

Automated Network Management for Large Networks

Nguyen Le, Florian Wiedner*

**Chair of Network Architectures and Services*

School of Computation, Information and Technology, Technical University of Munich, Germany

Email: nguyen.le@tum.de, wiedner@net.in.tum.de

Abstract—The management of large-scale networks remains complex and largely manual despite advances in network management solutions. This paper examines the key challenges in automating network management at scale, focusing on complexity arising from heterogeneous and legacy environments, the lack of reliable pre-deployment validation for configuration changes, and scalability challenges related to telemetry overhead and bandwidth fairness. It then assesses established approaches, namely Software-Defined Networking and script-based automation, and discusses how they reduce operator effort and improve traffic control. Next, the paper analyzes emerging paradigms, namely Network Digital Twin (NDT) and Intent-Based Networking (IBN), as promising directions for addressing the remaining challenges. NDT enables safer pre-deployment validation and can support predictive analysis. IBN raises the abstraction level of operations by translating high-level intents into device-level configurations and supporting closed-loop automation. Building on this comparison, the paper argues that combining complementary paradigms can address a broader set of challenges than any single approach in isolation. Overall, the analysis indicates that all challenges identified in this work, except the lack of standardized, vendor-agnostic data models, are being actively addressed by current research and approaches discussed.

Index Terms—Network management automation, large-scale networks, network digital twin, intent-based networking, software-defined networking

1. Introduction

Despite advances in network management tools, operating large-scale networks remains largely manual. As networks grow in size and heterogeneity (i.e., diversity across device types, vendors and application requirements), this manual operation leads to inefficiency and scalability challenges. Collecting and storing operational data at scale can require extremely large storage [1]. At the same time, legacy infrastructure often lacks the interfaces required for automation [2]. Multiple management systems coexist, each tied to a specific management protocol [3], and effective automation requires deep expertise in both networking and the tools themselves [4].

These limitations have motivated ongoing research into more advanced network management paradigms that aim to reduce manual effort, improve scalability and efficiency. This paper begins by identifying the key challenges associated with automating network management

in large-scale networks. It then systematically analyzes widely adopted network management approaches, including Software-Defined Networking (SDN) and script-based automation (SBA), highlighting their limitations and the key challenges they aim to address. Subsequently, emerging paradigms such as Network Digital Twin (NDT) and Intent-Based Networking (IBN) are discussed as promising candidates for addressing the unresolved challenges.

Building on this analysis, the paper then examines how hybrid combinations of these approaches can help resolve key management challenges more effectively. Selected research works are also reviewed to illustrate potential directions for addressing the remaining challenges identified in this paper. Overall, the paper aims to assess whether any single approach provides a sufficiently complete solution for large-scale environments, or whether combining multiple paradigms is necessary to achieve scalable and reliable automation.

The remainder of this paper is organized as follows. Section 2 presents the key challenges in automating large-scale networks. Section 3 and Section 4 respectively review established and emerging approaches to network management and evaluate, for each approach, the challenges it addresses and its limitations. Section 5 explores how combining the discussed approaches can mitigate key challenges. Section 6 reviews related work, and Section 7 concludes the paper.

2. Challenges

This section identifies the key challenges that motivate the network automation paradigms discussed in the subsequent sections.

High expertise requirements for network management systems: Heterogeneous networks typically use different management protocols that are not mutually compatible. Therefore, network operators are required to master multiple network management tools, often at least one for each device type or network domain [3].

Pre-deployment validation of configuration changes: In large-scale networks, operators often cannot safely verify the impact of configuration updates (e.g., routing or firewall policy changes) on reachability and performance by virtually testing them in an isolated environment before rollout, so errors may only surface after the change is applied [1], [5].

Lack of standardized data models within the network: According to a survey conducted by Enterprise Management Associates (EMA) [2], limited standardization results in a smaller scale of network automation, as

operators must individually configure each device. For example, inconsistent vendor data representations for inventory and configuration make it difficult to build reusable automation workflows. A network source of truth (NSoT) has emerged as a key concept for enabling consistent and standardized representation of network state and network intent. It serves as a centralized repository that contains all related data that network administrators need for network configuration.

