Empowering 6G with Agent Protocols

Requirements, capabilities, and enablers

Huawei Technologies Canada

IETF 123 - Madrid
July 2025

Background

- 6G unlocks digital transformation with superior connectivity and provides built-in features such native AI and ISAC that pave the way for new revenue opportunities.

- Emergence of new use cases increases bring about competing requirements and increase 6G design complexities
  - Expanding Telco services scope while increasing operational efficiency.
  - Per user network customisation while decreasing service provisioning complexity.
  - Rapid service introduction and delivery and maintaining interoperability through standardisations.

- Agentic AI is a key technology for automation and streamlining operations,
- Empowering smarter and customised networks, and providing personalised user experience.

- ETSI & 3GPP are actively researching AI-powered telecom systems.

Characteristics of mobile network

- **Resource dynamicity**
  - Handling resource fluctuations and link unreliability through adaptive scheduling to ensure service continuity.
- **Guaranteed QoS**
  - Ensuring support for heterogeneous application e.g. voice, video, URLLC, etc.
- **Mobility**
  - Seamless handover to guarantee service continuity.
- **Multi-domain network interoperation**
  - Enables infrastructure sharing and coordinated service delivery (e.g. QoE in roaming).
  - Paves the way for advanced 6G services such as AI, sensing, and computing while enhancing performance and scalability.
- **Security, privacy and trust**
- **Session-level operation**
  - Bridges end-user's different flow QoS requirements with network resource optimisation
- **Multi-Release coexistence support**

Agent Protocols Requirements

- **Agent type agnostic covering agents at terminal, network and service level**
  - 6G must also be able to utilise agent protocols to enable interactions and collaborations between
    - Agents with the 6G network (6G internal agents)
    - UE and network agents
    - 3rd party and 6G network agents

- **QoS assurance for collaborative Multi-Agent communication**
  - Multi-modal, multi-part traffic needs tailored QoS handling.
  - QoS varies across and within messages/artifacts.
  - Burst traffic behaviour demands real-time resource adjustment.

- **Agent Service Continuity**
  - Agent service continuity must be maintained, especially during mobility events or network changes.
  - Agent mobility requires safe transfer and support for wake-up procedures.
  - Failure scenarios (e.g. device shutdown mid-task) require mechanisms to relocate agents and ensure task handover.

Agent Protocols Requirements

- ## Agent task-oriented resource management
  - Agent interactions are task-driven, with one agent requesting another to perform and report on tasks.
  - Mobile networks rely on session-based resource and billing management, traditionally tied to user equipment (UE).
  - In a multi-agent system, resources must be scheduled per task, not per UE, to meet QoS needs.
  - Task sessions are required, where each message carries task identifiers for proper resource coordination.

- ## Intent-based Interaction
  - AI agents autonomously perform tasks, using reasoning and decision-making to achieve goals.
  - For given goals, agents plan execution themselves via local AI models and tools.
  - Agent interactions should therefore be intent-driven.

- ## Security, Privacy and Trust
  - Use of followings should be investigated:
    - **Security control flow** enables secure access, transmission, and cross-domain collaboration.
    - **Cryptographic algorithms** enforce secure communication and policy-based protection levels.
    - **User consent & monitoring capabilities** keep subscribers informed about agreed data usage.

Agent Protocols Requirements

- **Agent Governance**
  - AI agents need efficient discovery and matchmaking across vendors and capabilities to collaborate effectively.
  - Agent protocols must support registration, updates, de-registration, and discovery via centralised or distributed authority.
  - Protocols must ensure trust, behaviour compliance, and accurate capability publishing to enable reliable collaboration.
  - Discovery mechanisms like publish-subscribe and request-response are essential.
  - Wake-up signaling for agent-hosting equipment supports service continuity.
- **Backward Compatibility and Forward Compatible**
  - Networks are diverse, with varied stakeholders, device types, and protocol versions.
  - Agent protocols must be adaptable, supporting backward/forward compatibility and future self-negotiation.
  - Design should build on capabilities of existing protocols while staying flexible for future network evolution.
- **Extensibility**
  - Agent protocols should support agents in different domains (UE, RAN, CN, DN) and networks.
  - Protocols should enable seamless and scalable integration of variety of agents across a wide range of systems and device types.

Agent Protocols Requirements

- **Flexibility**
  - AI agents can evolve over time, changing their skills, capabilities, and I/O modalities.
  - Agent protocols must be highly flexible, supporting rapid adaptation.
  - Data objects and profiles should use scalable, easily modifiable fields to avoid protocol redesign.

- **Isolation**
  - Mobile network messages vary in security and delay requirements (e.g. UE data requests demand stricter security).
  - Agent messages should be decoupled and isolated to support tailored, performance-based processing.

- **Tool invocation**
  - Agents autonomously execute tasks by invoking tools such as APIs, models, and interpreters.
  - Tool descriptions must be clear and rich, enabling accurate identification and usage by agents.
  - Protocols must support flexible invocation, accommodating diverse tool implementations and potential future support for plug-and-play.

