Intent-Driven Control System for Heterogeneous Agent-Oriented Networking (HaoNet)

Wang Bowen[1], Lu Lu[2], Li Huimin[1], Yang Chungang[1]
(1. Xidian University, Xi'an 710071, China;
2. China Mobile Research Institute, Beijing 100053, China)

DOI: 10.12142/ZTECOM.202602007

Manuscript received: 2026-03-11

Abstract: Empowered by advances in large language models, the growing integration of autonomous agents into industrial and daily-life sectors is turning them into new networking entities. Such agent-oriented networking features high interaction frequencies and emergent task-driven structures, necessitating strong network policy consistency and reliability within dynamic environments. To address these challenges, we propose a network control system that integrates Intent-Driven Network (IDN) into Heterogeneous Agent-Oriented Networking (HaoNet). IDN focuses on high-level task intents and provides flexible reconfiguration and adaptive optimization, thereby enhancing the effectiveness of agent-oriented networking. In this paper, we first summarize three key features of HaoNet: task-driven operation, distributed collaboration, and closed-loop intelligence. Furthermore, we propose a comprehensive system architecture, which includes the application layer, the intent layer, and the infrastructure layer, and investigate the associated key technologies. Finally, typical application scenarios are presented to demonstrate the practical value of the proposed system in enabling robust agent-oriented networking control.

Keywords: AI agent; agent-oriented networking; intent-driven network; network control

Citation (Format 1): Wang B W, Lu L, Li H M, et al. Intent-driven control system for heterogeneous agent-oriented networking (HaoNet) [J]. ZTE Communications, 2026, 24(2): 52 – 63. DOI: 10.12142/ZTECOM.202602007
Citation (Format 2): B. W. Wang, L. Lu, H. M. Li, et al., "Intent-driven control system for heterogeneous agent-oriented networking (HaoNet)," *ZTE Communications*, vol. 24, no. 2, pp. 52 – 63, Jun. 2026. doi: 10.12142/ZTECOM.202602007.

1 Introduction

The rapid advancement of large language models (LLMs) has fundamentally expanded the application landscape of artificial intelligence (AI), with the AI agent emerging as a dominant paradigm for translating LLM capabilities into actionable systems. AI agents are intelligent entities capable of environment perception, autonomous decision-making, and action execution, which leverage LLMs to accomplish tasks independently or collaboratively in complex environments while continuously optimizing their behavior[1]. In recent years, a diverse spectrum of agent-based applications has been developed in both industrial and consumer domains, exemplified by Manus[2] and Coze[3]. Agents can be broadly categorized into physical agents, which exist in physical forms such as robots and interact directly with the physical world, and digital agents, which operate within digital environments to provide more sophisticated decision support. To accomplish increasingly sophisticated tasks, heterogeneous agents across different domains must frequently collaborate, necessitating seamless inter-agent communication and robust networking. As agents proliferate across diverse environments, future networks will increasingly interconnect these agents rather than conventional end hosts, marking a paradigm shift in networked object composition and imposing new demands on network architecture, control mechanisms, and resource management.

To accommodate the evolving connectivity requirements of agents, Heterogeneous Agent-Oriented Networking (HaoNet) has significant research value. HaoNet aims to provide a globally interconnected and controllable network with on-demand enabling for information exchange and task collaboration among agents of various forms, capabilities, and users[4]. Unlike conventional networking paradigms, HaoNet is designed to support reliable agent access and interaction frequencies that far exceed those of traditional networks. Driven by the intrinsic autonomy of agents, HaoNet shifts from reactive connectivity to proactive, task-oriented networking, where links are established autonomously to fulfill specific task goals. Furthermore, HaoNet enables real-time, flexible network reconfiguration to navigate emergent tasks and dynamic environments.

Despite its potential, realizing the full capabilities of HaoNet presents significant technical challenges. First, addressing emergent and complex tasks requires deep semantic parsing of task-specific objectives and the subsequent mapping of these mission-level goals onto networking targets. Second, ensuring policy consistency and reliability across diverse agents is critical, as any misalignment in intent interpretation could lead to network instability in dynamic environments. Third, the proactive nature of agents leads to highly frequent and bursty interactions, requiring continuous network policy adaptation. Finally, the disparity in functional capabilities and computational resources among agents demands a sophisticated orchestration mechanism capable of balancing global intent fulfillment with resource efficiency.

The Intent-Driven Network (IDN) is a novel paradigm centered on user intent, utilizing AI to accurately identify and rapidly respond to high-level user requirements[5]. By integrating core technologies such as intent refinement, policy generation, and intent assurance, IDN achieves a closed-loop mapping from user intent to network configuration, enabling the network to dynamically adapt and optimize itself without manual intervention. This inherent flexibility and self-optimization capability offer a promising pathway to overcome the bottlenecks of HaoNet. Specifically, the intent-centered paradigm provides a robust framework for parsing the semantic objectives of emergent tasks, effectively bridging the gap between mission-level goals and networking targets. Moreover, the adaptive architecture of IDN, supported by advanced techniques such as multimodal processing and semantic communication, is uniquely suited to maintain policy consistency and handle the highly frequent interactions characteristic of agents.

This paper aims to develop a network control system for HaoNet that operates under an intent-driven paradigm, fundamentally bridging the gap between high-level task objectives and autonomous network execution. By integrating IDN technologies, the proposed system ensures policy consistency and reliability while enhancing networking effectiveness and task efficiency for emergent tasks in dynamic environments. The main contributions of this paper are summarized as follows:

1) Characterization of HaoNet: We summarize three key features of HaoNet, namely task-driven operation, distributed collaboration, and closed-loop intelligence, establishing a conceptual foundation for agent-oriented networking.

2) System architecture and key technologies: We propose a comprehensive system architecture that integrates IDN, and investigates the associated key technologies required to enable autonomous network control.

3) Illustrative application scenarios: We provide typical application scenarios to demonstrate the practical value and feasibility of the proposed system in enabling robust agent-oriented networking.

2 Related Work

The rapid proliferation of agents has spurred diverse paradigms of agent-oriented networking. Gupta et al. proposed AgNet[6], a novel architecture that draws parallels with the World Wide Web by introducing specialized components such as the Agent Name Server and the Agent Text Transfer Protocol to facilitate discovery and interoperability. From a broader ecosystem perspective, the Internet of Agents[7] has been envisioned as a foundational framework for the seamless interconnection of heterogeneous agents at scale, focusing on capability notification and adaptive communication, and introducing emerging agent communication protocols including Model Context Protocol (MCP)[8] and Agent to Agent Protocol (A2A)[9]. To further integrate intelligence into the network fabric, Xiao et al. proposed AgentNet[10], a 6G-oriented framework that connects task-oriented agents by deploying dedicated agents across the physical, network, and application layers to enable cross-layer collaborative intelligence. Despite these advancements, systematic analysis[11] reveals that siloed agent designs lead to significant architectural fragmentation, hindering cross-domain interoperability and manageable scaling. Compounding this issue, existing frameworks[6–7, 10] primarily focus on defining static structural components while paying insufficient attention to specific operational mechanisms, particularly under dynamic task execution.

IDN has emerged as a transformative paradigm that automates network management by translating high-level user objectives into actionable configurations[12]. Recent research has increasingly focused on integrating Agentic AI to overcome the rigidity of traditional rule-based IDN systems. For instance, Jiang et al.[13] proposed a hierarchical multi-agent framework for 6G, where LLM-based agents employ reasoning-action cycles to decompose natural language intents into technically feasible network slice configurations. Similarly, Lee et al.[14] introduced a Generative-AI-empowered architecture that utilizes a chief agent to transform intents into dependency-aware execution graphs for autonomous telecom service reconfiguration. In specialized domains such as industrial automation, Romero et al.[15] integrated Agentic AI into the intent-based paradigm, utilizing specialized sub-agents to translate high-level human goals into autonomous actions aligned with Industry 5.0 principles. While these studies predominantly leverage Agentic AI as a tool to enhance the autonomy of IDN, the maturation of IDN capabilities provides a natural foundation for extending its robust intent-centered and self-optimization mechanisms to support agent-oriented networking. Applying these enhanced capabilities enables a strategic shift from using agents to manage the network to leveraging the network's inherent intelligence to proactively sustain the task-driven, bursty, and heterogeneous interactions of the agents themselves.

