Exposing Optical Network Control Capabilities to AI Agents Using Model Context Protocol and TeraFlowSDN

Ricard Vilalta[1], Daniel Adanza[1], Allen Abishek[1], Lluis Gifre Renom[1], Pol Alemany[1],
Ramon Casellas[1], Ricardo Martínez[1], Raul Muñoz[1]
[1]*Centre Tecnològic de Telecomunicacions de Catalunya - CERCA (CTTC-CERCA), Castelldefels, Spain*
ricard.vilalta@cttc.es

***Abstract*—This paper presents an architecture that leverages the Model Context Protocol (MCP) to expose ETSI TeraFlowSDN control capabilities to AI agents, enabling intent-driven orchestration and closed-loop automation for cognitive and autonomous networks. MCP provides a standardized client–server interface that allows AI agents to discover and invoke network primitives—tools, resources, and prompts—while accessing contextualized information such as topology, spectrum availability, and performance telemetry. By integrating MCP with TeraFlowSDN, AI agents can perform advanced tasks including spectrum assignment, modulation optimization, proactive rerouting, and energy-aware resource allocation across multi-layer environments (IP over DWDM).**

We implement a modular MCP server within TeraFlowSDN and integrate an AI agent capable of intent parsing, context retrieval, and safe tool execution with schema validation. Experimental validation demonstrates robust intent classification and efficient execution of network operations, with latency breakdowns for topology retrieval, service listing, and intent-based provisioning.

***Index Terms*—Network Digital Twin, Scalable, Distributed, Resilient.**

I. Introduction

The increasing complexity of transport networks in the era of 6G requires a shift in how control and orchestration functions are designed and consumed. Traditional human-driven network management is no longer sufficient to meet the stringent demands of automation, resilience, and adaptability. Instead, AI agents are emerging as key enablers for closed-loop decision-making, capable of translating high-level intents into actionable configurations and optimizing resources in real time [1], [2], [3]. To unlock this potential, AI agents must be provided with direct yet structured access to network control capabilities. The Model Context Protocol (MCP) [4] is an open, client–server standard that lets an AI agent spin up MCP clients to connect to one or more MCP servers that expose three core primitives: tools (things the AI can run), resources (data to read), and prompts (reusable templates). The client discovers what the server offers (e.g., tools/list, resources/list), calls tools (tools/call) or reads resources.

In this context, ETSI TeraFlowSDN is an open-source SDN orchestrator for multi-domain and multi-technology transport networks. It provides features such as intent-based networking [5], dynamic slicing, multi-layer optimization, and secure orchestration of disaggregated IP/optical infrastructures [6]. Exposing these capabilities through MCP offers an opportunity to bridge network programmability with AI-driven autonomy, enabling agents to request services, enforce policies, and adapt the network according to evolving conditions. By extending TeraFlowSDN, networks can include important benefits: a) AI-driven optimization becomes feasible, as MCP allows AI agents to access contextualized optical network knowledge such as topology, spectrum availability, and performance telemetry, resulting in informed decisions on spectrum assignment, modulation formats, and path computation, achieving more efficient utilization of optical resources; b) MCP integration enhances resilience and service continuity, as optical layers are prone to degradation from fiber impairments, equipment drift, or failures such as fiber cuts. By exposing telemetry parameters (e.g., BER, OSNR, Q-factor) to AI agents, predictive models can detect degradations before they impact services, triggering proactive rerouting or reconfiguration of optical channels; c) MCP-enabled orchestration contributes to energy efficiency and sustainability, optimizing power consumption by selectively deactivating idle transponders, aggregating traffic onto fewer lightpaths, or choosing energy-aware routing strategies; d) MCP allows reasoning across multi-layer environments (IP over DWDM), where agents can decide whether to reconfigure optical lightpaths or reroute at the IP layer, enabling end-to-end optimization.