Limited automation interfaces in legacy infrastructure: Large-scale enterprise networks are often existing production environments in which older devices constrain automation. In practice, this can mean that vendor automation solutions do not fully support older products, forcing operators to fall back to manual or device-specific workflows and slowing down consistent rollout at scale [2].

Bandwidth unfairness arising from heterogeneous traffic: Different traffic classes commonly traverse the same physical medium to maximize bandwidth utilization. For example, surveillance video streams, regular data traffic and multicast discovery traffic often share the same links, which can cause contention when cameras continuously generate high-definition video frames to capture events inside a building [6]. Managing this contention is a network management challenge, since network operators need to define and enforce quality of service (QoS) policies to prevent one traffic class from starving others during congestion [7].

Collection and storage of data: With the increasing use of NDT to simulate real-world scenarios, the collection of telemetry plays a significant role, as these network modeling tools utilize machine learning (ML) concepts, such as neural networks (NNs), to generate insights. Moreover, in large-scale data centers, the majority of traffic consists of short flows, and storing fine-grained statistics for all flows could result in significant data volume [1].

For continuous monitoring of a single aggregate (e.g., the percentage of links above 50 % utilization), it is unrealistic to achieve maximum accuracy due to the huge amount of fine-grained statistics that need to be collected and processed. Instead, monitoring protocols could be designed to provide estimates with low overhead, small delay, and a high degree of robustness [8].

3. Current Approaches

This section reviews the currently adopted approaches for network management and discusses their limitations, as well as the challenges they aim to address. It focuses on SDN and SBA.

3.1. Software-Defined Networking

SDN is an approach that decouples the control plane from the data plane, thereby increasing the programmability of network devices and facilitating dynamic network management through software. The control plane denotes the software logic that provides traffic control functionality, whereas the data plane primarily performs packet switching and forwarding [9].

3.1.1. Architecture. The architecture of SDN is illustrated in Fig. 1, consisting of three layers [9].

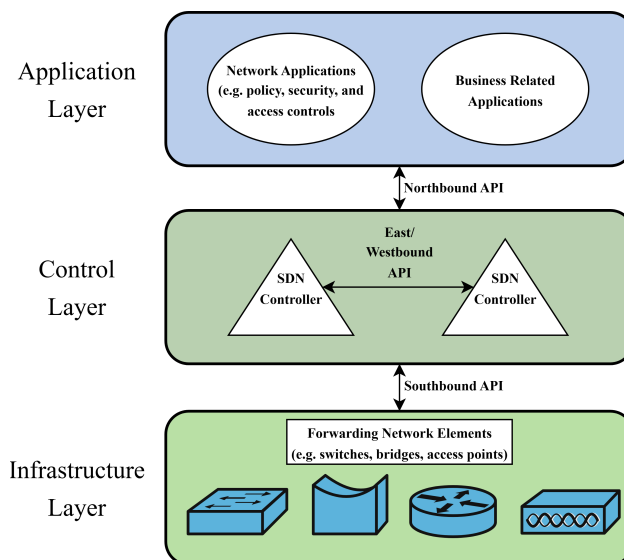


Figure 1: SDN Architecture (adapted from [9])

Application Layer: The Application Layer typically contains network applications as well as related business applications [9]. A relevant use case would be replacing a hardware firewall or load balancer with a software-based solution [10].

Control Layer: The Control Layer consists of the SDN controller software, which manages policies and supervises packet flow behavior [9]. In comparison, the control layer makes network decisions, while the application layer expresses resource needs [10]. It is a central component that uses three interfaces (northbound, southbound, and east/westbound) to communicate with its peers and the other two layers.

Infrastructure Layer: The Infrastructure Layer comprises physical switches or other devices within the network [10]. These devices perform packet switching and forwarding based on the entries inside the flow table. They may be populated in a proactive, reactive, or hybrid manner. When the proactive mode is used, the controller loads the traffic rules that match a particular switch in advance. In the reactive mode, the rules are loaded on demand [11].

3.1.2. Limitations. Based on the multi-domain emulation experiments in [9], latency in the legacy scenario is lower than in the SDN scenarios under the tested conditions. In particular, the legacy setup reports latencies mostly below 2.5 ms, whereas the SDN scenarios exhibit minimum latencies of 6.5 ms (SDN with Border Gateway Protocol) and 8 ms (SDN without Border Gateway Protocol), largely due to additional control-plane interaction and overhead from populating flow table entries (e.g., reactive flow setup adds delay for first packets in a flow).

