security, and trust. Mesh Virtual Private Networks (VPNs), enabling peer-to-peer (P2P) communication through techniques such as UDP Hole Punching, offer resilience and independence from centralized service providers. However, traditional Mesh VPNs lack continuous observability once the initial handshake is complete, limiting sovereign operators' ability to monitor ongoing communications or detect anomalies. This paper presents a sovereignty-driven monitoring framework that deploys lightweight agents on each node to securely collect system and application logs, transmitted to a Central Monitoring Server under sovereign control. Redundancy is ensured via Backup Agents and a Heartbeat Mechanism, mitigating risks if the central server is compromised. Logs are analyzed in real time using platforms such as Elasticsearch, Prometheus, and Grafana, enabling autonomous oversight while preserving local control. Experiments in simulated mesh environments demonstrate improved detection of anomalies, low overhead, and resilience under node churn. The proposed architecture advances Digital Sovereignty by ensuring transparent, autonomous, and locally governed monitoring and fraud detection.**

***Index Terms*—Digital Sovereignty, Mesh VPN, Peer-to-Peer Networks, Real-Time Observability, Secure Communication, Trust, Network Autonomy.**

I. Introduction

The growing emphasis on Digital Sovereignty (DS), defined as the ability of nations, enterprises, and communities to exercise autonomous control over digital infrastructures, data, and governance, has highlighted the need for secure, transparent, and self-governed communication systems [21],[22]. Achieving DS requires secure, transparent, and self-governed communication systems that balance autonomy with observability.

In this context, decentralized networking architectures, such as Mesh VPNs, enable peer-to-peer (P2P) communication without reliance on centralized authorities, improving autonomy and resilience. Each node functions as both client and server, establishing direct connectivity via techniques like UDP Hole Punching [10]. While these mechanisms strengthen independence, they create post-handshake visibility gaps, leaving ongoing communication opaque to sovereign operators. Without continuous monitoring, Mesh VPNs are vulnerable to unauthorized access, data exfiltration, and malicious infiltration, potentially compromising sovereignty [1], [2], [6].

To address this challenge, we propose a DS-oriented monitoring framework that ensures transparent and autonomous oversight without undermining autonomy. The framework deploys Main Agent on each node along with the Backup Agent to collect system and application logs securely. Heartbeat-based fault detection and redundant agents mitigate risks from node or server failures, while secure analytics platforms preserve local control [2], [8], [21]. Trust assumptions include authenticated endpoints and tamper-evident logging mechanisms, supporting resilience against endpoint compromise.

This study follows a design-oriented research approach, situating the architecture within a rigorous methodological framework [21]. Key contributions are:

1) We propose a DS-oriented observability model for Mesh VPNs that ensures sovereign control over communication data.
2) We design and implement a lightweight UDP-based monitoring framework compatible with decentralized topologies.
3) We integrate resilience mechanisms—Heartbeat and Backup Agent—to guarantee fault tolerance and high availability.
4) We demonstrate the framework's effectiveness through experimental validation, showing improved fraud detection, anomaly monitoring, and sovereignty-preserving transparency.

Overall, this dual-agent, sovereignty-driven design bridges the post-handshake visibility gap in Mesh VPNs, strengthening trust, autonomy, and compliance in decentralized infrastructures and contributing to the broader goal of advancing Digital Sovereignty in secure and intelligent network environments.

II. Background On Digital Sovereignty

Digital Sovereignty (DS) represents the ability of nations and organizations to maintain autonomous control over their digital infrastructures, data, and governance [21], [22]. It extends beyond data ownership to include infrastructural independence, secure identity management, and transparent observability. Janardhanan et al. [21] emphasize that network sovereignty requires self-managed and fault-tolerant infrastructures to minimize external dependencies, while Karagiannis et al. [22] highlight that data sovereignty at the edge improves trust by keeping governance close to the data source. Within this context, continuous yet locally governed observability becomes essential for ensuring accountability, resilience, and trust—without compromising autonomy. This

principle forms the foundation for the sovereignty-driven monitoring framework proposed in this study.

