Internet of Agents: Design of the Protocol System

Fu Yuexia, Liu Peng, Lu Lu, Duan Xiaodong
(China Mobile Research Institute, Beijing 100053, China)

DOI: 10.12142/ZTECOM.202602005
https://kns.cnki.net/kcms/detail/34.1294.TN.20260429.1856.004.html, published online April 30, 2026
Manuscript received: 2026-03-11

Abstract: With the rapid advancement of generative artificial intelligence (AI) and large language model (LLM) technologies, AI agents are gradually becoming the core service units in networks, and their communication mode is evolving from local collaboration to wide-area interconnection. The construction of the Internet of Agents (IoA) faces multiple challenges, such as identity management, dynamic networking, and semantic routing, which urgently requires the design of a network protocol system that adapts to its new traffic characteristics and collaboration needs. Based on the application scenarios of agent communication, this paper systematically analyzes the management, control, and routing requirements that multi-agent collaboration imposes on IP networks, proposes a three-layer functional architecture for the IoA, and designs a protocol suite covering management, control, and routing around key issues such as agent registration and identification, service discovery, capability sensing, and cross-domain traffic assurance. By extending existing Internet protocols and introducing a semantically aware routing mechanism, this paper provides a scalable, efficient, and secure approach to implementing a protocol for end-to-end agent collaboration, thereby contributing to the construction of an open, large-scale agent collaboration ecosystem.

Keywords: Internet of Agents; user agent; service agent; protocol framework

Citation (Format 1): Fu Y X, Liu P, Lu L, et al. Internet of agents: design of the protocol system [J]. ZTE Communications, 2026, 24(2): 33 - 42. DOI: 10.12142/ZTECOM.202602005
Citation (Format 2): Y. X. Fu, P. Liu, L. Lu, et al., "Internet of agents: design of the protocol system," *ZTE Communications*, vol. 24, no. 2, pp. 33 - 42, Jun. 2026. doi: 10.12142/ZTECOM.202602005.

1 Introduction

With the breakthrough in large language model (LLM) technology, LLM-based agents have demonstrated excellent autonomous perception, decision-making, and action capabilities across various tasks and are regarded as one of the important paths to achieve artificial general intelligence (AGI). While a single agent can handle specific tasks independently, complex scenarios often require the collaboration of multiple heterogeneous agents. The core challenge then lies in how to achieve efficient and reliable organization and interaction among these agents. This makes the Internet of Agents (IoA) protocol a key enabling technology.

Agent protocols are mainly divided into two categories. One is the agent interaction protocol, which is used to standardize the structured communication and collaboration mechanisms between agents[1 - 3]. For example, Anthropic's Model Context Protocol (MCP) focuses on the interaction between agents and external tools, while Agent Network Protocol (ANP), Agent-to-Agent (A2A), Agora, etc., are mainly oriented to direct communication and task coordination between agents[4 - 7]. Such protocols define message formats, session processes, and security rules, enabling multi-agent systems to achieve task allocation, knowledge sharing, and joint reasoning. The other is the agent network Internet protocol, which is responsible for connecting distributed agents in physical or virtual networks, and its foundation is the Transmission Control Protocol/Internet Protocol (TCP/IP) system and related underlying protocols, such as Wi-Fi, Ethernet, and Border Gateway Protocol/Interior Gateway Protocol (BGP/IGP) routing protocols[8 - 10]. With the development of agent applications towards real-time, collaborative, and cross-domain directions, the existing Internet architecture faces new requirements in terms of delay, reliability, and resource customization, promoting the continuous evolution of network layer protocols.

Nevertheless, most existing agent interaction protocols, including MCP, A2A, ANP, and Agora, focus on message exchange, tool invocation, and point-to-point coordination, while lacking unified specifications for agent service capability identifiers, agent capability sensing, task-driven network-

This work is supported by the National Key R&D Program of China under Grant No. 2024YFB2906701.

ing, and capability-oriented semantic routing. They rely heavily on traditional network addressing and lack support for semantic discovery, decentralized trust, and secure cross-domain scheduling in open, large-scale environments. The framework proposed in this paper comprises capability identification, capability control and management, and semantic routing as core components, which can effectively address the deficiencies of current protocols and be applied across different agent service providers and multiple domains. The standardization and optimization of the IoA protocol will be an important foundation for realizing a large-scale and open agent collaboration ecosystem.

The wide-area interconnection of agents may give rise to new behaviors such as message traffic management and transmission-reception control, thereby triggering new changes in Internet address naming, message forwarding, traffic control, and other aspects. Research on the IoA has emerged at home and abroad. Ref. [11] proposes that the IoA will be built based on cloud networks, with the bottom layer supported by the basic Internet, and looks forward to various types of agents relying on the Internet for interconnection and collaboration to complete tasks; Ref. [12] defines the concept of the IoA, proposes its architecture, including resource layer, management layer, collaboration layer, service layer, and user layer, and conceives five types of application scenarios and processes of the IoA.

On the basis of recent work, this paper further analyzes the protocol design challenges of the IoA, proposes a method for classifying agents into user agents and service agents, puts forward the protocol framework, and introduces the design of related key protocols, providing a reference for further application and promotion of the IoA.

