

Advancements and Challenges in IoT and Network Simulators

Subjects: Computer Science, Hardware & Architecture

Contributor: Reham Almutairi, Giacomo Bergami, Graham Morgan

The Internet of Things (IoT) has emerged as an important concept, bridging the physical and digital worlds through interconnected devices. Although the idea of interconnected devices predates the term “Internet of Things”, which was coined in 1999 by Kevin Ashton, the vision of a seamlessly integrated world of devices has been accelerated by advancements in wireless technologies, cost-effective computing, and the ubiquity of mobile devices. In this regard, Cloud infrastructures are crucial for orchestrating and collecting data from these ubiquitous devices.

Keywords: IoT simulators ; IoT simulation challenges ; IoT simulators performance

1. Current Simulation Platforms

IoTNetSim ^[1], is an advanced platform for modeling and simulating end-to-end Internet of Things (IoT) services and networks. It is a valuable tool for researchers and practitioners, offering a self-contained, multi-layered architecture to model IoT systems with different structures, application models, services, and network connections. Its distinguishing feature is the detailed modelling of IoT nodes and sensors, including power sources and mobility, enabling highly accurate simulations for testing configurations and algorithms. The platform covers a wide spectrum, from IoT networking, including wired and wireless connections and protocols, to services and applications, and is designed to be modular and extendable. It supports a broad spectrum of IoT components, including mobile nodes and gateways, facilitating the simulation of diverse IoT applications like environmental monitoring and disaster response. This platform enables realistic modelling across physical, networking, and application layers, incorporating cloud, fog, and edge computing models to simulate data flow and processing. IoTNetSim is equipped to model various protocols, including cellular, WiFi, LoRa, and ZigBee, enhancing studies on system performance. Its detailed simulation capabilities also hint at potential applications in creating virtual urban platforms and exploring urban IoT ecosystems. Although not explicitly mentioned, the platform's architecture suggests it could simulate SLoT-specific communication strategies. IoTNetSim emphasizes realistic simulations with features for simulating network and battery failures, highlighting its scalability for large-scale IoT infrastructures. Its modular design indicates compatibility with numerous IoT development tools and programming languages, making it a versatile tool for IoT system research and development. Real-world application examples, such as monitoring natural environments and responding to disasters, showcase IoTNetSim's utility in guiding users from conceptual design to detailed evaluation of complex IoT systems. While it demonstrates its prowess in facilitating the evaluation of large-scale systems, it currently has limitations in supporting certain sensor types due to the complexity of modelling their mobility.

The large-scale NB-IoT (Narrowband Internet of Things) simulator described in ^[2] is a robust machine-to-machine traffic simulator tailored for studying IoT application performance in extensive environments like smart cities. Its unique aspect is integrating real geographical data from smart city open data projects, creating a virtual urban landscape where devices interact with the telecommunications infrastructure. This approach enhances the realism of IoT application simulations and aids in understanding their performance and potential issues. It adeptly simulates a wide array of IoT components, including NB-IoT and LTE devices, capturing their interactions, energy consumption, and mobility within urban ecosystems. This simulator enhances the realistic modelling of IoT applications, from utility monitoring to urban mobility, by integrating real-world geographical data, allowing for precise device positioning within the simulated city. Its multi-layered architecture focuses on connection procedures and packet transmissions, efficiently simulating extensive networks without sacrificing speed or accuracy. Tailored for NB-IoT and LTE technologies, it aligns with 3GPP specifications, offering insights into network efficiency and scalability. The simulator's systematic approach in creating a virtual urban platform starts from real geographical data collection, facilitating the simulation of device interactions across the urban environment. While focusing on NB-IoT and LTE, its architecture suggests flexibility in supporting various IoT protocols and standards, showcasing exceptional scalability for city-wide simulations with numerous eNodeBs and UEs.

through a discrete-event simulation approach. Programmed in Python, it is compatible with other IoT development tools, encouraging integration with data analysis algorithms. The simulator provides a robust framework for examining IoT application performance in urban environments, enabling scenario creation that reflects the complex network behaviour at the city scale. However, since the simulator's focus on NB-IoT and LTE (Long Term Evolution technology) devices and its physical layer abstraction limit, its applicability to other IoT device types and communication protocols potentially provides an incomplete perspective on IoT application behaviour in real-world scenarios.

