Architecting Autonomous Network Management and Control via Intent-Driven Decoupled Network

Tong Li, Chungang Yang, Yanbo Song, Lin Cai, Ruirong Zheng, Xianglin Liu, Zeyang Ji, and Shuhan Liu

Abstract

The current coupled network management architecture and control protocols restrict the network flexibility and scalability. In order to achieve autonomous network management and control, it is necessary to decouple network functions across various layers, including application, control, data, and management layers. Although there exists emerging decouple concepts like Intent-driven Network, Software-defined Network, Kubernetes, and Network Function Virtualization, there is a lack of a generic network management architecture to achieve a full-lifecycle autonomous network management. To fill the gap, we design a generic decoupled network management architecture and interface that achieves higher flexibility and scalability by exploring and exploiting potentials of decoupled network management components. Given the decoupled network architecture, we present an intent-driven autonomous network management and control scheme, known as SAI, considering the network state, action, and potential intent. Finally, we build the intent-driven data network management prototype to prove the concept and evaluate the performance of the presented SAI management and control scheme.

Introduction

With the diversification of network services, the continuous evolution of network functions, and the high dynamism inherent in network environments, network management and control are facing new challenges, in particular with heterogeneous networks incorporating 5G cellular networks, computer networks, and the Internet of Things (IoT) [1]. The conventional network management protocols, including the simple network management protocol (SNMP) and policy-based network management (PBNM), struggle to meet the complex management requirements of the current network environment.

In order to ensure autonomous network operation and service delivery, network management and control should address various service requirements, including higher data rate, lower latency, and adaptive network configuration capabilities. The future network management architecture should pay more attention to the requirements of users in the network and realize more flexible network management and control [2]. Consequently, autonomous management needs to satisfy the following requirements.

- **Scalability:** With the continuous expansion of network scale and the increasingly complex network devices, conventional network management technologies often fail to effectively handle the intents of a large number of devices and the massive service requirements.
- **Adaptability:** High network dynamics requires the network management system to be highly adaptable. Conventional network management protocols adopt static management policies and Event-Condition-Action (ECA) rules, unable to adapt to changing network environments and service demands. It fails to guarantee timely and optimal network configurations.
- **Flexibility:** The multiplicity of network management services requires that the underlying network configuration can be flexibly configured according to different scenarios and requirements. The conventional network is designed according to the overall model, and the various functional modules of the network are tightly coupled, which makes the network very rigid and difficult to flexibly adapt to novel services and policies [3].

In summary, a more scalable, adaptive and flexible network management architecture is needed to enable autonomous network management considering the network state, configuration action, and potential service intent.

There are various decoupled concepts, such as Intent-driven Network (IDN), Software-defined Network (SDN), Kubernetes (K8S), and Network Function Virtualization (NFV), where the IDN emphasizes automating network configuration and management based on user intent or requirements [4]. The SDN enables more flexible and programmable network management and control by decoupling the network control plane from the data plane [5]. The K8S decouples application containers from the underlying system details, providing higher-level abstractions and automation capabilities, simplifying application deployment, management, and scalability [6]. The NFV aims to decouple network functions from dedicated

Tong Li, Chungang Yang (corresponding author), Yanbo Song, Ruirong Zheng, Xianglin Liu, Zeyang Ji, and Shuhan Liu are with the State Key Laboratory on Integrated Services Networks, Xidian University, Xi'an 710071, China; Lin Cai is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada.

Digital Object Identifier: 10.1109/MNET.2024.3399189
Date of Current Version: 18 November 2024
Date of Publication: 9 May 2024

This decoupled network management architecture provides higher flexibility and scalability, enabling the network to better adapt to evolving intents and challenges.

hardware and operate them in a virtualized form on hardware, providing more flexible and scalable network services [7]. In summary, although these decoupled concepts have achieved remarkable progress on specific aspects, there still lacks of a general decoupled network management architecture to realize the full-life cycle autonomous network management.

Therefore, it is critical to integrate various decoupled concepts into a generic network management architecture to achieve autonomous network management. By decoupling network functions from underlying hardware, control plane and data plane, task intent and policy execution, and configuration policy and configuration execution, network administrators can flexibly and efficiently configure the network according to intent. This decoupled network management architecture provides higher flexibility and scalability, enabling the network to better adapt to evolving intents and challenges.

The main contributions of this article are as follows:

- **We design a generic decoupled network management architecture with novel interface:** The generic decoupled network management architecture consists of the application layer, intent-enabled layer, control layer, and infrastructure layer. It decouples network functions from underlying hardware, control plane and data plane, task intent and policy execution, and configuration policy and configuration execution, respectively. This management architecture enables scalable services, achieving high management and control adaptability and flexibility.
- **We present an intent-driven autonomous network management and control scheme:** Based on the decoupled architecture, the intent-driven autonomous network management and control scheme considers network state, action, and potential intent separately, known as SAI. This SAI closed-loop scheme implements adaptive management with both upstream intent and downstream network state, further enhancing the adaptability of network management.
- **We build the intent-driven data network management prototype:** We built the prototype to prove the concept and evaluate the performance of the SAI management and control scheme. The results demonstrate that the intent-driven data network management prototype allows dynamic adjustments of the data network configuration based on real-time intents and network state, enhancing the intent success rate.