The integration of IDN into agent-oriented networking remains in its early stages. A preliminary attempt is GoAgentNet[16], which draws on the intent-driven paradigm to enable multi-agent collaboration by orchestrating their perception, communication, and computation. While it explores goal-oriented fulfillment, GoAgentNet lacks specific mechanisms to ensure dynamic intent-agent mapping and scalable collaborative orchestration of heterogeneous agents. To bridge these gaps, our research proposes an intent-driven control framework for HaoNet, which incorporates diverse mechanisms such as LLM fine-tuning and adaptive collaboration patterns to ensure high network policy consistency and reliability within dynamic environments.

3 Networking Scenario and Design of HaoNet

This section presents the core foundations of HaoNet, ranging from key features to architectural implementation. We introduce a general networking scenario to establish context, followed by a summary of key features that serve as a conceptual foundation. Finally, we present the system architecture for HaoNet.

3.1 Networking Scenario

To provide a foundational understanding of HaoNet, we introduce a general networking scenario centered on collaborative infrastructure maintenance tasks, which can be analogized to other networking scenarios, as shown in Fig. 1. This scenario incorporates a heterogeneous agent society comprising unmanned aerial vehicle (UAV) agents, ground robotic agents, and virtual assistant agents, all operating collaboratively to ensure the operational integrity and safety of 5G/6G network-based urban or unmanned ad-hoc network-based industrial infrastructures.

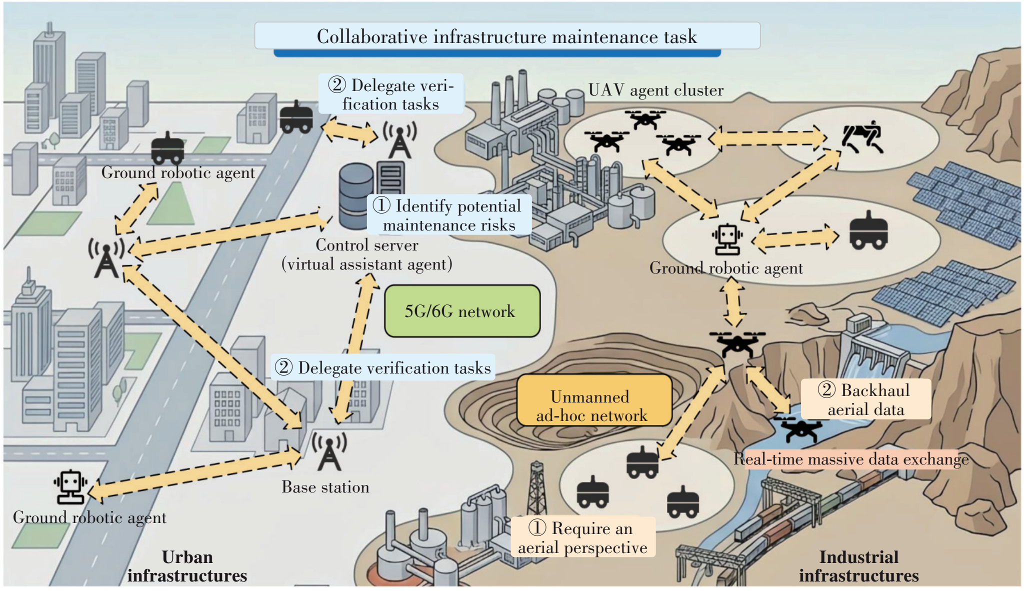
The collaborative infrastructure maintenance process typically initiates with high-level operational objectives, including comprehensive structural health inspections. These objectives are systematically decomposed into granular subtasks and allocated to multiple agents, which then form task subnets for collaborative execution based on their capabilities and real-time spatiotemporal status. During this collaborative process, their localized interactions give rise to specific networking requirements, as the underlying communication infrastructure must adapt its resources to support the synchronization and data-intensive demands of agents.

Beyond preplanned task allocation, the network must accommodate spontaneous, peer-to-peer interactions within this dynamic environment. For instance, a ground robotic agent requiring an aerial perspective for a localized assessment can autonomously initiate a collaborative request to a nearby UAV, triggering a real-time massive data exchange. These emergent communication requirements necessitate on-demand reconfiguration, where logical links and physical resources are rapidly reconfigured into transient, task-aligned subnets.

It is important to note that 5G/6G networks and unmanned ad-hoc networks exhibit fundamentally different characteristics. Specifically, 5G/6G networks rely on well-established infrastructure, providing relatively stable topology, reliable connectivity, and support for centralized control. In contrast, unmanned ad-hoc networks operate without fixed infrastructure, featuring highly dynamic topology, intermittent connectivity, and a stronger reliance on distributed and self-organizing mechanisms. Although the key technical requirements differ across scenarios in terms of control strategies, resource management, and communication patterns, the proposed architecture and key technologies are designed to address common requirements while enabling flexible adaptation to scenario-specific characteristics.

3.2 Key Features

By distilling the fundamental requirements of HaoNet, we summarize three key features: task-driven operation, distributed collaboration, and closed-loop intelligence.



UAV: unmanned aerial vehicle

Figure 1. A general networking scenario of collaborative infrastructure maintenance in HaoNet

3.2.1 Task-Driven Operation

In HaoNet, task intent originating from the user or application serves as the core driving factor, guiding the agent network to enable adaptive task orchestration. Upon receiving a task intent, HaoNet performs intent refinement and decomposition to identify a group of agents that satisfy the task requirements based on agent capabilities. The resulting subtasks are then assigned to the selected agents to form a task-oriented agent subnet.

From the perspective of adaptive coordination in various environments, HaoNet encompasses three task orchestration modes: centralized, distributed, and semi-distributed, as shown in Fig. 2. In the centralized mode, a central agent maintains a comprehensive global view of the network, covering agent capabilities, edge computing resources, network states, and related contextual information. Based on this view, it performs unified intent refinement and decomposition and assigns subtasks to appropriate agents. In the distributed mode, each agent is capable of intent refinement and decomposition; upon receiving a task intent, an agent autonomously processes the intent and collaborates with other agents through negotiation to accomplish the task. In the semi-distributed mode, a group of agents dynamically selects a central agent based on their intelligence levels. Task intents, together with relevant network information, are forwarded to the selected agent, which performs intent refinement and decomposition on behalf of the group.

These three orchestration modes are suitable for different scenarios and offer distinct advantages. Centralized orchestration is well-suited to relatively stable networks and tightly coupled tasks that require unified control. Distributed orchestration targets highly dynamic networks with numerous mobile nodes, where tasks exhibit weak dependencies and high requirements for flexibility and fault tolerance. Semi-distributed orchestration applies to agent networks that are locally centralized but globally distributed, aiming to balance the efficiency of centralized planning with the flexibility of distributed coordination.

3.2.2 Distributed Collaboration

Distributed collaboration signifies that agents are physically distributed and collaboratively accomplish tasks. Distributed agents execute assigned subtasks within a task subnet and continuously adjust their behaviors aligned with the task intent, collaborating and cooperating to achieve global task optimization.

To sustain such collaborative execution, the interplay between agents necessitates both information sharing and capability sharing. Information sharing encompasses the physical environment data perceived by agents, information regarding the capabilities of surrounding agents, and real-time network states. Through information exchange, agents within a task subnet can obtain a comprehensive view of subnet information. Capability sharing allows an agent to remotely leverage the capabilities of another agent to fulfill the current assignment.

Facing the challenge of massive data exchange among agents, distributed collaboration in HaoNet relies on semantic communication. To fulfill complex tasks, agents will exchange multi-modal information to achieve information unification, which inherently triggers massive data transmissions. Furthermore, the disparity in intelligence levels and functional capabilities leads to heterogeneous data modalities, creating a critical need for cross-modal transformation. To address these challenges, semantic communication is employed to provide expressive and flexible representations, such as embedding-based data transmission, ensuring efficient and seamless interoperability across agents.

