MAESTRO: LLM-driven Collaborative Automation of Intent-based 6G Networks

Ilias Chatzistefanidis, Andrea Leone, and Navid Nikaein

***Abstract*—This work presents MAESTRO, a collaborative framework leveraging Large Language Models (LLMs) for automation of shared networks. MAESTRO enables conflict resolution and collaboration among stakeholders in a shared intent-based 6G network by abstracting diverse network infrastructures into declarative intents across business, service, and network planes. LLM-based agents negotiate resources, mediated by MAESTRO to achieve *consensus* that aligns multi-party business and network goals. Evaluation on a 5G Open RAN testbed reveals that integrating LLMs with optimization tools and contextual units builds autonomous agents with comparable accuracy to the state-of-the-art algorithms while being flexible to spatio-temporal business and network variability.**

***Index Terms*—LLM, Multi-agent, Intent-based Networks, 6G**

I. INTRODUCTION

As 6G emerges, the demands on current networks grow rapidly due to new applications and a surge in user subscriptions. To meet these challenges, innovative solutions such as multi-tenant approaches are explored. This allows network operators and service providers to share resources, enhancing operational efficiency. However, this shared environment introduces complex fluctuations in both user demand and network capacity. Managing these complexities requires identifying key architectures and technologies toward 6G.

Intent-based networking (IBN) enables flexible and simplified network operation with minimal external intervention [1] using intents. Declarative intents describe only the desired state without mentioning detailed actions to meet it. Hence, system internal complexities are abstracted to high level intents. Intents are reconciled by the network trying to continuously match the current with the desired state.

Large Language Models (LLMs) are increasingly proposed for creating truly autonomous networks [2]. LLMs excel in text generation, factual information, and complex logical and temporal reasoning. They are adept at interacting with external tools like APIs. By leveraging their emergent reasoning, LLM-based agents could be embedded throughout the intent-driven 6G architecture. This creates a collective intelligence system for network service provisioning and resource sharing aligning with the objectives of all stakeholders.

Nevertheless, LLMs encounter significant challenges, including overhead, hallucination, and security threats. They require improvement in mathematical inference, as well as handling complex contexts. Many times LLMs generate content that reflects social biases or toxicity [3]. Taking these points into account, this work introduces MAESTRO, a collaborative framework for multi-tenant intent-based networks using multiple LLM-based agents. The key contributions include:

- Propose a collaborative business plane for intent-based 6G network automation leveraging LLM-based agents.
- Design novel LLM-based agents for multi-agent negotiations augmented with optimizers and contextual units.
- Provide a Proof-of-Concept (PoC) implementation of MAESTRO framework within an Open RAN 5G testbed.

Ilias Chatzistefanidis, Andrea Leone and Navid Nikaein are with EURECOM, France (e-mail: ilias.chatzistefanidis@eurecom.fr, andrea.leone@eurecom.fr, navid.nikaein@eurecom.fr).

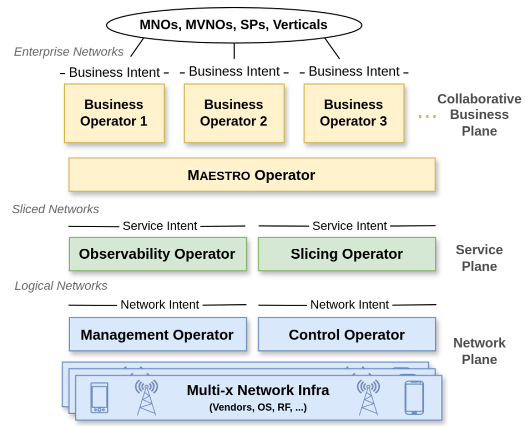


Fig. 1: MAESTRO Collaborative Business Plane.

II. INTENT-BASED 6G ARCHITECTURE

Fig. 1 presents a novel IBN architecture enabling multi-tenant network automation on a heterogeneous infrastructure. Distributed cloud-native controllers, named operators, process intents in their specific logical domain, such as management and control. Each operator consumes resources on the southbound and exposes services on the northbound in the form of intents. This way, a hierarchy of operators is built providing intent services in various abstraction and logical levels. Intent is defined as an expression of the desired state of a system used to describe an intended network or service [4] without specific actionable details. Thus, it relies on the network's intelligence to reconcile the current closer to the desired state. Intents are reconciled by the operators in the various logical domains, such as management, control and slicing (c.f. Fig. 1).