This paper implements a modular MCP server for ETSI TeraFlowSDN, which exposes topology, service, and intent primitives. Lastly, while the integration of LLMs with TeraFlowSDN has been explored in recent literature, such as in [2], our work differentiates itself by leveraging the Model Context Protocol (MCP), which provides a standardized, extensible interface for AI agents to discover and invoke network primitives, offering advantages over standard REST APIs or custom tools in terms of modularity, interoperability, and ease of integration. We integrate an AI agent that performs intent parsing, context retrieval, and safe tool execution with schema validation. We evaluate on intents and report classification accuracy and latency breakdowns, highlighting failure modes and easy mitigations.

978-3-903176-78-2 © 2026 IFIP

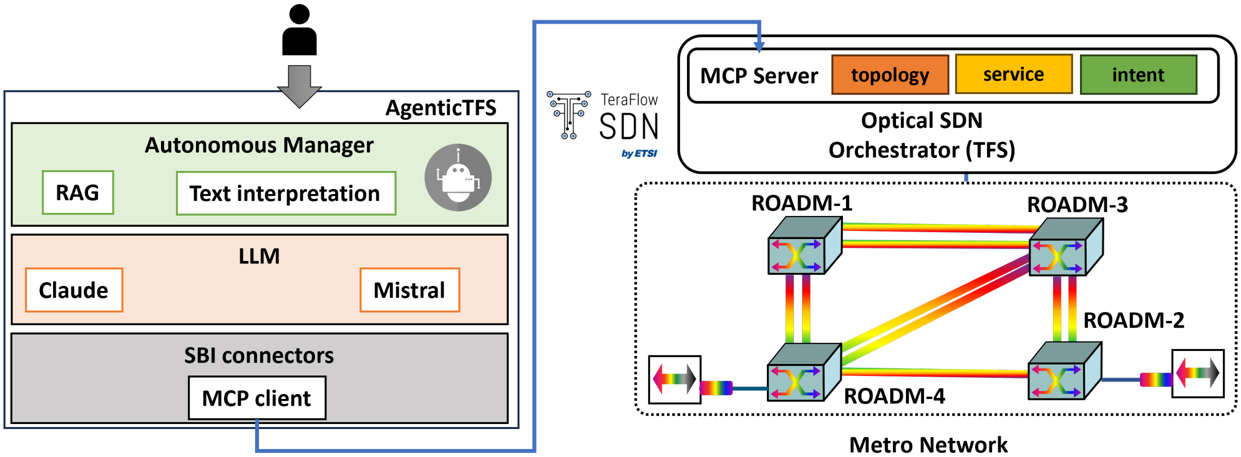

Fig. 1: Proposed architecture for exposing Optical SDN controller functionalities to AI agent.

II. Proposed Architecture

Fig. 1 shows the proposed architecture. On the left, an AI agent (AgenticTFS [2]) interprets a human request using text interpretation and Retrieval-augmented generation (RAG), then reasons with LLMs (e.g., Claude, Mistral). When it needs to act on the network, it uses an MCP client in the SBI (southbound) layer to connect to an MCP server hosted by the Optical SDN Orchestrator (based on TeraFlowSDN). MCP lets the client discover and call server-exposed primitives—tools, resources, and prompts—so the manager can, for example, read topology or service data and submit intent requests through well-defined methods. On the right, the Optical SDN Orchestrator (ETSI TeraFlowSDN) controls a wavelength-routed transport network (the colored links represent DWDM channels through ROADMs and transceivers). TeraFlowSDN is an open, cloud-native controller/orchestrator that can program end-to-end connectivity—across optical and IP domains—and interface with optical controllers (e.g., via Transport API). In this setup, the orchestrator's MCP server exposes topology/service/intent so the AI manager can read network state and request services by intent, which the controller translates into concrete optical paths/wavelengths.

The TeraFlowSDN architecture has been extended through a modular MCP server, which is a single component that mediates between AI agents (including MCP client(s)) and TeraFlowSDN's internal modules. Future enhancements should enable MCP server to plug into additional components beyond topology and service management, such as telemetry collectors, security managers, or energy-efficiency optimizers. By doing so, AI agents could access richer contextual views (e.g., real-time performance indicators, security states, or energy footprints) and act on a wider range of control knobs, expanding the scope of automation beyond basic provisioning. Finally, enhancements in scalability and multi-domain support are essential. As optical and IP transport infrastructures grow in scale and heterogeneity, the MCP Server should be capable of federating context from multiple TeraFlowSDN instances or interacting with peer domain controllers. This would provide AI agents with a unified, cross-domain view, enabling end-to-end orchestration across disaggregated multi-operator environments.