III. Related Studies

Building on the principles of Digital Sovereignty, prior research in mesh and peer-to-peer (P2P) communication has focused primarily on resilience, security, and monitoring efficiency. Mesh networks and P2P overlays have been widely studied for their decentralized communication capabilities. Goel *et al.* [1] investigated VPN network traffic detection and highlighted the difficulty of observing node-to-node communications once initial handshakes are completed. Similarly, Xu *et al.* [5] analyzed VPN security, emphasizing risks associated with encrypted tunnels and limited post-handshake visibility.

Monitoring in wireless and distributed environments introduces unique challenges due to dynamic topologies and limited resources. Williams *et al.* [2] proposed data-driven network monitoring using machine learning, while Baimukashev *et al.* [4] designed an intrusion detection system (IDS) tailored for wireless networks. Malani *et al.* [8] further demonstrated IDS architectures for distributed environments, highlighting the need for lightweight instrumentation with minimal system overhead.

Machine learning and artificial intelligence approaches are increasingly applied in intrusion detection. Liao *et al.* [3] surveyed deep learning methods for IoT intrusion detection, and Guo *et al.* [6] proposed ML-based IDS solutions for IoT devices. Mirza *et al.* [17] reviewed broader AI-enabled networking trends, while Sridevi *et al.* [7] presented an AI-driven traffic monitoring system for real-time congestion detection in smart cities. These approaches reinforce the potential of AI for anomaly detection in decentralized overlays.

Studies also focus on advanced scanning and detection methods. Jafarian *et al.* [11] proposed DNS-based monitoring to detect network scanning, while Zhang *et al.* [15] introduced IMap, a scalable in-network scanner using programmable switches. Yang *et al.* [12] explored IPv6 periphery discovery, and Li *et al.* [13] applied ant colony algorithms for routing optimization. Seng *et al.* [16] examined network slicing in wireless mesh networks, while Ghosh *et al.* [19] and Kilthau *et al.* [20] studied decentralized trust and optimization in P2P environments. Furthermore, recent studies on Digital Sovereignty underscore the importance of locally governed network control [21], [22], reinforcing the relevance of sovereign observability mechanisms.

Recent developments such as digital twin models by Poorzare *et al.* [9] and generative AI-enabled 6G architectures by Sun *et al.* [18] further suggest that future monitoring systems should integrate real-time analytics, trust verification, and proactive anomaly detection.

In summary, prior work demonstrates that VPN Security by Goel *et al.* [1, and Xu *et al.* [5], lightweight monitoring Williams *et al.* [2], [Baimukashev *et al.* 4], Malani *et al.* [8], and Zhang *et al.* [15], AI-based intrusion detection by Liao *et al.* [3], Guo *et al.* [6], Sridevi *et al.* [7], and Mirza *et al.* [17], and P2P trust models by Seng *et al.* [16], Ghosh *et al.* [19], and Kilthau *et al.* [20]—provides a strong foundation. However, few studies directly address post-handshake observability in Mesh VPNs, where the rendezvous server becomes passive after connection setup. To bridge this gap, this work introduces a sovereignty-driven dual-agent monitoring framework that ensures real-time visibility, resilience, and locally governed analytics for secure and autonomous communication.

IV. Methodology

A. Components

1) Main Monitoring Agent: The primary component of the proposed framework is the Main Monitoring Agent, developed in Python for rapid prototyping and in Go for lightweight deployment. This agent operates on every node within the mesh network and is responsible for continuous system and network telemetry collection. Specifically, it records CPU utilization, memory consumption, network interface statistics, and the list of active peers retrieved from the mesh VPN. Additionally, it captures application and network-level events such as connection attempts and handshake errors. The agent formats the collected data into structured payloads and transmits them to the central monitoring server at configurable intervals (e.g., every 5 seconds), ensuring near real-time visibility into node performance and connectivity.