Existing Decoupled Concepts and Network Management Protocols

We adopt a top-down approach to review the existing decoupled techniques systematically and summarize the current network management protocols by incorporating intelligent advancements.

Existing Decoupled Concepts

The diversification of network requirements, the complexity of network equipment, and the continuous evolution of network functions have drawn much attention to decoupled network [4], [5], [6], [7]. At present, many decoupled concepts have emerged, such as IDN, SDN, K8S, and NFV. These decoupled concepts have brought significant progress and changes in their respective fields, opening a new vision for network development.

1) Intent-Driven Network, IDN: The concept of IDN was initially introduced by the Open Networking Foundation (ONF) in 2015, outlining the key characteristics of "intent" in network management and defining the paradigm, role, attributes, and basic implementation structure of intent interfaces [8]. The IDN is a self-intelligent network that uses decoupled control logic and closed-loop orchestration technology to automate application intents [9]. It depends on various closed-loop technologies such as intent refinement-policy generation-policy verification-state collection, which makes the network more autonomous and reliable throughout the full-life cycle.

2) Software-Defined Network, SDN: The SDN decouples conventional networks into two functional planes of the data plane and the control plane. It employs a centralized controller to manage various networks [5]. The emergence of SDN has made it easier to drive network innovation and development, enabling the network to generate policies more flexibly, efficiently, independently, and dynamically. The authors in [10] introduced a policy conflict detection mechanism to address conflicts in the SDN, aiming to identify inconsistent policies within the network. They also adopted a policy merging approach to consolidate conflicting policies, eliminating conflicts and generating consistent network policies.

3) Kubernetes, K8S: Google announced the K8S, an open-source platform for container orchestration and management. It decouples application containers from the underlying system details, simplifying application deployment and management [6]. This decoupled method simplifies application development, where users only need to request abstract resources, facilitating data center operations. The K8S can manage and orchestrate container instances of applications, providing functions such as service discovery, load balancing, auto-scaling, and storage management.

4) Network Function Virtualization, NFV: The heavy reliance of the network on its underlying hardware and the presence of various specialized hardware devices in network infrastructure pose challenges for network service providers. As a novel approach, the NFV decouples network functions from proprietary hardware devices [7]. With the NFV, network functions can be instantiated at different locations, such as data centers, network nodes, and end-user premises, according to network requirements. The NFV provides the flexibility to allocate dynamically and scale network resources, enabling on-demand provisioning and optimization of network services [11].

Although these decoupled concepts have wide applications for different network scenarios, there is independent development among them without effective integration and collaboration mechanisms. It involves challenges across diverse network scenarios. Therefore, there needs a comprehensive approach that can achieve top-down decoupled network management architecture, laying the foundation for end-to-end self-management.

Network Management Protocols

With the increasing number of network devices and the diversification of network requirements, the future of network management undoubtedly moves toward automation and intelligence. Conventional network management protocols, such as SNMP, are statically configured, requiring manual intervention and adjustments by network administrators. However, this approach has become inadequate to meet the demands of modern complex networks. The rapid growth of the network and increasingly complex application scenarios require the network to make adaptive decisions and adjustments. A novel approach to network management, IDNM, is emerging in response to these challenges.

1) Simple Network Management Protocol, SNMP: The SNMP is a widely used network standard protocol proposed by the Internet Engineering Task Force (IETF). It adopts a centralized management approach. The SNMP-based network management system follows a Client/Server (C/S) architecture. The Network Management System (NMS) retrieves device state and information by sending SNMP request messages. At the same time, the managed objects communicate with the NMS through the agent processes, providing the required data and responses [12]. However, it is difficult for the SNMP to support complex network management tasks in the face of increasingly large-scale and complex network environments.

2) Policy-Based Network Management, PBNM: The PBNM represents an alternative to network management from focusing on "how to implement" to "what to achieve". This approach allows administrators to concentrate more on the goals of network management. By applying a set of policies across the entire network, the PBNM decouples network management into two distinct processes: policy definition and policy enforcement [13]. However, the PBNM requires finegrained policy design and implementation. Incorrect policies can result in misconfigured network resources or misguided restrictions. Moreover, the PBNM may require higher skills and knowledge to define and enforce policies effectively.

3) Intent-Driven Network Management, IDNM: The IDNM is a network management approach based on user intent to manage networks more flexibly and intelligently. Network administrators or users express their intent through a user-friendly interface without needing to focus on underlying network details [4]. The method refines and understands the intent and automatically executes the corresponding policies to fulfill their requirements. This approach decouples network intent from policy enforcement, reducing the need for manual operations. The IDNM method achieves full automation of operations. It is with great potential in lowering network management overhead and automating recovery operations.

However, limited literature worked on the relationships among intent, policy, and configuration. There is a lack of an integrated scheme to consolidate these elements effectively. The relationship of intent, policy, and configuration can elevate network management from a configuration-based level to an intent-driven level, thus enhancing the intelligence level of network management protocols.

Decoupled Network Management Architecture and Interface Design

With the continuous growth of network applications and the increasing complexity, conventional network management architectures can no longer meet the demands of flexibility and adaptability. Decoupled networks have emerged as a new trend. As illustrated in Fig. 1, we design a decoupled network management architecture with novel interface to provide scalable applications and services while ensuring network management and control flexibility and adaptability. The decoupling between different layers enables a more modular network management architecture, allowing for flexible combinations and customization based on specific requirements.