3.2.3 Closed-Loop Intelligence

Closed-loop intelligence refers to the internal architecture of the agent that enables it to autonomously perform the full lifecycle of task intents, including environment perception, data processing, decision-making, and action execution. As shown in Fig. 3, a general agent consists of four core functional modules (perception, analysis, planning, and execution) supported by an internal knowledge base for data storage and inter-module interaction. To enhance the capabilities of each module, the agent integrates additional components such as LLMs, external knowledge bases, and external tools, thereby supporting comprehen-

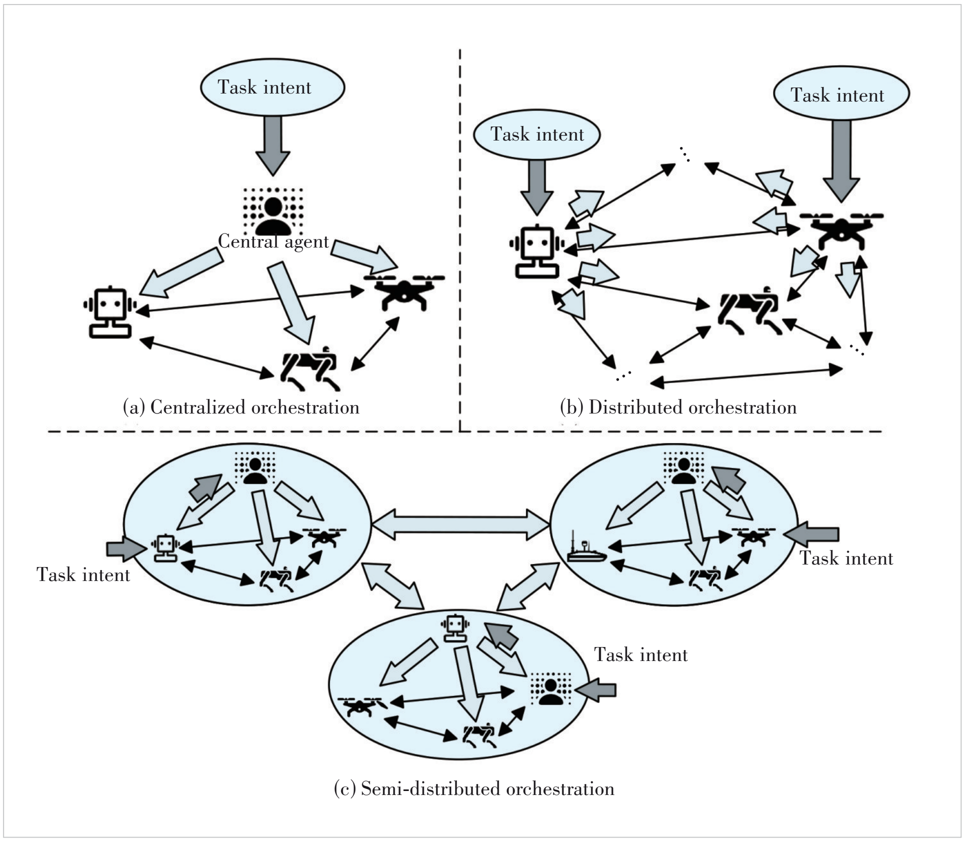


Figure 2. Three task orchestration modes in HaoNet

sive task completion[17].

The perception module monitors network connectivity, the physical environment, and agent capabilities. The analysis module handles intent recognition, conflict detection, intent refinement, and intent decomposition by utilizing LLMs and related processing tools. The planning module generates policies and resolves conflicts, drawing on analyzed data and other related tools to produce a new policy compatible with existing network policies. The execution module implements these policies using related execution tools to convert them into concrete configurations or actions. Interaction among these modules ensures reliable task execution and efficient collaboration with other agents.

3.3 Design of Architecture

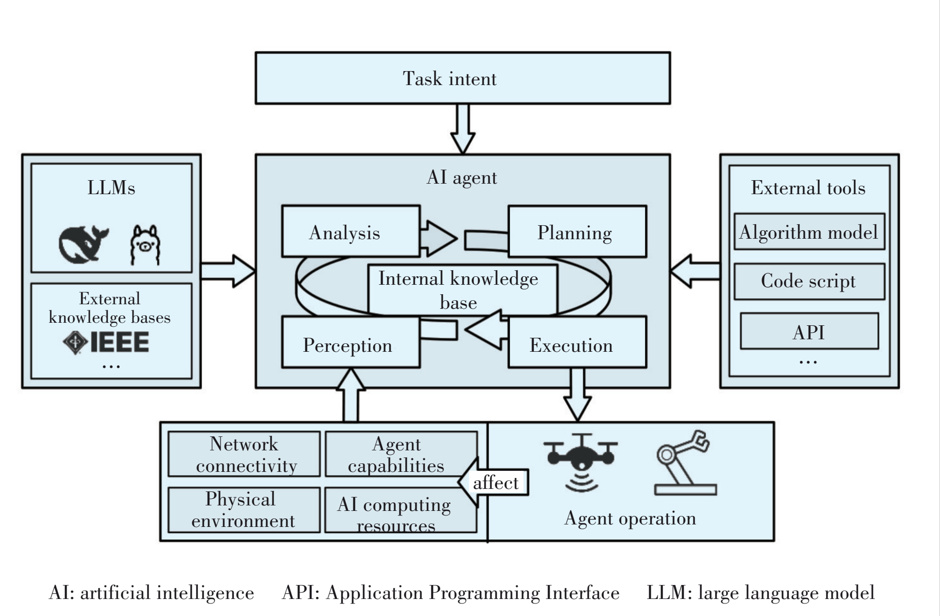
Based on the above summarization, an architecture of an intent-driven control system for HaoNet is proposed, as shown in Fig. 4. The architecture comprises three layers: the application layer, the intent layer, and the infrastructure layer. The application and intent layers reside within each agent, while the agents and their surrounding physical and network environments form the infrastructure layer. The separation between the application and intent layers ensures modularity and functional clarity. The application layer handles user interaction and intent input, while the intent layer performs intent processing and decision-making, acting for closed-loop operation.

1) Application layer

The application layer is responsible for receiving task intents from users or applications. It supports flexible intent expressions through multi-modal inputs, including voice, text, and graphical interactions, thereby providing a user-friendly and intuitive interface. Serving as the entry point of the overall architecture, this layer abstracts the complexity of the underlying network.

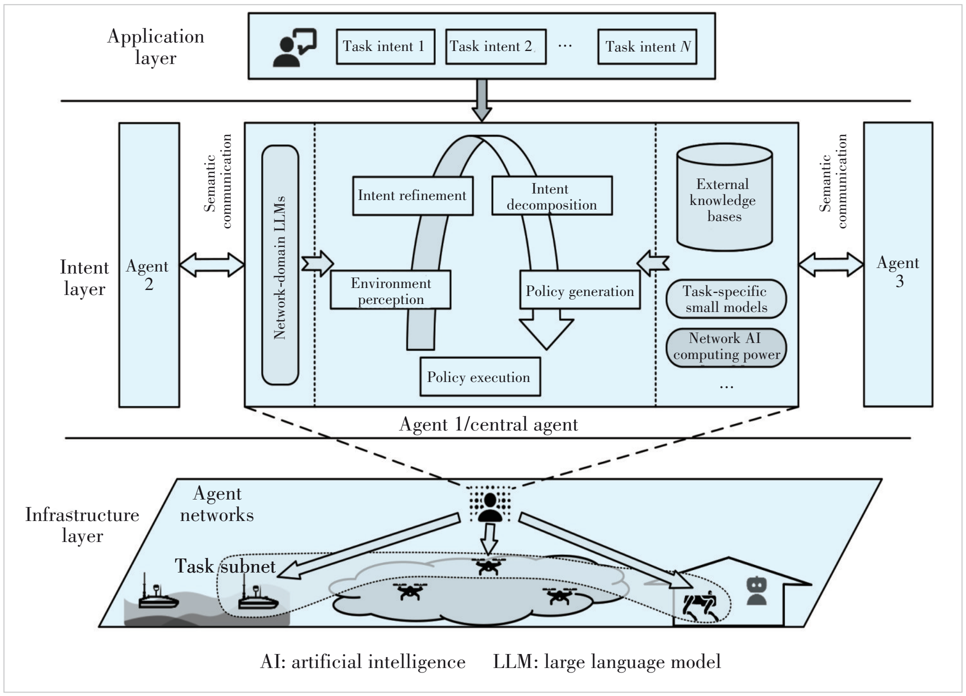
2) Intent layer

The intent layer is responsible for interpreting task intents and transforming them into executable policies. It performs a complete technical workflow, including environment perception, intent refinement, intent decomposition, and policy generation, based on the agent’s internal functional modules of Perception, Analysis, Planning, and Execution.