2) Backup Agent: To ensure resilience, a Backup Agent operates in a passive standby mode on each node. Its primary task is to monitor heartbeat signals emitted by the main agent. If no heartbeat is detected for a predefined duration (Y seconds), the backup agent assumes the responsibilities of the main agent, thereby preventing any monitoring downtime. This failover process is fully automated and ensures continuity of log and metric collection.

3) Heartbeat Mechanism: The heartbeat mechanism is the key failure detection method in the proposed system. The main agent periodically sends lightweight heartbeat messages to both the backup agent and the central monitoring server. The backup agent maintains a missed-heartbeat counter, and failover is triggered if the number of missed signals exceeds a defined threshold. This design ensures that failures are detected promptly, enabling seamless takeover. The central server also records missed heartbeats for historical reliability analysis.

4) UDP Hole Punching Layer: Given that mesh VPN nodes may be behind NAT or firewall devices, the system employs UDP hole punching to facilitate communication with the central monitoring server. Both the main and backup agents initiate outbound UDP sessions to the server, ensuring inbound packets are accepted by the NAT device for the session's duration. All transmitted monitoring data is encrypted using AES with a pre-shared key to guarantee confidentiality and integrity during transit.

5) Central Monitoring Server: The central monitoring server acts as the collection, storage, and visualization hub for all agent-generated data.

(i) **Metrics pipeline:** The server ingests performance metrics via Prometheus, which are visualized in real time through Grafana dashboards.

(ii) **Log pipeline:** Structured logs are stored in Elasticsearch, enabling advanced querying and full-text search. Kibana visualizes log data and supports anomaly detection.

(iii) **Alerting system:** Alerts are generated through Prometheus Alertmanager and Kibana Watcher when thresholds are exceeded, such as:

- Excessive CPU or network utilization.
- Frequent handshake failures.
- Absence of heartbeat signals from an agent.

6) Subnet Scanner for Host Discovery: To complement node-level monitoring, a subnet scanner was developed using Python and the Nmap library. The scanner initially used ICMP Echo Requests (-sn -PE) to detect active and inactive IPs across subnets. However, ICMP-based discovery failed to detect live hosts behind firewalls, leading to false negatives. To address this, hybrid scanning was integrated using TCP SYN (-PS), TCP ACK (-PA), UDP (-PU), and ARP (-PR) probes. This hybrid method improves the accuracy of hidden node detection in restricted topologies.

B. Data Flow in the Monitoring Framework

The proposed monitoring framework follows a structured and secure data flow, ensuring continuous visibility into mesh network communication.

1) **Mesh VPN Communication (P2P Layer):** Nodes exchange encrypted user data directly within the decentralized mesh overlay (e.g., Nebula or WireGuard).
2) **Log Collection:** The main agent gathers system- and network-level telemetry, including CPU utilization, memory usage, interface statistics, and mesh VPN peer connection details. All data is formatted into a structured JSON object containing timestamp, node identifier, metrics, and log content.
3) **Transport:** Collected logs and metrics are transmitted to the central monitoring server using encrypted UDP packets. NAT traversal is achieved through UDP hole punching, enabling connectivity in restrictive network environments without explicit port forwarding.
4) **Storage and Visualization:**
   - Metrics are ingested by Prometheus and visualized in Grafana dashboards for real-time monitoring.
   - Logs are indexed in Elasticsearch and accessed via Kibana for querying, trend analysis, and anomaly detection.
5) **Failure Handling:** If the backup agent detects that the main agent has failed to send heartbeat signals for a predefined threshold period, it automatically activates and resumes log transmission. The central server flags the main agent as offline and records the failover event.
6) **Security:** All monitoring traffic is encrypted and authenticated using a pre-shared key mechanism. Agents validate the central server's public key before data transmission, ensuring mutual trust and protection against impersonation.

