StackNet: IoT Network Simulation as a Service

Samir Si-Mohammed[1,2], Zakaria Fraoui[2], Thomas Begin[1],
Isabelle Guérin Lassous[1], and Pascale Vicat-Blanc[2,3]

[1]Univ Lyon, ENSL, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France
[2]Stackeo, Lyon, France
[3]Inria, France
[1]{samir.si-mohammed, thomas.begin, isabelle.guerin-lassous}@ens-lyon.fr
[3]pascale.vicat—blanc@inria.fr

***Abstract*—The Internet of Things (IoT) is transforming all economic sectors by connecting physical assets to the virtual world. In this context, the IoT network plays a critical role. It must meet the communication requirements and business constraints of the targeted application to deliver the true value of a smart solution. Small scale technical experimentation is not sufficient to comprehensively evaluate the future behavior of the network. To deeply analyze the performance and the scalability of a solution design, one proven method is to rely on simulation. Simulation is a cost- and time-effective approach to secure infrastructure choices and convince business decision-makers to industrialize an IoT solution at large scale. However, the network simulation process is too complex for most IoT teams. In this article, we show how IoT network simulation can help to future-proof an IoT connectivity design, without the burden of installing a lot of hardware and writing complicated scripts. Then, we propose a no-code online IoT network simulation platform, hiding the simulation process complexity, to make this powerful tool accessible to IoT teams. In particular, we explain how we transform, via relevant abstractions, the network simulation workflow into intuitive interactions. We illustrate the usage of our approach and tool to easily evaluate what-if scenarios in order to answer a set of questions that may arise all along the lifecycle of a smart connected solution.**

***Index Terms*—Internet of Things; Smart Industry; Connected Solutions, No-code; Software as a service; Network; Simulation.**

I. Introduction

The Internet of Things, or IoT, defined as the convergence of the digital and physical worlds, is a fundamental pillar of the digitization of industrial enterprises and is becoming the beating heart of their operations. Amongst popular IoT solutions, we can list asset tracking for smarter transportation, smart building for more energy-efficient commercial areas, predictive maintenance for Industry 4.0, smart metering for efficient utilities, or video-surveillance for safer cities.

For more than ten years, remarkable progress has been made in network technologies and protocols to serve the increasing number of such connected applications. The range of connectivity options offered to IoT architects keeps growing. From new low-power communication technologies like LoRaWAN, or Sigfox, Wi-Fi HaLow, 6LowPAN, NB-IoT to Wi-Fi, 5G or satellites, the choice is huge. These technologies differ from each other in terms of radio spectrum, hardware, protocols, medium access and sharing, environmental footprint, parameters, complexity, operations and capacities.

The variety of network technologies is continuously widening the range of possible IoT use-cases. However, too many possibilities often make it hard for industrial specialists to choose the right technology and configuration settings, yet these are crucial decisions. In particular, an under-sized network can be prejudicial to the quality of experience and consequently to the solution adoption. On the other hand, over-sizing can challenge the whole solution's profitability and environmental impact, thus the stakeholder's commitment. The current approach to test the adequation of a technology for a specific use-case is to deploy a Proof of Concept in labs and potentially a small pilot in real situation with a sample of instrumented assets. However, these experiments do not strongly secure a production deployment at large scale. Analytical models can be of a great use to overcome the deployment complexity of real experimentation. For instance, [1] proposes a graph-based model for estimating the attained throughput in a Wi-Fi network. This model is then used as a decision-support tool by a network architect for channel allocations and frame aggregation. A stochastic geometry framework for modeling the performance of a single gateway with LoRa is, for instance, proposed in [2]. Nonetheless, we argue that analytical models sometimes lack precision and neglect important parameters for the network technologies analysis.

To address the need for in-depth evaluation, network simulation appears as a key enabler. Indeed, it can provide good insights about the performance of a technology and permits to test what-if scenarios at scale to trade-off cost, QoS and energy efficiency. Network simulation complements small scale PoC for large scale assessment and is cost efficient since there is no need to massively deploy real IoT equipment.

However, simulators like ns-3 [3] require network expertise and knowledge in C++ programming to design experiments, to run them but also to analyze and exploit the results. Network simulators have been designed by network experts targeting network researchers and programmers, not product managers, industrial teams or even IoT architects, who do not have the time and skill to perform sophisticated simulations.