2 Protocol Design Requirements for the Internet of Agents

With the development of AI agent technology, the independent work of single agents, local collaboration, and wide-area collaboration will become the mainstream interconnection modes, supporting various scenarios from personal tasks to cross-domain collaboration. The large-scale hybrid networking of physical and software agents will pose new systematic requirements for the management, control, and routing systems of the existing Internet.

The first is the agent management requirement, whose core is to realize unified identity management and full-life-cycle supervision. AI agents can be divided into user agents (which act on behalf of users to perform tasks) and service agents (which provide functions or services), and a unified registration and identification system should be established to be compatible with multi-source heterogeneous agents. The key points include standardized identification and registration of capabilities, performance monitoring and reliability assurance, and a charging mechanism for service calls and resource consumption.

The second is the IoA control requirement, which focuses on dynamically adaptive and policy-driven collaboration scheduling. To support flexible and secure collaboration, it is necessary to realize real-time discovery of capabilities and services, sensing of the status and load of service agents, dynamic networking, and fine-grained definition and cross-domain synchronization of user agent permissions, so as to ensure task-driven adaptive collaboration.

The third is the IoA routing requirement, which needs to realize semantically aware and differentiated traffic assurance. The new type of "implicit traffic" generated by the autonomous interaction of agents requires the routing mechanism to go beyond the traditional address-based strategy, support dynamic routing based on capabilities and task content, and integrate multiple forwarding methods such as unicast, anycast, and multicast. At the same time, it is necessary to distinguish between user-service interaction traffic and inter-service collaboration traffic, and implement intelligent scheduling and collaborative management according to their differences in bandwidth, delay, packet loss, etc.

3 Protocol Framework of the Internet of Agents

The IoA is a new type of overlay network built based on the IP network for agent interconnection, which meets the needs of flexible networking, autonomous communication, massive concurrency, etc., between agents. Based on the IoA architecture[13], this paper proposes a protocol view, which serves as the basis for the construction and implementation of its core protocols. Considering various factors such as the service object, main function, and permission of agents, this paper classifies agents into two categories: user agents and service agents.

The user agent parses the user's service intentions, automatically executes subsequent operations for the user, and interacts with other agents on behalf of the user, enabling the user to be unaware of the implementation details. Limited by human interaction, the interaction frequency is relatively low, and the traffic is mainly used for interaction instructions and execution results. The current Internet protocols can be adopted, and the information transmission content includes the instructions issued by users to agents and the feedback results of agents. In addition, such agents often form a deeply bound relationship with users, so authentication and authorization only occur in the early stage of establishing the relationship and regularly in the later stage, with a low frequency.

The service agent is mainly responsible for providing specific services, and interacts with user agents about service demand information, service execution result information, etc. Service agents need to register the services and capabilities they provide in the network for discovery and invocation by other agents. Their traffic is mainly the interac-

tion information between agents, including instructions, information, etc., and special protocols for inter-agent communication are adopted, including identification, addressing, etc. In addition, due to the task-driven agent collaborative networking, the initial authentication and authorization need to be triggered after each task group is established, so the authentication and authorization between service agents will be more frequent.

To meet the needs of agent identification, discovery, collaboration, and networking, the IoA protocol framework, which combines the networking view and architecture view, is divided into three major protocol suites: IoA Management Protocol, IoA Control Protocol, and IoA Routing Protocol, as shown in Fig. 1.

The IoA Management Protocol enables agents to interact with three types of platforms. First, they interact with the unified registration platform during network access registration to obtain agent identification information. Second, they interact with the unified service registration to register agent service capability information and its enabling information. Third, they interact with the management platform that obtains the performance information provided by services to evaluate the historical performance information of agent services. There are two kinds of identifiers in IoA: one is the agent identifier, used to identify the agent; the other is the agent service capability identifier, used to identify the agent service capability. Both identifiers need to be unified and verifiable. The IoA Management Protocol also includes protocols such as agent service orchestration, agent service operation, and agent service charging, supports the new task-based charging for agents, constructs an agent service evaluation system to support the performance and reputation evaluation of various agent service capabilities, and provides an agent capability exposure protocol to provide a unified capability calling platform, serving as a unified service operation entrance for agent users.

The IoA Control Protocol is used to support the sensing of agent service capability status. At the same time, it supports centralized agent service discovery to obtain information of agents participating in collaboration; supports task-driven networking implementation of multiple agents and performs traffic engineering according to service requirements. In addition, considering the implementation of multi-sessions between multiple agents, a transmission control protocol is used to ensure the transmission quality between multiple agents. Moreover, the control protocol can realize unified identification distribution, service discovery, networking routing, and other capabilities regardless of the access methods, supporting the agent communication requirements of 6G network and the integrated access and services under the trend of space-air-ground integration.

The IoA Routing Protocol is used to support the routing and forwarding of data between agents, including the intra-domain and inter-domain routing and forwarding protocols. It supports the semantic routing protocol based on the agent service capability identifier. It needs to support the adaptive routing protocol for multiple access methods such as mobile access, fixed access, and satellite access, which can realize the integrated and unified access of various access methods. In addition, it needs to support multi-modal routing protocols for voice, text, video, etc.

Based on the characteristics of the information to be transmitted by user agents and service agents, their service objects, and networking situations, Table 1 analyzes the demand characteristics of the two types of agents for management protocols, control protocols, and routing protocols.