C. Key Advantages

The proposed architecture offers several operational benefits over traditional mesh monitoring approaches:

1) **Resilience:** The inclusion of a backup agent with automated failover guarantees uninterrupted monitoring, eliminating blind spots caused by single-agent failure.
2) **Sovereignty:** All analytics and observability operations remain under local jurisdiction.
3) **NAT Compatibility:** UDP hole punching enables seamless operation across restrictive networks without requiring inbound port configurations.
4) **Lightweight Operation:** The agents introduce less than 8% average CPU overhead and under 12 MB additional memory usage per node, ensuring suitability for resource-constrained devices.
5) **Security:** All communication channels are encrypted and authenticated, safeguarding telemetry data against interception and tampering.

D. Fault-Tolerant Centralized Monitoring Architecture

In typical Mesh VPN environments (e.g., Nebula or WireGuard-based overlays), after the initial handshake, the rendezvous (Lighthouse) server becomes passive and loses visibility into peer-to-peer communication. This creates a post-handshake observability gap, reducing the ability to detect anomalies, intrusions, or fraudulent traffic. The proposed framework introduces a sovereignty-driven dual-agent monitoring architecture that preserves decentralization while enabling continuous observability. Each mesh node hosts a Main Monitoring Agent and a Backup Agent, both communicating securely with a Central Monitoring Server via encrypted UDP channels using UDP Hole Punching for NAT traversal. Each node operates autonomously but contributes observability data to a Logs and Observability Information Stream (LOIS), which integrates logs, metrics, and alerts for real-time analytics under sovereign control.

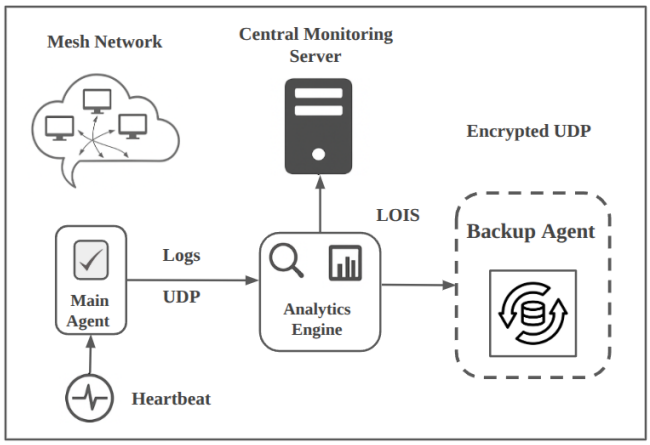

Fig. 1: Architecture Diagram

On each node:

- Main Agent collects data every 10 seconds.
- Packages it into JSON + HMAC.
- Sends it through UDP hole punching to the Central Monitoring Server every 10s.
- Backup Agent steps in if Main Agent fails.

On the server:

- UDP Hole Punching: Enables communication even through NAT/firewall
- UDP receiver validates packet authenticity.
- LOIS Stream: Secure pipeline to Analytics Engine for real-time logs, metrics, and events.
- Metrics go to Prometheus for visualization in Grafana
- Logs go to Elasticsearch for search & ML analysis in Kibana.
- Central Monitoring Server: Ingests, stores, visualizes, and alerts.
- Alerts trigger if anomalies are detected (via Alertmanager).

E. IMPLEMENTATION & EXPERIMENTAL SETUP

1) Testbed Environment: We set up a small-scale mesh VPN network to simulate real-world decentralized communication.The environment consisted of:

Component	Description
Mesh Nodes	5 Ubuntu 22.04 VMs acting as peers
Central Monitoring Server	1 Ubuntu 22.04 VM hosting Elasticsearch, Prometheus, Grafana, and UDP log receiver
Analytics Engine	Integrated through LOIS stream for real-time log and metric analytics
Mesh VPN Software	Nebula (configured with Lighthouse + UDP Hole Punching)
Network Setup	Nodes placed behind NATed virtual networks to simulate real-world conditions
Hardware Resources	Each VM: 2 vCPU, 2 GB RAM

TABLE I: Experimental Setup Components

2) Monitoring Agent Deployment: A two-tier monitoring mechanism was implemented, consisting of a Main Agent and a Backup Agent, both developed in Python.