AI: artificial intelligence API: Application Programming Interface LLM: large language model

Figure 3. General internal architecture of agent



AI: artificial intelligence LLM: large language model

Figure 4. Architecture of an intent-driven control system for HaoNet

Environment perception collects heterogeneous contextual information from both the physical and network environments, providing essential inputs for subsequent processing. Intent refinement extracts explicit task requirements from user intents. Intent decomposition integrates the task requirements with environmental information to identify appropriate agent sub-groups, generate executable subtasks for individual agents, and coordinate their execution to optimize the fulfillment of the overall task intent. Based on the agent's current state and assigned subtask requirements, the policy generation function produces executable policies and delivers them to the execution module for concrete actions.

The intent layer incorporates semantic communication to support multi-modal and cross-modal information exchange among heterogeneous agents. Multi-modal inputs are encoded into unified semantic embeddings via modality-specific encoders, preserving task-relevant information for intent understanding, agent matching, and subtask coordination. These embeddings can be further transformed across modalities to ensure consistent interaction among agents with different capabilities. Structured interfaces based on MCP and A2A support context exchange, tool invocation, and task-level coordination, reducing redundant transmission and improving interoperability.

3) Infrastructure layer

The infrastructure layer is responsible for constructing task-specific subnets based on the matching results between agents and assigned subtasks, thereby enabling multi-agent collaboration. After receiving their respective subtasks, agents within a task subnet exchange information and adapt their execution policies in response to environmental dynamics, collectively working toward the optimal fulfillment of the overall task intent.

Built on the IDN framework, the architecture incorporates inter-agent semantic communication and implements workflows such as intent refinement and policy execution within the agents. Through continuous interactions within the three-layer architecture, the north – south direction realizes a full-lifecycle closed loop of task intent, from input and execution to assurance, while the east – west direction supports efficient collaboration among multiple agents, thus comprehensively ensuring the dynamic adaptability and flexible scheduling of HaoNet.

4 Key Technologies

To realize the intent-driven control system for HaoNet, this section presents the key technologies from three perspectives: Monitor-Analyze-Plan-Execute over a shared Knowledge (MAPE-K) based intent closed-loop and adaptive modes; LLM for intent recognition, refinement, decomposition, and mapping; and Coordination, Collaboration, and Cooperation (CoX) and large-small model collaboration mechanisms. These technologies form a general framework that can be flexibly adapted to different networks such as 5G/6G and unmanned ad-hoc networks.

4.1 MAPE-K Based Intent Closed-Loop and Adaptive Mode

To handle diverse task intents, agents must exhibit task adaptability and support closed-loop policy generation to ensure intent fulfillment. MAPE-K, an adaptive IDN architecture, consists of five components—Monitor, Analyze, Plan, Execute, and Knowledge—enabling autonomous management and intelligent optimization[18]. The internal MAPE-K architecture of the agent is shown in Fig. 5.

The Monitor, Analyze, Plan, and Execute components constitute the primary functional components of the agent. Knowledge serves as the data storage module, providing storage space for data interaction among the primary functional components. The Monitor captures environmental information through integrated sensors and data filtering tools. The Analyze component processes this data to perform intent refinement, decomposition, and conflict detection, leveraging LLMs and external knowledge bases to extract task requirements and generate subtasks while checking for conflicts with existing network policies. The Plan component generates executable policies by reconciling task requirements and detected conflicts, using LLMs and specialized small models. Finally, the Execute component implements these policies, employing code scripts and physical tools to carry out concrete task operations.

The MAPE-K architecture supports multiple patterns to accommodate different task scenarios, including coordinated control, information sharing, master-slave control, regional planning, and hierarchical control patterns[19]. In multi-agent collaboration, an MAPE-K based agent can dynamically select patterns according to task complexity, environmental conditions, and its own capabilities. The schematic diagrams of these five control modes across three task orchestration modes are shown in Fig. 6. While each agent in the figure contains all MAPE-K components, the diagrams highlight only the primary components utilized in each mode.

In the centralized orchestration mode, the main patterns are hierarchical control and master-slave control patterns. The hierarchical control pattern features a top-down structure, where the high level maintains a global network view and ensures long-term adaptive objectives, the middle level coordinates and manages multiple lower-level agents, and the low level handles direct execution, status monitoring, and short-term adaptive decisions. This pattern is well-suited for large-scale management systems, such as massive data centers, providing clear hierarchies but increasing management complexity. In contrast, the master-slave control pattern is simpler and easier to implement. The master agent handles analysis and planning, while slave agents focus on monitoring and execution. The master agent directly governs all slave agents, imposing higher stability requirements on the master and lacking the long- and short-term goal differentiation inherent to hierarchical control, making it better suited for small-scale agent networks.

In the distributed orchestration mode, the main patterns are coordination control and information sharing. The coordination control pattern emphasizes system consistency in intent execution. It requires the four MAPE-K components of each agent to

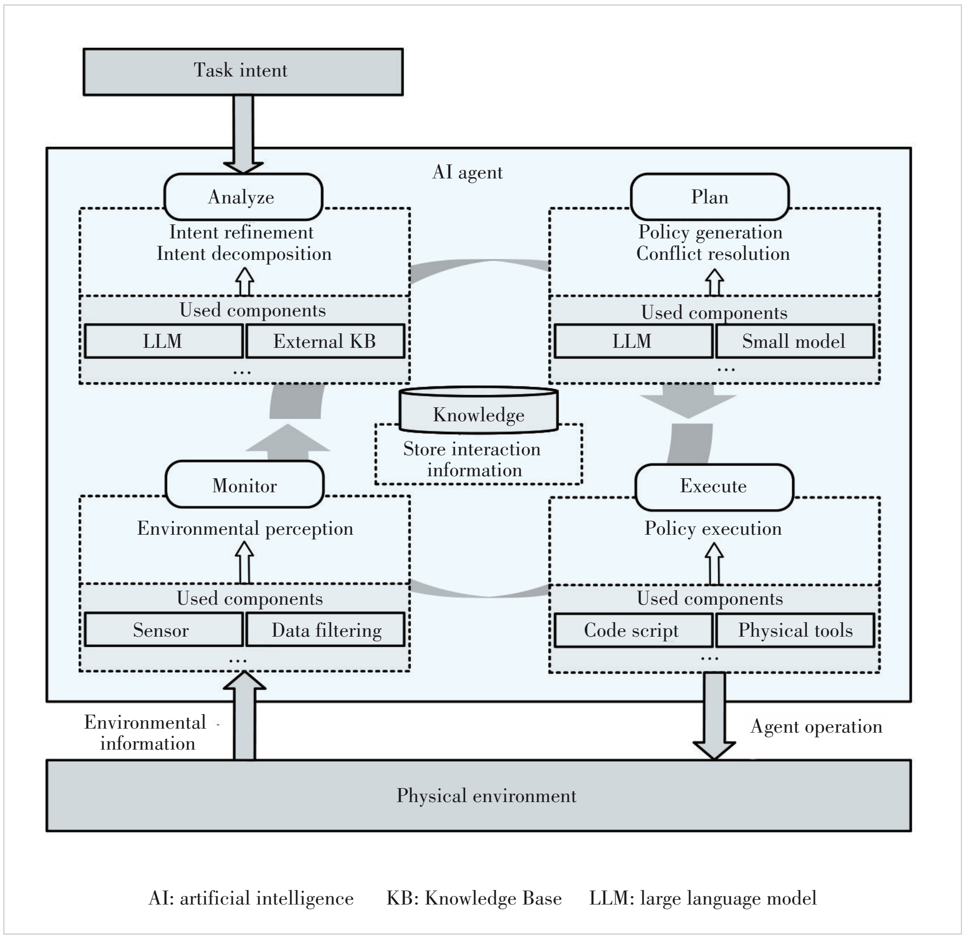


Figure 5. Internal MAPE-K architecture of the agent

interact with their counterparts in other agents. This pattern is suitable for tasks with high precision coordination requirements, but it also entails higher coordination costs. The information sharing pattern, by contrast, limits interaction to the Monitor component, while other components operate independently. It reduces complexity and inter-agent interactions, making it suitable for tasks such as distributed environmental monitoring, where information synchronization is prioritized over behavior.