ASSIST (Agent Simulator for Social Smart Things), outlined in [3], creates a unique simulation tool for the Internet of Things (IoT), aiming to establish a Social Network-inspired environment for IoT entities. It empowers users to define smart Internet of Everything (IoE) entities, characterize their attributes, and establish social connections among them. ASSIST's foundation is a cognitive middleware containing a Social Network of Agents, including SIoT Agents and a Broker Agent, representing the instantiated IoE entities. The SIoT Agents are autonomous, intelligent entities that employ ontologies for knowledge representation and employ a Publish/Subscribe pattern for service exposure and resource consumption. The Broker Agent oversees communication between SIoT Agents and manages their status. The simulator employs a flooding mechanism for SIoT Agents to find required services and a deterministic approach to forming social IoT clusters, distinguishing it from probabilistic methods. It integrates diverse IoT components like sensors and actuators as agents, allowing for rich interactions and social connections based on shared goals or ownership. The simulator excels in modelling IoT applications across scenarios, from environmental monitoring to energy management, using ontologies for knowledge representation and supporting dynamic social connections through a Publish/Subscribe pattern. ASSIST accommodates common IoT communication protocols such as CoAP, MQTT, and HTTP, suggesting its utility in virtual urban platform development and smart city solutions. It adheres to IoT standards and employs semantic web technologies for interoperability, testing IoT environments under various conditions to assess network resilience. Demonstrating scalability, ASSIST can manage extensive networks, indicating compatibility with semantic web technologies and various IoT development tools. A use case involving cultural heritage protection through WSNs, UAVs, and anomaly detection apps showcases its capability to simulate complex social interactions among IoT devices for effective service discovery and collaboration. However, it primarily supports Social Internet of Things (SIoT) environment simulation and SIoT Agent behaviour.

GVSoc [4] is an event-driven simulator tailored for RISC-V-based IoT processors, balancing accuracy and speed. It combines efficient C++ models with flexible Python configuration scripts, making it a valuable tool for researchers focusing on highly parallel and heterogeneous RISC-V-based IoT processors. Unlike the slow but accurate cycle-by-cycle simulators or the fast but less informative simulators, GVSoc delivers reasonably fast simulation with accuracy, enabling rapid exploration of diverse configurations, which is crucial for Design Space Exploration (DSE). It is highly configurable and flexible, with open-source availability for the research community. It supports a comprehensive range of IoT components, including processors and peripherals, enabling the simulation of complex systems with multicore and multi-memory levels. GVSoc excels in IoT application modelling, particularly for evaluating performance and energy efficiency in applications using deep neural networks and near-sensor data analytics. Its multi-layered architecture facilitates high-fidelity modelling of system components, including detailed simulations of the PULP platform. While focusing on structural and functional capabilities, GVSoc implies support for various communication protocols and standards, which are essential for architectural exploration. Its simulation features are particularly suited for Design Space Exploration, offering rapid configuration exploration with a significant speed-up over cycle-accurate simulations and maintaining errors typically below 10 per cent. GVSoc's scalable architecture supports extensive IoT systems simulation and is adaptable for urban infrastructure modelling. Utilizing C++ and Python, it provides a flexible and open-source development environment conducive to community research and IoT processor simulation. Documentation showcases real-life application examples, underscoring GVSoc's role in accurate performance estimation and design exploration for low-power IoT system design. However, it currently focuses on RISC-V processors, lacking support for other architectures like ARM or MIPS, and is designed for near-sensor data analytics applications, including Deep Neural Networks (DNNs).

LoRa-MAB [5] is a versatile Python-based simulator tailored for decentralized learning resource allocation within IoT networks, specifically focusing on LoRaWAN applications. It employs an event-driven simulation framework using the Simply library to replicate LoRa link behaviour across various network scenarios, including the capture effect and inter-SF collision. After a simulation, the simulator provides valuable insights into the network's packet delivery ratio and overall energy consumption. LoRaWAN, a low-power, wide-area network technology commonly employed in IoT applications, constitutes the simulation's primary area. This technology encompasses end devices, gateways, and network servers, forming a star-of-stars network topology. Gateways connect to the network server through IP-based backhaul networks, and LoRa modulation enables energy-efficient, long-range communication. Additionally, the LoRaWAN protocol offers security features such as encryption and authentication to safeguard data integrity and privacy.