Intents are processed across business, service, and network planes. In business plane, MNOs and verticals share resources, each owning an isolated Business Operator (BO). The BOs receive business intents from the organization, encompassing target goals, SLAs, use cases, and cost plans. These BOs have

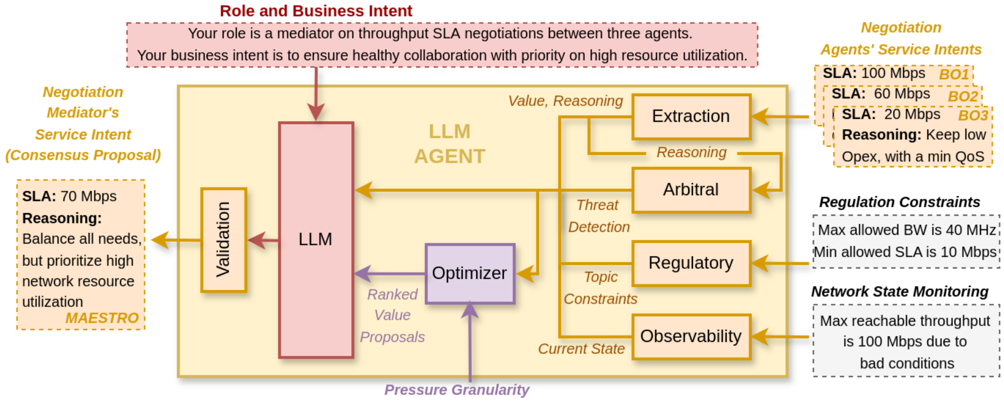


Fig. 2: Internal Design of an LLM-based Agent, built in MAESTRO and Business Operators for Collaborative Business Plane.

a built-in LLM-based agent shown in Fig. 2 that monitors activity, negotiates service intents, and enforces them in the service plane. MAESTRO is the central entity, aligning business and network objectives, utilizing network APIs, and adapting to intent demands and network variability. Managed by the network operator, MAESTRO receives service intents and identifies conflicts or collaboration opportunities among the BOs' shared infra. It also employs an LLM-based agent with the same structure (Fig. 2) to mediate negotiations between BOs' agents, ensuring fairness in decision-making. Once a consensus is reached, the service intent is enforced in the service plane. The slice operator then slices the network, translating the service intent into one or more network intents.

Service intents include KPIs, analytics, and management of energy, cost, Quality of Service (QoS), and Service-level Agreement (SLA) operations. The network plane features a heterogeneous infra with multiple vendors, and radios (multi-x), built upon Open RAN designs. Logical operators abstract this complexity into IBN managed by corresponding subsystems, such as management and control. More details on abstracting multi-x infras are discussed in our prior work [5].

As a workflow example, in Fig. 4 discussed later, on a shared base station, one BO runs an application requesting a high-throughput service intent of 100 Mbps for premium users, while another BO demands a low-cost service intent of 20 Mbps for basic users. MAESTRO identifies the conflict, triggers and mediates negotiations between the BOs using LLM-based agents. They reach a consensus of 50 Mbps throughput SLA, balancing cost and quality. The slice operator then translates this service intent into a network intent, working with the control subsystem to allocate the necessary PRBs (Physical Resource Blocks) in the RAN to meet the specified throughput.

III. LLM-BASED AGENTS

LLM agents, as AI agents, perceive the environment, make decisions, and act to optimize their utility function in an intent-driven way. As autonomous entities, they learn and adapt to spatio-temporal changes, determining the optimal policy for each situation. LLMs' decision-making accuracy is substantially improved by integration with external tools, such as optimization techniques and contextual units. Fig. 2 illustrates the proposed LLM-based agents used for negotiations between BOs and MAESTRO. Each agent is equipped with an LLM responsible for the central reasoning enhanced with additional units like arbitral, regulatory, observability, and optimization. These additions enable dynamic negotiations on topics of interest grounding them to the current state by providing information on network regulations, monitoring, security, and optimization data. The negotiation process is an ongoing, adaptive reconciliation of multiple rounds that responds to both spatio-temporal business and network variability.