4 Management Protocol Design for the Internet of Agents

4.1 Agent Registration and Identification Protocol

Both user agents and service agents require unified identification, but their capability types, registration information, permission information, and dependent trust roots are differ-

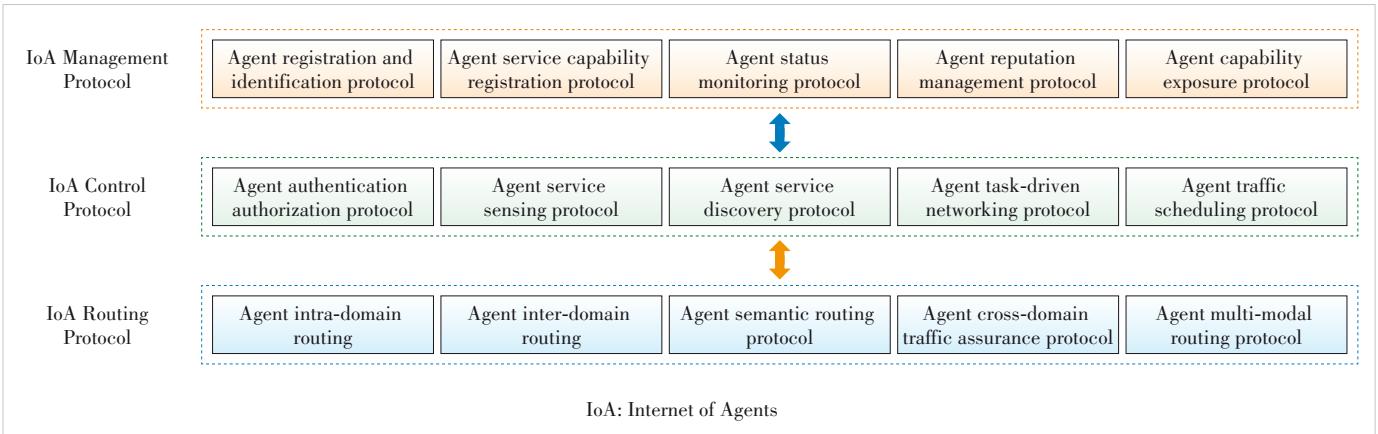

Figure 1. Protocol framework of the IoA

Table 1. Analysis of protocol requirements for user agents and service agents

Protocol Category	Specific Protocol	User Agent	Service Agent
IoA Management Protocol	Registration and Identification Protocol; Service Registration Protocol	Distribution of user agent identification information; association of user binding information	Distribution of service agent identification information and agent service capability identification information; registration and update of capability identification information
	Status Monitoring Protocol; Reputation Management Protocol	More direct user perception; demand for fixed-duration charging	Demand for service performance monitoring; task-based charging
IoA Control Protocol	Authentication and Authorization Protocol	Low frequency; consideration of user permission information	High frequency; consideration of agent service capability information and task group information
	Service Discovery Protocol	Small selection range, simple discovery mechanism; relatively fixed association with users	Large selection range; discovery based on service identification, combined with performance for optimization
	Service Sensing Protocol	Simple sensing information; infrequent status changes	Complex sensing information; frequent status changes
	Task-Driven Networking Protocol	Networking with users: local area, static network; networking among user agents: driven by multi-user collaboration, relatively static networking	Networking among service agents: task-driven networking, multi-agent networking; involving cross-domain networking
IoA Routing Protocol	Semantic Routing Protocol	Supporting IP-based routing and agent service capability identifier-based routing	Supporting IP-based routing and agent service capability identifier-based routing
	Cross-domain Traffic Assurance Protocol	Intra-domain routing	Inter-domain routing; demand for traffic engineering
	Multi-Modal Routing Protocol	Mainly transmitting interaction information between humans and agents, and task results of service agents	Mainly transmitting instructions, task related information, multi-modal, etc., among agents

IoA: Internet of Agents

ent. In the initial access and registration management stage, the management of user agents focuses on unified identity management, association with the affiliated user, and permission authentication, laying a foundation for subsequent service access and charging; the management of service agents focuses on service registration, service capability access verification, and maintenance of service capability dependency relationships, laying a foundation for subsequent service capability discovery and supply.

As shown in Fig. 2, user agents obtain identifiers assigned by the network operator they access through association with the user. User agents of different operators establish trust relationships through operators and follow a unified identification mechanism; service agents obtain identifiers assigned by their respective agent service providers. Service agents of different service providers establish trust relationships through the service providers and follow a unified identification mechanism; in addition, operators and service providers will also establish trust relationships, thereby enabling user agents across operators and service agents across service providers to establish trust relationships and realize the uniqueness and verifiability of identifiers.

As shown in Fig. 3, the access network operator issues a unified registration template to the user agent. After the user agent reports information according to the registration template, identifier distribution is carried out. In addition, after obtaining the identifier of the user agent, it is necessary to establish a binding relationship with the user and report information of the user agent, such as user association information, valid time, permission information, user agent type, and location information. Considering the security and privacy of user information, the relevant sensitive information is uniformly maintained by the access network operator. A new type of user agent registration template information can be defined. The issuance of such registration templates can be realized by extending existing registration protocols (e.g., Netconf), or a new user agent registration protocol can be defined to realize the flexible issuance and interaction of registration templates.