MyiFogSim ^[6] is a virtual machine (VM) migration simulator in fog computing that extends the iFogSim simulator. The simulator is designed to support VM migration policies for mobile users, which involves migrating VMs to cloudlets based on user position. This approach can result in lower latencies and better quality of experience (QoE) for users. MyiFogSim incorporates a range of new classes, notably the Coordinate class that depicts map coordinates on a Cartesian plane, and the ApDevice class, an extension of the FogDevice class from iFogSim, equipped with the functionalities and duties of a wireless network access point. Additionally, the simulator introduces a VM migration strategy for mobile users within a fog computing environment, contrasting it with a scenario lacking VM migration. The outcomes of the simulations indicate that implementing the migration policy can lead to reduced latency compared to a setup that does not employ the migration policy.

The co-simulator discussed in ^[7] is a versatile platform developed for assessing the impact of emerging technologies on smart grids. It integrates two established simulators, Gridlab-D for power systems and CORE for communication networks. To tackle synchronization and interaction challenges between these components, the co-simulator employs a Graphic User Interface (GUI) for efficiency, software emulation for fidelity, and an Ethernet-tunnel-based distributed module for scalability. The co-simulator serves as a robust instrument for utility companies and policymakers to implement new IoT devices or strategies in upcoming smart grid frameworks. This platform offers flexibility and scalability, enabling experiments across various scenarios. It can run Linux applications on virtual nodes, using lightweight virtualization, and supports real-time and non-real-time modes, making it a valuable tool for studying the effects of smart grid technologies, particularly when developing a new simulator from scratch is complex and time-consuming.

The simulation platform presented in ^[8] offers an efficient Java-based solution for simulating large-scale IoT systems in urban settings. Its strength lies in simulating thousands of geographically distributed devices, a crucial aspect of extensive IoT deployments that require rigorous testing. Unlike traditional IoT simulators focused on low-level networking, this platform provides a high level of generality, modelling devices with multiple network interfaces and various mobility, network, and energy consumption patterns. This encourages code reuse and efficient development. However, it primarily concentrates on the application-layer perspective of IoT systems, which may limit its suitability for testing low-level networking aspects, as it abstracts device interactions more than specific data transmission and routing details.

The IoT software infrastructure proposed in ^[9] facilitates energy management and simulation in a city district by enabling the integration of diverse data sources and IoT devices. It incorporates real-time building energy profiles, environmental sensor data, and building/grid models, allowing for comprehensive energy consumption monitoring and management. The platform employs REST-based request/response and MQTT-based publish/subscribe communication paradigms, leveraging the LinkSmart OpenSource Middleware. It excels in modelling real-world IoT devices and ICT systems, integrating heterogeneous data sources, and simulating energy policies for district-level energy optimization while following the microservices paradigm. Nevertheless, further expansion is required to support additional IoT devices and data sources, like smart meters, weather sensors, and traffic sensors.

The dynamic co-simulation of Internet-of-Things (IoT) components using a multi-agent system ^[10] is an innovative approach for simulating complex IoT systems in a modular and adaptable manner. This approach enables the separate simulation of IoT components in different simulation tools, with agents representing each component, allowing them to join a running co-simulation dynamically. The connection between agents and simulation tools is facilitated through an interface concept. While promising, this approach involves using multiple simulation tools and the development of agents for each IoT component, which can be complex and time-consuming. Additionally, it lacks a complete solution for adding intelligence to the models, as it mainly optimizes the simulation process by reducing message traffic rather than enhancing agent intelligence.

RelIoT ^[11] is a framework for end-to-end reliability simulation in IoT networks, focusing on energy efficiency and reliability optimization. It integrates power, performance, temperature, and reliability modules into the widely used ns-3 network simulator. This framework estimates device temperature and reliability, a unique feature compared to other network simulators. It aims to balance energy consumption and Quality-of-Service (QoS) constraints, considering reliability as a design parameter that can be optimized. It offers runtime adjustments for the trade-off between performance and reliability and has minimal performance overhead for scalability. While validated with real-world experiments, RelIoT could benefit from support for more complex reliability models and diverse device types and applications, particularly those in challenging environments.

