An Intelligent Intent-Based Network Management System for Automated Configuration and Governance

DR. B. Anuja Beatrice, Head and Associate Professor,

Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

Srivarshini K, Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

Abstract

Modern enterprise networks operate within highly dynamic environments characterized by virtualization, cloud-native architectures, distributed applications, and Software Defined Networking (SDN). As infrastructures expand, manual configuration of routers, switches, firewalls, and access devices through command-line interfaces becomes increasingly inefficient and error-prone. Configuration inconsistencies, human errors, policy conflicts, and compliance violations can severely affect network availability, confidentiality, and integrity. Traditional network management approaches primarily operate in reactive modes, identifying misconfigurations only after deployment, which increases operational risk and administrative overhead.

This paper presents an Intent-Based Networking Controller designed to automate policy deployment and compliance verification within enterprise networks. The proposed framework enables administrators to define high-level business intent rather than low-level configuration commands.

The system translates structured intent into device-specific configurations, performs conflict detection prior to deployment, executes secure automation procedures, and validates compliance post-deployment. The architecture integrates intent parsing, policy mapping, command generation, automated configuration execution, and centralized logging within a modular backend environment.

The system provides real-time monitoring, structured event logging, and audit-ready compliance reporting. Experimental validation demonstrates reduced configuration latency, improved policy consistency, minimized manual errors, and enhanced traceability. The modular architecture ensures scalability and extensibility for future enhancements. The proposed framework contributes to the advancement of intelligent, autonomous, and policy-driven enterprise networking systems.

Keywords: Intent-Based Networking, Network Automation, Policy Deployment, Compliance Verification, Conflict Detection, Software Defined Networking, Enterprise Network Management, Secure Configuration Automation, Network Governance, Hybrid Policy Framework.

1. Introduction

The rapid advancement of digital technologies has led to the widespread adoption of cloud computing, virtualization, Internet of Things (IoT), edge computing, and distributed enterprise applications. Modern organizations depend heavily on interconnected network infrastructures to deliver services, ensure secure communication, and support mission-critical operations. As networks expand in size and complexity, managing configurations across multiple routers, switches, firewalls, and virtual devices becomes increasingly challenging.

Traditional network management approaches rely primarily on manual configuration through command-line interfaces (CLI). Although CLI-based configuration provides granular control, it introduces significant operational challenges in large-scale environments. Administrators must configure each device individually, which increases deployment time and the probability of configuration errors. Even minor mistakes in syntax, access control lists, routing rules, or VLAN assignments can lead to service disruptions, security breaches, or compliance violations.

Several limitations of traditional network management systems can be identified:

• **Manual Configuration Overhead** – Device-level configuration requires repetitive manual commands, increasing operational workload.
• **Policy Inconsistency** – Ensuring uniform policy enforcement across multiple devices is difficult without centralized automation.
• **Configuration Drift** – Over time, deployed configurations may deviate from intended policies due to incremental changes.
• **Limited Scalability** – As the number of network devices increases, manual configuration becomes inefficient and unsustainable.
• **Delayed Compliance Verification** – Traditional systems validate configurations only after deployment, increasing risk exposure.
• **Higher Probability of Human Error** – Manual input increases the likelihood of misconfigurations and operational disruptions.

To address these challenges, Intent-Based Networking (IBN) has emerged as a transformative paradigm in network management. IBN abstracts low-level technical complexity by enabling administrators to define high-level business intent rather than device-specific commands. The system automatically interprets the defined intent and translates it into appropriate configuration policies. This approach reduces manual intervention and enhances automation.

An effective intent-based framework should incorporate the following capabilities:

• Automated translation of business objectives into device configurations
• Proactive conflict detection before deployment
• Secure automated configuration execution
• Continuous compliance validation
• Centralized logging and audit tracking
• Scalable and modular system architecture

Despite advancements in Software Defined Networking (SDN) and policy-based automation, many existing systems lack integrated compliance verification and structured conflict detection within a unified platform. There remains a need for a modular, secure, and scalable framework that unifies intent abstraction,

automated policy deployment, and post-deployment validation.