In this paper, we present the work we have done to make the network simulation power accessible and at the service of IoT teams. We propose an online no-code IoT network simulation platform to democratize the use of simulation and reach a community of non-network-experts. For this, we have developed a model-based approach and its associated intuitive interface for setting up, running and analyzing simulations without writing a single script. We believe that such a tool can help accelerate the standardization of IoT practice to boost industry digitization and encourage contributions to open source IoT network simulators.

The remainder of this paper is organized as follows: Related works are analyzed in Section II. The problem statement is presented in Section III. An overview of our approach is given in Section IV. Section V illustrates how our proposal can be leveraged to answer typical IoT network questions around a smart building use-case. Conclusion and future works are given in Section VI.

II. Related Works

Numerous academic works on evaluating the performance and the scalability of IoT protocols or applications have been carried out and analyzed in [4] The authors stress that the design of complex IoT systems requires the implementation of testbeds on a very large scale, or the use of scalable simulation tools. A substantiated study of the challenges of IoT modeling and simulation is offered in [5]. In [6], the authors proposed a framework for the evaluation of IoT network technology using simulation, giving special attention to the energy consumption efficiency. [7] compares between Wi-Fi HaLow, LoRaWAN and NB-IoT for smart city applications, using simulation. In [8], the authors discuss the different types of IoT simulators used in the academic world. The full-stack simulators like iFogSim [9] provide support for simulating the end-to-end IoT chain, not specifically the network performance. Other simulators focus on the big data processing aspects of IoT applications like IOTSim [10] and SimIoT [8]. Classical network simulators were not created specifically for the IoT, but have progressively evolved to include aspects necessary for IoT studies. The survey conducted in [8] includes more than 30 network simulators like OMNeT++[11], ns-3 [3], Cooja [12]. It is worth noting that in all these studies, the accessibility of network simulation to non-network-experts like IoT architects has never been taken into account. Therefore these powerful tools are rarely used by IoT teams. On the other hand, the concept of no-code is becoming a very popular trend in IoT, as it permits manufacturers and operation managers to program their IoT applications while reducing the time and expertise needed. According to analysts, 70% of IoT applications are going to be coded via no-code in 2025 compared to 25% of them in 2020 [13]. Ultimately, IoT architects want to be able to program their IoT solution from building blocks without the need of having a deep expertise in networking, or facing the IT human resource bottleneck with its project backlog. In this perspective, [14] proposes an end-to-end low-code mechanism for managing the relationship between heterogeneous hardware sensors and IoT platform. [15] is a first attempt to simplify the programming of IoT network simulations. This tool proposes ns-3 templates but requires network expertise. In our work, we address IoT architects to let them seamlessly benefit from network simulation via an interactive tool, for what-if scenario creation and comparison. This will enable them to future-proof the IoT infrastructure while developing IoT applications. We show how the online no-code approach simplifies the network design and test phases of these complex projects, making the non-programmer community able to run IoT network simulations and gather performance insights in an easy way.

III. Problem Statement

A. Evaluating an IoT network technology at scale

Selecting and tuning the network, that best fits the needs of an envisioned smart application, is one important and complex problem the IoT architect faces. To examine this issue, let us use a smart building example. We consider the case of the instrumentation of a commercial building already equipped with a BMS (Building Management System) in charge of the remote control of HVAC (heating, ventilation, and air conditioning) systems as well as water and energy regulation. The facility manager wants to add a smart solution to finely monitor the building, to gain better visibility on its real usage and to better adapt the building services. For this, the facility manager would like to deploy a range of sensors: Entrance detectors, occupancy monitors, air quality sensors, temperature sensors, smart lighting and other end-devices. This customer is working with an IoT architect, who is proposing several sensors supporting various communication technologies. To demonstrate the feasibility of the project and to qualify the sensors for the solution, the solution architect has developed a proof of concept (PoC) of the smart solution that shows how sensors collect data and how the prototype application exploits and visualizes them.