In the semi-distributed orchestration mode, the main patterns are regional planning and master-slave control. In the regional planning pattern, a central agent is assigned the planning function within each region. Agent subnets achieve intra-domain objectives through local closed loops, while inter-domain objectives are accomplished via collaboration among the central agents of different regions. The master-slave control pattern integrates features of multi-regional planning, establishing master-slave control within each region while enabling collaboration among central agents across regions. Compared to master-slave control, regional planning provides greater autonomy to ordinary agents and reduces the computational burden on central agents, whereas master-slave control is more centralized, emphasizing the achievement of overall objectives. Both patterns are suitable for geographically dispersed tasks, balancing local adaptation with cross-regional optimization.

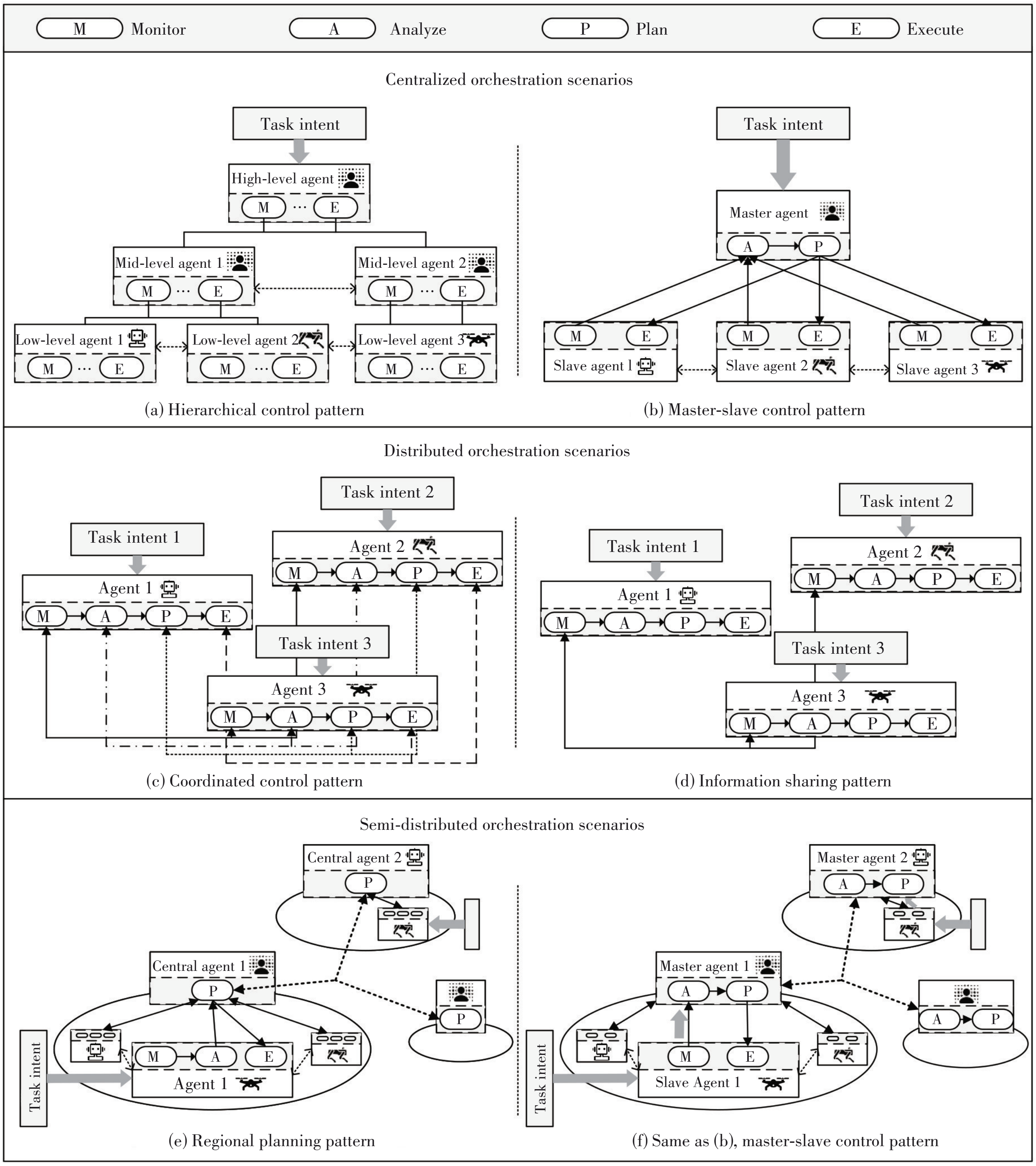
In large-scale agent-oriented networking scenarios, the scalability of the proposed control system is primarily enabled by the flexible adaptability of MAPE-K control patterns. Selecting appropriate patterns according to environmental characteristics ensures stable system operation. For example, in stable environments, hierarchical control provides large-scale structured collaboration, while in dynamic environments, information sharing and regional planning may enhance adaptability.

4.2 LLM for Intent Recognition, Refinement, Decomposition, and Mapping

For each task intent received by an agent, the processing workflows are performed in conjunction with its dependent LLM. The processing capability of the LLM largely represents the intelligence level of the agent. Therefore, to enhance the application of the LLM in specialized fields, this paper investigates the LLM for intent recognition, refinement, decomposition, and mapping, with the proposed technical architecture shown in Fig. 7.

In recent years, LLMs have attracted significant attention due to their powerful natural language processing capabilities, spurring extensive research on their application in networking[20–25]. However, deploying LLMs in highly specialized domains often faces challenges such as opaque internal mechanisms, limited interpretability, and hallucinations[26]. To improve reliability in professional network scenarios, mainstream approaches include fine-tuning[27] and prompt engineering[28]. Fine-tuning adjusts all or part of the model parameters on smaller, domain-specific datasets to enhance specialized performance, whereas prompt engineering operates at the input level, augmenting the user intent with additional information and instructions to elicit more accurate and comprehensive outputs.

This paper enhances the domain-specific performance of LLM through Parameter-Efficient Fine-Tuning (PEFT) and GraphRAG. PEFT trains only a subset of the LLM's parameters, with popular schemes including Prompt Tuning[29], LoRA[30], and Adapter Tuning[27]. An external knowledge base is constructed from network-domain documents, research papers, and expert knowledge, and is continuously updated as technology