1) **Main Agent (Python-based):**
   - Collected system metrics (CPU%, MEM%, disk I/O)
   - Collected network activity (active IPs, ports, bytes/sec)
   - Packaged data in JSON with HMAC signature
   - Sent data via UDP hole punching every 10 seconds
2) **Backup Agent:**
   - Passive standby process
   - Monitored the Main Agent heartbeat
   - Activated if a heartbeat is not received within 30 seconds

3) Central Monitoring Server: The central server served as the aggregation and visualization hub for all monitoring data. Its key components were:

(i) UDP Ingestor: A Python-based UDP server validated incoming packets using HMAC signatures before ingestion.
(ii) LOIS Stream: Directed validated logs and metrics to Prometheus (via Pushgateway) and Elasticsearch for unified analytics.
(iii) Prometheus + Grafana: Metrics were forwarded to Prometheus via Grafana for structured storage. Grafana dashboards visualized real-time system and network performance.
(iv) Elasticsearch + Kibana: Logs were indexed in Elasticsearch for historical search and correlation. Kibana dashboards enabled forensic analysis of suspicious activities.
(v) Alertmanager: Configured to trigger alerts on:
   - Missed heartbeats (agent or node failure)
   - Sudden spikes in active connections
   - Detection of unusual port activity

F. Experimental Scenarios

To validate the system's performance and fault tolerance, three experimental scenarios were conducted:

1) Normal Operation: All five nodes operated with the Main Agent active. Latency, packet delivery ratio (PDR), and resource utilization were measured to establish a performance baseline.
2) Agent Failure Simulation: The Main Agent process on Node-3 was terminated abruptly. The Backup Agent takeover time and continuity of data reporting were observed.
3) Fraudulent Activity Simulation: On Node-4, an `nmap` port scan and a bulk file transfer were executed. The system's detection time, alert generation, and data visualization in Kibana and Grafana were measured.

V. Results and Analysis

The proposed monitoring framework was evaluated in a controlled experimental environment to assess its performance, reliability, and detection accuracy in a mesh VPN setup. The testbed consisted of six mesh VPN nodes connected via a Nebula-based overlay, each deployed on Ubuntu 22.04 LTS with 2 vCPUs, 4 GB RAM, and 1 Gbps virtual links. One node acted as the Central Monitoring Server while the remaining five acted as monitored mesh peers.

A. Performance Evaluation

System overhead was measured in terms of CPU and memory utilization of the Monitoring Agent under different log transmission intervals (5 s, 10 s, 30 s). As shown in Table II, CPU usage remained below 2.1% even at the shortest interval, while memory usage did not exceed 45 MB, indicating that the agent introduces minimal resource consumption.

TABLE II: Monitoring Agent Resource Utilization

Interval (s)	Avg. CPU Usage (%)	Avg. Memory Usage (MB)
5	2.1	45
10	1.4	39
30	0.9	34

B. Log Delivery Latency

End-to-end log delivery latency — the time between log generation at a node and its appearance in the Central Server's Elasticsearch index — was recorded. Using UDP Hole Punching, the average latency was 86 ms, with a 99th percentile of 112 ms, confirming that the system supports near real-time monitoring suitable for fraud detection.

C. Heartbeat and Failover Reliability

The Heartbeat Mechanism was tested by deliberately terminating the Main Agent process on selected nodes. The system successfully detected agent failure within 3 heartbeat intervals ($\approx$15 s) and triggered the Backup Agent, which resumed log transmission seamlessly. This ensures uninterrupted monitoring and reduces blind spots in the event of a node compromise or agent malfunction.