3.1.3. Addressed Challenges. By decoupling the control and data planes, network switches become simple packet-forwarding devices, while the control logic is moved to a logically centralized controller, thereby simplifying policy enforcement and network configuration and evolution [11].

SDN provides network-wide data-flow control, enabling administrators to define flows that meet specific requirements of different traffic classes [12]. This facilitates more effective traffic engineering and fairer bandwidth allocation across heterogeneous traffic types.

3.2. Script-based automation

SBA utilizes programming languages such as Python, Java, or Bash to automate repetitive networking tasks. A script might rely on protocols such as SSH and SNMP, or use command-line interfaces (CLIs) [13], and utilize available binaries on the remote target device to execute tasks such as configuring interfaces, editing routing tables, and adjusting firewall rules. These scripts are highly customizable and operate at a higher level of abstraction, enabling network professionals to perform ad hoc configuration across multiple devices without having to deal with low-level or platform-specific details.

3.2.1. Common languages. As discussed in [14], Python is a powerful language that comes with numerous libraries for network configuration, such as Netmiko or Paramiko. The authors report that automating the configuration of 36 Cisco routers takes, on average, only 6 s when the code includes multithreading and queueing. In contrast, other authors report that performing the same task manually takes 5797 s [15].

Another popular tool among network engineers is Ansible, which Red Hat characterizes as the de facto language for enterprise network automation, with broad adoption across diverse network domains. This is largely due to the extensive ecosystems of certified content and integrations that cover multi-domain and multi-vendor architectures [16].

3.2.2. Limitations. Network administrators often use scripts to handle one-off or small automation tasks, which is feasible in small-scale network environments. Nevertheless, once a system scales to hundreds or thousands of servers, the use of scripts without adequate error handling or documentation could lead to system-wide misconfigurations, since such scripts are typically applied uniformly across a large number of machines [13].

3.2.3. Addressed Challenges. Script-based approaches can be adopted at low cost using freely available scripting languages and libraries. Some libraries, such as Netmiko and NAPALM (Network Automation Programmability Abstraction Layer with Multivendor support), abstract underlying connection and device-specific details, enabling network administrators to automate tasks without handling low-level protocols directly. As a result, network management tasks can be performed more easily and efficiently [17].

Overall, SDN and SBA can reduce operator effort and improve traffic control, thereby addressing the high expertise requirements and bandwidth unfairness challenges to a significant extent. However, several challenges remain open in large-scale environments. Accordingly, the next section reviews emerging approaches such as NDT and IBN and analyzes which of these remaining challenges they address.

4. Emerging Approaches

This section reviews NDT and IBN and evaluates which remaining challenges they address.

4.1. Network Digital Twin

A digital twin is a virtual representation of a physical object, system, or process within a digital environment. In this work, an NDT is a virtualized replica of the essential components of the network infrastructure, enabling operators to perform what-if analyses, troubleshoot issues, and plan network upgrades in a safe and controlled environment [1].

The NDT is a data-driven way to represent the current state of the network through network-related data. This data could come from various sources, including real-world traffic and simulated traffic from testbeds or tools [1].

Currently, the market adoption of NDT is mostly limited to simpler capabilities such as policy validation, migration to cloud, or single-vendor what-if analyses. More advanced, autonomous, multi-vendor deployments that support complex what-if scenarios remain mostly aspirational because they are expensive to implement and still lack the ability to reliably predict outcomes in dynamic and complex networks [18].

4.1.1. Architecture. At present, there is no standardized definition of the characteristics of an NDT. The IETF draft in [5] proposes five key elements (data, models, mapping, interface, and logic) to characterize it. Fig. 2 illustrates these elements.