In addition, when the enabling information of the user agent or the associated user information changes (such as a current fault that makes it unable to provide services or changes in the bound user information), it is necessary to

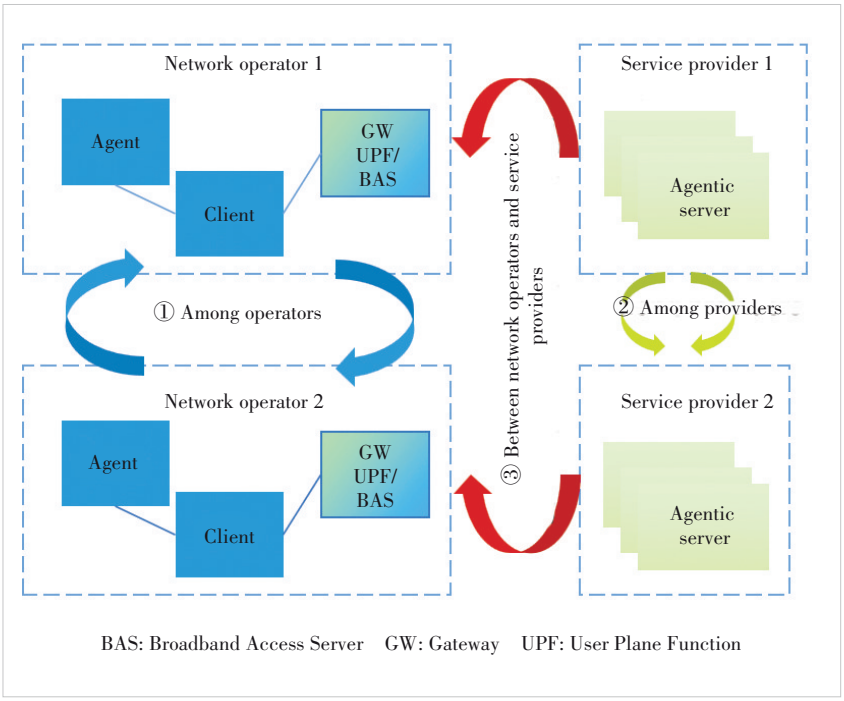

Figure 2. Classification view of agents in the IoA

synchronize the changes to the registration platform in a timely manner. A new type of update management message can be defined to update the changes in the attribute information of registered agents.

Service agents can be uniformly assigned agent identifiers by various agent service providers, which issue unified registration templates to the service agents. After the service agents report information according to the registration templates, identifier distribution is carried out. Optionally, a new type of service agent registration template can be defined, and its issuance can be realized by extending existing registration protocols, e.g., by extending interfaces based on the Representational State Transfer (REST) over HTTP. Various service agents establish mutual trust through service providers and maintain capability registration and verification information.

On the other hand, in addition to considering the identifier allocation for newly added agents, it is also necessary to support the identifier recovery mechanism for existing agents. For example, when an agent goes offline and no longer provides services, to ensure the effective allocation and utilization of identifiers, a new type of withdrawal message can be defined to recover the identifier of the agent for subsequent distribution.

4.2 Agent Service Capability Registration Protocol

After the initial identifier allocation of agents is completed, it is necessary to maintain the global agent service capability view of user agents and service agents in real time. Agents' services can be divided into two categories: self-services and open services[13]. Self-services can be used for agent maintenance, management, monitoring, etc., and are called by the registration or management platforms; open services are based on unified agent service capability identifiers to identify the same type of agent service capabilities.

Based on the unified identifiers of service agents, the initial registration and dynamic update of agent service capabilities need to be carried out. The service agent initiates a service capability registration request to the capability exposure platform, and the capability exposure platform sends an information template required for registration. According to the unified template, when registering service capabilities, the service agent needs to carry the agent identifier, IP address, service capability list, service capability binding information, and agent service capability dependency relationships, among which:

• Agent service capability binding information: This information is used to distinguish whether the agent service capability is an exclusive agent service capability or a shared agent service capability. It can be registered in the capability exposure platform. For example, if it is an exclusive service capability, its current binding relationship needs to be registered with no need to maintain its performance information, and the duration of the exclusive service capability can be further registered; if it is a shared service capability, its performance information needs to be main-

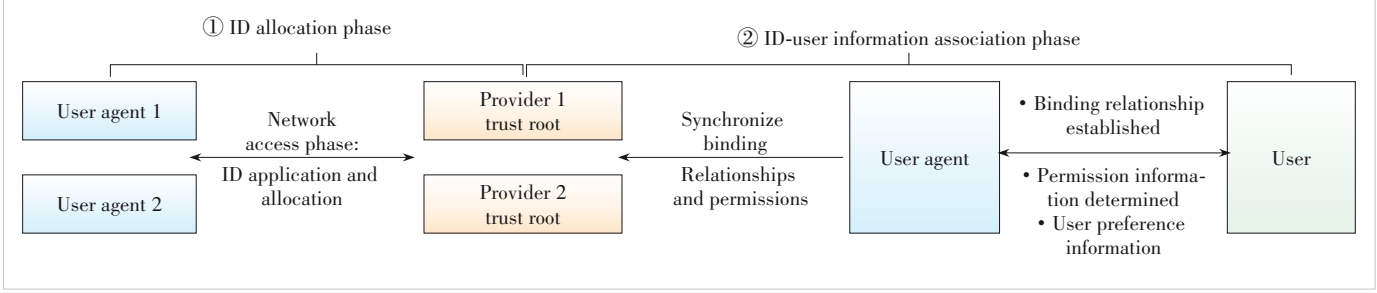

Figure 3. Registration diagram of user agents

tained in the later stage.