1) Asking the right questions: After having convinced the customer, the architect has to select and define the detailed configuration of the network together with the sensors for the targeted deployment, and integrate the final solution. They have several options for the network, typically LoRaWAN, Wi-Fi, Wi-Fi HaLow and 6LowPAN and a range of questions that the small PoC has not truly answered:

- How would LoRaWAN and Wi-Fi capacities compare for this application?
- How long will the battery last for this traffic workload?
- What are the best settings for maximizing the solution quality in this specific commercial building?
- How many devices would one gateway support?

2) Analyzing the network behavior under application workload: To answer this type of questions, network experts would typically simulate the behavior of network candidates under application workload, considering the environmental conditions and the application traffic and topology (number of sensors, their location, etc.). They will evaluate if the traffic sent by all devices will be properly supported by the network, then estimate what percentage of packets will be successfully transmitted and how much energy will be consumed in different scenarios.

To analyze their smart building scenario in depth and at scale, the IoT architect would have to adopt the same approach: Define the application requirements as well as the network setup to be tested using a network simulator, then run a first simulation to establish the base-line and validate the parameter assumptions with one technology and a basic application scenario. Then, they would have to proceed with what-if scenarios comparison. This process requires specific skills as we detail below.

B. Coding a simulation

The network expert generally codes network experiments into the appropriated language for their simulator, for example, in C++ for the ns-3. This code globally works as follows: (i) It takes input parameters, (ii) creates and executes the corresponding network nodes and traffic, (iii) calculates the Key Performance Indicators (KPIs) obtained from the simulation.

1) Technology comparison: After the base-lining step, one has to conduct technology scenario comparisons. This step consists in setting up and running a set of simulations for: (i) Various network technologies to select the best of them and (ii) different network configurations to select the best network settings for the best network technology.

2) Analyzing the scalability: After the network choice has been established, the scalability analysis consists in analyzing how an increase in load in terms of number of devices or traffic intensity affects the application performance of the best network technology. This analysis is crucial because the applications and device deployments supported by the connectivity infrastructure are supposed to evolve and grow over time. For example in the smart building case, the number of sensors is expected to increase and the existing topology to change when various floors of the building are equipped. Estimating in advance the KPIs at the limits of the system capacities will ensure that the choice of the technology will survive the solution's scaling and that the cost or the footprint will not explode.

C. Dealing with simulation complexity

As we can see, this simulation workflow is far from the reach of IoT architects, who are not necessarily network experts. Consequently, they naturally tend to overlook the systematic analysis of the technology they deploy. This often leads to project failure at short, mid or long term [16].

To change this, we propose to abstract as much as possible the complexity of the simulation process and guide IoT architects in the in-depth performance evaluation of IoT network technologies for ensuring their IoT solutions design and evolution.

IV. Proposed Solution

In this section we explain how we transform the complex simulation process detailed above and encapsulate the simulation and analysis code described in a no-code automated network simulation software named StackNet.

A. Principles

Guiding the IoT architect in the methodological evaluation via a high-level, easy-to-use interface, requires i)to simplify and structure the network performance evaluation process, ii) to propose pre-built components for application and network parameter settings, iii) to automatically code simulation scripts, iv) to deploy the simulator underneath and give online access to it, v) to gather and organize the simulation results and vi)to automatically create easy-to-understand, explainable and interactive dashboards.

B. Methodology

To analyze the performance and the scalability of various network options with StackNet framework, an IoT architect will perform the tasks listed below:

1) Identify key questions to answer for correctly supporting the targeted application.
2) Describe the application by initializing its important parameters.
3) Define a set of simulation scenarios to compare various networking alternatives.
4) Define a set of simulations to analyze the impact of the potential increase of fleet size or of workload in the future.

C. Modeling

The role of a no-code network simulator is to abstract what is an IoT scenario in order to be able to capture any IoT use-case and "code" it automatically in the simulator (for example in ns-3) language. To achieve this, and as illustrated in Figure 1, we segregate the description of an IoT scenario in two parts: The application part and the network part. Indeed, application parameters can be easily defined by the IoT architect or the application developer while the network settings are the difficult aspects of the simulation.