Decoupled Network Management Architecture

As illustrated in Fig. 1, the management architecture consists of four layers, namely the application

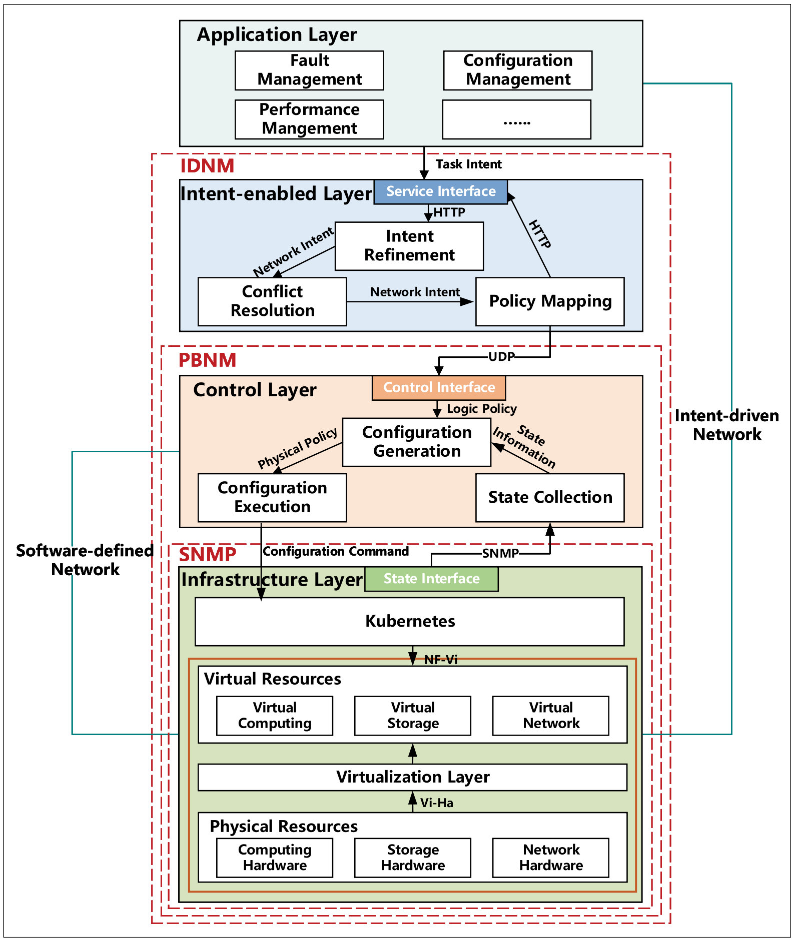

FIGURE 1. Decoupled network management architecture with novel interface.

layer, intent-enabled layer, control layer, and infrastructure layer. Moreover, standard interfaces are defined between and within layers, which can achieve interoperability and platform independence between different modules and reduce the coupling between modules.

- **Application Layer:** It consists of various management objects within the network, aiming to achieve effective management and control of the network. The ISO/IEC 7498-4 document encompasses different management aspects, such as fault management, configuration management, performance management, etc. The application layer is responsible for defining and managing specific task intents. The sources of task intents can be categorized into two types: intents input by administrators and potential intents feedback from the network state.
- **Intent-Enabled Layer:** It plays a crucial role in a decoupled network management architecture, including technologies such as intent refinement, conflict resolution, and policy mapping. Task intents from the application layer are received based on the HyperText Transfer Protocol (HTTP). These intents are processed using intent refinement techniques, such as Named Entity Recognition (NER), to obtain standardized intent tuples. Multiple input intents are merged and processed through conflict resolution techniques, resulting in conflict-free network intents. These network intents are further transformed into corresponding logic policies using policy mapping techniques, such as finite state machines (FSM). These logic policies are distributed to the control layer via the User Datagram Protocol (UDP), enabling automatic response to the task intents. This approach reduces the complexity of network management, enhances automation, and allows the network to be more flexible, customizable, and intelligent in meeting various intents.
- **Control Layer:** The control layer primarily encompasses configuration generation, configuration execution, and state collection. It is decoupled into configuration policy and configuration command execution to achieve network control and optimization. Firstly, the control layer receives logic policies from the intent-enabled layer through the UDP and generates corresponding physical policies through configuration generation module. Then, the configuration execution module implements policy deployment by running the protocol source code. Additionally, the control layer collects network state information using the SNMP to obtain the state and availability of network resources. The collected network state information is reported to the configuration generation module for policy adjustment and optimization. The network can dynamically configure and optimize itself based on intents and changes in the network state, thereby realizing advanced network management and control.
- **Infrastructure Layer:** It provides a flexible, scalable, and manageable infrastructure that supports the deployment and operation of the decoupled network management. The NFV technology transforms physical resources into the Virtual Network Functions (VNFs) on commodity hardware. The K8S is employed to manage and orchestrate container instances of the VNFs, offering functions such as service discovery and load balancing. This approach enables automated management and flexible deployment, providing higher abstraction and control capabilities for a decoupled network management architecture.

Therefore, the proposed decoupled network management architecture decouples network functions from underlying hardware, control plane and data plane, task intent and policy execution, and configuration policy and configuration execution. This decoupled network management architecture enables the independent evolution of network functions, facilitating rapid adaptation to evolving intents and achieving advanced network management and control at a higher level. In the section "Intent-Driven Autonomous Network Management and Control Sai Scheme," we will describe the function modules included in each layer and explain their implementation methods.