• Agent service capability dependency relationships: These are dependency relationships between some agent service capabilities, that is, the current service capability must cooperate with specific service capabilities to complete specific services. The dependency relationships need to be registered in the unified capability platform for reference when querying and discovering agent service capabilities.

As for the agent service capability identifier allocation, a standardized method for the identifier needs to be adopted. For different types of agent service capabilities, different metrics may be needed. The name space of the identifier employs a hierarchical, globally unique, and semantically extensible structured design compliant with unified agent capability identifier specifications, which explicitly characterizes the domain, capability type, interface version, and security attributes of agent service capabilities. Furthermore, a distributed trusted registration and resolution mechanism can be adopted to achieve dynamic identifier allocation, conflict detection, access control, and version evolution.

The capability exposure platform completes the agent service capability registration after verifying the identification information of the service agent. The above process can be realized by extending the interaction based on existing protocols, such as via the RESTful interface over HTTP, or by defining a new service capability registration and update protocol to implement processes such as service capability addition, update, and deletion. The capability exposure platform further generates a service capability information database based on the dimension of agent service capabilities. However, agent service capabilities may be added, deleted, or experience performance changes, so the service capability information needs to be updated, which can be realized through periodic, trigger-based, or active subscription methods. In a specific implementation, the network side can consider the popularity of the service capability demand on the request side and subscribe to updates for service capabilities with high popularity; or consider whether there is additional payment, and for paid agent capabilities, subscribe to update information.

• Periodic update: It only synchronizes the information of newly added/deleted agent service capabilities, and regularly synchronizes with the capability management platform.

• Trigger-based update: It only synchronizes the information of newly added/deleted agent service capabilities, or actively updates to the management platform when their service capability binding relationships change.

• Subscription-based update: The capability management platform can subscribe to the target service capability update information from the agent, and specify to receive active updates of agent service capabilities of specific types.

5 Control Protocol for the Internet of Agents

5.1 Agent Service Discovery Protocol

How to quickly and accurately discover suitable agents for task collaboration is an important issue in IoA. For the discovery of agent service capabilities, two technical routes have emerged currently: based on existing Internet infrastructure or building a new agent discovery mechanism.

Route 1: Build a new mechanism to obtain the agent identifier and agent IP, respectively, which can flexibly discover the required agent service capabilities without modifying the current infrastructure.

Route 2: Reuse the Domain Name System (DNS)[8] capability, initiate an "agent domain name request" to directly obtain the Agent IP, reuse the dynamic resolution and hierarchical mechanism of DNS, and require little modification on the user side; however, it needs to rely on DNS for ultra-large concurrent dynamic service discovery, as shown in Fig. 4.

As for Route 2, when a task request is initiated to the agent proxy, the proxy decomposes the instruction into multiple subservices and needs to connect to the corresponding service agents. The agent needs to initiate a DNS request based on the unified agent service capability identifier to achieve dynamic mapping from it to the IP address. This involves enhancing multiple functionalities, including the unified agent service capability identifier, adding agent service query and response packets, a hierarchical DNS mechanism, and dynamic resolution mapping.

Unified agent service "domain name": The unified agent service "domain name" could design identifiers similar to "domain name" to identify different agent service capabilities. Compatibility with the current system needs to be considered, and one alternative method could be considered: adding new information resource records based on the agent service "domain name" to store the mapping relationship between the agent service "domain name" and IP addresses. In this case, as shown in Fig. 5, a new fully qualified domain name (FQDN) can be defined, for example, based on a URL; a new type is defined to identify this as an agent service re-

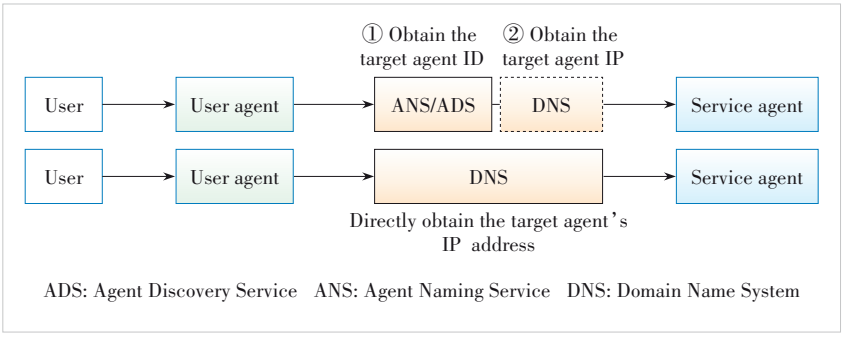

Figure 4. Two routes of agent service discovery

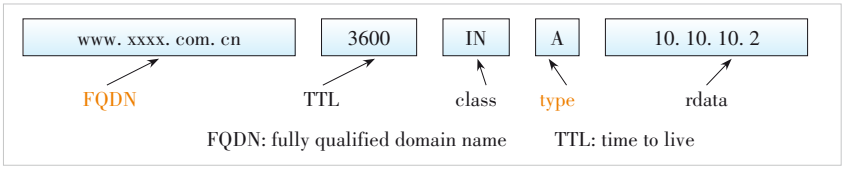

Figure 5. Example of the format of DNS domain name resource record

cord; the class defines whether the agent service capability identifier is for the local area or the Internet. The DNS reads the FQDN of the query packet and the packet type to determine whether it needs to query the information resource record of the corresponding agent service capability identifier.