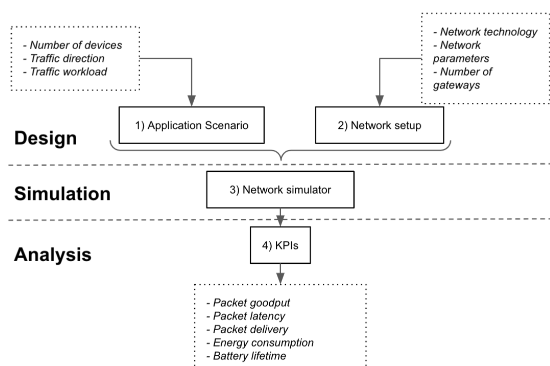


Fig. 1: Scenario simulation building blocks.

1) Abstracting the application: An IoT solution often relates to sensors/actuators sending/receiving data to/from a remote application. The application is characterized by the number of connected sensors or actuators and the workload, which is the traffic they submit to the network. For example, in the smart building use-case described above, the application will be modeled by the number of sensors, the size and frequency of temperature samples sent to an IoT Hub, then to analytics pipeline and finally to the application's back-end, generally located in the cloud. This similar flow applies to other types of sensors. From a network simulation perspective, the number of sensors as well as the profile of data collected and transmitted over the communication channels have to be defined by the IoT architect. They are the best person to define these application characteristics. They have the knowledge of the sensors' details, the application and usage of the data. All this information should be formalized and initialized with a set of valued parameters, typically the traffic direction and workload. The external conditions affecting the radio signal also matter: We distinguish indoor and outdoor environments. The latter can be rural, urban or semi-urban presenting various behaviors in terms of radio propagation. In StackNet, a collection of pre-defined IoT application templates are proposed to accelerate the initialization process.

2) Abstracting the network: The specification of an IoT network simulation scenario needs a network model defined by a list of parameters representing the network technology and topology. These parameters can be divided as follows: (i) The IoT network technology as defined by its physical and mac layers, (ii) the number of gateways and their location (topology) and (iii) the low-level parameters related to the network technology (configuration). In StackNet, pre-built network models are made available to the IoT architect, so they can integrate them in simulation scenario without network expertise.

D. Designing the no-code interface

A simulation scenario is defined by the application model, which is set once for several experiments, and the network model and settings on the other side. Various network settings can be selected to compare what-if scenarios. To automatically generate the scripts for running a simulation, our framework provides a dynamic interface to initialize the values of the application model's parameters and the selected network settings of a scenario, as illustrated in the left column of the StackNet's interface of Figure 2. Pre-defined network settings with default values are proposed for each supported technology. These predefined settings can be easily uploaded then adjusted manually. The interface enables to create and save multiple scenarios.

Once the application model and the network settings are initialized, the simulator is invoked. The simulation script is automatically generated, the simulation executed, and the resulting output metrics (KPIs) interactively visualized in scenario dashboards. To permit the user to analyze and explain one particular scenario in context, he can open the input parameters panels together with the results dashboard.

The results dashboard gathers and highlights the KPIs related to the IoT connectivity solution under study in the specified application context as illustrated in the Figure 2. The network performance evaluation focuses on five KPIs. These KPIs are: (i) Packet goodput, (ii) packet delivery, (iii) packet latency, (iv) energy consumption and (v) battery lifetime. We consider that, together, these metrics provide a fair representation of the performance of an IoT network technology for a given scenario.

If the application requires high quality data, like for precision air quality monitoring in the operating room of a hospital, then, estimating the packet delivery is of utmost importance. For a use-case like real-time equipment location application, packet latency is what matters the most. If the deployment of sensors and the battery change are difficult, energy consumption and battery life time have critical importance.

As the IoT architect is looking for scenario comparison, a dedicated and intuitive dashboard (*e.g.,* Figure 4) is proposed to display comparative results and charts, and to let the best technology and configuration immediately stand out.

The evaluation methodology has been seamlessly integrated within the interactive front-end while the programming and networking expertise have been encoded and hidden within the back-end. This makes the simulation experience smooth and quick, allowing the IoT architects to focus on the data and decision process rather than bothering them with programming and results collection complexity. Figure 3 illustrates the no-code workflow for creating and setting various scenarios for comparison, and the corresponding mapping to the simulation workflow activated underneath.