Decoupled Network Management Architecture Interface

The internal components of the decoupled network management architecture, along with their internal interface and workflows, are illustrated in Fig. 2. The architecture includes a closed management loop that continuously monitors the quality of upper-layer services and the underlying network state, generating new configuration decisions in real time. In the closed-loop management process, intent serves as a high-level abstract policy encompassing task intent, network intent, logic policy, and physical policy. Task intent represents the advanced management intents input by network controllers in the form of text or voice. Network intent refers to the network requirements corresponding to task intent obtained through intent refinement and conflict resolution. Logic policy refers to the adjustment actions that nodes need to take. Physical policy involves specific network configuration parameters that align with the task intent for network demands.

Firstly, the front-end interface of the application receives the task intent, obtains the network intent through intent refinement and conflict resolution, and feeds the network intent back to the front-end interface. Secondly, the logic policy is obtained through policy mapping. Thirdly, the logic policy is transformed into corresponding physical policy through configuration generation. Finally, executing the protocol source code deploys the physical policy through configuration commands, issuing it to the corresponding infrastructure. In the process, network state information is obtained through technologies such as network telemetry to assist network policy generation and optimization. This real-time monitoring and feedback mechanism ensures that the network management system can quickly and accurately respond to changing network requirements and environments.

In order to promote the intelligent level of network management, we design the service

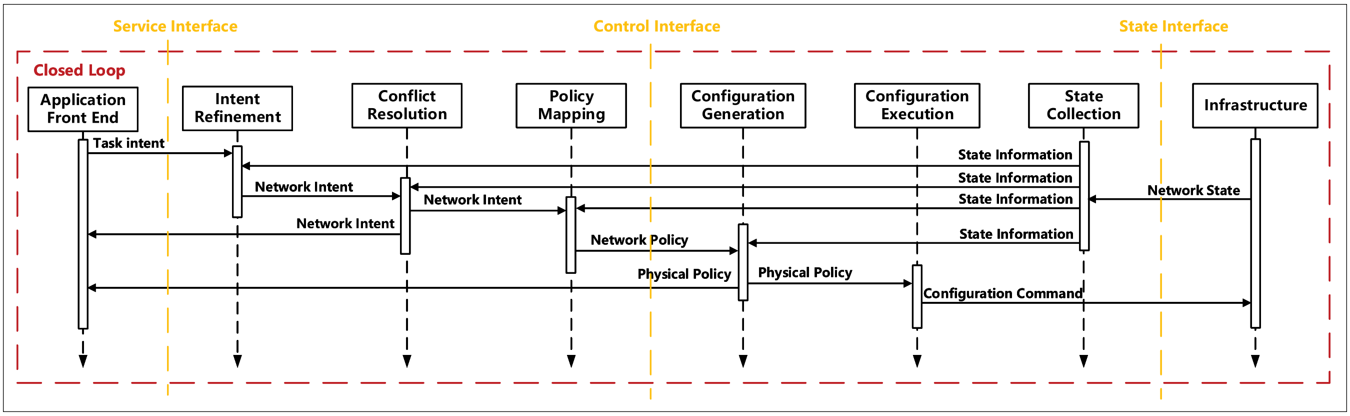

FIGURE 2. Decoupled network management architecture internal interface and workflows.

interface, control interface, and state interface in the decoupled network management architecture. Through these interfaces, the IDNM is implemented based on the SNMP and the PBNM, thereby promoting the intelligence level of network management.

- **Service Interface:** First, the service interface is designed, which provides essential support for realizing the task intent input and transformation of the IDNM. The user enters the task intent through this interface and uses the HTTP to send the intent to the intent-enabled layer. The task intent is converted into a network intent through intent refinement and conflict resolution, and the logic policy is obtained through policy mapping. It introduces a higher level of intelligent decision-making and policy optimization for network management.
- **Control Interface:** Second, the control interface is designed to receive logic policy through the UDP and utilize configuration generation and configuration execution processes to create and execute physical policies. In the decoupled network management architecture, the physical policy of network management can be defined, deployed, and executed according to the logic policy to perform network management according to the intent flexibly.
- **State Interface:** Last, the state interface is designed to collect network state information. This interface allows for the retrieval of network information, device configuration, and underlying state, facilitating the monitoring and management of network state.

By designing service interface, control interface, and state interface within the decoupled network management architecture, these interfaces offer network administrators a unified approach to access, configure, and monitor network devices and services. They ensure efficient communication and data exchange between modules, ultimately elevating network management intelligence.

Intent-Driven Autonomous Network Management and Control SAI Scheme

As described in the section "Decoupled Network Management Architecture and Interface Design," the decoupled network management architecture integrates a comprehensive list of decoupled concepts and technologies. It elevates the level of management intelligence, combines SNMP-PBNM-IDNM, an allows network management to focus on satisfying task intents without concerning underlying network configurations and operations. This realization leads to an intelligent level of network management and control. Furthermore, the management process within this architecture constitutes a closed-loop self-adaptive management.

> The closed-loop process involves the key technologies including intent refinement, conflict resolution, policy mapping, configuration generation, configuration execution, and state collection.