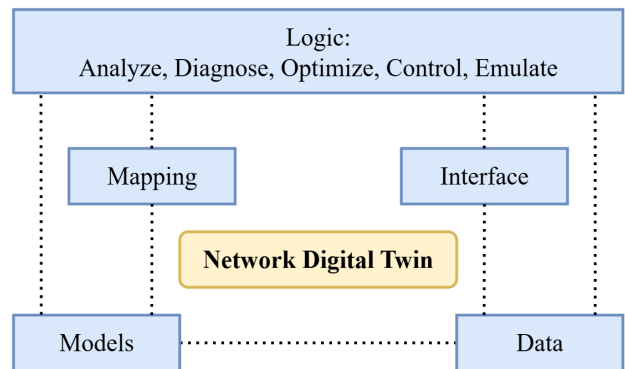


Figure 2: Key Elements of NDT (adapted from [5])

Data: Data is considered the single source of truth and is stored in a data repository. It contains historical and/or real-time data (e.g., configuration, metric data) of the real-world twin that are required by the models to understand real-world behaviors.

Models: Models enable the emulation of changes in the configuration, state, and resource usage by capturing the behaviour of a real network and generating reasoning information that may be useful in operational decision-making.

Interfaces: There are two types of interfaces. The first interface is between the NDT platform and the real network infrastructure, enabling the collection of real-time data. The second interface connects the NDT platform

to the network management applications (e.g., network optimization, visualization tools [19]), which enables bidirectional communication, exposing numerous platform functionalities to applications, as well as transmitting application requests to the NDT platform.

Mapping: Mapping defines the relationship between the NDT and the underlying network entities, enabling synchronization between the real network and its virtual counterpart, or among multiple NDT instances, which is relevant in multi-domain deployments where separate NDT instances are maintained per administrative domain. The observability provided by this mapping allows the NDT to correctly reflect and analyze the state of the real network.

Logic: Logic is the decision-making component of the NDT, responsible for tasks such as optimizing resource usage and diagnosing issues based on information produced by the models. Additionally, it is a key component in planning, as it enables a dry run to be performed and anticipates the end result before implementation.

4.1.2. Addressed Challenges. NDT enables network operators to perform what-if analyses on the virtualized twin environment without having to do it directly in the production environment, which would be expensive and difficult to reverse [1]. This capability directly addresses the challenge of pre-deployment validation of configuration updates (Section 2). Previously, simulation platforms (e.g., NS-3) were used for such purposes. However, they suffer from the problem of low fidelity and a lack of real-time interaction with the real network [5].

Recent advances in ML are key enablers for NDT, enabling faster execution with lower resource overhead compared to traditional network simulation technologies [1]. Moreover, their strong predictive capabilities enable NDTs to extend beyond reactive service deployment and management, focusing on predictive maintenance (e.g., anticipating link congestion before service degradation occurs) [5].

4.2. Intent-based Networking

IBN is a software-based automation paradigm in which network operators specify high-level intents, and the network translates them into a set of configurations that can be understood by the underlying network components, thereby achieving the desired outcome without requiring the low-level execution of individual tasks [20].

Recent progress in Large Language Models has made these models particularly suitable for intent extraction and translation in next-generation networks, thanks to their ability to generate fast and consistent output in text, code, or other forms of writing [21].

4.2.1. Architecture. The authors in [22] present a reference three-layered architecture for integrating intent-based technology into existing network management systems (see Fig. 3). The Business Layer captures high-level intents based on Key Performance Indicators that align with the Service Level Agreements objectives. The Intent Layer continuously evaluates the current network state after every execution step based on the available data. The novel aspect of this approach is that all actions are

not executed blindly in a single operation; instead, the system dynamically replans after every step [22]. At the lowest layer is the Network Layer, which contains the physical infrastructure and is responsible for translating raw network data into a format comprehensible to the Intent Layer.

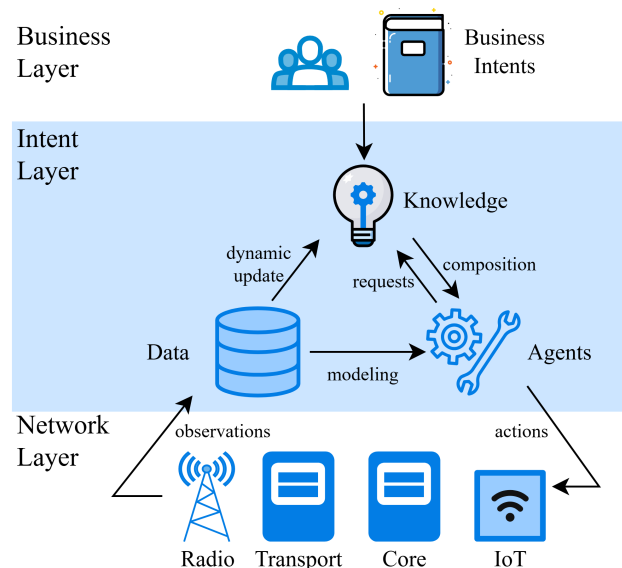


Figure 3: IBN Architecture (adapted from [22])