The MoSIoT framework ^[12] is an innovative approach designed for modelling and simulating Internet of Things (IoT) healthcare monitoring systems, particularly for people with disabilities. It utilizes the principles of model-driven engineering (MDE) to facilitate the creation of customizable and efficient healthcare monitoring scenarios. This framework allows

healthcare professionals to easily simulate complete IoT healthcare scenarios tailored for various disabilities and diseases and subsequently generate final IoT systems. Key features of MoSIoT include a set of models for scenario simulation, integration with enterprise cloud architecture for data simulation, and support for commercial IoT hubs like Azure IoT Central. The framework's emphasis on customization, ease of use for non-technical users, and its potential for improving remote healthcare monitoring make it a significant contribution to IoT and healthcare technology.

The hybrid simulation-based testing approach in ^[13] combines simulation and real-life testing to evaluate large-scale IoT applications, aiming to simulate the interactions between local entities (LEs) effectively. It utilizes the IEEE-standardized Parallel and Distributed Simulation (PADS) methodology, particularly implemented through the Gaia/Artis IoT simulator, which can run thousands of virtual LEs in parallel. The main challenge addressed is the scalability required to manage real-time interactions among numerous LEs, achieved partly through cloud-based infrastructure and efficient synchronization mechanisms between real-life and simulation environments. However, the approach is primarily focused on testing large-scale Internet of Things (IoT) applications at the system level, and its main goal is to effectively facilitate interactions between local entities (LEs) in large IoT environments. So, it does not directly impact the use of renewable energy sources.

SimulateloT ^[14] is a Domain Specific Language (DSL) that streamlines the design, code generation, and execution of IoT simulation environments. It offers time and cost savings for developers by enabling the creation of scalable IoT systems without substantial hardware and software investments. Developers define models with numerous Node elements, generating code that adheres to recognized software architecture patterns like publish-subscribe and Docker containers. SimulateloT's flexibility allows for easy adaptation to different target technologies. Key features include Model-Driven Development for handling IoT system complexity, the ability to define various nodes and policies with granularity values, and support for both CloudNode and FogNode. However, the node mobility in this simulator has been partially developed, and the hardware simulation is only managed by the size attribute at ProcessNode, which implies several constraints to avoid creating specific software elements. Second, the current version of the simulator IoT environment only allows defining connected nodes by TCP/IP, and it assumes that connectivity is guaranteed.

SimulateloT-FIWARE ^[15] is a domain-specific language (DSL) that enables the design, code generation, and execution of IoT simulation environments on the FIWARE platform. The language is based on the SimulateloT metamodel ^[14], which defines the concepts and relationships required to model IoT systems. SimulateloT-FIWARE extends the SimulateloT to include FIWARE-specific concepts and relationships, such as context brokers, IoT agents, and IoT devices. The language also includes a set of M2T transformations that generate code for specific FIWARE technology, such as Orion Context Broker and IoT Agents. SimulateloT-FIWARE is flexible and scalable and based on open-source technology, allowing developers to choose the necessary components and integrate them with other technologies as required. However, SimulateloT-FIWARE is tailored to the FIWARE platform, which may limit its applicability to other IoT platforms.

EdgeCloudSim ^[16] is a simulation framework tailored for evaluating IoT services over Edge and Cloud systems. It excels in its fine-grained analysis, covering service time, resource utilization, energy consumption, and service reliability. Notably, it accommodates mobile devices with diverse hardware characteristics, expanding its applicability. The framework stands out for its dual consideration of computational and network aspects, encompassing WLAN and WAN communication models, device mobility, load generation, and virtual machine usage models. EdgeCloudSim's capacity to detail IoT service provisioning and the Edge/Cloud trade-off empowers researchers and practitioners. Furthermore, it introduces novel functionalities, like dynamic device hardware configuration, improved output, and expanded service simulation. However, its limitation lies in potentially missing nuances of diverse hardware features, potentially leading to inaccuracies in simulation results.