Based on the architecture, we propose an intent-driven autonomous network management and control scheme, known as SAI, as shown in Fig. 3. It effectively incorporated network state, action, and potential intent. This closed-loop management and control allows adaptive handling both the upper-level task intents and lower-level network states. The closed-loop process involves the key technologies including intent refinement, conflict resolution, policy mapping, configuration generation, configuration execution, and state collection. Firstly, task intents are input through the frontend interface, and intent refinement is utilized to understand the requirements of the task intents. Secondly, conflicts arising from multiple intents are resolved. Thirdly, combining real-time network state information and expert knowledge, network intents are mapped to logic policies. Fourthly, corresponding physical policies are generated to implement task intents through pre-designed policy generation functions. Fifthly, configuration execution is achieved by running protocol source code to deploy physical policies. Sixthly, the state collection module dynamically perceives network state information, providing state information support for policy mapping and configuration generation, and enabling dynamic adjustments to policies based on state information. The critical technologies with the SAI closed-loop adaptive management are introduced as follows:

- **Intent Refinement:** Intent refinement is the first step in realizing decoupled network management. It involves converting

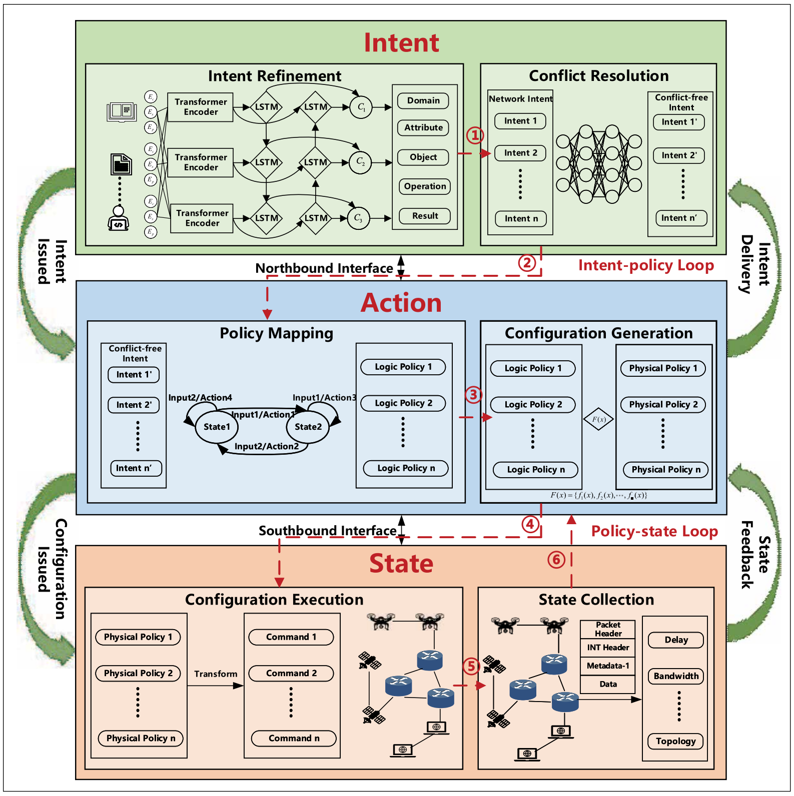

FIGURE 3. Intent-driven autonomous SAI network management and control scheme.

task-level intents input into the network in various forms, such as voice or text, into intelligible intents for the network. Intent refinement leverages the Transformer Encoder model and the multi-layered BiLSTM network, to extract critical entity information from unstructured intent expressions. By applying morphological rules for intent tuples, it parses task intents and extracts accurate semantic information, representing them in a standardized intent format to better meet user requirements [14].

- **Conflict Resolution:** Conflict resolution is a critical process for ensuring the smooth operation of the decoupled network management. Ensuring consensus among these intents is crucial in complex network scenarios where multiple task intents are input. First, priorities or weights need to be assigned to different intents. Then, when multiple intents are input, the mathematical model created is solved by neural network to achieve conflict resolution between the intents to ensure their execution. Conflict resolution allows to handle intent degradation or error feedback intelligently, ultimately providing an enhanced user experience and application performance [15].
- **Policy Mapping:** Policy mapping is a crucial component of the decoupled network management, facilitating the mapping of "intent-policy" within the network. In various network scenarios, pre-defined ECA rules and policies are determined based on domain knowledge, including guidelines for bandwidth allocation, traffic control, service prioritization, and more. These ECA rules and policies guide the actions under different intents. Subsequently, through the design of FSM, the management system defines various states it can be in and the logic policies for each state. It enables the mapping of network intents to the corresponding logic policies, thereby achieving precise mapping between the intent requirements of the upstream and the network state information of the downstream.
- **Configuration Generation:** Configuration generation is a vital step in ensuring intelligent management within the decoupled network management. Utilizing Artificial Intelligence (AI) algorithms, in conjunction with current network state information, intelligent configuration policy generation functions are designed for different parameters, and network policies are transformed into executable configuration policies. This enables the network to adjust and optimize in response to real-time states. For example, this may involve crafting optimal routing policies based on the latest topology information and traffic analysis or automatically adjusting bandwidth and configuration parameters based on requirements and network performance.
- **Configuration Execution:** Configuration execution is the execution phase within the decoupled network automated management process. The configuration policies are transformed into specific commands applicable to network devices and systems by executing protocol source code. The network devices include multi-domain nodes such as satellites, UAVs, edge nodes and end users. These commands are then applied to involve modifications or optimizations of network parameters and configurations.
- **State Collection:** State collection is the foundational component for ensuring the realization of task intents within the decoupled network management. Real-time monitoring and analysis of network state information are conducted through various means, such as network telemetry for collecting performance metrics like bandwidth utilization from network devices and sensor-based monitoring of physical equipment and environmental states. By establishing an effective feedback mechanism, the system can gain a more comprehensive understanding of the network state, enabling it to make informed decisions and promptly correct errors, thereby ensuring the accurate execution of intents.

