# Digital Assistant for Online Capacity Planning

Pham Tran Anh Quang, Pierre Bauguion, Paolo Medagliani, Jérémie Leguay
Huawei Technologies, Paris Research Center

***Abstract*—Service providers must strategically plan network capacity over time to achieve an optimal balance between investments and customer satisfaction. Given the vast amounts of monitoring data collected by modern network controllers and the growing use of compute-intensive AI, we demonstrate an online capacity planning module that utilizes traffic prediction and network behavior models derived from historical data to accurately recommend capacity expansions. Additionally, we show how network administrators can benefit from a conversational assistant powered by a large language model (LLM), which can recognize their intents, automatically suggest expansions, and provide detailed explanations when needed.**

## I. Introduction

Efficient network capacity planning tools are a must for service providers to improve user experience while mastering costs. Commercially available software (e.g., WAE Design, Paragon Planner) can load network configuration, to retrieve the topology and routing policies, and they offer the possibility to test traffic and failure scenarios (designed separately). A number of research works have been done on traffic prediction [1] and their use for capacity planning [2]. However, these solutions are completely offline and do not take advantage of the huge amount of data available in network controllers. Therefore, we argue that a tighter integration with network controllers would help operators to take more accurate and informed decisions with online capacity planning.

In this demo, we present a capacity planning module that is embedded into the controller. As a rich set of data can be accessed, it can leverage past traffic information and accurate traffic predictions, and also build a behavioral network model for better QoS assurance. Implemented as a digital assistant, a conversational LLM-based agent can recognize operator's intents and timely provide recommendations and explanations. The agent also operates with operational constraints, such as a maximum budget, and provides insights about the risks associated with under-investment in terms of performance degradations. This demo will showcase how generative AI combined with optimization can help network management [3].

## II. Online Capacity Planning

To generate relevant capacity recommendations based on historical data, traffic predictions are issued to produce a set of representative Traffic Matrices (TMs). To keep this set minimum, we only keep the dominant TMs, i.e. the matrices that may require a network expansion. More formally, we consider a network $G = (V, A)$ where $V$ is the set of nodes and $A$ is the set of arcs. Let $C : A \to \mathbb{R}^+$ (resp. $P : A \to \mathbb{R}^+$) be an arc-capacity (resp. arc-cost) function. We denote as $T$, the set of TMs, and for each $t \in T$, we use the triplet $(s^k, d^k, b^k)$, for each commodity $k \in K^t$, where $s^k$ denotes the source, $d^k$ the destination, and $b^k$ the bandwidth consumption.

We have two levels of decision variables: (a) $c_a \in R^+, \forall a \in A$, i.e., the amount of additional installed capacity, and (b) $x_a^k \in \{0, 1\}, \forall k \in K^t, \forall t \in T$ to decide if the commodity $k$ is routed along the arc $a$. We point out that while the routing decisions are different for each TM, the installed capacities will be the same among all TMs.

Due to space limitation, we introduce an over simplified optimization model, which we have extended in our implementation to include a QoS model and additional system constraints, such as budget and non-linear installation costs. The MILP can be formulated as follows:

$$\min \quad \sum_{a \in A} c_a P_a$$

$$\sum_{k \in K^t} x_a^k b^k \leq MLU(C_a + c_a) \quad \begin{matrix} \forall a \in A, \\ \forall t \in T, \end{matrix} \tag{1}$$

$$\sum_{a \in \delta^+(i)} x_a^k - \sum_{a \in \delta^-(i)} x_a^k = \begin{cases} 1 & \text{if } i = s^k, \\ -1 & \text{if } i = d^k, \\ 0 & \text{otherwise} \end{cases} \quad \begin{matrix} \forall i \in V, \\ \forall k \in K^t, \\ \forall t \in T, \end{matrix} \tag{2}$$

$$x_a \in \{0, 1\} \quad \forall a \in A, \forall k \in K^t, \forall t \in T,$$
$$c_a \in R^+ \quad \forall a \in A.$$

The objective here is to minimize the capacity installation costs subject to two sets of constraints. Constraints (1) force the load (link utilization) of each arc $a$ to be lower than a target Maximum Link Utilization (MLU) for every possible TM, while constraints (2) are typical flow conservation constraints.

Depending on user intents, the previous model is adjusted. For example, the objective can be to minimize MLU, while an installation budget is given. Other use cases include end-to-end QoS constraints, where a delay model can be introduced. In this case, the problem becomes non-linear and optimization techniques such as outer approximations are used. Once solved, a "what-if" analysis is performed to evaluate the solution considering based on past and predicted TMs, and failure scenarios to provide feedback and insights to the user. The whole architecture is depicted in Fig. 1.

## III. Digital Assistant

In this section, we introduce the architecture of the digital assistant that can recognize the intent of users and find an appropriate answer or solution for them.

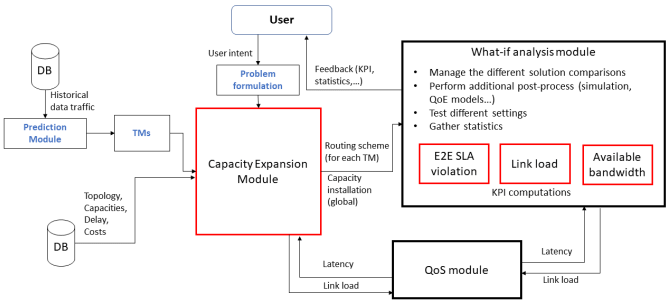


Fig. 1: Capacity planning module architecture

**Intent and Entity recognition.** To interpret user intents and extract relevant parameters, we perform prompt engineering telling an LLM, such as OpenAI ChatGPT or Llama, to behave as an intent classifier, and we use few-shot learning to teach it how to extract information. Indeed, we provide the LLM with a small set of examples (e.g., user questions or commands). Based on these examples, the LLM can classify user intents into predefined categories and extract associated parameters.