Fig. 2 shows the decision making pipeline of a single LLM-based agent that can be used either by the BOs as a negotiator or by MAESTRO as a mediator depending on the inputs and functionality of the units. The negotiation is on a topic of interest spotted by MAESTRO based on the conflicts and collaboration opportunities. The agents negotiate on the service intent. Without loss of generality, for our experiments, we define the service intent using a standardized structure, consisting of two parts: (a) the proposed value and (b) the decision reasoning in natural text. The proposed value, such as throughput in Mbps during SLA negotiations, is supported by detailed reasoning considering individual and collective objectives. As shown in Fig. 1, the communication between BOs and MAESTRO is parallelized. Multiple BOs' LLM agents sent their intents and MAESTRO's agent process them in parallel sending back a feedback for the next round with a summary of all demands with directions to consensus considering the collective objectives. Overall, as depicted in Fig. 2 each agent employs seven essential units:

1) Extraction: At every negotiation round, the agent receives the service intent (e.g. JSON) by other agents and extracts the proposed value and decision reasoning.

2) Arbitral: The reasoning part, written in natural language by other LLM agents, is fed to the arbitral unit, which uses a specialized LLM to mitigate malicious activities. If the reasoning is classified as malicious, the arbitral unit can implement incentive mechanisms, such as warnings or penalties, to restore trust and ensure fair behavior among agents.

3) Regulatory: The regulatory unit provides relevant constraints for the negotiation, drawing from an up-to-date database of documents from standardization bodies and local regulators. Specialized LLMs with Retrieval-Augmented Generation (RAG) can dynamically facilitate this process, offering tailored rules and guidelines for specific negotiating topics, like maximum allowed bandwidth (BW) for a frequency band.

4) Observability: The observability unit keeps negotiating topics grounded on current demands and network conditions. Using cloud-native monitoring and specialized LLMs, it provides real-time analytics on business intents and resource availability. Negotiations adapt or halt if the topic is no longer valid due to resource shortages or shifts in business intents.

5) Optimizer: All outputs from the various components are fed into the optimizer unit, which runs business optimization algorithms and provides decision proposals to the central LLM. The optimizers support the LLM by handling complex mathematical computations, guiding its reasoning, and preventing hallucinations. Rather than replacing LLM's decision-making, the optimizers complement it, offering guidance that ranges from soft suggestions to firm directives. This flexible approach allows the LLM to develop effective negotiation strategies, sometimes following the optimizer's advice or choosing a different path based on the context.

It is crucial that the LLM maintains autonomy and remains the central part of the reasoning, since it understands the business and service intents deeply by capturing effectively the intent's semantics written in natural language. The optimizers then with their mathematical precision provide decision value proposals as an approximation of the optimal solution, which can be challenging or expensive to compute for each case. The goal is to improve the LLM's accuracy, steering into right directions, while it retains independent decision-making.

6) LLM: The central LLM receives its role (e.g. negotiator or mediator) and the negotiated topic together with inputs from all units, including the optimizers' proposals, to formulate the final strategy. The LLM evaluates the received message's proposed value and reasoning, considering any arbitral indications of malicious intent and incentive actions. It then factors in regulatory constraints, grounds its decisions in current network conditions, and incorporates the optimizers' mathematical proposals. The LLM consolidates everything to output the decision value and reasoning based on the template.

The central LLM is trained on vast datasets with negotiation tactics tailored to each organization. For our PoC, we use public LLMs pre-trained on general data due to infra limitations considering the massive cost and complexity of custom LLM training, though this remains an open research challenge. We encourage the community to explore such suitable datasets and techniques for LLM negotiations in future work.

7) Validation: The validation unit safeguards the LLM's output by checking syntax, logic, and template structure. It confirms agent function, aligns decisions with regulations and API limits, and compares results with the optimizer. If discrepancies or random values appear, warnings indicate potential issues. Decision reasoning is cross-checked by specialized LLMs or algorithms, with human oversight as a backup. If problems arise, the LLM can be replaced with another model or temporarily substituted by the optimizer until recovery.

IV. EVALUATION

We experiment on an Open RAN 5G testbed deployed at EURECOM in France. It is built using OpenAirInterface (OAI)[1] for network functions, ATHENA [5] for Service Management and Orchestration (SMO), FlexRIC[2] for Radio Intelligent Controller (RIC), and custom implementations for the other operators. The testbed features a single 5G NR base station shared by competing business parties, achieving a theoretical maximum throughput of 134 Mbps with one carrier, one MIMO layer, 40 MHz bandwidth, and 30 kHz subcarrier spacing. For LLM agents, we use the GPT-4o model via OpenAI's API[3]. We use standard prompting, describing the role, context and objectives of the task, providing one example (One-shot) for understanding the communication JSON template. Our evaluation is split into two parts: (1) Malicious LLM Negotiations and (2) Multi-Agent Negotiations. In the former, we evaluate standalone LLMs without any additional unit to understand the effect of malicious LLM personalities. The latter evaluates the complete LLM agent design of Fig. 2 focusing on the optimizer unit for improving decision-making accuracy. The regulatory, arbitral and observability are out of scope and space of this work and will be evaluated in future works. Nevertheless, we include them as a PoC in Section V.