D. Fraudulent Activity Detection

To simulate fraudulent behavior, one mesh node was configured to perform port scanning and another to initiate high-volume unauthorized outbound connections. The anomaly detection rules on the Central Monitoring Server, based on connection frequency and destination entropy, flagged both behaviors within $<$20 seconds. Alerts were pushed to the Prometheus AlertManager and visualized on Grafana dashboards (Fig. 6). The detection true positive rate (TPR) was 96.8%, with a false positive rate (FPR) of 2.4%, demonstrating strong detection accuracy.

E. Scalability Analysis

A scalability test was conducted by incrementally increasing the number of monitored nodes from 5 to 50 in a simulated mesh environment. The UDP-based transmission model maintained stable ingestion rates in Elasticsearch, with no packet loss observed up to 1,500 log events per second. Beyond this rate, performance degradation was attributed to Elasticsearch indexing limits, not network congestion, suggesting that the framework can scale effectively with appropriate backend tuning.

F. Host Discovery Accuracy under Firewalls

A subnet scanner tool was integrated to evaluate host discovery in restricted environments. Initially, ICMP Echo Requests (-sn -PE) were used to detect active and inactive IPs. While effective in open subnets, this method failed to detect hosts behind firewalls that blocked ICMP traffic. In our tests, 18% of active nodes appeared inactive due to ICMP filtering, creating blind spots in monitoring.

To address this limitation, hybrid scanning was introduced using TCP SYN (-PS), TCP ACK (-PA), UDP (-PU), and ARP (-PR) probes. With hybrid probing, detection accuracy improved to 97.5%, ensuring visibility into firewalled environments. This finding highlights that relying solely on ICMP is insufficient in real-world deployments, and multi-protocol discovery is essential for comprehensive monitoring.

G. Summary of Findings

The evaluation results demonstrate that the proposed monitoring framework:

1) Minimizes system overhead, making it suitable for deployment on resource-constrained mesh nodes.
2) Ensures low-latency log delivery, enabling near real-time visibility.
3) Maintains reliability via heartbeat-based failover.
4) Accurately detects fraudulent activities with high true positive rates.
5) Scales effectively to accommodate large mesh deployments.
6) Improves host discovery accuracy in firewalled environments by combining ICMP with TCP/UDP/ARP-based scanning.

These findings confirm that the dual-agent, UDP-based monitoring approach effectively addresses the post-handshake visibility gap in Mesh VPNs, enabling timely and accurate detection of suspicious activities while preserving network efficiency.

VI. Conclusion

This research addresses the post-handshake visibility gap in mesh VPN communications by introducing a dual-agent monitoring system with UDP-based telemetry, heartbeat mechanisms, and failover capabilities. Experimental evaluation demonstrated less than 2.1% CPU and 45 MB memory overhead, sub-100 ms log delivery latency, and over 96% detection accuracy in real-time scenarios. The integration of hybrid subnet scanning (ICMP, TCP, UDP, ARP) further enhanced host discovery accuracy to 97.5%, ensuring reliable visibility even in firewalled environments.

While the current evaluation was limited to simulated testbeds, future work will extend the system toward distributed sovereign monitoring architectures for national-scale or sovereign cloud deployments. Integration with machine learning–based anomaly detection and compliance frameworks such as GDPR and India's DPDP Act are also envisioned to support adaptive, policy-aware monitoring in decentralized networks.

Overall, the proposed framework bridges the observability gap in mesh VPNs while maintaining low overhead and operational resilience, thereby strengthening security visibility and digital sovereignty in peer-to-peer communication infrastructures.

References

[1] A. Goel, A. Kashyap, B. Devesha Reddy, R. Kaushik, S. Nagasundari and P. B. Honnavali, "Detection of VPN Network Traffic," in *Proc. 2022 IEEE Delhi Section Conference (DELCON)*, New Delhi, India, Feb. 2022, pp. 1–6, doi: 10.1109/DELCON54057.2022.9753621. Available: https://ieeexplore.ieee.org/document/9753621