V. USE-CASE EXAMPLE

This section details how an IoT architect would use StackNet for the aforementioned smart building use-case. The application model is defined by the following: 100 end-devices are placed in a building. They are separated by a distance of around 200 meters. They send one packet of 100 bytes every 2 minutes. For the scalability analysis, the network density is scaled (up to 600 end-devices) and the traffic workload is increased (one packet of 110 bytes is sent every 90 seconds). The IoT architect introduces the application model inputs. Then, they create various scenarios for different network technologies, typically Wi-Fi, 6LoWPAN, 802.11ah and LoRaWAN. Let us consider how our IoT architect can leverage the network simulation service to easily answer the questions asked in Section III:

1) How would LoRaWAN and Wi-Fi capacities compare for this smart building application?: Figure 4 displays the performance synthesis of the different technologies explored. We can easily observe that all the technologies perform the same in terms of goodput and packet delivery, *i.e.* all the packets are correctly received. However, LoRaWAN clearly outclasses the other technologies, including Wi-Fi, in terms of battery lifetime (up to 700 days). Therefore, we can say that LoRaWAN is more relevant than Wi-Fi in the basic version of this use-case.

2) How long will the battery last?: The battery lifetime of Wi-Fi can last approximately 3 months (90 days) while 6LoWPAN can go up to 5 months (150 days). The difference is not that big, but it shows that 6LoWPAN is better designed and suited for this IoT applications than Wi-Fi. For LoRaWAN, it can go up to approximately 2 years.

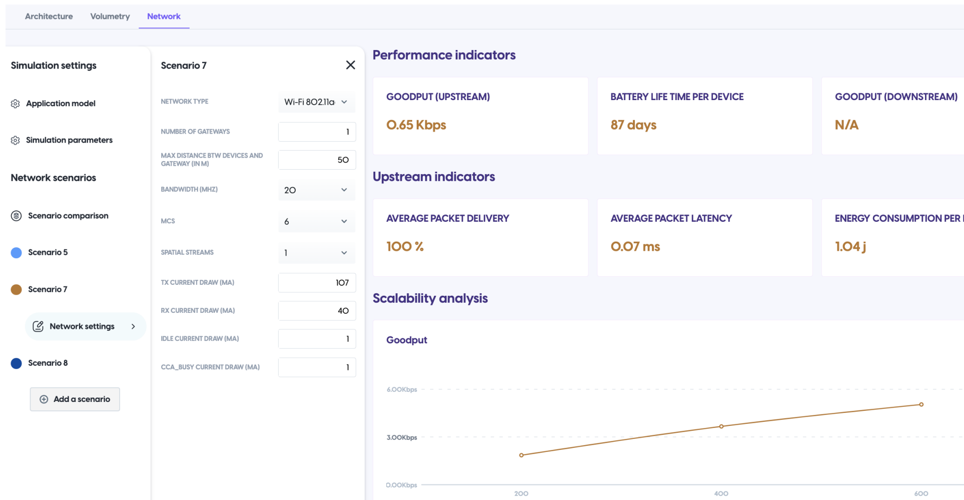


Fig. 2: Dynamic interface for setting and analyzing a scenario.

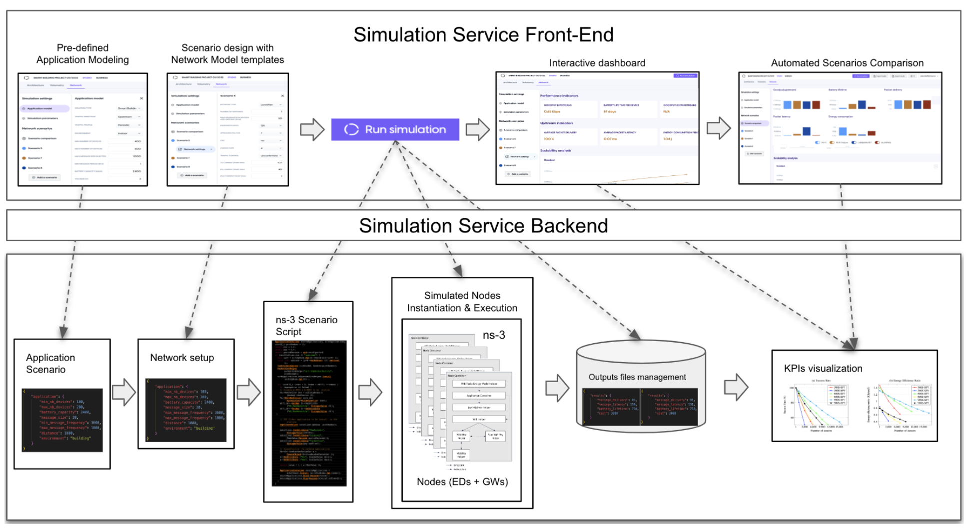


Fig. 3: Illustration of the no-code workflow and its mapping to expert simulation for IoT scenarios analysis.

