Internet Engineering Task Force H. Yu
Internet-Draft China Internet Network Information Center
Intended status: Informational 2 June 2026
Expires: 4 December 2026

IPv6 Networking Considerations for AI Agent Communication
draft-yu-ai-agent-ipv6-networking-considerations-01

Abstract

AI agents are increasingly expected to communicate across platforms, organizations, clouds, edge environments, and administrative domains. Current agent-related mechanisms mainly focus on description, discovery, identity, authentication, authorization, tool invocation, and application-layer collaboration. These mechanisms are important, but they often treat the IP network as a transparent connectivity substrate.

This document describes networking problems that arise when AI agents perform cross-domain communication and continuous tool, API, data, and agent-to-agent calls. It focuses on problem description rather than a protocol solution. In particular, it discusses gaps related to agent identity visibility at the network layer, path control, audit continuity, the separation of discovery from addressing and authorization, and privacy and privilege risks introduced by highly autonomous agents.

The document positions Agent6 as a problem space for network support of cross-domain AI agent collaboration. It does not define a new agent discovery protocol, a new application-layer collaboration protocol, a new authentication or authorization mechanism, or a new IPv6 extension header.

This Internet-Draft will expire on 4 December 2026.

Table of Contents

1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Introduction

AI agents are moving from isolated application components toward large-scale, cross-domain collaboration. An agent may plan a task, invoke tools, access data, call APIs, communicate with other agents, and act on behalf of a user, organization, or system. As these behaviors become more continuous and autonomous, agent communication becomes not only an application-layer issue, but also an Internet infrastructure issue.

Recent discussions in the Internet operations community have raised a broader question: whether the Internet infrastructure is ready for the next phase of AI. A key observation from these discussions is that the main pressure is shifting from centralized model training toward continuous inference and agent operation. Future traffic may be shaped by high-frequency small flows, chained API calls, background agent access, local inference, local policy enforcement, local audit, and edge-side coordination.

AI infrastructure also includes compute, power, data centers, data placement, scheduling systems, and governance. However, the network remains a necessary substrate for agent interaction. If agents become a common way to access services, tools, data, and other agents, the Internet will increasingly carry communication that is initiated, chained, and adapted by software agents rather than directly by human users through browsers or applications.

In this context, Agent6 is not intended to be a closed new protocol or a replacement for existing agent frameworks. It is better understood as a problem space for the network support layer required by cross-domain AI agent collaboration. The central question is how the network can remain observable, governable, and interoperable when agent communication becomes more autonomous, distributed, and cross-domain.

3. Scope and Non-Goals

This document focuses on problem description for AI agent networking. It discusses why existing application-layer mechanisms, by themselves, may be insufficient to provide network-level identity visibility, path accountability, audit continuity, discovery-to-reachability association, and privacy-aware governance.

This document does not define:

* a new agent discovery protocol;

* a new application-layer agent collaboration protocol;

* a new authentication, authorization, or delegation protocol;

* a new replacement for DNS, HTTPS, OAuth, OIDC, DID, WIMSE, RATS, SCITT, A2A, MCP, RDAP, or audit frameworks;

* a general-purpose semantic routing protocol;

* a requirement that ordinary Internet routers understand agent semantics, prompts, tasks, or application payloads;

* a new IPv6 extension header, SRv6 behavior, or packet encoding.

4. Terminology

Agent:
: A software entity that can perceive context, reason, plan, invoke tools, communicate with other agents, and perform tasks on behalf of a user, organization, or system.

Agent Identifier:
: A stable identifier used to identify an agent independently from its current network location.

Agent Gateway:
: A gateway or mediation point that provides controlled access to one or more agents, agent services, tools, APIs, or related policy functions.

Agent Domain:
: An administrative or operational domain in which agents, gateways, registries, policy systems, audit systems, or network functions are managed under a common authority.

Agent Communication:
: Communication initiated by, received by, or mediated for an agent, including tool calls, API calls, data access, agent-to-agent calls, and gateway-mediated interactions.

5. Background: From Training Networks to Agent Operation

Early AI infrastructure discussions often emphasized the network requirements of large-scale model training. Training workloads may involve very large data transfers inside data centers or between training clusters. Agent operation introduces a different traffic pattern. It may involve many small, frequent, chained, and automatically generated requests across tools, APIs, data sources, gateways, and other agents.

This shift changes the operational problem. Traffic volume remains relevant, but flow shape, locality, observability, access control, auditability, and policy correlation become equally important. Agent traffic may be generated in the background, may traverse multiple administrative domains, and may depend on intermediate gateways or registries that are not visible in traditional endpoint-based measurements.

Locality is also becoming more important. Local agent gateways, local inference, local audit, local policy execution, and local data handling may reduce latency, reduce cross-domain exposure, and support jurisdictional or enterprise controls. At the same time, excessive localization or incompatible agent-specific infrastructures could increase fragmentation if there is no interoperable Internet-facing model.

These observations suggest that the key issue is not whether a single new protocol can solve all agent communication problems. The issue is whether the existing Internet protocol stack, identity systems, registries, gateways, audit systems, and network policy mechanisms can be coherently related when autonomous agents communicate across domains.