IoTSim-Edge ^[17] is a simulation framework addressing IoT and edge computing challenges. It enables data-driven decision-making in smart environments like homes and transport. The framework models various aspects, including device diversity, communication protocols, mobility, and battery features. It supports mobile IoT devices and handles handoffs between edges for consistent communication. IoTSim-Edge is built on existing simulators, capturing the complete behavior of IoT and edge computing systems. It allows for modelling various IoT protocols and their energy consumption profiles and provides a new abstraction for IoT application graph modelling. However, it does not consider the energy consumption of the entire IoT and edge computing infrastructure.

SimulateloT-Mobile, detailed in ^[18], is a model-driven development tool to simplify the simulation of complex Internet of Things (IoT) environments with mobile nodes. Developers can employ it to model, validate, generate, and simulate IoT systems with mobility characteristics. The tool's metamodel, built on the Eclipse modelling Framework (EMF), defines IoT system structure, while the Graphical Concrete Syntax provides a visual representation for enhanced understanding. This

tool is an extension of SimulateIoT [14] and effectively addresses the absence of mobile node modelling capabilities. It utilizes the MQTT protocol for publish/subscribe communication and integrates an MQTT mobility management model to handle mobility within IoT systems. However, the tool has limitations, such as the assumption of guaranteed connectivity, which may not reflect real-world conditions where connectivity can be intermittent or disrupted.

PlOT [19] is a large-scale simulator designed to assess the network performance of IoT devices in a city-wide context. It includes a front-end for simulation configuration, a back-end for modelling millions of IoT devices using cellular infrastructure, and geographical and application databases. The architecture employs a grid structure based on a realistic city model and supports various elements like NB-IoT, network slicing, MEC, and beamforming. PlOT enables the testing of applications and KPI data extraction for AI and ML algorithm development. However, its primary focus is on network performance evaluation, with less emphasis on IoT device energy sources.

ABS-SmartComAgri [20] is a novel simulator for smart communication protocols in the context of precision agriculture. This is an open-source, agent-based tool designed to efficiently manage pesticide usage in agriculture by implementing smart communication protocols. It allows the simulation of various strategies, including broadcast, neighbour, and low-cost neighbour protocols, to optimize electric power, crop health, pesticide consumption, and overall network performance. The simulator is unique in its application to precision agriculture, explicitly targeting the reduction in pesticide usage and energy consumption while maintaining crop health. It is a valuable tool for testing and developing communication protocols before deploying them in real-world wireless sensor networks, contributing significantly to advancing smart agricultural practices.

FS-IIoTSim [21] is an innovative network simulation framework specifically designed for the performance evaluation of Industrial Internet of Things (IIoT) systems. Characterized by its flexibility and scalability, the tool adeptly supports both existing and modified communication protocols, addressing the complexities inherent in IIoT environments. FS-IIoTSim is structured into three integral components: scenario modelling, performance evaluation, and a user interface. Scenario modelling facilitates the creation of detailed industrial sensor network models and the generation of trace files containing device operation logs. The performance evaluation component analyzes these trace files, focusing on key metrics such as throughput, latency, and energy consumption.

IoTSim-Osmosis, as detailed in [22], is an advanced simulation framework designed for deploying Internet of Things (IoT) applications within integrated edge-cloud environments. Rooted in osmotic computing principles, it enables dynamic workload transfer between cloud data centres and edge devices, driven by performance and security triggers. This framework uniquely caters to the complexities of IoT applications and the heterogeneity present in integrated edge-cloud environments. Notably, it is the first framework to offer unified modelling and simulation for intricate IoT applications in such diverse settings. While existing frameworks support cloud-edge integration, none can directly accommodate osmotic computing due to IoT application intricacies and environmental diversity. IoTSim-Osmosis empowers researchers to evaluate end-to-end IoT application performance comprehensively using osmotic computing concepts. A case study on electricity management and billing demonstrates its potential. However, the framework has limitations, including expanding the wireless communication layer and incorporating support for security and privacy simulation. Assuming fixed IoT device locations impacts accuracy, necessitating further research to address varying signal factors and mobility. Additionally, research efforts are required to develop models and algorithms for security and privacy simulation.