(a) Hierarchical control pattern

(b) Master-slave control pattern

(c) Coordinated control pattern

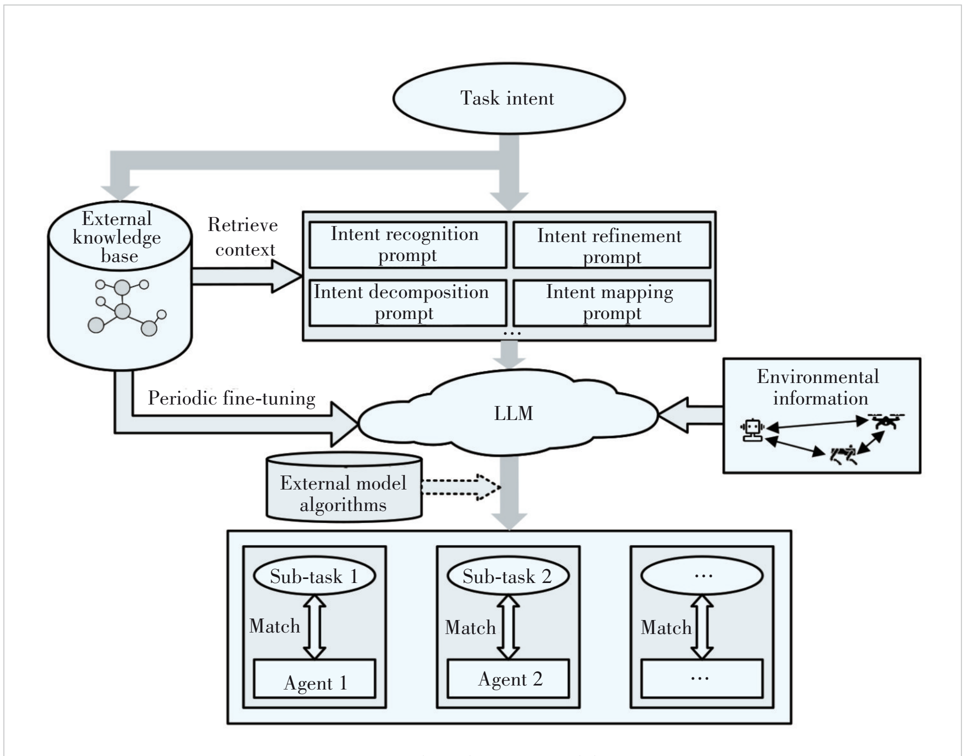
(d) Information sharing pattern

(e) Regional planning pattern

(f) Same as (b), master-slave control pattern

Figure 6. Five control patterns of MAPE-K across three task orchestration modes

evolves. The LLM is periodically fine-tuned using this knowledge base to achieve ongoing capability enhancement. GraphRAG[31–32], an extension of standard RAG and a prompt engineering technique, retrieves intent-related triplets from the knowledge base and performs graph-based expansions to enable contextual retrieval. The task intent, retrieved context, and relevant prompt templates are sequentially provided to the LLM according to the workflow. Combined with external model algorithms, this allows the LLM to perform task intent recognition, refinement, decomposition, and mapping. Additionally, environmental information collected by the agent's perception module is fed into the LLM to support comprehensive analysis and



LLM: large language model

Figure 7. Technical architecture of LLM-based intent recognition, refinement, decomposition, and mapping

achieve optimal task-agent matching.

4.3 CoX and Large-Small Model Collaboration Mechanisms

To address challenges such as the difficulty of orderly governance over multiple agents, the design of the CoX mechanism is essential for ensuring reliable task intent execution. A schematic of the CoX mechanism within HaoNet is shown in Fig. 8.

Within the CoX mechanism, Coordination organizes and orchestrates agents, assigning subtasks to achieve complex objectives. Collaboration ensures that agents with distinct tasks work together to fulfill the overall mission, as individual efforts alone are insufficient. Cooperation allows idle agents to assist others when a subtask exceeds an agent's capacity[33]. In HaoNet, coordination refers to the rational partitioning of task intents into subtasks and their assignment to appropriate agents. Collaboration denotes that agents receiving subtasks form task subnets and interact according to orchestration specifications to ensure subtask execution and efficient realization of the overall task intent. Cooperation refers to scenarios in which an agent, when facing a subtask whose workload exceeds its reliable execution capability, seeks assistance from other idle agents with similar capabilities to jointly complete the subtask. Based on the CoX mechanism, task intents can be executed by multiple agents in a rational and orderly manner, ensuring the efficient operation of the AI-Agent Communication Network (ACN).

Within a single agent, a collaboration mechanism between large and small models is necessary to ensure efficient task execution. LLMs offer strong language understanding and generalization but incur high computational costs, slow inference, and limited specialization in decision-making. In contrast, small models are computationally efficient, fast, and excel at specialized decision-making, though they have weaker language processing and generalization. Additionally, Ref. [34] highlights that LLMs are insensitive to changes in numerical environmental conditions and lack precise decision-adjustment capabilities, motivating a symbiotic reinforcement learning approach with small models to achieve continuous optimization of the agent network. The large-small model collabora-

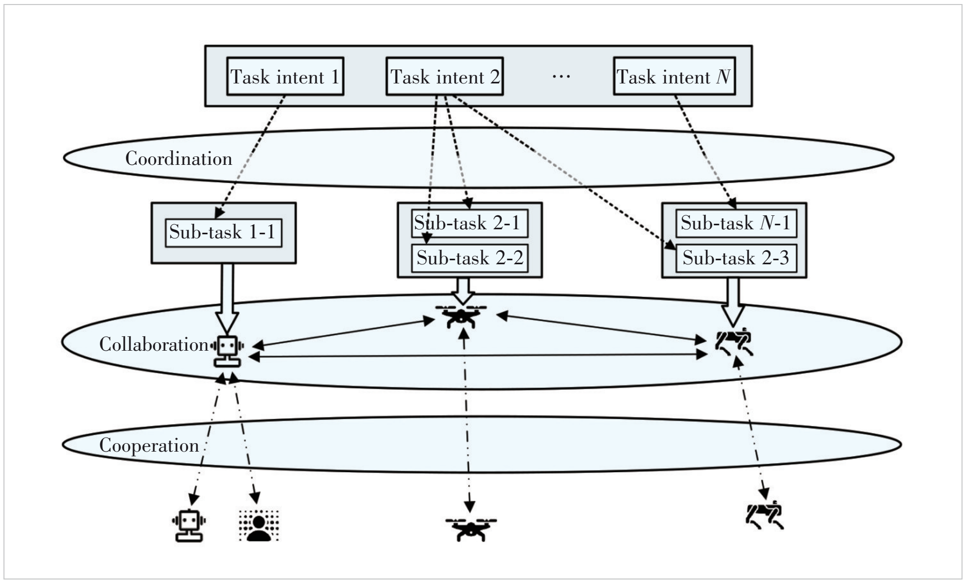


Figure 8. Schematic of CoX mechanism

tion mechanism is illustrated in Fig. 9.

The internal architecture of the agent integrates a domain-specific LLM for networking[23–24] and task-specific small models. Leveraging its natural language processing capabilities, the LLM identifies highly specialized tasks that require the expertise of a small model, initiates an invocation command, and forwards the request. By utilizing rule-based algorithms or pre-trained neural networks specific to concrete tasks within these small models, the system generates task policies that are professional and possess high real-time performance. Conversely, tasks involving extensive data analysis or language processing remain handled by the LLM itself. The final executable policy can either be directly generated by a task-specific small model or synthesized by the LLM based on feedback received from the small models. The large-small model collaboration mechanism combines the strengths of both, achieving adaptive selection of task processing strategies to support diverse tasks.

From a scalability perspective, the scalability of the proposed control system is primarily enabled by multiple MAPE-K control patterns. In addition, supporting mechanisms such as large-small model collaboration further improve system efficiency and responsiveness, facilitating practical deployment in complex scenarios. Together, these mechanisms enable the system to maintain performance as the system scales.

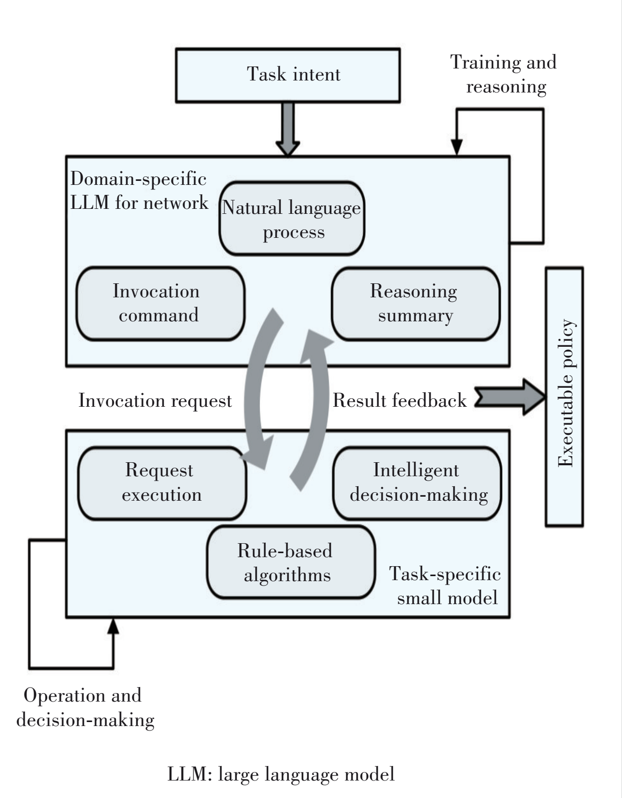


Figure 9. Schematic of large-small model collaboration mechanism

5 Application Scenarios

To demonstrate the practical value of the proposed system, we provide two representative application scenarios, as shown in Fig. 10.