Agent service query and response packets: New DNS query packet and response packet types can be introduced to identify the DNS request initiated by the agent, and to consider a DNS-side authentication mechanism that determines whether the current agent has permission to initiate a query. One alternative method is shown below:

• Identify the DNS message type, which can carry information in the Opcode field or Z field of the DNS message header, as shown in Table 2.

• When sending updates, send one or more update messages to the local DNS, with QCOUNT=0 (the number of questions) and ACOUNT (the number of answers) ⩾ 1; multiple agent service query information can be carried.

Hierarchical DNS mechanism: To improve response and resolution speed, agent service "domain name" resolution can also be carried out hierarchically and regionally, supporting the local DNS to store mapping information of nearby agent service capability identifiers, ensuring the stability and scalability of domain name resolution. The freshness and hit rate of locally cached mapping information need to be optimized using the AI/machine learning (ML) methods.

Dynamic resolution mapping: The DNS reads whether the query packet carries additional demand information, such as agent selection preferences (e. g., requesting an agent from the same agent provider), and whether there are capability dependency relationships to define priorities. In addition, agent capability performance information can be maintained, and selection can be made by combining demand information, geographical location, preference requirements, etc. The demand information translation and the mapping process need to consider multi-dimensional information, which may require the integration of AI and ML into these functionalities to improve accuracy and resolution performance.

The Agent Service Discovery Protocol adopts a centralized or DNS-like architecture for identifier-to-location resolution, emphasizing unified control and good compatibility, making it suitable for intra-domain, controllable and relatively static agent discovery.

Table 2. Format of DNS message header

Field	Description
QR (1 bit)	Query/response flag; 0 for query, 1 for response
Opcode (4 bit)	Operation code; 0 for standard query, 1 for reverse query, 2 for server status request; others are reserved bits
AA, TC, RD, RA	Refer to RFC 1035[14] for definitions
Z	Reserved bits
QCOUNT	Number of questions
ACOUNT	Number of answers

AA: authoritative answer flag RA: recursion available flag TC: truncated flag
DNS: Domain Name System RD: recursion desired flag

5.2 Agent Service Sensing Protocol

Considering that different users may have different requirements for the same type of agent service capability (e.g., recognition accuracy, response delay, preference for agents with a higher number of service occurrences, and agents with good service records), it is necessary to obtain the performance information of agent service capabilities to achieve differentiated matching.

Different grading dimensions of agent service capabilities can be determined according to different types of capabilities, such as capability response delay, capability response accuracy, maximum concurrent service number, capability calling cost, performance stability, and historical performance level. First, according to the agent service capability binding information, the list of currently available shared agent service capabilities is screened.

Two sensing methods, active performance reporting or on-demand pulling, can be supported, which can be realized through the RESTful interface with HTTP, etc.

When a capability actively reports performance information to the capability control platform, it can be registered at the new capability registration stage. Three methods of obtaining capability performance information can be supported: periodic, trigger-based, or active subscription.

• For the periodic acquisition method, different cycles can be set according to the service type: a longer cycle is set for long-term service agents, and a shorter cycle is set for short-term service agents.

• The trigger-based update method triggers an update when the agent capability change reaches a certain threshold. Different thresholds can be set according to the service type, such as a lower threshold for performance-sensitive agents and a higher threshold for performance-insensitive agents.

• The subscription-based update method allows the capability control platform to subscribe to target capability update information from the agent and specify to receive timely updates of the agent capabilities of specific capability types.

To save the maintenance cost of performance information,

the capability control platform can first maintain the capability list information. When a user requests a certain agent service capability, the platform can be triggered to actively initiate a capability performance query request to the corresponding agent service capability list. After receiving the performance information fed back by all capabilities in the capability list, further screening is carried out in combination with user requirements. Alternatively, the user's performance requirements are first extracted to generate a capability list, and after the capabilities in the capability list receive the demand information, they locally judge whether they meet the demand. If so, they return the specific capability performance information to the capability control platform.

In practice, agent service sensing can be performed by agent service providers with their own controllers and the required status information is then synchronized to the controller of the IoA. In this case, the interfaces between the controller of the IoA and other agent service providers are essential.

5.3 Agent Task-Driven Networking Protocol

Considering that when the performance of an agent deteriorates or a fault occurs, triggering a service agent to switch to a new agent requires the agent task proxy to initiate a new agent service discovery process and to carry additional information, such as the address information of the peer agent for collaboration. Before and after the switch, the agents perform the required context synchronization using task group identification information, service capability identifier, etc.

Therefore, during the agent task execution process, it is necessary to introduce an agent task status monitoring protocol to monitor the current agent's session connection status (including network status) with other agents, as well as the status of each agent, such as service response delay and uplink and downlink rates. When the network status information or agent status information deteriorates to a preset threshold, the switch process is triggered. The agent proxy then re-initiates a service discovery request and additionally carries demand information, which includes the target agent capability identification information, address information (which can carry the address information of the peer agent interconnected with the agent), etc.