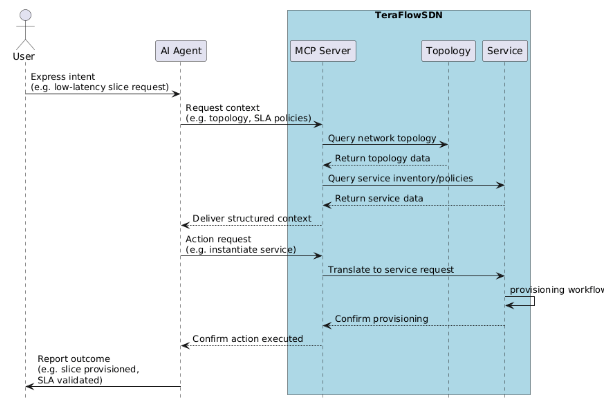

Fig. 2: Sequence diagram between Agentic AI and Optical SDN Controller

The workflow (Fig. 2) begins when the User expresses a high-level intent to the AI Agent, for example requesting the instantiation of a low-latency ultra-high bandwidth connection between two endpoints. The user does not interact directly with the network infrastructure; instead, the AI Agent acts as the mediator, responsible for interpreting the intent and determining the corresponding network actions. To make informed decisions, the AI Agent requires detailed knowledge of the transport network state. For this purpose, it issues a context request to the MCP Server within TeraFlowSDN. The MCP Server serves as the standardized interface between AI-driven applications and the orchestration platform. Upon receiving the request, the MCP Server queries two internal modules of TeraFlowSDN: the Topology Component, which provides information on the network graph, available spectral resources, and optical connectivity; and the Service Component, which returns service-related information such as existing connectivity instances, SLA definitions, and applicable policies. The responses from these components are aggregated by the MCP Server and exposed to the AI Agent as structured, machine-

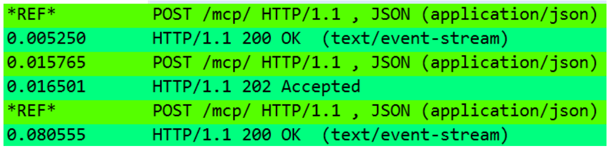
REF POST /mcp/ HTTP/1.1 , JSON (application/json)
0.005250 HTTP/1.1 200 OK (text/event-stream)
0.015765 POST /mcp/ HTTP/1.1 , JSON (application/json)
0.016501 HTTP/1.1 202 Accepted
REF POST /mcp/ HTTP/1.1 , JSON (application/json)
0.080555 HTTP/1.1 200 OK (text/event-stream)

Fig. 3: Wireshark captures of MCP protocol.

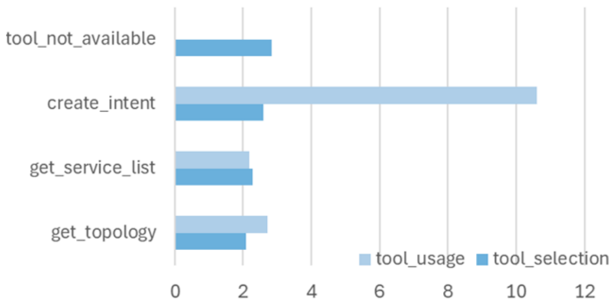

Fig. 4: Tool selection and tool usage delays (in ms).

readable context in compliance with the MCP model. With this contextual information, the AI Agent applies its reasoning or optimization algorithms to derive an appropriate course of action. For instance, the agent may determine the most suitable path and spectrum allocation for a connectivity service that meets the requested SLA. The agent then issues an action request to the MCP Server, specifying the service to be instantiated or modified. The MCP Server translates this request into a TeraFlowSDN-native workflow and delegates the execution to the Service Component. The Service Component carries out the provisioning workflow, configuring the necessary network resources and orchestrating multi-layer operations as needed. Once the requested service has been successfully provisioned, the Service Component returns confirmation to the MCP Server. The response is forwarded back to the AI Agent, which in turn informs the User that the requested service has been created and validated against the specified requirements.