4.2.2. Addressed Challenges. IBN enables network automation with minimal or no human intervention, a concept commonly referred to as Zero-Touch Network and Service Management. By accepting high-level intents as input from network operators, either through voice, text, or any other form of information, the operators do not need to specify the low-level configuration details imperatively [21]. This not only reduces the likelihood of human-induced errors, but also lowers tooling proficiency required to operate large-scale networks reliably.

If an issue appears during a deployment push, network operators can immediately roll back using the previously defined versioned intent specification. Moreover, because the system is aware of the expected outcomes of a configuration, it can still achieve those outcomes even if a commonly used method is unsuccessful, by employing a different mechanism to achieve the same result [20].

Overall, NDT and IBN address additional open challenges in large-scale network management. In particular, NDT supports safer pre-deployment validation by enabling what-if analyses and planning without direct interaction with the production network [1]. IBN further reduces expertise requirements for different management systems by allowing operators to express high-level intents that are automatically translated into device-level configurations [21].

5. Hybrid Approaches

This section examines how combining the approaches discussed above helps address key network management challenges.

5.1. Combination of SDN and SBA

Legacy network devices often lack the necessary Application Programming Interfaces to be controlled by SDN systems and only support management through a CLI [23]. In situations where network operating systems do not support RESTCONF or NETCONF, the Python library Netmiko can be used to manage these older devices through SSH [24]. Alongside legacy hardware, organizations could deploy SDN-enabled devices that act as intermediaries between the legacy infrastructure and the SDN controller, allowing the SDN controller to indirectly control legacy devices [23]. This combination helps mitigate the challenge of limited automation interfaces in legacy infrastructure identified in Section 2.

5.2. Combination of SDN and IBN

IBN is often associated with SDN and relies on SDN monitoring capabilities to continuously assure that declared intents are met [25]. This combination, together with ML, can achieve a closed-loop automation cycle [25]. The authors in [26] proposed an intent-based SDN network slicing framework that continuously observes the current state of the slices to ensure correct behavior based on the objectives throughout the intent lifecycle.

Intents are also used as the means for defining north-bound interfaces of SDN controllers. These intent-driven interfaces reduce the need for deep networking expertise when interacting with SDN controllers and make SDN systems easier to integrate with other systems [27].

5.3. Combination of NDT and IBN

When IBN is used, applying the generated configurations directly to a production network carries inherent risk. By using an NDT, intents can be tested and verified before real deployment. For example, Juniper Apstra's integration with Digital Twin Lab is a powerful combination that combines both NDT and IBN to eliminate this gap between testing and production [28].

6. Related Work

This section reviews related work that addresses additional open challenges in automated network management, complementing the approaches discussed earlier.

As discussed in Section 2, collecting exhaustive telemetry in large-scale networks does not scale efficiently. Without appropriate pre-processing or edge-based summarization, the resulting data volumes may overwhelm artificial intelligence models [29]. To address this issue, ML inference can be offloaded onto the programmable data plane (PDP), as demonstrated in [30]. This work uses model compression to facilitate convolutional neural network inference on the PDP, achieving 97.3 % accuracy on an anomaly detection task while using only 22.7 % of the SRAM available on the resource-limited Intel Tofino ASIC.

The challenge of bandwidth unfairness arising from heterogeneous traffic has not been solved universally. However, within the context of industrial private 5G networks, whose adoption is increasing as factories evolve

toward Industry 5.0, the QoS-Aware Proportional Fairness scheduling algorithm proposed in [31] shows strong potential. The results demonstrate that the algorithm improves fairness while achieving better deadline adherence compared to Maximum Carrier-to-Interference and Static Priority scheduling algorithms.