1) Smart home scenario

Future smart homes are evolving from simple remote control and preset automation into complex ecosystems composed of multiple specialized agents. Consider a common return-home scenario in daily life: The system may include digital agents, such as home steward agents and health agents, as well as physical agents, such as robotic agents and quadruped robots. Before returning home, the user informs the home steward agent of the current intent: “Steward, I will be home in 30 minutes and I need a warm meal.” Based on this intent, the home steward agent might assign a health agent to curate a nutritionally balanced recipe, command a quadrupedal robot to prepare the necessary ingredients, and finally delegate the cooking process to a robotic agent.

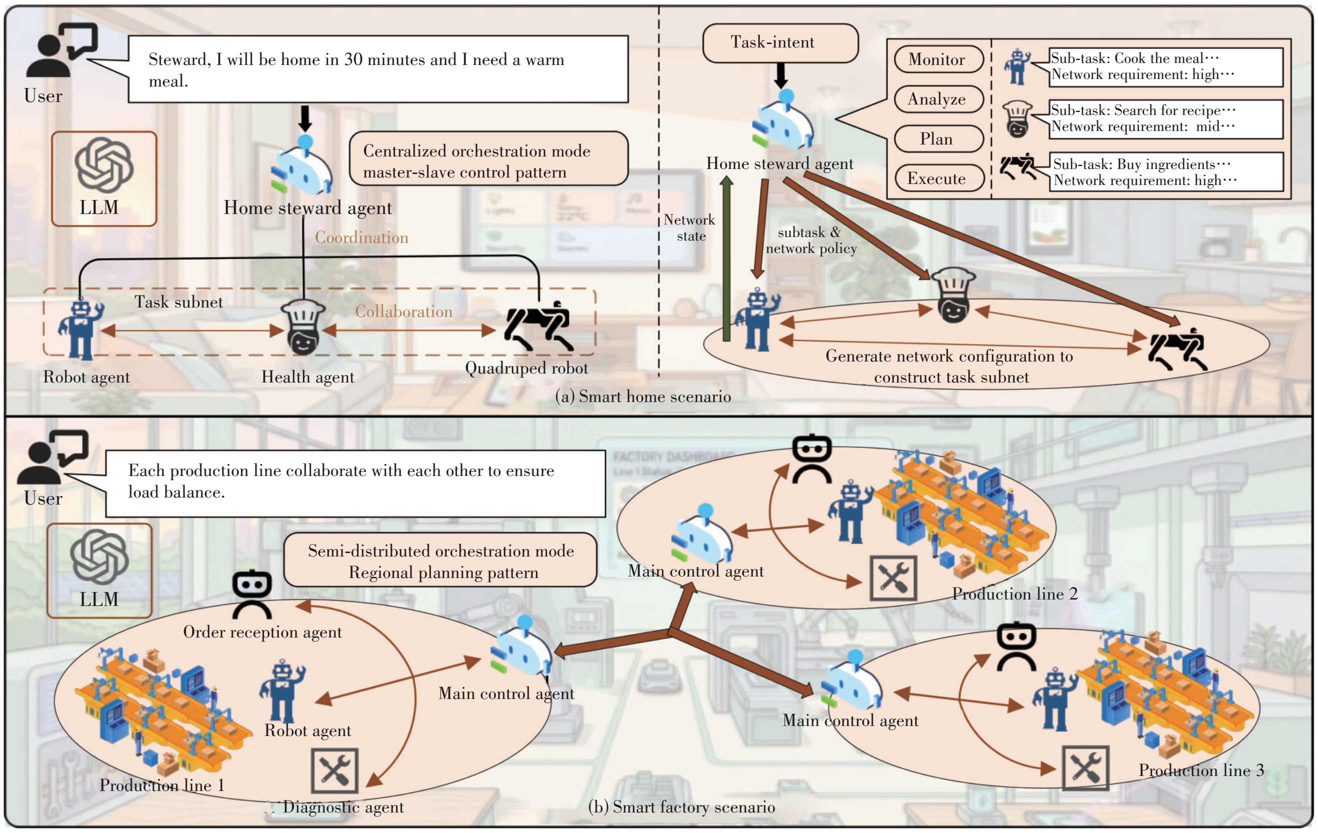
In this process, the proposed system plays a central role. When an unstructured intent is received, the intent layer performs intent refinement to identify specific task requirements. By referencing the real-time availability of agents based on the environment perception, the system executes an automated agent matching process and intent decomposition accordingly. Given the relatively stable and small-scale nature of smart home environments, the system utilizes the centralized task orchestration mode and selects the master-slave control pattern. Therefore, the home steward agent acts as the master agent, responsible for intent processing at the intent layer and for generating corresponding network policies according to the communication requirements among slave agents, thereby ensuring reliable communication during task execution.

Upon the successful formation of this task-specific subnet, the slave agents collaborate to complete their respective subtasks. After the overall task intent is fulfilled, the system systematically deallocates the dynamically configured resources and dissolves the task-specific subnet.

2) Smart factory scenario

Manufacturing factories are undergoing a digital transformation toward device-level intelligence, as they evolve into highly collaborative multi-agent networks. Consider a typical multi-production-line manufacturing scenario: Several production lines operate simultaneously, each managed by a dedicated main control agent. Under each main control agent, different types of agents are deployed to support production operations, including order reception agents, diagnostic agents, and robot agents. Meanwhile, the main control agents of different production lines exchange information and coordinate their actions to ensure balanced workloads and efficient collaboration across the entire factory.

Through continuous environment perception, the system obtains real-time information about the operational status of production lines, the availability of equipment agents, and the workload of embodied robots. Based on this information, the



LLM: large language model

Figure 10. Two application scenarios of HaoNet: smart home and smart factory

system performs agent matching and intent decomposition, generating subtasks that are distributed to the corresponding agents within each production line. Considering the relatively large scale of industrial production systems and the need for both local autonomy and global coordination, the system adopts a semi-distributed task orchestration mode with a regional planning pattern. In this architecture, each production line is managed by its own main control agent responsible for local decision-making and for generating corresponding network policies according to the communication requirements among task agents, while inter-line communication among main control agents enables collaborative scheduling, load balancing, and global production optimization.

After the task-oriented subnet for a given production objective is established, the agents within each production line collaborate to complete their respective subtasks under the coordination of the local main control agent. Diagnostic agents continuously monitor production conditions, and robot agents perform the required physical manipulation tasks. Meanwhile, the main control agents exchange status information and coordinate resource allocation across production lines to maintain system-wide efficiency and stability. Through this mechanism, the proposed system supports flexible and intelligent coordination across heterogeneous agents, thereby improving production efficiency, operational reliability, and overall manufacturing adaptability in smart factory environments.

6 Conclusions

Heterogeneous agent-oriented networking aims to enable an interconnected and efficiently collaborative network for a large-scale population of heterogeneous agents. Leveraging the technical strengths of intent-driven networking, the proposed control system enhances networking effectiveness and task execution efficiency for heterogeneous agent-oriented networking, while enabling intent recognition, translation, and decomposition. By incorporating key technologies such as MAPE-K and the large-small model collaboration mechanism, it offers a novel architectural solution to advance future agent-oriented networking.