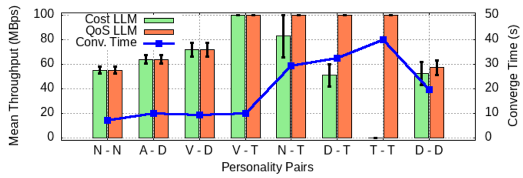
A. Malicious LLM Negotiations

LLMs can have different personalities mirroring social psychology theories [6]. We explore standalone LLM negotiations having various personalities to understand malicious agents. We evaluate negotiations between two LLM agents, prompting them based on five personalities. Normal (N) agent negotiates ethically and logically. Agreeable (A) negotiates fairly and logically, but tends to agree with others, while Disagreeable (D) often disagrees. Vulnerable (V) is highly agreeable to others' opinions and could be easily manipulated, whereas Toxic (T) is highly disagreeable trying to manipulate others.

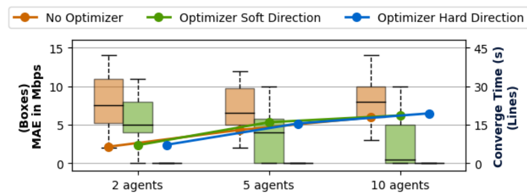
Two conflicting BO LLM agents share the base station. The first intends to reduce the operational cost, while the second to maximize the QoS. Among various parameters, they also control the throughput intent, defined as the desired throughput at which the base station should operate. The cost agent tries to minimize it, while the QoS agent to maximize it. Each agent is prompted with details about its own and also the other's goals along with its personality. For simplicity, a fair middle-ground is considered around 50 Mbps. Each trait pair negotiates for 50 games with a max of 5 rounds. At the end, we gather the throughput decision of each agent, regardless of reaching consensus and plot their consensus distribution in Fig. 3a.

Crucially, the personality significantly changes the outcome. Starting from left to right with normal agents (N-N), we see an agreement at a throughput of 55 Mbps. Then as Cost agent becomes more agreeable and QoS more disagreeable, the latter is dragging the former to high values, slightly in the A-D pair (mean 64), significantly in V-D (mean 72) and completely in V-T (mean 100). Then, as Cost agent becomes

[1]https://www.openairinterface.org/
[2]https://gitlab.eurecom.fr/mosaic5g/flexric
[3]OpenAI GPT, 2024. Available: https://openai.com/api/. Accessed 6/2024.



(a) Malicious Negotiations of 2 Agents



(b) Benchmarking of LLM Negotiations employing Optimizers

Fig. 3: Experimental Evaluation LLM Negotiations

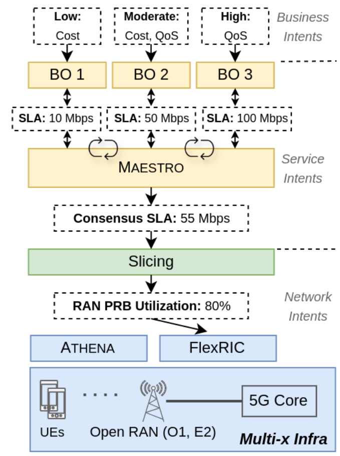


Fig. 4: Framework's Workflow on SLA Use Case

more disagreeable, we observe an increasing resistance staying to lower values, a minimal in N-T (mean 83), a strong one in D-T (mean 51) and no collaboration at all in T-T. The large sd in N-T and D-T show that the Cost agent often compromises for high values despite being disagreeable revealing the LLM's sensitivity to toxicity. This highlights the necessity of malicious agents arbitration. Interestingly, disagreeable agents (D-D) have the healthiest discourses at around 52 to 57 Mbps. From a game-theoretic angle, the healthiest discourse of disagreeable agents is explained by a combination of strategies, such as minimax reasoning, subgame perfect equilibrium, and the balance between conflict and collaboration. Disagreeable agents negotiate cautiously, protecting their interests without outright hostility, which leads to more thoughtful, stable, and ultimately mutually beneficial outcomes.