3) What are the best settings with the LoRaWAN technology (spreading factor) to maximize the solution quality in this specific commercial building?: Figure 5 shows how the IoT architect can further explore some configurations of LoRaWAN with different spreading factors (7, 8 and 9) to determine the impact of this parameter on the performance. Except covering longer distances, we observe that higher SF is less interesting in terms of all the considered KPIs. Packet latency is increased, which makes collisions more likely, and therefore lessens packet delivery. Additionally, end-devices spend more time to send a packet, and thus consume more energy, leading to lower battery lifetime. We see that a lower SF results in lower message latency, better message delivery and longer battery lifetime. In this case SF7 is the best setting.

4) How many devices would one gateway support?: Figures 6a and 6b highlight the various network technologies' performance in terms of packet delivery and packet latency for the scaled settings. We see that Wi-Fi HaLow, 6LoWPAN and Wi-Fi behave well in terms of packet delivery. However, it drops faster for LoRaWAN (down to 60% for 600 end-devices). Setting and running these experiments only took a dozen of minutes. The scalability study shows that on one hand, one LoRaWAN gateway can handle up to 200 sensors before deteriorating the performances, while on the other hand one Wi-Fi HaLow gateway can handle 400 sensors without noticing any degradation in the performances. The node's placement and the radio environment in this use-case make it impossible to use only one gateway for Wi-Fi and 6LoWPAN.

VI. Conclusion and Future Works

In this work, we have presented StackNet, a no-code framework to ease and automate IoT network simulation and its integration in a SaaS (Software as a Service) platform. This approach enables IoT architect to test and compare IoT network technologies without learning and deploying any network

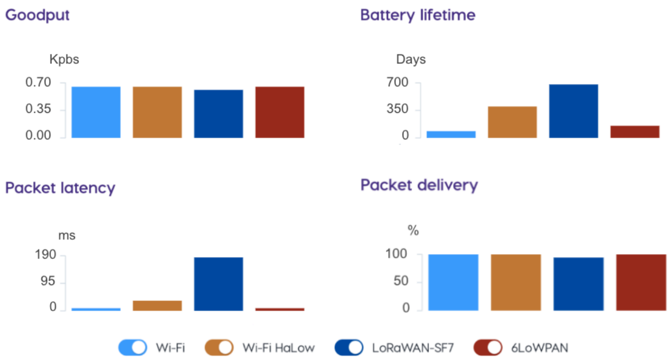


Fig. 4: Results for the network technologies comparison (Q1 & Q2).

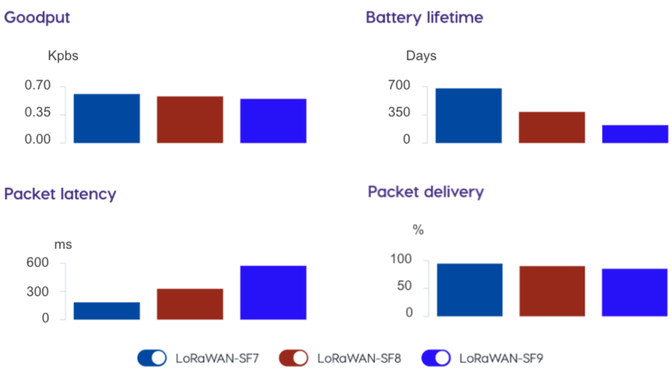
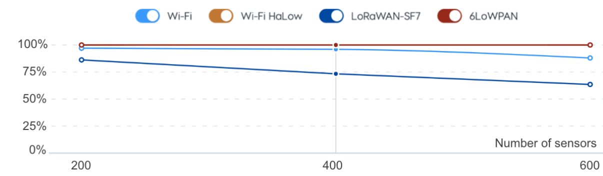


Fig. 5: Results for the LoRaWAN configuration (Q3).