As an extension of IoTSim-Osmosis, IoTSim-Osmosis-RES [23] encompasses a range of features pertinent to sustainable and autonomic IoT ecosystems, offering an assessment of various factors such as solar radiation levels, the utilization of renewable energy sources, the adoption of low-emission sources, and the battery capacity of IoT devices. It aims to simulate osmotic computations within the context of renewable energy sources and autonomic agents, thereby facilitating the analysis of distributed management algorithms. IoTSim-Osmosis-RES distinguishes itself as a unique IoT simulator capable of encompassing energy management, diverse power sources, and network infrastructure. In contrast to other simulators, it can simulate fluctuating weather conditions and the utilization of renewable energy sources, while also affording an easily extensible system that empowers researchers to specify their own virtual machine (VM) and power management policies. This framework is further characterized by its support for Software-Defined Networking (SDN), IoT devices, IoT device batteries, and renewable energy sources. This simulator has been linked with traffic simulation to simulate VANET environments in [24].

SimulatorBridger [24] is a novel simulator extended IoTSim-Osmosis-RES to bridge IoT simulation with traffic simulation. This simulator integrates IoTSim-OsmosisRES, an IoT simulator, with SUMO, a traffic simulator, to enable realistic simulation of Vehicular Adhoc Networks (VANETs) in urban mobility scenarios. The key innovation lies in the simulator's ability to handle dynamic and heterogeneous environments by managing the mobility of vehicles and the communications

of IoT devices. SimulatorBridger tackles the gap in current VANET research that lacks a simulation framework focused on IoT infrastructure. It facilitates seamless mobility integration, IoT devices, diverse technologies, and battery management within dynamic settings. This simulator addresses the difficulty of replicating the behaviour of highly mobile nodes within traffic networks that have constrained energy resources and the absence of centralized control. It allows for evaluating energy efficiency in vehicles and communication within VANETs, utilizing IoT infrastructure to enhance both realism and efficiency. However, the simulator does not support direct communications between IoT devices. Authors highlight their future works to extend the simulator's applications, including load balancing in traffic lights and accommodating new technologies like 5G in VANETs.

2. Classifications of IoT and Network Simulators

IoT simulators are categorized into distinct types based on their operational level within the overarching IoT architecture and the specific location where both data processing and simulation tasks are executed in cloud, fog, and IoT device simulators [41].

2.1. Cloud Simulators

Cloud IoT simulators primarily emulate cloud-based IoT platforms and services. Their design is rooted in understanding the intricate interactions between IoT devices and cloud services. They help evaluate the performance and efficiency of cloud resources, scheduling algorithms, and application deployments. This includes the mechanisms of data storage and processing in the cloud, as well as the intricacies of cloud-based analytics. Typically focuses on large-scale simulations involving multiple data centres and thousands to millions of VMs. They are used to evaluate cloud service models (IaaS, PaaS, SaaS), resource allocation strategies, and energy efficiency. Simulators of cloud computing emphasize the complexity of data centres, virtualization, and the management of large-scale resources. Examples of such simulators include the CloudSim Simulator. The foundation of these simulators lies in the services provided by cloud platforms, encompassing aspects like device management, data storage, analytics, and machine learning. The following are some examples of cloud simulators:

- IOTSim [37] is a versatile simulator implemented on top of CloudSim, primarily focused on modelling and simulating multiple IoT applications in shared cloud data centres. It covers various modules, including IoT application modelling, MapReduce data processing, cloud data centre management, and network simulation. It offers a layered architecture that facilitates the simulation of various IoT components, including VMs on data centre nodes with diverse hardware configurations, enabling detailed modelling of IoT applications from smart cities to healthcare systems with big data technologies like MapReduce. IOTSim excels in processing large volumes of data using parallel processing technologies, simulating network and storage delays critical for IoT applications, thereby providing a realistic model for the execution of parallel and distributed applications. Its protocols are tailored for efficient big data processing in cloud computing environments, supporting both batch and stream models and demonstrating scalability and adaptability for complex IoT environments. The architecture integrates CloudSim's core engine with layers for big data and user code, enhancing the simulation of network and storage delays. IOTSim's design is suited for a range of IoT applications, from building automation to wearable tech, highlighting its ability to simulate batch-oriented applications with high accuracy and offering a cost-effective solution for IoT solution development in cloud computing environments. IOTSim excels in accurately simulating IoT application performance and resource scalability in cloud environments but lacks support for stream-oriented IoT applications.
- SimIoT [25] is a simulation toolkit designed to analyze cloud computing systems in the context of IoT. It extends SimIC by including user submissions from IoT devices like sensors and smartphones. It models various entities and interactions, including users, data centres, and virtual machines, with a focus on real-time constraints. The toolkit offers message-exchanging optimization and supports heterogeneity, making it versatile for various cloud configurations. Although valuable for healthcare and information processing scenarios, SimIoT lacks a real-world IoT implementation, physical device simulation, mobility support, and explicit energy efficiency measures.