This study proposes an Intent-Based Networking Controller that addresses these gaps by integrating structured intent processing, policy mapping, conflict detection, secure deployment, compliance validation, and centralized logging within a cohesive architecture. The objective is to transform traditional reactive network management into a proactive, automated, and policy-driven governance framework suitable for modern enterprise environments.

2. Literature Review

The concept of network automation has evolved significantly over the past decade, driven by the increasing complexity of enterprise infrastructures. Early network management systems relied on manual configuration and device-specific command execution. Although effective in small-scale environments, these approaches lacked scalability and centralized governance mechanisms. As enterprise networks expanded, researchers began exploring programmable networking paradigms to address these limitations.

2.1 Traditional Network Management Approaches

Traditional networking frameworks depend on manual configuration using Command Line Interfaces (CLI) and static configuration files. Administrators configure routing protocols, VLANs, firewall rules, and access control policies individually on each device. While this provides fine-grained control, it introduces several operational inefficiencies such as configuration redundancy, version mismatch, and increased risk of human error.

Research indicates that configuration errors are a major cause of network outages in enterprise systems. Studies on configuration management highlight the prevalence of policy inconsistencies and configuration drift over time. These issues emphasize the need for centralized and automated management frameworks.

2.2 Software Defined Networking (SDN)

Software Defined Networking (SDN) introduced a paradigm shift by separating the control plane from the data plane. SDN controllers enable centralized

management of network flows and provide programmability through APIs.

Researchers have demonstrated that SDN improves flexibility and simplifies network orchestration. However, SDN primarily focuses on flow control and programmability rather than business-level intent abstraction. Administrators still need to translate high-level requirements into technical flow rules manually.

While SDN enhances automation capabilities, it does not inherently provide structured conflict detection or compliance validation mechanisms within enterprise environments.

2.3 Policy-Based Network Management

Policy-based management systems were introduced to ensure consistent rule enforcement across multiple devices. These systems define policies in structured formats and apply them automatically across network elements.

Although policy-based systems improve configuration consistency, many implementations lack real-time validation and proactive conflict detection. Policies may be deployed successfully but remain unverified against compliance standards. Additionally, these systems often do not provide audit-ready logging mechanisms necessary for governance and regulatory requirements.

2.4 Intent-Based Networking (IBN) Paradigm

Intent-Based Networking extends policy-based management by abstracting device-level configuration into business-level intent. In IBN architectures, administrators define desired outcomes rather than specific commands. The system interprets the intent, translates it into configuration rules, deploys them automatically, and continuously verifies network state alignment.

Recent studies in IBN emphasize automation, scalability, and simplified network orchestration. However, several existing frameworks remain theoretical or vendor-specific. Many implementations lack open and modular architectures suitable for academic experimentation and performance evaluation.

Moreover, existing intent-based solutions often focus on deployment automation

while providing limited insight into structured conflict detection, compliance verification logic, and centralized audit management.

2.5 Configuration Conflict Detection and Compliance

Configuration conflicts such as duplicate VLAN assignments, overlapping IP address ranges, and inconsistent access control lists can significantly impact network stability. Research in configuration verification suggests the use of rule-based validation engines to detect logical inconsistencies prior to deployment.

Compliance management tools verify whether deployed configurations align with predefined standards. However, these tools frequently operate as post-deployment validation mechanisms rather than integrated components within automation frameworks. The absence of proactive validation increases risk exposure.

3. Proposed Framework

The proposed system introduces a modular Intent-Based Networking Controller that integrates network modeling, automated configuration deployment, and compliance verification within a unified architecture.

The network is logically represented as:

$$\boldsymbol{N = (D, P)}$$

Where:

- Drepresents network devices
- Prepresents policy definitions

The system workflow includes:

1. Intent submission via dashboard
2. Intent parsing and validation
3. Policy mapping and rule structuring
4. Conflict detection analysis
5. Command generation
6. Secure SSH-based deployment
7. Post-deployment compliance verification
8. Centralized logging and reporting

The architecture includes the following modules:

- Authentication Module
- Intent Processing Module
- Policy Mapping Engine
- Conflict Detection Module
- Automation Engine
- Compliance Validation Module
- Centralized Logging and Audit Module

The modular design ensures extensibility and scalability.