The agent proxy receives the new agent information and synchronizes it to the relevant interconnected agents, re-establishing a session connection for task collaboration. When the agent service discovery function receives a new service discovery request, it re-selects a new agent in combination with the additional demand information, which could be carried via the extension of the DNS packet as shown in Fig. 6. To ensure the service continuity, the agent proxy issues an application synchronization instruction to the two agents. The synchronization instruction includes: the identifier of the other agent, task identifier, and agent service capability identifier. The old and new agents then negotiate and confirm the information to be synchronized, including the task identifier and agent service capability identifier. After verification, the information synchronization starts.

Furthermore, to improve the service efficiency of agents during multi-agent collaboration, if an agent completes the task it is responsible for, it can flexibly exit the task group in advance. To ensure the performance of the original task after such an early exit, the agent can add a new intermediate state, which is a transition state. In this state, the current agent can accept new task requests while still retaining the relevant information and context of the original task. The scheduling priority of agents in the transition state is higher than that of occupied agents and lower than that of idle agents. This can realize more efficient scheduling and utilization of each agent, make full use of the capabilities of each agent, and allows more users to be served.

6 Routing Protocol for the Internet of Agents

6.1 Agent Semantic Routing Protocol

Considering the agent service capability discovery mechanism, agent routing can be based on traditional routing systems, supporting flexible task group construction and multiple routing mechanisms such as unicast, anycast, and multicast. By supporting the agent routing protocol based on the agent service capability identifier, service discovery based on agent capabilities can be realized. A candidate list of agents that can provide services is searched through the agent service capability identifier, and the target agent is then determined in a distributed manner according to other dimensions.

By extending the computing-aware traffic steering routing protocol[15], a capability-aware agent routing mechanism can be realized. By advertising the supported capability information in the network, the anycast mechanism based on the

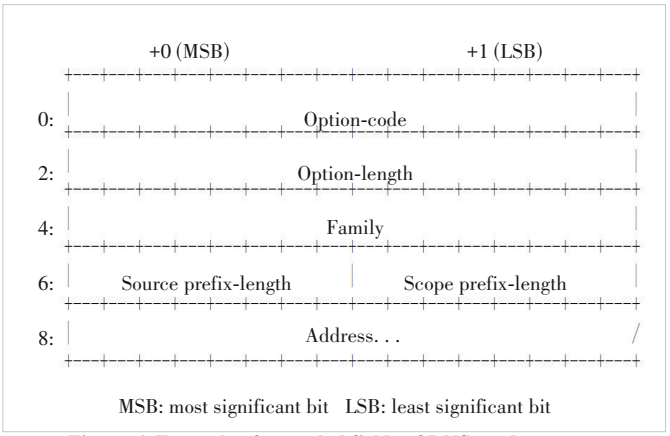

Figure 6. Example of extended fields of DNS used to carry demand information

agent service capability identifier could be realized. The agent proxy only needs to carry the target agent service capability identifier, and the entry gateway selects the optimal capability for it by combining location information, capability information, demand information, etc., which can ensure real-time and dynamic distributed capability discovery.

In addition, the agent routing protocol needs to support flexible task-oriented networking and construct a new task-based routing table. A collective communication group can be created based on the published task requirements to complete agent team formation, and further adaptively complete the task tree construction. According to the task tree, task release operator functions such as multicast broadcast and scatter are performed, and task execution results are summarized based on operators such as "gather" or "reduce" to achieve distributed task collaboration.

The Agent Semantic Routing Protocol is used in distributed agent collaboration scenarios, enabling related agents to discover target capabilities within a limited controlled scope. It performs capability-based routing directly via agent service capability identifiers in a decentralized manner, with high scalability and dynamics, and is more applicable to open, multi-domain, large-scale and dynamic agent interconnection scenarios.

6.2 Agent Cross-Domain Traffic Assurance Protocol

The agent wide-area Internet will bring a new traffic pattern. Different from the traditional client-server (C-S) communication traffic pattern, in the agent-centered traffic pattern, communication between agents may trigger call traffic between agents and tool libraries, introducing implicit traffic[13]. This may evolve into multi-segment traffic between agents, as well as between agents and tool libraries, among others. Therefore, the routing protocol needs to optimize such multi-segment traffic between agents, between agents and tool libraries, etc., in the same agent task to ensure unified service level agreement (SLA) assurance and to prevent the performance of any single traffic segment from affecting the overall task performance.

For the same service agent, when it interacts with different service agents, there may be different SLA assurance requirements. Therefore, it is necessary to adjust the SLA requirements for interacting with the peer agent and the corresponding tools according to the performance requirements of the different tasks in which the agent is involved. The SRv6 policy can be extended. The SRv6 controller identifies different segment sessions based on the network policy, performs session SLA association, and maps them to different policies. When the current agent interacts with other agents, if agent-invoked API traffic is introduced, the new session (identified by the agent-agent session) is maintained with the same priority as the current session. This can be identified by a triplet: the policy is uniquely identified by <Head end, Color, Endpoint>. The Endpoint can represent multiple destinations sharing the same policy group, supporting cross-domain traffic scheduling and ensuring that multi-segment sessions run under the same policy framework.