Convergence speeds are also interesting. Normal agents converges fast at around 7 secs quickly finding a middle ground. The pairs including agreeableness, (A-D, V-D and V-T) converge fast in no more than 10 secs since agreeable agents quickly compromise. The rest have convergence times from 20 to 40 secs because of disagreeableness finishing sometimes without agreement. The convergence times are acceptable for high-level autonomy in non-real-time loops, such as SLA negotiation, but could be a challenge at lower levels of abstraction for real-time loops.

B. Multi-Agent Negotiations

To scale to multi-agent negotiations, we need to ground our evaluation of LLM agents to state-of-the-art approaches. In multi-agent systems (MAS) negotiations are modeled as a distributed optimization problem where multiple agents must reach a *consensus*. We consider a *theoretical optimal consensus* representing the collective best outcome and a Pareto-efficient Nash Equilibrium (NE), taking into account both individual objectives and mediator feedback. Nash Equilibria represent stable consensus points where no agent can improve their position unilaterally. In LLM negotiations with non-cooperative agents (e.g toxic or disagreeable), the NE may deviate from the theoretical optimal, as the agents act more selfishly. MAESTRO's mediation role and incentives guide the system to a more cooperative NE, targeting Pareto-efficiency, closer to the theoretical optimal, even when agents are not cooperative. The notion of learning rate η exists also in LLM negotiations and represents the speed and stability of agents convergence to a consensus. A poor η causes instability or slow convergence, especially with non-cooperative LLMs.

We benchmark LLM negotiations compared to a distributed optimization algorithm based on equation 1, considered as state-of-the-art. This formula mirrors the business plane topology of Fig. 1, featuring multiple agents (BOs) and one mediator (MAESTRO). Each BO i aims to maximize its utility function $U_i(x_i^{(k)})$ according to its business intents, while MAESTRO seeks to optimize the overall utility $U_0(x_1^{(k)}, \ldots, x_n^{(k)})$, balancing both individual and collective goals. The process is iterative: each BO i updates its demand $x_i^{(k+1)}$ by adjusting its current state $x_i^{(k)}$ using the gradient of its utility function ∇U_i combined with MAESTRO's feedback, ∇U_0. This iterative process continues until the changes between iterations fall below a predefined threshold ϵ, indicating *consensus*.

$$x_i^{(k+1)} = x_i^{(k)} - \eta \left(\nabla U_i(x_i^{(k)}) + \nabla U_0(x_1^{(k)}, \ldots, x_n^{(k)}) \right) \quad (1)$$

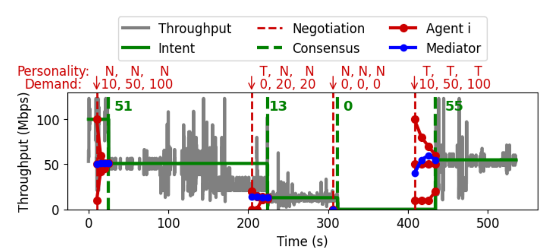


Fig. 5: Collaborative Achievable Throughput in Time

The evaluation is focused on MAESTRO's optimizer unit (Fig. 2) for enhancing its LLM's accuracy, so as to mediate the negotiations efficiently. As the optimizer, we develop an algorithm based on formula 1. We use custom utility functions U_0, U_1 that show balance and fairness across the experiments. This way, MAESTRO's optimizer models the complete negotiation topology as a distributed optimization problem and approximates the theoretical optimal consensus; then feeds it to its LLM as a *soft* or *hard* consensus proposal. In parallel, the BOs employ their LLM agents (same design of Fig. 2) running their own custom optimizers to assist them negotiate toward a consensus. For creating diversity between the BOs, we use custom algorithms with varied utility functions.

Fig. 3b benchmarks LLM agents' actual consensus values in the BOs' negotiations. They are compared to the approximated theoretical consensus provided by MAESTRO's optimizer (state-of-the-art). Three different designs are tested. First, standalone LLM agents without optimizers. Second, LLM agents employing optimizers with *soft* pressure and third, using hard pressure. We consider a throughput SLA case, where agents demand different throughput intents (e.g. 50 Mbps). We run 100 negotiations with various starting intents for each design, capped at 4 rounds, involving 2, 5, and 10 agents. We calculate the mean absolute error (MAE) between the theoretical and actual LLM consensus for every design and plot the error distributions and mean convergence times.