Therefore, based on the decoupled network management architecture, the SAI adaptive management and control scheme can adjust to changing intents and network states adaptively. This closed-loop design not only facilitates rapid adaptation and adjustment of the network when facing decision errors, ensuring the stability and resilience of network operations, but also provides a higher level of intelligence and automation capability for network management.

Prove-of-Concept and Evaluation of SAI Management and Control Scheme

In this section, we have built the intent-driven data network management prototype to prove the concept and evaluate the performance of the SAI management and control scheme. The prototype can respond to complex task intents and network states autonomously, improving the management efficiency and performance of the data network.

Evaluation Environment

To evaluate the effectiveness of our proposed scheme, we have designed an evaluation environment as depicted in Fig. 4. The platform is composed of the following components:

- The intent module presented in the section "Intent-Driven Autonomous Network Management and Control Sai Scheme" is deployed in a personal computer using ubuntu 16.04. The personal computer is characterized by an Intel i9-13900KF and 64GB of memory.
- The action module presented in the section "Intent-Driven Autonomous Network Management and Control Sai Scheme" is deployed in a personal computer using window 10. The personal computer is characterized by an Intel i5-1660v4 and 32GB of memory.
- The state module presented in the section "Intent-Driven Autonomous Network Management and Control Sai Scheme" is deployed in a personal computer using window 10 and USRP B210. The personal computer is characterized by an Intel i510210U and 16GB of memory.
- The network infrastructure uses miniaturized terminals to complete parameter configuration and wireless message forwarding.

FIGURE 4. Intent-driven data network management prototype.

Evaluation Results

In the intent-driven data network management prototype, the design of the intent module is accomplished using tools including Postman (interface debugging and testing tools), Neo4j (a graph database), and Py2neo (the toolkit of Python). This module parses task intent and transforms it into network intent for managing and configuring the data network. Furthermore, the action module is designed based on the Qt software, featuring a user-friendly graphical interface and integrating configuration decision algorithms. This module enables agile decision-making in response to user requirements. Additionally, we utilize technologies such as USRP hardware and Matlab simulation to collect spectrum state information, which is essential for managing data networks and making decisions.

Finally, the decision parameters, such as time slots, frequency, power, and rate, are distributed to the underlying infrastructure platform by issuing configuration commands, facilitating the adaptive configuration of network parameters.

During the experimental process, we defined two distinct task intents, namely "Nodes 2, 7, 17, 42, 57, 75 perform a patrol task to target area 37." and "Nodes 2, 7, 17, 42, 57, 75 perform a rescue task for target area 37." These two task intents are deployed under the same and different spectrum states respectively. By using intent-driven data network management prototype, node configuration parameters can be adaptive to different spectrum states, adjusting the data network to realize task intents.

Fig. 5 shows the configuration results generated by the intent-driven data network management prototype for the two task intents under the same and different spectrum states. Fig. 5(a) and (b) depict the configuration results obtained under different intents, while Fig. 5(b) and (c) demonstrate the prototype's ability to generate distinct configuration policies in various spectrum states promptly. These results vividly confirm that the SAI scheme can achieve adaptability to different intents and network states, and the flexibility to generate configuration policies in real-time dynamic intents.

As shown in Fig. 6, we compare the intent success rate of the SAI scheme and the fixed