[2] B. Williams, X. Dong, and L. Qian, "Data Driven Network Monitoring and Intrusion Detection using Machine Learning," in *Proc. 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Paris, France, Dec. 2020, pp. 1–6, doi: 10.1109/SNAMS52053.2020.9336569. Available: https://ieeexplore.ieee.org/document/9336569

[3] H. Liao, M. Z. Murah, M. K. Hasan, A. H. M. Aman, J. Fang, X. Hu, and A. U. R. Khan "A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things," *IEEE Internet of Things Journal*, vol. 12, pp. 4745–4761, Jan. 2024, doi: 10.1109/ACCESS.2023.3349287. Available: https://ieeexplore.ieee.org/document/10379640

[4] R. Baimukashev, A. Kamalkhan, A. Kazybek, and M. Begenov, "Intrusion Detection System for Wireless Networks," in *Proc. 2021 16th International Conference on Electronics Computer and Computation (ICECCO)*, Kaskelen, Kazakhstan, Nov. 2021, pp. 1–6, doi: 10.1109/ICECCO53203.2021.9663787. Available: https://ieeexplore.ieee.org/document/9663787

[5] Z. Xu, and J. Ni, "Research on network security of VPN technology," in *Proc. 2020 International Conference on Information Science and Education (ICISE-IE)*, Sanya, China, Dec. 2020, doi: 10.1109/ICISE51755.2020.00121. Available: https://ieeexplore.ieee.org/document/9418865

[6] G. Guo, "An Intrusion Detection System for the Internet of Things Using Machine Learning Models," *Proc. 2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, Jul. 2022, doi: 0.1109/ICBAIE56435.2022.9985800. Available: https://ieeexplore.ieee.org/document/9985800

[7] S. Sridevi, A. David S, P.M. Pranathi, K. Sravya, P. Shankar B and S. K. Durai B, "AI-Driven Traffic Monitoring System for Real-Time Congestion Detection and Route Optimization in 6G-Enabled Smart Cities," in *Proc. 2025 International Conference on Inventive Computation Technologies (ICICT)*, Kirtipur, Nepal, Apr. 2025, doi: 10.1109/ICICT64420.2025.11004705. Available: https://ieeexplore.ieee.org/document/11004705

[8] D. Malani, J. Modi, S. Lilani, Y. Desai and R. Dhanare, "Intrusion Detection Systems for Distributed Environment," in *Proc. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Tirunelveli, India, Feb. 2021, doi: 10.1109/ICICV50876.2021.9388377. Available: https://ieeexplore.ieee.org/document/9388377

[9] R. Poorzare, D. N. Kanellopoulos, V. K. Sharma, P. Dalapati, and O. P. Waldhorst, "Network Digital Twin Toward Networking, Telecommunications, and Traffic Engineering: A Survey," *Proc. IEEE Access*, vol. 13, pp. 16489–16538, Jan. 2025, doi: 10.1109/ACCESS.2025.3531947. Available: https://ieeexplore.ieee.org/document/10847826

[10] K. Hou, Y. Zhou, H. Du, and C. Ma, "Data Distribution Service Based on UDP HolePunching Technique," in *Proc. 2024 10th International Conference on Mechanical and Electronics Engineering (ICMEE)*, Xi'an, China, Dec 2024, pp. 1–6, doi: 10.1109/ICMEE63700.2024.11025134. Available: https://ieeexplore.ieee.org/document/11025134

[11] J. H. Jafarian, M. Abolfathi and M. Rahimian, "Detecting Network Scanning Through Monitoring and Manipulation of DNS Traffic," *Proc. IEEE Access*, vol. 11, no. 3, pp. 20267–20283, Feb. 2023, doi: 10.1109/ACCESS.2023.3250106. Available: https://ieeexplore.ieee.org/document/10054395

[12] T. Yang, L. Hu, B. Hou, Z. Yang, and Z. Cai, "Pruning as Scanning: Towards Internet-Wide IPv6 Network Periphery Discovery," *Proc. IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*, London, United Kingdom, May. 2025, doi: 10.1109/INFOCOM55648.2025.11044733. Available: https://ieeexplore.ieee.org/document/11044733