6. Implications for Agent6

The above background leads to several implications for the Agent6 problem space.

First, the focus needs to move from training networks alone to inference and agent collaboration networks. Future networks may need to support not only large flows between model training facilities, but also agent-to-agent interactions, tool calls, API access, and cross-domain reachability.

Second, the discussion needs to move from total traffic growth alone to traffic shape change. Agent traffic may consist of large numbers of small flows, high-frequency calls, chained invocations, and automated background access. This makes measurement, observability, and policy recognition more difficult.

Third, the architecture needs to account for local inference and Agent Gateways. Not all agent collaboration will return to a remote central cloud. Some interactions will depend on local exchange, local policy execution, local audit, local data controls, and edge-side inference.

Fourth, the problem is a multi-technology governance problem rather than a single-protocol design problem. DNS, HTTPS, OAuth, WIMSE, RDAP, A2A, MCP, DAWN, AUDIT, SCITT, RATS, IPv6, SRv6, and related mechanisms may all appear in different parts of the agent interaction lifecycle.

Fifth, Agent6 should be considered in terms of open interoperability. A closed agent networking system could create new agent islands and contribute to Internet fragmentation. The problem space therefore needs to consider distributed deployment, cross-domain operation, and fallback to existing Internet mechanisms.

Agent6 should therefore not be framed as an AI use case invented to promote IPv6. It is more accurately framed as the network support layer problem created by cross-domain AI agent collaboration.

7. Current Agent Access and Invocation Flow

Current agent access or invocation is usually not a single network transaction. It is a compound process that involves application entry points, discovery mechanisms, registries, credentials, gateways, network connections, logs, and later investigation.

A simplified flow is as follows:

1. A user, application, browser, mobile application, or agent client identifies a task objective or service need.

2. The client finds a stable entry point using mechanisms such as DNS, DNSSEC, SVCB or HTTPS resource records, or well-known locations.

3. The client reads an agent description, such as an A2A Agent Card, ANP Agent Description, MCP server metadata, or a similar description object.

4. The client or platform queries a registry or identity registration system, such as an MCP Registry, Agent Registry, DID Registry, or enterprise directory.

5. The client obtains runtime credentials and authorization material through mechanisms such as OAuth 2.0, OIDC, WIMSE, SPIFFE or SPIRE, JWT, or mutual TLS.

6. The client selects a reachable instance, Agent Gateway, Discovery Service, Service Mesh endpoint, or API Gateway.

7. The client establishes network connectivity using IP, TLS, QUIC, HTTP, gateway mediation, service mesh mechanisms, or other transport and network mechanisms.

8. The involved systems generate logs, telemetry, traces, or audit records.

9. If a dispute, incident, or compliance review occurs, operators attempt to reconstruct what was called, by whom, through which path, under which policy, and with what result.

This flow shows that agent access is assembled from multiple protocols and platform mechanisms. Discovery, description, registry lookup, credential issuance, gateway selection, network reachability, and audit are often handled by separate systems. The networking problem appears most clearly at the point where an authorized and discovered agent interaction becomes actual network traffic.

8. Core Problems in Current Agent Interconnection

Based on the flow described above, this document identifies five core problems in current mainstream agent interconnection.

8.1. Lack of Agent Identity Visibility at the Network Layer

Current identity and authentication mechanisms mainly operate at the application layer. Examples include accounts, API keys, OAuth tokens, OIDC assertions, JWTs, workload identities, and TLS certificates. These mechanisms can authenticate a user, workload, service, client, or session, but the network commonly observes only IP addresses, ports, transport protocols, and flow behavior.

As a result, the network often cannot distinguish the agent subject behind a connection, the organization responsible for that agent, the type of task being performed, or the trust level associated with the interaction. A gateway or application may know this information, but ordinary network elements and cross-domain operators usually do not have a consistent way to correlate it with the flow.

This creates a separation between identity authentication and network operation. A connection may be authenticated at an application endpoint while path control, network policy enforcement, traffic measurement, and audit reconstruction remain based on locator-level information. For agent communication, this separation becomes more significant because a single agent may perform many delegated actions across many services and domains.

8.2. Limited Control over IP Paths

Traditional IP forwarding is primarily based on destination reachability and dynamic routing. This model is efficient and scalable, but it is not naturally organized around agent task levels, data sensitivity, latency objectives, safety requirements, audit requirements, or compliance constraints.

In cross-cloud, cross-network, and edge scenarios, a high-risk or high-value agent task may need assurance that traffic has traversed expected control points, such as authenticated gateways, audit collection systems, data-loss prevention systems, content filtering systems, security inspection systems, or compliance boundaries. In current deployments, this assurance is often provided by overlay gateways, service meshes, private interconnects, or application policy systems, each with its own operational boundary.

The resulting path behavior may be difficult to express, verify, or compare across domains. An application may know the intended task class, while the network may only know where packets are going. A network may know the selected route, while the application may not know whether the route satisfied a task-specific policy expectation.