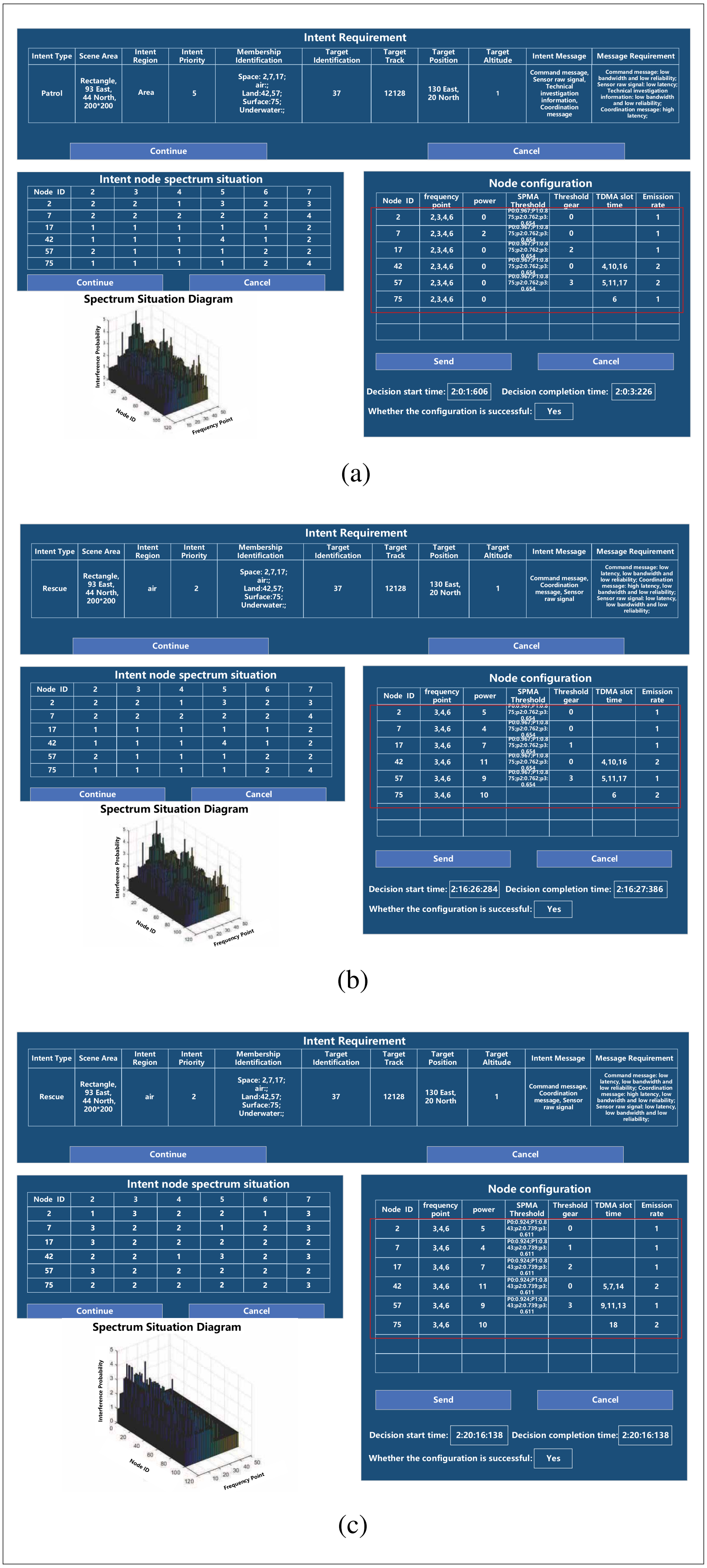

FIGURE 5. Decision results for different intents and different spectrum states. a) The decision result with spectrum state 1 and patrol intent. b) The decision result with spectrum state 1 and rescue intent. c) The decision result with spectrum state 2 and rescue intent.

allocation scheme. The evaluation results will be analyzed in terms of scalability and adaptability.

- Scalability: Fig. 6 illustrates that as the simulation time increases, the network resources gradually become insufficient, making it impossible to process the intent effectively, and the number of services that the network needs to implement will also increase, causing the intent success rate to show a downward trend. However, comparing the SAI scheme and fixed allocation scheme, it can be seen that when the number of intents increases to 25, the intent success rate of the SAI scheme still maintains more than 80%, while that of the fixed allocation scheme is close to 60%. When the number of intents increases to 45, the SAI scheme continues to maintain an intent success rate above 70%, whereas that of the fixed allocation scheme drops to 55%. This discrepancy is attributed to the SAI scheme, which solves the problem of multi-intent conflicts, can fulfill multi-intent requirements to a great extent, and achieves scalability of network management.
- Adaptability: Fig. 6 demonstrates that the intent success rate of the SAI scheme decreases slower over time compared to that of the fixed allocation scheme. When there are 25 intents, and the simulation time reaches 100s, both schemes achieve a 100% intent success rate. However, at 250s, the intent success rate for the SAI scheme remains at 80%, in contrast to the 70% success rate for the fixed allocation scheme. The SAI scheme exhibits adaptive adjustment characteristics, and can dynamically adapt policies based on intent and network state. This dual decision-making scheme enables the prototype to respond to changes in intent more intelligently, thereby enhancing intent success rate.

Therefore, the proposed SAI management and control scheme effectively improves the scalability and adaptability of network management. It can automatically adjust and optimize network configurations based on evolving intents and network states, delivering high-quality services and performance. This management scheme enhances the flexibility of network management.

Challenges and Future Work

This article proposes a generic decoupled network management architecture with novel interface, presenting the SAI scheme, and building the intent-driven data network management prototype. Despite much notable progress, there are some obvious challenges to be tackled. Firstly, there is a lack of a generic representation model for characterizing task intent, network intent, logic policy, and physical policy within the decoupled management architecture. Secondly, it is imperative to extend the application of the SAI scheme to a broader range of network scenarios. In order to address these challenges, it is crucial to establish a generic representation model. This model will provide a clearer depiction of the SAI mechanism, facilitating intelligent transformation and adaptive implementation of intents in diverse network scenarios.

Conclusion

To achieve advanced network management and control, we proposed a generic decoupled network management architecture with novel interface that effectively decouples network functions from underlying hardware, control plane and data plane, task intent and policy execution, and configuration policy and configuration execution, enabling high-level network management and control. Second, based on the decoupled network management architecture, we presented an intent-driven autonomous network management and control scheme, known as SAI, considering network state, action, and potential intent. Finally, we built the intent-driven data network management prototype to prove the concept and evaluate the performance of the SAI management and control scheme. The evaluation results demonstrated that the proposed management scheme could effectively manage and regulate the network by achieving more intents and dynamically adjusting the configuration policy in response to real-time intents and network states.

Acknowledgment

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1807700, in part by the Fundamental Research Funds for the Central Universities, and in part by the Innovation Fund of Xidian University.