For this demonstration, we defined two categories: (a) *Expansion* if users wish to expand their network, and (b) *Explanation* if users seek insights about network behavior and recommended expansion. If an expansion is requested, the digital assistant calls the capacity planning module, after extracting the relevant parameters. For instance, the input "Compute the new network expansion to guarantee link utilization below 50%" is recognized as a capacity expansion intent with parameter LU $<$ 50%, while "How many links will have LU around 70% in two days?" is recognized as an explanation.

**Explanations with RAG.** When users seek explanations, the digital assistant must process large volumes of data (historical data, predictions, recommendations). To efficiently extract relevant information, the Retrieval-Augmented Generation (RAG) framework is employed. A crucial step in RAG is text vectorization, which ensures precise retrieval of pertinent data. For instance, the intent "How many links will have LU around 70% in two days?" is mapped into ['Explanation', 'no', 'link', 'failure', 'time', '2', 'current', 'network', 'link', 'utilization', '70%']. To mitigate this challenge, we leveraged two models for similarity search, BAAI/bge-m3 and all-MiniLM-L6-v2, available from Hugging Face. As each model returns the two closest chunks with the associated indexes, the one with the highest cumulated score is selected and returned to the LLM together with the user's query, in order to generate accurate and comprehensive explanations. The whole workflow is presented in Fig. 2.

## IV. Demonstration

Fig. 3 presents the graphical interface based on Streamlit and Chainlit. For each recommended expansion, it performs a what-if analysis and shows statistics about utilization, installed capacity, expected SLA (Service Level Agreement) violations. Thanks to a chat area, the operator can send natural language requests to the assistant. Manual control is also possible.

In the demonstration scenario, the SLA for the actual TM is met, while the SLAs for past TMs have been violated due to

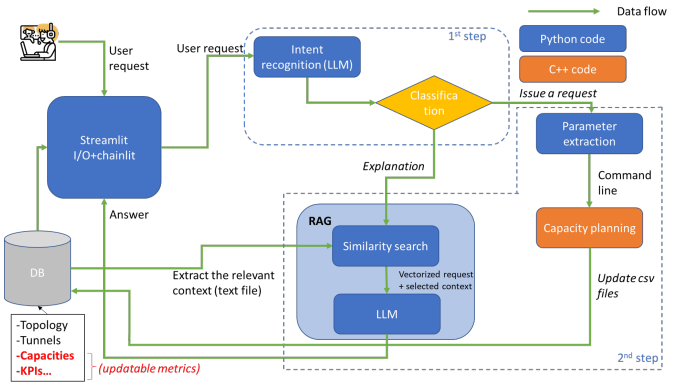


Fig. 2: Digital assistant architecture

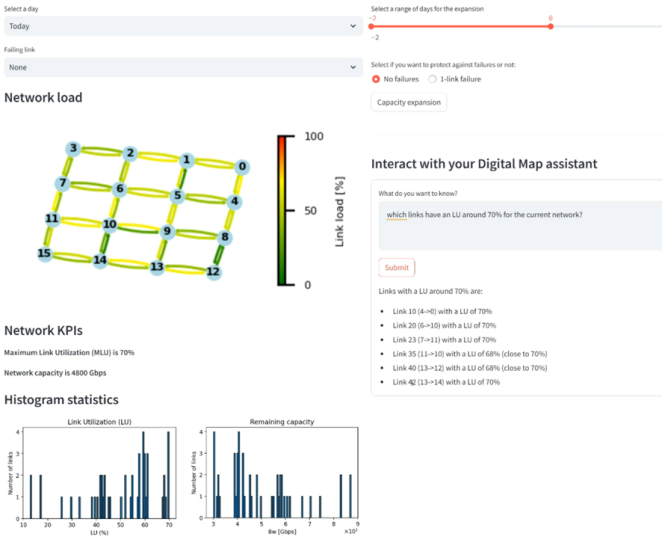


Fig. 3: Demonstrator GUI

latency issues for certain tunnels. To resolve this, we ask the digital assistant to recommend a proper expansion for the past TMs, ensuring that SLA violations are eliminated. After these adjustments, the SLAs for both the current and past TMs are fully met. However, the SLAs for future TMs are not satisfied. To address this, we ask the assistant to consider both past and future TMs. Then, we tell the assistant that we finally have a budget constraint. It automatically recognizes operators intents and adjusts the recommendation. At different steps of the demo, we ask the assistant to provide explanations about utilization, installed capacities and SLA violations. The demo fully shows the power of LLMs to extract knowledge and even make calculations when explaining.

A video is available here: https://tinyurl.com/capacityLLM

## References

[1] Gabriel O Ferreira, Chiara Ravazzi, Fabrizio Dabbene, Giuseppe C Calafiore, and Marco Fiore. Forecasting network traffic: A survey and tutorial with open-source comparative evaluation. *IEEE Access*, 11:6018–6044, 2023.

[2] Alexandra Verhagen. Technical report from KPN - Capacity planning by Network Traffic Prediction. 2024.

[3] Hao Zhou, Chengming Hu, Ye Yuan, Yufei Cui, Yili Jin, Can Chen, Haolun Wu, Dun Yuan, Li Jiang, Di Wu, et al. Large Language Model (LLM) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities. *IEEE Communications Surveys and Tutorials*, 2024.