References

[1] ETSI GR ENI 051 v4.1.1: 2025 Experiential networked intelligence (ENI); study on AI agents based next-generation network slicing [S]
[2] Monica. Manus [EB/OL]. [2026-03-10]. https://manus.im
[3] Bytedance. Coze [EB/OL]. [2026-03-10]. https://www.coze.com
[4] China Mobile Think Tank. AI-agent communication network white paper [R]. Beijing: China Mobile, 2024

[5] Yang C G. Intent-driven autonomous network [M]. Beijing: Science Press, 2024

[6] Gupta M, Acharya V. Agnet: a novel AI agent network architecture [PP/OL]. SSRN [2026-03-10]. https://ssrn.com/abstract=5108385

[7] Wang Y T, Guo S L, Pan Y H, et al. Internet of agents: fundamentals, applications, and challenges [J]. IEEE transactions on cognitive communications and networking, 2026, 12: 4476 – 4501. DOI: 10.1109/TCCN.2025.3623369

[8] Anthropic. Model context protocol (MCP) [EB/OL]. [2026-03-10]. https://www.anthropic.com/news/model-context-protocol

[9] Google. Agent to agent protocol (A2A) [EB/OL]. [2026-03-10]. https://a2a-protocol.org/latest/

[10] Xiao Y, Shi G M, Zhang P. Toward agentic AI networking in 6G: a generative foundation model-as-agent approach [J]. IEEE communications magazine, 2025, 63(9): 68 – 74. DOI: 10.1109/MCOM.001.2500005

[11] Du C G, Wang C Y, Chao Y H, et al. AI agent communication from Internet architecture perspective: challenges and opportunities [PP/OL]. V1. arXiv (2025-09-02)[2026-03-10]. https://doi.org/10.48550/arXiv.2509.02317

[12] Minhas S, Jaswal R, Sharma A, et al. Revolutionizing networking: a comprehensive overview of intent-based networking [C]//Proceedings of International Conference on Emerging Innovations and Advanced Computing (INNOCOMP). IEEE, 2024: 463 – 468. DOI: 10.1109/INNOCOMP63224.2024.00081

[13] Jiang G Z, Wang K Z, Chen X M, et al. Agentic AI empowered intent-based networking for 6G [PP/OL]. V1. arXiv (2026-01-10)[2026-03-10]. https://doi.org/10.48550/arXiv.2601.06640

[14] Lee K Y. Intent-driven autonomous reconfiguration for telecom OSS: a generative-AI-empowered hierarchical multi-agent architecture [C]//Proceedings of 25th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 2025: 1 – 6. DOI: 10.23919/APNOMS67058.2025.11181316

[15] Romero M L, Suyama R. Agentic AI for intent-based industrial automation [C]//Proceedings of 16th IEEE International Conference on Industry Applications (INDUSCON). IEEE, 2025: 437 – 444. DOI: 10.1109/INDUSCON66435.2025.11241317

[16] Chen S T, Liao Q, Aijaz A, et al. Goal-oriented multi-agent semantic networking: unifying intents, semantics, and intelligence [PP/OL]. V2. arXiv (2026-03-24)[2026-03-10]. https://doi.org/10.48550/arXiv.2512.01035

[17] Tong J W, Guo W, Shao J W, et al. WirelessAgent: large language model agents for intelligent wireless networks [PP/OL]. V1. arXiv (2025-05-02) [2026-03-10]. https://doi.org/10.48550/arXiv.2505.01074

[18] Dzeparoska K, Beigi-Mohammadi N, Tizghadam A, et al. Towards a self-driving management system for the automated realization of intents [J]. IEEE access, 2021, 9: 159882 – 159907. DOI: 10.1109/ACCESS.2021.3129990

[19] Weyns D, Schmerl B, Grassi V, et al. On patterns for decentralized control in self-adaptive systems [M]//Software Engineering for Self-Adaptive Systems II. Berlin, Heidelberg: Springer, 2013: 76 – 107. DOI: 10.1007/978-3-642-35813-5_4

[20] Jiang F B, Peng Y B, Dong L, et al. Large language model enhanced multi-agent systems for 6G communications [J]. IEEE wireless communications, 2024, 31(6): 48 – 55. DOI: 10.1109/MWC.016.2300600

[21] Saad W, Hashash O, Thomas C K, et al. Artificial general intelligence (AGI)-native wireless systems: a journey beyond 6G [J]. Proceedings of the IEEE, 2025, 113(9): 849 – 887. DOI: 10.1109/JPROC.2025.3526887

[22] Cui H W, Du Y Y, Yang Q, et al. LLMind: orchestrating AI and IoT with LLM for complex task execution [J]. IEEE communications magazine, 2025, 63(4): 214 – 220. DOI: 10.1109/MCOM.002.2400106

[23] Chen Y X, Li R P, Zhao Z F, et al. NetGPT: a native-AI network architecture beyond provisioning personalized generative services [PP/OL]. V4. arXiv (2024-03-09) [2026-03-10]. https://doi.org/10.48550/arXiv.2307.06148

[24] Zou H, Zhao Q Y, Tian Y, et al. TelecomGPT: a framework to build telecom-specific large language models [J]. IEEE transactions on machine learning in communications and networking, 2025, 3: 948 – 975. DOI: 10.1109/TMLCN.2025.3593184

[25] Wang X C, Ji S W, Li C. Research on intelligent computing network technology for large-scale pre-trained models [J]. Telecommunications science, 2024, 40(6): 160 – 172. DOI: 10.11959/j.issn.1000-0801.2024167

[26] Huang B, Wu S A, Wang W G, et al. KG-LLM-MCom: a survey on integration of knowledge graph and large language model [J]. Journal of Wuhan University (natural science edition), 2024, 70(4): 397 – 412. DOI: 10.14188/j.1671-8836.2024.0040

[27] Hu Z Q, Wang L, Lan Y H, et al. LLM-adapters: an adapter family for parameter-efficient fine-tuning of large language models [C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: ACL, 2023: 5254 – 5276. DOI: 10.18653/v1/2023.emnlp-main.319

[28] Mekrache A, Ksentini A. LLM-enabled intent-driven service configuration for next generation networks [C]//Proceedings of IEEE 10th International Conference on Network Softwarization (NetSoft). IEEE, 2024: 253 – 257. DOI: 10.1109/NetSoft60951.2024.10588881

[29] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning [C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. ACL, 2021: 3045 – 3059. DOI: 10.18653/v1/2021.emnlp-main.243

[30] Hu E J, Shen Y, Wallis P, et al. LoRA: low-rank adaptation of large language models [C]//International Conference on Learning Representations. ICLR, 2022: 1 – 20. DOI: 10.48550/arXiv.2106.09685

[31] Han H Y, Wang Y, Shomer H, et al. Retrieval-augmented generation with graphs (GraphRAG) [PP/OL]. V2. arXiv (2025-01-08) [2026-03-10]. https://doi.org/10.48550/arXiv.2501.00309

[32] Edge D, Trinh H, Cheng N, et al. From local to global: a graph RAG approach to query-focused summarization [PP/OL]. V2. arXiv (2025-02-19) [2026-03-10]. https://doi.org/10.48550/arXiv.2404.16130

[33] Ou Y Y. Intelligent intent policy refinement of space-terrestrial network [D]. Xi‘an: Xidian University, 2023

[34] Liu Y Q, Liu G Y, Wang J C, et al. LAMeTA: intent-aware agentic network optimization via a large AI model-empowered two-stage approach [J]. IEEE journal on selected areas in communications, 2026, 44: 2462 – 2478. DOI: 10.1109/JSAC.2025.3642840

Biographies

Wang Bowen is currently pursuing his doctoral degree with the GUIDE family at Xidian University, China. His research interests include intent-driven networks, large AI models, and software-defined networks.

Lu Lu received her master’s degree from Beijing University of Posts and Telecommunications, China. She is currently the Deputy Director of the Department of Basic Network Technology, China Mobile Research Institute; the leader of the Core Network Group, CCSA TC5; and the Vice Chair of ITU-T SG13. Her main research interests include 5G/6G network architecture and next-generation IP technology.

Li Huimin is currently pursuing a master’s degree in the GUIDE family at Xidian University, China. Her research interests include intent-driven networks, agents, and software-defined networks.

Yang Chungang (guideyang2050@163.com) is a full professor at Xidian University, China. His research interests are AI-enabled 6G wireless mobile networks, intent-driven networks, space-terrestrial networks, and game theory for emerging communication networks.