In addition, a single agent may establish connections with multiple agents at the same time to conduct sessions. Therefore, there is resource competition between multiple sessions of a single agent, including competition for network, computing and storage resources. It is necessary to clarify the priority of sessions and conduct resource planning for a single agent. Different types of agent sessions will correspond to different types of traffic. For short-term high-frequency traffic and long-term low-frequency large-data traffic, agents can negotiate different communication protocols. Therefore, a single agent needs to support the ability to adapt to different protocols and to support the simultaneous operation of different protocols.

7 Security Considerations

The introduction of agents brings new security and privacy challenges, including fake agent capability registration, discovery poisoning, performance signal spoofing, cross-domain trust issues, malicious identity spoofing, unauthorized service invocation, semantic routing hijacking, privacy leakage during capability discovery, and model or data attacks. These threats severely hinder the compliance of service calls and privacy protection of data interactions, which are crucial for building an open collaborative ecosystem. To address these issues, future research should focus on several key directions: developing trust evaluation and dynamic access control mechanisms for agent capabilities, designing decentralized cross-domain authentication and authorization systems, exploring privacy-preserving capability discovery and semantic routing technologies, establishing threat detection methods based on agent behavior analysis, constructing trusted execution environments for verifiable service invocation, and promoting the security standardization of capability identifier and namespace management.

8 Conclusions

The rise of agent technology marks a profound shift in Internet service models and communication paradigms, promoting the evolution of networks from "connecting information" to "connecting intelligence". Aiming at the management, control, and routing challenges faced by the IoA, this paper proposes a protocol framework for IoA. Through mechanisms such as unified identification registration, dynamic service discovery, and semantically aware routing, it provides a systematic protocol solution for efficient, secure, and scalable collaboration among multiple agents from different agent service providers and across multiple domains. In the future, with the continuous expansion of agent application scenarios, the actual deployment of the protocol still needs further ex-

ploration and verification in terms of security, real-time performance, and resource scheduling optimization.

Future work will focus on the detailed implementation of the IoA architecture, protocol performance simulation, and cross-domain experiments, aiming to advance the IoA from theoretical design to practical application and lay a foundation for the development of the next-generation intelligent network.

References

[1] Cheng Y H, Zhang C Y, Zhang Z W, et al. Exploring large language model based intelligent agents: definitions, methods, and prospects [PP/OL]. V1. arXiv (2024-01-07) [2026-01-10]. https://doi. org/10.48550/arXiv.2401.03428
[2] Yang Y X, Chai H C, Song Y Y, et al. A survey of AI agent protocols [PP/OL]. V1. arXiv (2025-04-23) [2026-01-10]. https://doi. org/10.48550/arXiv.2504.16736
[3] Chen W Z, You Z M, Li R, et al. Internet of agents: weaving a web of heterogeneous agents for collaborative intelligence [PP/OL]. V2. arXiv (2024-07-10) [2026-01-10]. https://doi.org/10.48550/arXiv.2407.07061
[4] Introducing the Model Context Protocol [EB/OL]. (2024-11-25) [2026-01-10]. https://www.anthropic.com/news/model-context-protocol
[5] Agent network protocol technical white paper [R/OL]. 2025-07-24. https://agent-network-protocol.com/specs/white-paper.html
[6] Agent2Agent (A2A) Protocol [EB/OL]. [2026-01-10]. https://google.github.io/A2A
[7] Agora [EB/OL]. [2026-01-10]. https://www.agora.io/en
[8] IETF RFC 8484. 2018 DNS queries over HTTPS (DoH) [S]
[9] IETF RFC 4271. 2006 A border gateway protocol 4 (BGP-4) [S]
[10] IETF RFC 2328. 1998 OSPF version 2 [S]
[11] Cisco. The internet of agents white paper [R/OL]. 2025: 1 – 14. https://outshift.cisco.com/the-internet-of-agents
[12] Liu J, Yu K, Chen K L, et al. ACPs: agent collaboration protocols for the internet of agents [PP/OL]. V1. arXiv (2025-05-18) [2026-01-10]. https://doi.org/10.48550/arXiv.2505.13523
[13] Duan X D, Sun T, Lu L, et al. Internet of agents: conception, architecture and key technologies [J]. Telecommunications science, 2025, 41 (10): 1 – 10. DOI: 10.11959/j.issn.1000-0801.2025221
[14] IETF RFC 1035. 1987. Domain names - implementation and specification [S]
[15] IETF. Official website of the IETF computing-aware traffic steering (cats) working group [EB/OL]. [2026-01-10]. https://datatracker.ietf.org/wg/cats

Biographies

Fu Yuexia is a researcher at the China Mobile Research Institute and serves as the Vice Chairman of FG AINN in ITU-T SG13. Her main research interests include computing force networks and new technologies of future networks.

Liu Peng is a senior engineer of the China Mobile Research Institute. His main research interests include next-generation IP technologies, computing-aware traffic steering routing, and new technologies and applications of integrated computing and networking.

Lu Lu (lulu@chinamobile.com) is the Deputy Director of the Department of Basic Network Technology at the China Mobile Research Institute, a leader of the core network group of CCSA TC5, and a Vice Chairman of ITU-T SG13. Her research interests include 5G-A/6G network architecture and computing force networks.

Duan Xiaodong is the Vice President of the China Mobile Research Institute and a leader of the network technology group of IMT-2030 (6G). His research interests include 5G/6G architecture, computing force networks, and new IP technologies.