8.3. Incomplete Audit Chains

Agent invocation can generate records in many places, including application logs, model platform logs, gateway logs, identity system logs, service mesh logs, tool logs, API provider logs, and network telemetry systems. These records often use different formats, identifiers, retention policies, timestamps, and trust assumptions.

The audit chain is therefore frequently discontinuous. A task may start in one domain, call a tool in another, trigger an agent in a third domain, and access data through a gateway in a fourth domain. Each participant may see only a local fragment of the interaction. Cross-domain visibility is limited, and the correlation between application-level events and network-level flow evidence is often weak.

This problem becomes more serious for high-risk agent tasks. If a network path cannot require traversal of an audit point, or if the audit point cannot correlate a flow with the relevant agent session, then later reconstruction may be incomplete. Determining the actual path, responsible subject, delegated authority, policy applied, and tool behavior may require manual correlation across independent systems.

8.4. Separation of Registration, Discovery, Addressing, and Authorization

Agent ecosystems use multiple mechanisms for registration, discovery, description, naming, identity, authorization, and capability publication. DNS, DID systems, certificates, API gateways, service registries, capability directories, agent cards, MCP metadata, enterprise catalogs, and internal access-control systems may all participate in the same interaction.

These mechanisms answer different questions. One mechanism may describe what an agent can do. Another may identify who operates it. Another may provide a network endpoint. Another may issue runtime credentials. Another may decide whether a caller is allowed to invoke a specific capability. Another may select a gateway or reachable instance.

Because these questions are answered by separate systems, an agent that has discovered "what can be done" still needs to determine whether the target is trustworthy, where it should be reached, which path or gateway should be used, and whether the invocation is authorized for the current caller, task, data, and context. The lack of consistent linkage among registration, discovery, addressing, and authorization creates operational ambiguity.

8.5. Privacy and Privilege Risks from Highly Autonomous Agents

Highly autonomous agents may plan multiple steps, execute continuously, invoke tools, access data, and adapt their behavior without a human approving each individual network interaction. If such an agent has broad privileges, it can create risks of privacy leakage, unauthorized access, mistaken execution, excessive data retrieval, unintended delegation, or uncontrolled behavior.

The network and platform environment may have difficulty distinguishing a normal chain of delegated agent actions from an excessive or unintended chain of actions. Traditional access-control decisions may be made per API, per token, or per service, while the overall agent task spans many calls and may accumulate risk over time.

At the same time, governance mechanisms that rely on full content inspection, prompt capture, or complete logging of every interaction can themselves increase privacy exposure. Recording all prompts, tool inputs, tool outputs, user data, and intermediate reasoning may create new sensitive data stores. The governance problem is therefore not only how to detect misuse, but also how to avoid expanding the privacy attack surface while attempting to govern agent behavior.

9. Summary of the Agent6 Problem Space

The five problems above can be summarized as five missing or insufficiently connected capabilities in current agent interconnection.

* Agent identity is not sufficiently recognizable from the network perspective.

* Agent communication paths are not sufficiently expressible or verifiable according to task, data, security, latency, audit, or compliance constraints.

* Agent invocation audit chains are not sufficiently continuous across application, gateway, platform, tool, and network layers.

* Agent registration, discovery, addressing, and authorization are not sufficiently associated across mechanisms and domains.

* Agent governance does not yet sufficiently balance autonomy, privilege control, auditability, and privacy minimization.

In short, Agent6 is concerned with five problem areas: recognizable identity, accountable paths, auditable invocation, associated discovery and addressing, and governance that accounts for privacy. This document intentionally stops at the problem description level. It does not specify a protocol design or deployment architecture.

10. Security Considerations

The problems described in this document are security-relevant because agent communication can involve delegated authority, cross-domain access, tool invocation, and data retrieval. Weak linkage between agent identity, network flows, authorization decisions, and audit evidence can make impersonation, misuse, excessive privilege, and post-incident reconstruction more difficult to manage.

This document does not introduce a new protocol mechanism and therefore does not introduce protocol-specific security requirements. Any future mechanism in this problem space would need to consider authentication, authorization, replay protection, downgrade risks, gateway trust, cross-domain policy consistency, telemetry integrity, and the risk that metadata itself can become sensitive.

11. Privacy Considerations

Agent communication can reveal sensitive information about users, organizations, tasks, data sources, tools, models, policies, and business relationships. Privacy risk may arise both from agent actions and from the records created to govern those actions.

The problem space therefore includes a tension between observability and minimization. More complete logs, traces, metadata, or inspection may improve governance, but may also create additional exposure. Any future work in this area would need to avoid assuming that full content capture or universal visibility is an acceptable default.

12. IANA Considerations

This document does not request any IANA action.

13. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

14. Informative References

[I-D.pioli-agent-discovery] Pioli, L., "Agent Registration and Discovery Protocol", <https://datatracker.ietf.org/doc/draft-pioli-agent-discovery/>.

[Agent2Agent-Archive] IETF, "IETF Agent2Agent Mailing List Archive", <https://mailarchive.ietf.org/arch/browse/agent2agent/>.

Author’s Address

Haisheng Yu
China Internet Network Information Center
Email: yuhaisheng@cnnic.cn