2.2. Edge Simulators

Edge simulators specifically focus on the computing resources that are located at the edge of the network, closer to the source of data generation, which are IoT devices. Unlike cloud or fog computing, edge computing aims to process data at or near the source, significantly reducing latency and bandwidth use. These simulators are crucial for evaluating the performance, reliability, and operational efficiency of edge computing environments. They model the dynamics of data processing, storage, and application execution at the network's edge, enabling researchers and developers to explore various edge deployment scenarios, resource management strategies, and application behaviours in real-time conditions.

Edge simulators are designed to address challenges such as limited computing resources, network connectivity variations, and the seamless integration of edge computing with cloud and fog layers. This allows for the optimization of IoT applications and services that require immediate data processing and decision-making capabilities. Examples of edge computing simulators include EdgeCloudSim and IoT-Sim-Edge, which offer detailed modelling tools for edge computing scenarios, from single-edge nodes to complex, multi-tier architectures involving edge and cloud collaborations. The following are some examples of edge simulators:

- CupCarbon, introduced in [26], is a cutting-edge platform for designing and simulating IoT Wireless Sensor Networks (SCI-WSN) in Smart Cities. It addresses the increasing prevalence of radio communication systems and technological advancements enabling IoT. CupCarbon stands out from traditional simulators by offering realistic modelling of radio channels and interferences, accounting for deployment environments, supporting mobile nodes, and enabling behavioural analysis in practical scenarios. It is a versatile tool with a multi-agent environment for mobility scenarios and event generation. It offers realistic modelling of radio channels and interferences, supports dynamic mobile nodes, and includes a comprehensive range of IoT components such as microcontrollers, sensors, batteries, and radio modules optimized for energy efficiency. The platform supports ZigBee 802.15.4 for WSN applications and long-distance communication technologies like LoRa and SigFox, facilitating direct communication to base stations and reducing the need for extensive sensor node networks. CupCarbon's novel architecture integrates environmental factors, mobility, and accurate radio propagation models for urban settings, enhancing network deployment studies and interference detection. It supports physical layer communication standards (ZigBee, WiFi, LoRa) for evaluating link quality and transmission conditions. The simulator's 2D/3D virtual urban platform aids in planning smart city projects by simulating sensor network deployments, node mobility, and radio propagation. With its discrete event simulation kernel, CupCarbon scales to simulate dense sensor networks, which is crucial for future smart cities. Its modular, Java-based design ensures compatibility with various IoT development tools, facilitating customization and integration. CupCarbon's utility in simulating smart city environments, showcasing sensor deployment, mobility modelling, and communication link visualization underscores its comprehensive toolset for designing, simulating, and visualizing IoT network dynamics in urban settings. Although it represents a significant advancement in wireless sensor network simulation for IoT and Smart City applications, it has limitations, including integrating only two radio propagation models and potential challenges with complex scenarios.
- Contiki-Cooja [27] is a network simulation tool derived from the Contiki operating system. Developed in Java, it enables users to specify both large and small Contiki motes (sensor network nodes) for deployment across the network. After running a simulation, users can access crucial network data, including mote outputs and timelines. It is worth noting that motes can be custom-defined using template options. These simulations primarily emphasize hardware and network challenges rather than IoT scenarios or communication models like publish-subscribe.
- EdgeMiningSim in [28], offers a simulation-driven approach for IoT data mining within edge computing. This methodology addresses IoT applications' unique challenges in data mining, aiming to empower domain experts with actionable insights for decision-making in dynamic IoT scenarios. The simulator is adaptable to various IoT applications, encompassing algorithmic, infrastructural, and contextual aspects often studied in isolation. It employs an interactive and iterative approach, balancing technical objectives and business interests. It simulates a diverse IoT ecosystem, including sensors and edge servers, integrating algorithmic, infrastructural, and contextual aspects for a comprehensive architectural design, network modelling, and topology definition. Employing a multi-layered architecture, EdgeMiningSim offers realistic modelling of device and application characteristics, such as mobility and energy consumption. While not extensively detailed, the simulator accommodates various IoT communication protocols and strategies, including task offloading and edge server management. It highlights the role of simulation in the early stages of IoT deployment, supporting the planning and evaluation of virtual urban platforms. The simulator includes mechanisms like flooding for efficient information dissemination among nodes and supports lightweight, decentralized data processing techniques. Demonstrating high scalability, EdgeMiningSim is suitable for large-scale IoT deployments, with a modular architecture indicating compatibility with numerous development tools. Through a smart environmental monitoring case study, EdgeMiningSim exemplifies its ability to simulate real-life IoT data mining scenarios, bridging the gap between theory and practice, and underscoring its value for researchers and practitioners. While promising, EdgeMiningSim does require substantial computational resources, which may pose challenges for resource-constrained IoT devices with limited processing power and memory.
- Mercury [29] is an open-source framework tailored for simulating real-time fog computing scenarios, emphasizing low latency, high data throughput, and 5G capabilities. It provides a detailed structural and behavioural model, offering insights into edge infrastructure and quality of service (QoS) optimization. Mercury is particularly suitable for data stream analytics applications and federated computation offloading. However, it is important to note that in the first