4. System Implementation and Maintenance

4.1 Architectural Design Approach

The system follows a layered and modular architectural design to ensure separation of concerns and maintainability. The architecture is divided into presentation layer, processing layer, automation layer, and persistence layer. Each layer performs specific responsibilities and interacts through structured interfaces.

The presentation layer consists of a web-based dashboard that enables administrators to submit structured network intent. The processing layer handles intent validation, parsing, policy mapping, and command generation. The automation layer establishes secure communication with network devices for configuration deployment. The persistence layer stores policies, execution logs, compliance results, and timestamps in a centralized relational database.

This layered architecture improves scalability and allows independent upgrades of individual modules without affecting the overall system functionality.

4.2 Intent Processing Workflow

The implementation workflow begins when a user submits high-level network intent through the dashboard. The submitted intent undergoes structured validation to ensure syntactic correctness and logical consistency. Mandatory parameters such as VLAN ID, IP address range, access policy type, and target device are verified before further processing.

After validation, the intent parsing module extracts key attributes and converts them into structured policy objects. These structured objects are then passed to the policy mapping module, where predefined templates translate business-level requirements into technical configuration rules.

This structured workflow ensures accuracy and minimizes ambiguity in configuration generation.

4.3 Command Generation and Automation

The command generation module dynamically constructs device-specific configuration commands. Since different network devices may require unique syntax formats, the module ensures compatibility by applying device-type-specific templates.

For secure deployment, the automation module establishes SSH-based communication using network automation libraries. It executes generated commands sequentially and captures device responses. The system monitors execution success or failure and records status details in the database.

Error handling mechanisms are implemented to manage connectivity issues, authentication failures, and invalid command execution scenarios. If an error

occurs, the system logs detailed diagnostic information for troubleshooting.

4.4 Conflict Detection Mechanism

Before deployment, the system performs proactive conflict detection to prevent configuration inconsistencies. The conflict detection module analyzes existing database records and compares new intent parameters with previously deployed policies.

Examples of detected conflicts include:

- Duplicate VLAN identifiers
- Overlapping IP address ranges
- Redundant or conflicting access control rules
- Policy redefinition on the same device

If a conflict is detected, the system generates a warning message and prevents deployment until corrective action is taken. This proactive validation significantly reduces operational risk.

5. Results & Analysis

The proposed Intent-Based Networking Controller was evaluated in a simulated enterprise network environment consisting of multiple virtual devices, including routers and switches configured through automated deployment mechanisms. The evaluation focused on deployment efficiency, conflict detection accuracy, compliance validation success rate, and overall system stability.

5.1 Deployment Efficiency Analysis

The automation-driven deployment mechanism significantly reduced configuration time compared to manual CLI-based configuration. In traditional environments, configuring VLANs, routing rules, and access control policies across multiple devices requires sequential device access and repetitive command execution.

With the proposed controller, high-level intent was translated into configuration commands automatically and deployed simultaneously across devices. The average deployment time per policy was reduced substantially due to centralized automation and structured processing. This demonstrates the operational advantage of intent abstraction and automated orchestration.

5.2 Conflict Detection Performance

The conflict detection module was tested with scenarios involving:

- Duplicate VLAN ID requests
- Overlapping IP address ranges
- Redundant access control policies
- Reassignment of policies to previously configured devices

The system successfully identified logical conflicts before deployment in all test scenarios. This proactive validation mechanism prevented configuration inconsistencies that could otherwise lead to network instability.

The conflict detection accuracy remained high across repeated testing cycles, demonstrating robustness in rule-based validation logic.

5.3 Compliance Verification Accuracy

Post-deployment compliance verification was evaluated by comparing intended configurations with actual device states. The system retrieved configuration details from network devices and matched them against structured policy records.

Results indicate that compliance verification successfully identified both fully compliant and partially mismatched configurations. In cases where manual device alterations were introduced after

deployment, the system correctly flagged deviations as non-compliant.

This continuous verification capability reduces configuration drift and strengthens network governance.

6. Discussion

The integration of intent abstraction, automation deployment, conflict detection, and compliance validation transforms traditional reactive network management into a proactive governance framework.