References

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.

[2] X. Shen et al., "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 1st Quart., 2021.

[3] F. Salahdine et al., "A survey on sleep mode techniques for ultra-dense networks in 5G and beyond," *Comput. Netw.*, vol. 201, Dec. 2021, Art. no. 108567.

[4] C. Yang et al., "SMART intent-driven network management," *IEEE Commun. Mag.*, vol. 61, no. 1, pp. 106–112, Jan. 2023.

[5] A. Liatifis et al., "Advancing SDN from OpenFlow to P4: A survey," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–37, Sep. 2023.

[6] C. Carrión, "Kubernetes scheduling: Taxonomy, ongoing issues and challenges," *ACM Comput. Surv.*, vol. 55, no. 7, pp. 1–37, Dec. 2022.

[7] H. Hawilo et al., "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov. 2014.

[8] D. Lenrow. (2015). *Intent: Don't Tell Me What to Do! (Tell Me What You Want)*. [Online]. Available: https://www.sdxcentral.com/articles/contributed/network-intent-summit-perspective-david-lenrow/2015/02/s

[9] L. Pang et al., "A survey on intent-driven networks," *IEEE Access*, vol. 8, pp. 22862–22873, Jan. 2020.

[10] Y. Zhou et al., "Raze policy conflicts in SDN," *J. Netw. Comput. Appl.*, vol. 199, Mar. 2022, Art. no. 103307.

[11] P. Soto et al., "Network intelligence for NFV scaling in closed-loop architectures," *IEEE Commun. Mag.*, vol. 61, no. 6, pp. 66–72, Jun. 2023.

[12] H. Wang, "Improvement and implementation of wireless network topology system based on SNMP protocol for router equipment," *Comput. Commun.*, vol. 151, pp. 10–18, Feb. 2020.

[13] F. Deng et al., "An efficient policy evaluation engine for XACML policy management," *Inf. Sci.*, vol. 547, pp. 1105–1121, Feb. 2021.

[14] Y. Ouyang et al., "A brief survey and implementation on refinement for intent-driven networking," *IEEE Netw.*, vol. 35, no. 6, pp. 75–83, Nov. 2021.

[15] Y. Song et al., "Full-life cycle intent-driven network verification: Challenges and approaches," *IEEE Netw.*, vol. 37, no. 5, pp. 145–153, Sep. 2023.

It can automatically adjust and optimize network configurations based on evolving intents and network states, delivering high-quality services and performance.

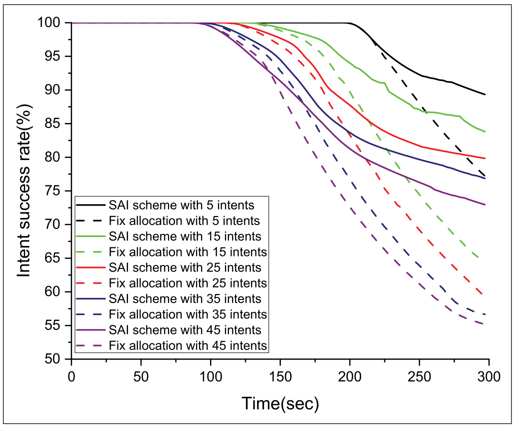

FIGURE 6. Comparison of intent success rate.

Biographies

Tong Li (tongli1018@gmail.com) received the bachelor's degree from Changan University. She is currently pursuing the Ph.D. degree in information and communication engineering. Her research interests include intent-driven network management, wireless ad hoc network, and communication protocol.

Chungang Yang (guideyang2050@163.com) is currently a Full Professor with Xidian University, where he leads there search team of "GUIDE, Game, and Utility, artificial Intelligent Design for Emerging communications." His research interests include artificial intelligent 6G wireless mobile networks, intent-driven networks (IDN), space-terrestrial networks (STN), and game theory for emerging communication networks.

Yanbo Song (songyanbo94@163.com) received the Ph.D. degree in information and communication engineering from Xidian University. His research interests include intent-driven network security and network management.

Lin Cai (Fellow, IEEE) (cai@ece.uvic.ca) is currently a Professor with the Department of ECE,University of Victoria. She is an NSERC E.W.R. Steacie Memorial Fellow, an Engineering Institute of Canada (EIC) Fellow, and a College Member of the Royal Society of Canada. Her research interests include network protocol and architecture design supporting multimedia traffic and the Internet-of-Things.

Ruirong Zheng (zhengruirong1221@163.com) received the bachelor's degree from the North University of China. He is currently pursuing the master's degree in information and communication engineering. His research interests include intent-driven network routing protocols and wireless ad hoc network communication protocols.

Xianglin Liu (1x108080627@163.com) received the bachelor's degree from the Qingdao University of Science and Technology. He is currently pursuing the master's degree in information and communication engineering. His research interests include the intent refinement and intent negotiation.

Zeyang Ji (jzy15835274090@163.com) received the bachelor's degree from Taiyuan Normal University. He is currently pursuing the master's degree in electronic information. His research interests include the topology control in wireless ad hoc network.

Shuhan Liu (liushuhan200005@163.com) received the bachelor's degree from the Shanghai University of Electric Power. She is currently pursuing the master's degree in information and communication engineering. Her research interests include intent-driven network intelligent decision-making and multi-dimensional resource situational awareness.