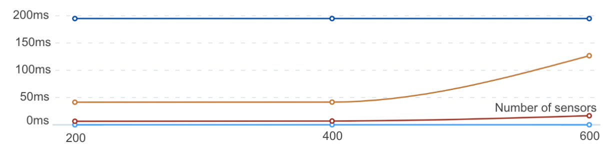
simulator nor coding any script. We began by identifying the network evaluation problems that IoT architects face on their journey, and the difficulty of running network simulations to overcome them. Then, we described our approach by highlighting the salient aspects that need to be taken into consideration for hiding the complexity of IoT simulation while empowering architect with an intuitive solution to design and compare alternative scenarios.

An application of our methodology and tool on a smart building use-case has been presented. We put the emphasis on the ease of simulation initialization, on results visualization, on what-if scenarios comparison. We show how the no-code framework helps IoT architects ask and answer their own design questions. The tool currently supports simulations with Wi-Fi, LoRaWAN, Wi-Fi HaLow, 6LoWPAN and 5G mmWave, but is limited by the availability of validated simulation modules in ns-3 for other new technologies. As the no-code and as a service approach can democratize the usage of cost-effective evaluation methods like simulation in IoT, we hope it will encourage the systematic development of ns-3 simulation modules for other IoT network technologies.

As the user interface is simulator-agnostic, we project to adapt the back-end to support other simulators like Omnet++ and even experimental test-beds to leverage the same user-friendly tool for technology and configuration comparison. In



(a) Packet delivery.



(b) Packet latency.

Fig. 6: Results for the scalability study (Q4).

terms of future works, we will also extend StackNet with a decision support engine to generate recommendations for network selection and configuration. We will also combine network simulation with end-to-end solution simulation for precise workload estimation, cost and environmental impact analysis. StackNet can be freely accessed via the Stackeo platform here https://app.stackeo.io/.

References

[1] M. Stojanova, T. Begin, and A. Busson, "Conflict graph-based model for ieee 802.11 networks: A divide-and-conquer approach," Performance Evaluation, 2019.

[2] O. Georgiou and U. Raza, "Low power wide area network analysis: Can lora scale?" IEEE Wireless Communications Letters, 2017.

[3] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in Modeling and tools for network simulation. Springer, 2010.

[4] G. D'Angelo, S. Ferretti, and V. Ghini, "Simulation of the internet of things," in 2016 International Conference on High Performance Computing & Simulation (HPCS). IEEE, 2016.

[5] Kecskemeti, Gabor et al., "Modelling and simulation challenges in internet of things," IEEE cloud computing, 2017.

[6] S. Si-Mohammed, T. Begin, I. Guerin Lassous, and P. Vicat-Blanc, "Adiperf: A framework for application-driven iot network performance evaluation," in International Conference on Computer Communications and Networks. IEEE, 2022.

[7] Verhoeven, Richard et al., "Performance evaluation of wi-fi halow, nb-iot and lora for smart city applications," in PE-WASUN, 2022.

[8] D. Pandey and V. Kushwaha, "Experimental tools and techniques for wireless sensor networks," International Journal of Recent Technology and Engineering (IJRTE) ISSN, 2017.

[9] Gupta, Harshit et al., "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," Software: Practice and Experience, 2017.

[10] Zeng, Xuezhi et al., "Iotsim: A simulator for analysing iot applications," Journal of Systems Architecture, 2017.

[11] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in 1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems, 2010.

[12] Osterlind, Fredrik et al., "Cross-level sensor network simulation with cooja," in IEEE conference on local computer networks. IEEE, 2006.

[13] Gartner, https://www.gartner.com/en/newsroom/press-releases/.

[14] Pantelimon, Silviu-George et al., "Towards a seamless integration of iot devices with iot platforms using a low-code approach," in IEEE World Forum on Internet of Things (WF-IoT). IEEE, 2019.

[15] S. Si-Mohammed, M. Janumporn, T. Begin, I. Guerin Lassous, and P. Vicat-Blanc, "Sifran: Evaluating iot networks with a no-code framework based on ns-3." in 2022 Latin America Networking Conference (LANC). ACM, 2022.

[16] A. Aramayo-Prudencio et al., "Digital-manufacturing-escaping-pilot-purgatory," McKinsey Global Institute research report, 2018.