Demonstration of Automation of Network Configuration Generation using Generative AI

Supratim Chakraborty
Software Engineer
Mobility Group
Cisco Systems Inc.
Bengaluru, India
supratch@cisco.com

Nithin Chitta
Principal Engineer
Mobility Group
Cisco Systems Inc.
Bengaluru, India
nithin.chitta@gmail.com

Rajesh Sundaresan
Professor, ECE
Associate Faculty, RBCCPS
Indian Institute of Science
Bengaluru, India
rajeshs@iisc.ac.in

***Abstract*—A network service provider (SP) often requires months of planning and testing to launch a new service in the form of a tariff plan. This is because SPs traditionally used operation support systems and business support systems which are large multi-vendor systems with custom implementations that are not easily amenable to launching new digital services such as internet, voice, SMS, IoT. In another detailed work, we have highlighted the challenges and proposed a faster approach to provisioning such services on the network. The method used a deep neural network framework and a large language model to automate the generation of network configurations. In this submission, we propose to demonstrate our solution framework.**

***Index Terms*—Artificial Intelligence, Network Provisioning, Generative AI**

I. Introduction

This paper is a demonstration of a solution framework [1] that translates a tariff plan and other additional parameters (not a part of the tariff plan) like Public Land Mobile Network (PLMN) identifier, IP pools, DNS server list and other Network Services and Polices (NSPs) to a working network configuration. We provide a short summary of the motivation and methodology, which are detailed in [1]. We then discuss the setup and demonstration in the subsequent sections.

Mobile network operators offer various connectivity services like voice, SMS bundles, data allowances, hotspot, roaming services, and other related value-added services with their respective pricing structures in the form of an agreement called tariff plan. Service providers (SP) or mobile network operators serve diverse market demands by offering plans tailored to various customer segments, such as Consumer, Enterprise, and IoT, each with distinct offerings and pricing structures. Segmentation of the market refines these offerings further with the introduction of various tiers of plans. For example, while with Gold plans, a premium experience is offered, certain offerings are masked for Silver or Bronze plans. Student plans, on the other hand, prioritise data affordability, and enterprise plans focus on high quality of voice, leading to various features and value propositions. In addition to this, government regulations also change the offerings across different SPs. Technological advances in telecommunications driven by changes in infrastructure and market adoption also continue to define tariffs. An examination of different tariff plans across 27+ operators, across geographies revealed a high amount of variation.

Additionally, service differentiation can sometimes lead to ambiguity in the use of terms. The phrase "unlimited data" might imply an upper limit of 50 GB before throttling or unlimited usage with a maximum speed of 50 Mbps. Clear interpretation is essential to assess perceived value.

The complexity highlighted above suggests that turning a tariff plan into a configuration is not easy. The context is well-poised for use of AI and large language models in enabling configuration automation and management of complexity.

II. Setup Design

As indicated in Fig. 1, a graphical user interface (GUI) allows the service provider to input the tariff plan and any other additional parameters (NSPs) on the network. This GUI is connected to BLOCK-2, the AI engine, which further processes the input from BLOCK-1, and is discussed in the following sections.

A. Input filtration

As discussed in [1], erroneous inputs to large language models may generate wrong configurations of network elements leading to network outages and security vulnerabilities. Therefore, an input filter as shown in BLOCK-2A in Fig. 1 is necessary to filter out invalid tariff plans to avoid a cascading effect of misconfiguration on the network. Within the proposed solution, the input filter is implemented as a deep neural network comprising four layers: an embedding layer that transforms textual data into embedding vectors, given the substantial textual content within a tariff plan; a global average pooling layer that reduces the dimensions of the input data while retaining crucial information; and two dense layers, with the terminal dense layer responsible for classifying whether the input is a valid or invalid tariff plan. Consequently, the input filter ensures that only valid tariff plans are forwarded to the fine-tuned GPT-3.5 model.

B. Identifying and extraction of keywords from tariff plan

After studying tariff plans across 27+ different service providers, five information elements or keywords are identified