As discussed in Section 2, a NSoT has been proposed as a solution to address the challenge of standardization. However, the NSoT market remains in its early stages [32]. According to EMA, an NSoT should document network intent data rather than network state data. The actual network state can instead be retrieved from network systems and used to validate alignment between the intended configuration and the operational state [32].

7. Conclusion

This paper examined the key challenges in automating network management for large-scale networks, and systematically reviewed both established and emerging approaches that aim to address them. The analysis indicates that all identified challenges, with the exception of the lack of standardized data models within the network, are actively being addressed by current research or by approaches including SDN, SBA, NDT, and IBN. In particular, the emerging approaches such as NDT and IBN introduce new capabilities for pre-deployment validation, predictive maintenance and closed-loop automation.

Evaluating the approaches individually and in combination suggests that no single approach provides a sufficiently complete solution. Instead, hybrid designs are preferable, because each approach targets a different subset of challenges.

Future work should focus in particular on advancing NDT toward more autonomous, multi-vendor deployments for complex what-if analyses. In addition, further research on standardized, vendor-agnostic data models is needed, since such models are a key enabler of a higher degree of automation in large-scale networks.

References

- [1] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong, V. Ram, S. Xiao, X. Shi, X. Cheng, A. Cabellos-Aparicio, and P. Barlet-Ros, "Network digital twin: Context, enabling technologies, and opportunities," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 22–27, 2022.
- [2] S. McGillicuddy, "Enterprise network automation: Emerging from the dark ages and reaching toward netdevops," 2024, available online at <https://efficientip.com/wp-content/uploads/2024/04/EMA7235-Network-Automation-RR-Summary-EfficientIP.pdf>; last accessed on 2026/02/22.
- [3] R. Hassan, R. Razali, S. Mohseni, O. Mohamad, and Z. Ismail, "Architecture of network management tools for heterogeneous system," 2010. [Online]. Available: <https://arxiv.org/abs/1001.1967>
- [4] U. H. Rao and S. Mohapatra, "Deploying network management solutions in enterprises," in *INC2010: 6th International Conference on Networked Computing*, 2010, pp. 1–6.
- [5] C. Zhou, H. Yang, X. Duan, D. Lopez, A. Pastor, Q. Wu, M. Boucadair, and C. Jacquenet, "Network Digital Twin: Concepts and Reference Architecture," Internet Engineering Task Force, Internet-Draft draft-irtf-nmrg-network-digital-twin-arch-11, Jul. 2025, work in Progress. [Online]. Available: <https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/11/>