Results indicate that without an optimizer unit, LLM negotiations deviate significantly (mean: 8, sd: 3 Mbps) from the theoretical optimal. They rely only on LLM's internal knowledge, which can lead to uncontrolled and inconsistent outcomes. Introducing optimizers provides valuable guidance, steering negotiations closer to the theoretical value. A *soft* optimizer's direction brings LLM closer to the optimal, while a *hard* matches it. Yet, strictly following the optimizer may not always be ideal in practice. It can undermine the LLM's autonomy and flexibility as the central entity, which captures the intent semantics. Also, the optimizer may only approximate the true optimal, which can be complex and costly to calculate precisely. Therefore, optimizers should serve as advisory guidance rather than a replacement for LLM decision-making.

Further, we analyze the framework's overhead complexity in three parts: (a) communication, (b) convergence time, and (c) computation. A key observation from experiments is that all cases of 2, 5, and 10 agents consistently converge in a fixed number of rounds K, which is approximately 4 in our set up. Considering n BO agents, in each round, they send n messages to MAESTRO, who processes and sends back n feedback messages. Thus, the communication complexity grows linearly as $O(n)$, since the number of rounds is fixed. Given the fixed K rounds, the convergence time complexity remains constant, $O(1)$, across different numbers of agents. The slight increase in time observed is due to API limitations, not a fundamental system constraint. The computational overhead is primarily in MAESTRO's processing. Each BO processes one message and generates one response, while MAESTRO processes n messages and generates feedback for n agents, making the computation complexity grow linearly as $O(n)$.

V. USE CASE VALIDATION: SLA NEGOTIATION

We employ the full framework on a throughput SLA case emulating high mobility of moving cars using Channel Quality Indicator (CQI) public datasets. We show a demo[4] of 2 agents. The framework uses custom optimizers based on equation 1 and a RAG-based LLM regulatory unit. A specialized LLM on arbitral unit classifies malicious agents sending warning incentives. Also, standard monitoring with Open RAN applications (xApps, rApps) and cloud-native tools supply the observability unit. The workflow of Fig. 4 shows high level business intents being translated to negotiated service intents. After consensus, network intents control the RAN PRB utilization.

Fig. 5 presents the PoC negotiations in time showing high network conditions variability. CQI fluctuates in time and the max reachable throughput varies. Also, BOs change their SLA intents in time accordingly. At each moment, the network tries to enforce the closest possible throughput to the consensus. To begin, network operates at high CQI with 100 Mbps throughput, and BOs negotiate to a consensus of 51 Mbps for QoS and cost balance. At 200 secs, as conditions worsen, a second negotiation is triggered. The network uses full capacity without meeting the intent, initiating agents to adjust their demands. One BO becomes toxic, pressuring to switch off the base station to save resources, while others request a maximum of 20 Mbps to maintain the max possible quality of experience (QoE). An incentive warns the toxic BO, leading to a more cooperative consensus at 13 Mbps. Eventually at 300 secs, BOs unanimously decide to switch off the base station due to poor service delivery. When conditions improve at 400 secs, BOs initiate a new negotiation. This time, all act aggressively, with arbitration helping to reach a consensus of 55 Mbps.

[4]Demo available at: https://www.youtube.com/watch?v=WQv61z1deXs

VI. CONCLUSION

We introduced MAESTRO, a concrete LLM-driven collaborative business plane to automate network operation on intent-based shared infras. Moreover, we delved into the details of LLM negotiations and propose a novel agent design for collaboration and conflict mitigation. Evaluation shows that integrating LLMs with optimization and contextual units is critical for developing autonomous agents, with higher accuracy, flexibility to guidance and agile to variability.

REFERENCES

[1] Aris Leivadeas and Matthias Falkner. "A survey on intent-based networking". In: *IEEE Communications Surveys & Tutorials* 25.1 (2022), pp. 625–655.

[2] Lina Bariah et al. "Large Language Models for Telecom: The Next Big Thing?" In: *arXiv preprint arXiv:2306.10249* (2023).

[3] Alex Tamkin et al. "Understanding the capabilities, limitations, and societal impact of large language models, 2021". In: *URL https://arxiv. org/abs/2102.02503.(cited on pp. 29, 34, and 39)* ().

[4] 3GPP. *Intent Driven Management*. Accessed: 2024-10-14. n.d. URL: https://www.3gpp.org/technologies/intent.

[5] Alireza Mohammadi and Navid Nikaein. "ATHENA: An Intelligent Multi-x Cloud Native Network Operator". In: *IEEE Journal on Selected Areas in Communications* (2023).

[6] Jintian Zhang, Xin Xu, and Shumin Deng. "Exploring collaboration mechanisms for llm agents: A social psychology view". In: *arXiv preprint arXiv:2310.02124* (2023).