978-3-903176-66-9

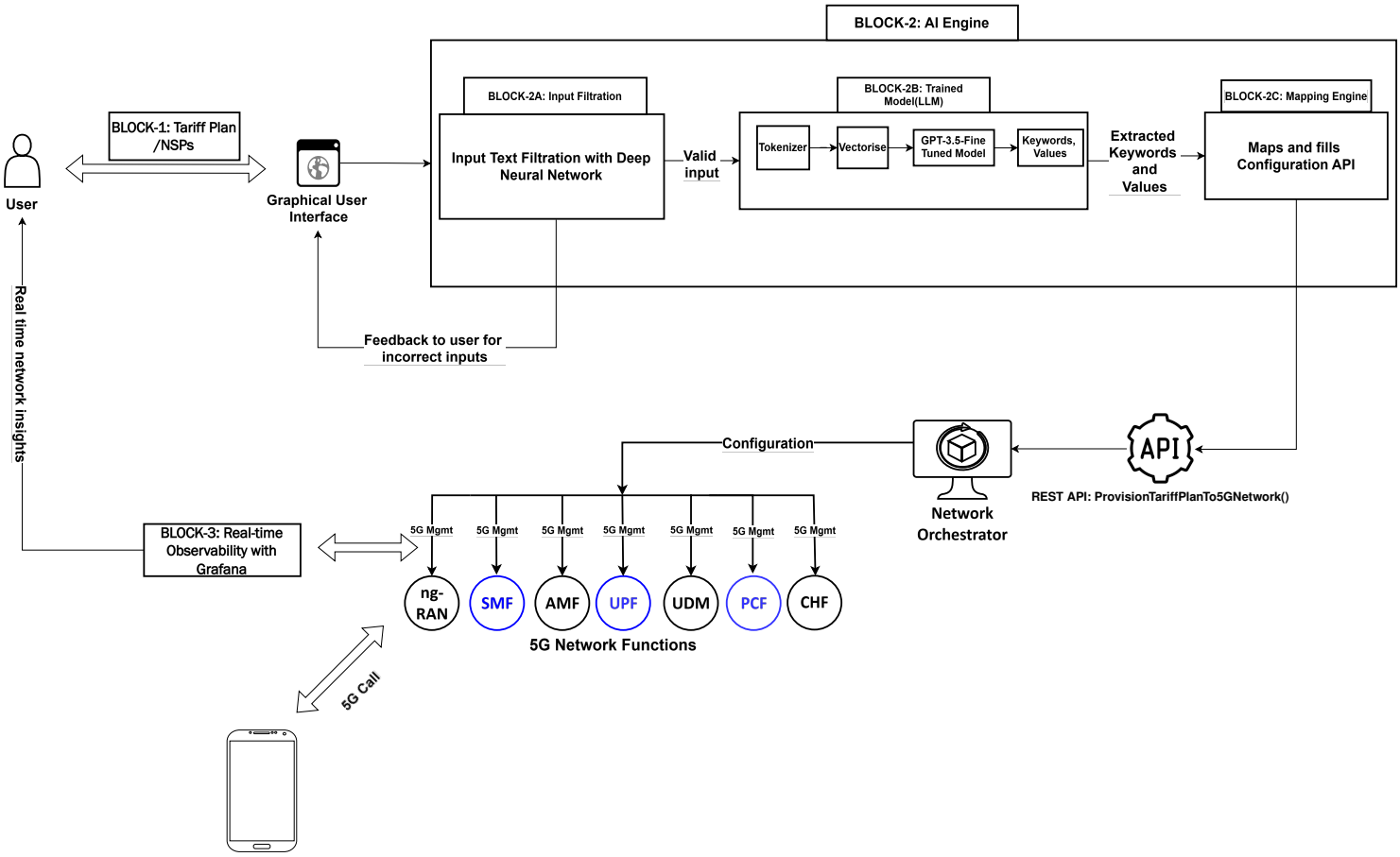

Fig. 1: End-to-end tariff plan to configuration across network nodes. As part of the demonstration, we shall use SM, UPF and PCF (circled in blue) as the target network node for provisioning.

based on serviceability, tariff quota, fair usage policies (FUPs), and quality of services (QoS), which bear the necessary configuration information within a tariff plan. The keywords are *'dnn', "data allowance", "avg speed data allowance', "data speed video stream" and "FUP redirection url"*. After the valid tariff plan is passed through the input filter, it is then tokenised and converted into vectors, depending on the size of the tariff plan. Then the vectorised tariff plan is passed through the fine-tuned GPT-3.5 model [2] to extract the keywords and values in the form of a dictionary, shown in BLOCK-2B. Before concluding with the choice of fined-tuned GPT-3.5 model, we have conducted a comparative analysis of various approaches for keyword (entity) extraction, including Named Entity Recognition (NER) with entity linking [3] and fine-tuned generative models (GPT-3.5 and Llama2-7B [4]) capable of contextual keyword extraction. To evaluate the performance of the various models with input filter augmentation, 50 tariff plans were analyzed to extract the five specific keywords. The results indicated that the fine-tuned GPT-3.5 model achieved the highest F1-score [5], which reflects the harmonic mean of precision and recall, outperforming other methods across datasets (tariff plan and PLMN identifier). Furthermore, the fine-tuned GPT-3.5 model also exhibited a competitive keyword extraction time amongst all the tested approaches as indicated in Table I.

C. Keyword to Configuration Mapping

The Mapping Engine (BLOCK-2C in Fig. 1) is responsible to populate a REST [6] based configuration API with the aforementioned keywords (Section II-B). The output of this mapping engine is the desired configuration API. The network orchestrator then consumes the API and forms the relevant configuration, which is then provisioned across the network elements [7] as shown in the bottom left of Fig. 1.

TABLE I: Comparative results across different approaches tried with both tariff plan and PLMN identifier datasets for evaluation. The metric execution time here is measure in terms of processing time for a single input of tariff plan or PLMN identifier.