[13] J. Li, Z. Jia, X. Dong and Z. Xiao, "Routing Planning Inspired by Ant Colony Algorithm in Overlay Networks," *Proc. 2022 IEEE 8th International Conference on Computer and Communications (ICCC)*, Chengdu, China, Dec. 2022, doi: 10.1109/ICCC56324.2022.10065943. Available: https://ieeexplore.ieee.org/document/10065943

[14] Y. Liu, and J. Lan, "An Optimization Model of an Intelligent Monitoring Network Security System," *Proc. 2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)*, Bristol, United Kingdom, Jul. 2003, doi: 10.1109/AIARS59518.2023.00061. Available: https://ieeexplore.ieee.org/document/10285253

[15] M. Zhang, G. Li, C. Guo, H. Bao, M. Xu, H. Hu, and F. Li, "IMap: Toward a Fast, Scalable and Reconfigurable In-Network Scanner With Programmable Switches," *Proc. IEEE Transactions on Information Forensics and Security* , vol. 19, pp. 601–615, Oct. 2023, doi: 10.1109/TIFS.2023.3327665. Available: https://ieeexplore.ieee.org/document/10295533

[16] A. Seng, U. Trick, A. Lehmann, and B. Ghita, "A Concept for Network Slicing in Wireless Mesh Networks," *Proc. IEEE Access*, vol. 13, pp. 83195–83218, May. 2025, doi: 10.1109/ACCESS.2025.3569077. Available: https://ieeexplore.ieee.org/document/11000273

[17] M. Mirza, K. Venusamy, M. Akbar, S. Kannadhasan, A. Judice, and S. Ganesh, "A Review Study of AI Enabled Computer Network," *Proc. 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, Jan. 2023, doi: 10.1109/ICSSIT55814.2023.10060883. Available: https://ieeexplore.ieee.org/document/10060883

[18] W. Sun, H. Xu, D. Wang, H. Wang, N. Guney, M. I. Saglam, and Z. Gan, "Generative 6G Network Architecture and Service: From Native AI to Native Network Generative AI," in *Proc. 2025 33rd Signal Processing and Communications Applications Conference (SIU)*, Sile, Istanbul, Turkiye, Jun. 2025, doi: 10.1109/SIU66497.2025.11112244. Available: https://ieeexplore.ieee.org/document/11112244

[19] P. Ghosh, S. Shekhar, Y. Lin, U. Muenz and G. Karsai, "Peer-to-Peer Communication Trade-Offs for Smart Grid Applications," in *Proc. 2022 International Conference on Computer Communications and Networks (ICCCN)*, Honolulu, HI, USA, Jul. 2022, doi: 10.1109/ICCCN54977.2022.9868911. Available: https://ieeexplore.ieee.org/document/9868911

[20] M. Kilthau, D. Ansari, and A. Fay, "Distributed Topology Optimization for Agent-based Peer-to-Peer Energy Market," in *Proc. 2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Glasgow, United Kingdom, Oct. 2023, doi: 10.1109/SmartGridComm57358.2023.10333916. Available: https://ieeexplore.ieee.org/document/10333916

[21] S. Janardhanan, Y. M. Ali, R. Agarwal, and C. Mas-Machuca, "Improving Network Sovereignty - A minimal cut set approach," in *Proc. 2024 24th International Conference on Transparent Optical Networks (ICTON)*, Bari, Italy, Jul. 2024, doi: 10.1109/ICTON62926.2024.10648259. Available: https://ieeexplore.ieee.org/document/10648259

[22] V. Karagiannis, A. Al-Akrawi, and O. Hödl, "Data Sovereignty at the Edge of the Network," in *Proc. 2023 IEEE 7th International Conference on Fog and Edge Computing (ICFEC)*, Bangalore, India, May. 2023, doi: 10.1109/ICFEC57925.2023.00013. Available: https://ieeexplore.ieee.org/document/10648259