Agent Protocols Requirements

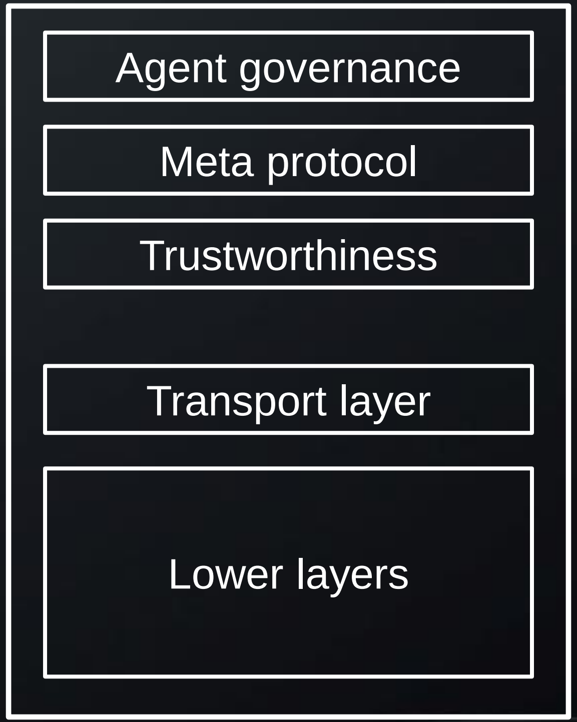
- **Knowledge retrieval**
  - AI agents in 6G require flexible access to external/internal data and knowledge sources that fall beyond their initial training.
  - Key design considerations should include:
    - Metadata filtering to exclude irrelevant data (e.g., by time or type).
    - Efficient data representation formats (e.g., vectors, graphs).
    - Smart retrieval methods using keywords, tokens, similarity, or natural language.

Enabling Features at Different Layers

- **Agent governance**
  - Agent profile contains key information about the agent, such as agent role description, skills, capabilities, identifier, input and output schema, vendor information, URL, etc.
  - **An authority object/entity**
    - Stores and maintains agents’ profile and provides for
    - Agent profile registration and deregistration
    - Profile update (e.g. as a result of changes in agents’ skills or capabilities)
    - Discovery mechanism by matching requirements to registered agent’s capabilities and skills.
  - **Publish/Subscribe**
    - Agents can subscribe to real-time updates from the authority object and employ filters like agent ID, role, skill, etc.
    - Authority object/entity notifies subscribers when an agent’s profile changes or is de-registered (e.g. preventing invalid task requests).
    - Discovery fallback is triggered when no suitable agent is available for a task.
      - Agents post task interests to await suitable agent registration.
      - Notification is triggered once a matching agent joins the ecosystem.

Enabling Features at Different Layers

- **Agent governance**
  - **Communication**
    - Agents communicate with other agents, tools/data resources, infrastructure (e.g. base stations), and users (e.g. UEs, apps).
    - Semi-structured data objects allow flexible, intent-based interaction. Fields are labeled but values aren’t fixed.
    - Agent-to-agent protocols should enable task initiation, updates, cancellation, status queries, and subscriptions using “Task” objects.
    - Task data includes identity, requirements, context, state, output, QoS info, and network/resource status.
    - Agents may reject tasks (e.g., limited buffer, horse power, etc.).
    - Agent-to-resource protocols enable tool invocation, data retrieval (e.g., network logs, operator data).
    - Agent-to-user protocols allow service requests, updates, deletions, and status/result access.



Enabling Features at Different Layers

- **Multimodal data management**
  - Agent communication is multi-modal and involves text, image, audio, video, structured data etc.
  - Protocol design should support mapping modalities to distinct data structures (e.g., TextPart, AudioPart).
  - Networks assign differentiated QoS and may establish separate sessions for each modality.
  - Multi-modal data processing and analysis is challenging due to its heterogeneity.
    - Data from different modalities differ in structure, semantics and representation.
  - Processing and analysis should consider multiple impact factors such as data alignment, feature extraction, etc. Factors to considered include
    - Data Pre-processing: cleaning (e.g. noise removal), standardisation (e.g. format unification for modalities), augmentation (e.g. use of rotation, scaling, etc. to increase diversity).
    - Feature Extraction: modality-specific feature mining.
    - Data Alignment: ensuring data temporal, spatial, semantic consistency.
    - Fusion: integration of diverse modalities for richer and more usable data.

Enabling Features at Different Layers

- **Meta protocol**
  - Agent protocols standardisation must allow for customisation.
  - Protocols should enable tool and data invocation via network functions.
  - Support for protocol release control should ensure smooth coexistence of multiple versions across operators and domains.
  - Meta protocol layer design should support self-negotiation, and consider longer term capabilities such as self-generation, and self-evolution.
  - Standardised and negotiated protocols should be allowed to coexist to support diverse, cross-domain agent systems.

- **Security, Privacy and Trust Layer**
  - Technologies such as Digital Identity can be considered to offer rapid authentication, key establishment based on multiple roots of trust, fine-grained authorisation, and selective privacy disclosure.
  - Distributed ledger ensures immutability, keeping transaction records tamper-proof. Ciphertext-based computing enables secure data processing, allowing computations without revealing underlying data (“computing without knowing”).
  - Quantum-safe technologies including Post-Quantum Cryptography (PQC) and Quantum Key Distribution (QKD). can offer protection against future quantum-based threats.

Enabling Features at Different Layers

- **Transport**
  - Synchronous or asynchronous version of stream multiplexing raise the need for striking a balance between the complexity of transport connection establishment and scheduling complexity.
  - Should support reliable and efficient transmission of agent traffic.
  - Native support for connection migration and quick link establishment is necessary for guaranteed agent service continuity(e.g. in mobility scenarios)

  - Support for burst agent traffic is needed.
    - Burst traffic information can be encapsulated in protocol header for better QoS guarantee and real-time resource scheduling and adjustment.
    - Aspects such as latency and timeliness required to be guaranteed over the entire agent burst data transmission.

IETF Draft

https://datatracker.ietf.org/doc/draft-hw-ai-agent-6g/

Thank You.