III. Experimental Validation

To validate the proposed integration of AgenticTFS with TeraFlowSDN through the MCP protocol (as shown by the Wireshark capture in Fig. 3), we conducted a series of experiments where natural language intents were issued to AgenticTFS. The intents covered a diverse set of requests, including topology retrieval, including available and used resource spectrum, connectivity service listing, and intent-based connectivity service creation. Each request was classified and executed through the MCP Server connected to TeraFlowSDN. The results were evaluated in terms of system response times, including initialization, connection, response, and processing delays, and classification accuracy.

The performance analysis focuses only on protocol delay between MCP client and server (Fig. 4). It reveals clear differences depending on the type of request. For tool selection of simple topology and service list retrievals, times were generally low, with mean delays of 2.09ms and 2.29ms, and standard deviation of 1.44 and 0.81ms, respectively. The intent creation tool selection exhibited slightly higher latency, at 2.58ms (stdev 0.81ms). Significant differences appear in tool usage, where get topology mean is of 2.71ms and get service list is of 2.17ms, stdev 1.47 and 1.24ms. The triggering of a new intent, requires a processing delay of 10.6ms (excluding provisioning), stdev 2.95ms. Finally, unsupported queries were handled gracefully. All tool_not_available cases returned valid responses with moderate latency (mean 2.84ms, stdev 0.84ms), ensuring that the system rejected irrelevant inputs without timeout or crash. This confirms the robustness of the classification layer as an integral part of the MCP-TeraFlowSDN integration.

The classification results demonstrate that the MCP-enabled system can distinguish between supported and unsupported queries. Intents such as "retrieve the topology architecture of network" or "get the list of services for the network" were correctly mapped to the get_topology and get_service_list categories, respectively. Similarly, more complex requests such as "create an intent service low-latency ultra-high bandwidth between A and B" were accurately mapped to the create_intent category. Queries outside the network management domain were consistently flagged as tool_not_available, demonstrating that the classification module can filter unsupported intents without attempting invalid orchestration actions.

IV. Conclusion

In summary, the experimental validation shows that the proposed architecture can successfully classify intents, retrieve network information, and provision services through MCP. The results indicate that retrieval actions are efficient, while provisioning workflows incur higher but acceptable overheads. Future work will focus on optimizing workflows for intent creation, particularly in reducing redundant operations, and on further refining the integration of telemetry feedback for closed-loop automation. Furthermore, ensuring safety measures against AI hallucinations is essential to maintain the reliability and security of autonomous network operations.

Acknowledgements

This work was partially supported by: the European Union's Horizon Europe Research and Innovation Program under grant 101119602 in the Project COBALT; Spanish SHINE (PID2024-159781OB-I00) project financed by MCIU/AEI/10.13039/501100011033/ and by ERDF, UE.

References

[1] H. Zaid and other, "Multi-agent design for llm-assisted network management," in *2025 Optical Fiber Communications Conference and Exhibition (OFC)*, 2025, pp. 1–3.

[2] D. Adanza *et al.*, "Leveraging generative ai for intent-based networking operations in network slices," *Com. Net.*, 2025.

[3] D. Wang *et al.*, "Large language model for optical network automation: Prospects and challenges," in *OFC*, 2025.

[4] X. Hou *et al.*, "Model context protocol (mcp): Landscape, security threats, and future research directions," 2025.

[5] P. Alemany *et al.*, "Defining intent-based service management automation for 6g multi-stakeholders scenarios," *IEEE OJCOMS*, 2025.

[6] L. Gifre *et al.*, "Dynamic adaptation of ip virtual network topologies over multi-granular (wavelength and waveband) optical networks supported by etsi teraflowsdn multi-layer control," in *OFC*, 2025.