- [6] T. Shanmugam and B. Malarkodi, "Analysis of campus network management challenges and solutions," in *2019 TEQIP III Sponsored International Conference on Microwave Integrated Circuits, Photonics and Wireless Networks (IMICPW)*, 2019, pp. 312–316.
- [7] Cisco Systems, Inc., "Qos modular qos command-line interface configuration guide, cisco ios xe fuji 16.8.x," 2018, available online at https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_mqc/configuration/xe-16-8/qos-mqc-xe-16-8-book/qos-scheduling.html; last accessed on 2026/02/22.
- [8] A. Pras, J. Schönwälder, M. Burgess, O. Festor, G. Martinez Perez, R. Stadler, and B. Stiller, "Key research challenges in network management," *Communications Magazine, IEEE*, vol. 45, pp. 104 – 110, 11 2007.
- [9] F. X. A. Wibowo and M. A. Gregory, "Software defined networking properties in multi-domain networks," in *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*, 2016, pp. 95–100.
- [10] R. Awati, L. Rosencrance, and J. English, "What is software-defined networking (sdn)?" 2025, available online at <https://www.techtarget.com/searchnetworking/definition/software-defined-networking-SDN>; last accessed on 2025/12/12.
- [11] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
- [12] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Computer Networks*, vol. 72, pp. 74–98, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128614002588>
- [13] Selector Software Inc., "Network automation in 2025: Technologies, challenges, and solutions," 2025, available online at <https://www.selector.ai/learning-center/network-automation-in-2025-technologies-challenges-and-solutions/>; last accessed on 2025/12/12.
- [14] O. W. J. Altaiebi and A. A. Ibrahim, "Optimization of elapsed time of automation for large-scale traditional networks and proposing new automation scripts," in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2022, pp. 1–10.
- [15] A. M. Mazin, R. A. Rahman, M. Kassim, and A. R. Mahmud, "Performance analysis on network automation interaction with network devices using python," in *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2021, pp. 360–366.
- [16] D. Mendoza, "Accelerating your network automation journey with ansible network validated content," 2023, available online at <https://www.redhat.com/en/blog/accelerating-your-network-automation-journey-ansible-network-validated-content>; last accessed on 2025/12/13.
- [17] T. Slattery, "Compare paramiko, netmiko and napalm network automation," 2024, available online at <https://www.techtarget.com/searchnetworking/tip/Network-automation-with-Python-Paramiko-Netmiko-and-NAPALM>; last accessed on 2026/01/08.
- [18] K. Rohrecker, "Maximizing the impact of network digital twins through automated network assurance," 2024, available online at <https://ipfabric.io/blog/maximizing-network-digital-twins-with-assurance/>; last accessed on 2025/12/13.
- [19] D. Chen, H. Yang, and C. Zhou, "Requirements and Design for Interfaces of Network Digital Twin," Internet Engineering Task Force, Internet-Draft draft-chen-nmrg-dtn-interface-03, Mar. 2024, work in Progress. [Online]. Available: <https://datatracker.ietf.org/draft-chen-nmrg-dtn-interface/03/>
- [20] Hewlett Packard Enterprise Development LP, "What is intent-based networking?" 2025, available online at <https://www.hpe.com/us/en/what-is/intent-based-networking.html>; last accessed on 2025/12/13.
- [21] D. M. Manias, A. Chouman, and A. Shami, "Towards intent-based network management: Large language models for intent extraction in 5g core networks," in *2024 20th International Conference on the Design of Reliable Communication Networks (DRCN)*, 2024, pp. 1–6.
- [22] E. Zeydan and Y. Turk, "Recent advances in intent-based networking: A survey," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.
- [23] S. Cooper, "Sdn and legacy system integration: Challenges & solutions," 2025, available online at <https://www.comparitech.com/net-admin/sdn-legacy-integration/>; last accessed on 2026/01/14.
- [24] C. Uneze, "How to use network automation to ease cloud integration," 2024, available online at <https://www.techtarget.com/searchnetworking/tip/How-to-use-network-automation-to-ease-cloud-integration>; last accessed on 2026/01/14.
- [25] X. Zheng and A. Leivadreas, "Network assurance in intent-based networking data centers with machine learning techniques," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 14–20.
- [26] B. Martini, M. Gharbaoui, and P. Castoldi, "Intent-based network slicing for sdn vertical services with assurance: Context, design and preliminary experiments," *Future Generation Computer Systems*, vol. 142, pp. 101–116, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167739X2200437X>
- [27] P. H. Gomes, M. Buhrgard, J. Harmatos, S. K. Mohalik, D. Roeland, and J. Niemöller, "Intent-driven closed loops for autonomous networks," *Journal of ICT Standardization*, vol. 9, no. 2, pp. 257–290, 2021.
- [28] SynerComm and Juniper Networks, Inc., "Digital twin lab and intent-based networking: Building trust through network validation," 2025, available online at <https://20503945.fs1.hubspotusercontent-na1.net/hubfs/20503945/eBooks/Digital%20Twin%20Lab%20and%20Intent-Based%20Networking%20eBook.pdf>; last accessed on 2026/01/14.
- [29] O. Aramide, "Advanced network telemetry for ai-driven network optimization in ultra ethernet and infiniband interconnects," *SAM-RIDDHI A Journal of Physical Sciences Engineering and Technology*, vol. 17, p. 2025, 01 2025.
- [30] M. Zhang, L. Cui, X. Zhang, F. P. Tso, Z. Zhen, Y. Deng, and Z. Li, "Quark: Implementing convolutional neural networks entirely on programmable data plane," in *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*, 2025, pp. 1–10.
- [31] M. Seliem, U. Roedig, C. Sreenan, and D. Pesch, "Qos-aware proportional fairness scheduling for multi-flow 5g ues: A smart factory perspective," 2025. [Online]. Available: <https://arxiv.org/abs/2508.21783>
- [32] S. McGillicuddy, "Getting started with a network source of truth," 2025, available online at <https://blog.enterprisemanagement.com/getting-started-with-a-network-source-of-truth>; last accessed on 2026/01/14.