	F1-Score	Exec. Time(sec)
Tariff Fine-tuned GPT-3.5	1	5
Tariff Fine-tuned Llama2-7B	1	27
Tariff NER	1	2
PLMN Fine-tuned GPT-3.5	0.9612	3
PLMN Fine-tuned Llama2-7B	0.948	20
PLMN NER	0.738	1.5

D. Single pane of glass observability

BLOCK-3 in Fig. 1 allows for real-time observability with Grafana[1]. This ensures auxiliary protection against any kind of misconfiguration in the network by monitoring the network in real-time. Grafana enables continuous real-time network monitoring [8] with necessary insights, thus allowing the network operator to quickly adapt to changes that may be required in the network at various times.

III. Demonstration

In this demonstration, we will present a systematic approach for converting tariff plans and NSP parameters, such as the PLMN identifier, IP address pools, and list of DNS servers into an operational network configuration with minimal human intervention with the aid of fine-tuned GPT-3.5 generative model. Following this, the generated configuration will be reviewed and validated. The network service orchestrator will provision the validated configuration into the network elements in real-time. Subsequently, using standard network orchestrator APIs, we will verify the accuracy and correctness of the provisioned configuration. To assess the efficacy of the configuration, a simulated 5G call will be set up that uses the newly provisioned configuration. The 5G call status will be then validated through command line interfaces (CLI) and observability tool. The demo is split into two steps: 1) Provision NSPs like PLMN identifier, DNS servers, IP pools, and 2) Provision a plain text tariff plan. The detailed steps of the demo are as follows:

1) *Provision NSPs like PLMN identifer, DNS servers and IP pools.*
   a) User inputs NSPs like valid PLMN identifier, list of DNS servers and IP pools in plain text through the GUI.
   b) The AI Engine extracts required values from the input NSPs and populates the necessary REST based configuration API for each NSP and the same is validated.
   c) Then the network orchestrator invokes the configuration API on the 5G-Mgmt interface of SMF.
   d) SMF then applies the configuration and informs the success/failure of the configuration action to the network orchestrator which is then informed to the user through the GUI.
   e) The applied configuration is then further validated with use of the CLI commands at SMF.
2) *Provision a plain text tariff plan.*
   a) User inputs a plain text tariff plan through a GUI, which is passed on as an input to the AI Engine of BLOCK-2 in Fig. 1.
   b) AI Engine in BLOCK-2 extracts the discussed keywords, values in Section II-B from the tariff plan.
   c) AI Engine maps the extracted keywords, values to populate the REST based configuration API parameters. Then the filled REST based configuration API is validated.
   d) AI Engine invokes the REST based configuration API on the network orchestrator.
   e) Network orchestrator invokes configuration API on the 5G-Mgmt interface of SMF, UPF and PCF as circled in blue in Fig. 1.
   f) SMF, UPF and PCF apply the configuration and then asynchronously inform success/failure of the configuration action to the network orchestrator. This status is then propagated to the user through the GUI.
   g) A simulated 5G call is performed, following confirmation of a successful call establishment using sanity checks on call traces and Grafana (observability).
   h) Importantly, we also show that this 5G call is utilising, the AI Engine provisioned configuration.
   i) Next, we perform extended data browsing for a long enough duration with the simulated User Equipment (UE) so that the allocated data quota is exhausted. The UE is redirected to the Fair Usage Policy (FUP) redirection URL as per the tariff plan, for the purpose of recharging.

This demo uses operational and observability tools at various steps to demonstrate how the AI Engine generated parameters are being realised in the network configuration. This comprehensive two-step process rigorously validates the configuration provisioned in the network entities.

[1]https://grafana.com/

References

[1] S. Chakraborty, N. Chitta, and R. Sundaresan, "Automation of network configuration generation using large language models," in *Submitted to the 4th International Workshop on Analytics for Service and Application Management (AnServApp)*, 2024.

[2] "Preparing your dataset with gpt during fine-tuning," 2022. [Online]. Available: https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset

[3] P. H. Martins, Z. Marinho, and A. F. T. Martins, "Joint learning of named entity recognition and entity linking," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, F. Alva-Manchego, E. Choi, and D. Khashabi, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 190–196. [Online]. Available: https://aclanthology.org/P19-2026

[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, and B. Fuller, "Llama 2: Open foundation and fine-tuned chat models," Jul 2023. [Online]. Available: https://arxiv.org/abs/2307.09288

[5] D. M. W. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2020. [Online]. Available: https://arxiv.org/abs/2010.16061

[6] I. Ahmad, E. Suwarni, R. I. Borman, Asmawati, F. Rossi, and Y. Jusman, "Implementation of restful api web services architecture in takeaway application development," p. 132–137, Oct 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9649679/

[7] ETSI, *5G; System Architecture for the 5G System (3GPP TS 23.501 version 15.3.0 Release 15)*. ETSI, 2018.

[8] A. Mehdi, M. K. Bali, S. I. Abbas, and M. Singh, ""unleashing the potential of grafana: A comprehensive study on real-time monitoring and visualization"," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1–8.