6.1 Impact on Enterprise Network Management

By abstracting device-level complexity, the system allows administrators to focus on business objectives rather than technical command syntax. This reduces cognitive load and minimizes human error.

The centralized control model enhances visibility across network infrastructure. Administrators gain real-time insight into policy deployment status, compliance results, and historical execution logs.

6.2 Governance and Audit Advantages

The structured database logging mechanism provides accountability and traceability. Every policy submission, deployment attempt, compliance verification result, and timestamp is recorded.

This capability supports regulatory compliance requirements in enterprise environments where audit readiness is critical.

6.3 Conflict Prevention as a Security Enhancement

Proactive conflict detection not only prevents configuration errors but also

strengthens network security posture. By identifying overlapping IP ranges or conflicting access policies before deployment, the system reduces exposure to misconfiguration-based vulnerabilities.

6.4 Limitations

Despite its advantages, the proposed system has certain limitations:

• Device-specific templates must be maintained for different vendors
• Large-scale enterprise deployment may require distributed database support
• Real-time telemetry integration is not yet implemented
• AI-driven predictive policy optimization is not included

These limitations highlight opportunities for further enhancement and research expansion.

7. Future Work & Conclusion

7.1 Future Enhancements

Several enhancements can further improve the system:

• Integration of Machine Learning models for predictive conflict detection
• Real-time telemetry monitoring for adaptive policy enforcement
• Cloud-native deployment with distributed databases
• Support for multi-vendor device abstraction
• Graph-based network topology visualization
• AI-driven intent recommendation engine
• Automated rollback mechanism for failed deployments

Advanced deep learning models may also be integrated to analyze historical policy patterns and recommend optimized configurations.

7.2 Conclusion

The proposed Intent-Based Networking Controller presents a structured and practical approach to enterprise network automation. By translating high-level business intent into automated device configurations, the system reduces operational complexity and enhances deployment efficiency.

The integration of conflict detection and compliance verification ensures configuration accuracy and governance alignment. Centralized logging strengthens audit capabilities and accountability.

Experimental evaluation confirms that the system improves deployment speed, reduces manual errors, and enhances policy consistency compared to traditional CLI-based management approaches.

Overall, the proposed framework contributes toward intelligent, scalable, and policy-driven network automation. It provides a strong foundation for future development of autonomous networking systems capable of adapting to dynamic enterprise environments.

Appendices

References

- Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 17(1), 4–27.
- Feamster, N., Rexford, J., & Zegura, E. (2014). The Road to SDN: An Intellectual History of Programmable Networks. *ACM SIGCOMM Computer Communication Review*, 44(2), 87–98.
- Nadeau, T., & Gray, K. (2013). *SDN: Software Defined Networks*. O'Reilly Media.

- Goransson, P., & Black, C. (2017). *Software Defined Networks: A Comprehensive Approach*. Morgan Kaufmann.
- Zhang, H., Chen, X., & Li, Y. (2024). Artificial Intelligence–Driven Intent-Based Networking for Autonomous Network Operations. *IEEE Network*, 38(2), 45–52.
- Kim, J., Park, S., & Lee, H. (2024). Advancements in Intent-Based Networking: Architecture, Challenges, and Future Directions. *IEEE Communications Surveys & Tutorials*.
- Gupta, A., & Kumar, R. (2024). AI-Enabled Network Automation Using Software Defined Networking. *Springer International Publishing*.
- Li, Z., & Zhao, M. (2025). Towards Fully Autonomous Networks: Integrating SDN, NFV, and Intent-Based Systems. *IEEE Access*.
- Open Networking Foundation (ONF). (2024). Software-Defined Networking and Network Automation Framework. ONF Technical Report.
- Cisco Systems. (2024). Cisco Intent-Based Networking Architecture Guide. Cisco White Paper.
- Juniper Networks. (2024). AI-Driven Enterprise Networks and Intent-Based Automation. Juniper Technical Documentation.
- Microsoft Azure Networking Team. (2024). Cloud-Based Network Automation and Intent-Driven Infrastructure. Microsoft Technical Documentation.