approach of the scenario under study, cloud computing is not included in Mercury's fog model. The framework is primarily focused on investigating how location awareness and mobility impact the Quality of Service (QoS) and operational costs. This analysis aids in optimizing both the sizing and functioning of edge infrastructure necessary for supporting services that rely heavily on computation offloading.

2.3. IoT Simulators

IoT simulators simulate the behaviour of IoT devices, networks, and services. They are used to evaluate the performance, scalability, and reliability of IoT systems. Their main features are device modelling, network communication, data generation and processing, etc. They are used to test IoT protocols, device-to-device communication, and scalability of IoT networks. Simulators of IoT computing focus on the heterogeneity of devices, network protocols, and scalability challenges. They are designed to handle many IoT devices, potentially in the order of millions, although not all IoT simulators simulate the cloud. Thus, comprehensive simulation environments must be developed that represent the entire journey of data from its origins in IoT devices to its eventual processing and storage in a data centre in the cloud. The following are some examples of IoT simulators:

- In [30], the authors propose a multi-level IoT simulator designed for smart territories, emphasizing scalability and advanced modelling techniques. The approach aims to promote the development of sustainable services in non-metropolitan areas. The simulator offers a two-level simulation approach, encompassing IoT systems' physical (sensors and actuators) and logical (software components) aspects. While the methodology shows promise in enhancing services for decentralized regions, it faces challenges in handling large-scale IoT environments. Its distinguishing feature is the comprehensive modelling of both physical and logical scenarios, essential for developing sustainable services in non-metropolitan areas. Physical scenarios encompass the tangible components of IoT systems, such as sensors and actuators, which interact directly with the environment. An illustrative example is the implementation of a smart market, where various producers utilize these devices to advertise product availability, enabling customers to engage interactively with the marketplace in real-time. This scenario demands precise simulation of physical interactions and device deployments, ensuring the efficient operation of sensors and actuators within a dynamic market environment. Conversely, logical scenarios address the software-driven aspects of IoT systems, including data processing algorithms, communication protocols, and decision-making mechanisms. These scenarios are pivotal for simulating the complex, intangible interactions that govern the behaviour of IoT systems. In the context of the smart market, logical simulation involves modelling the publish/subscribe mechanisms that facilitate information exchange between producers and consumers. It also encompasses the detailed simulation of wireless communication protocols, which are crucial for providing customers with timely information on product availability and guiding them through the market. The multi-level simulation approach, integrating coarse-grained and fine-grained simulations, is instrumental in addressing the intricacies of such scenarios. At the coarse level (Level 0), the simulation provides a broad overview of the smart territory, encompassing general interactions and behaviours. The fine-grained level (Level 1) delves into detailed simulations of specific areas, such as the smart market, focusing on intricate wireless communications and proximity-based interactions. This dual-level methodology ensures scalability and maintains high detail levels where necessary, facilitating the intricate simulation of IoT environments to advance smart services in decentralized regions. This simulator designed to support the development and optimization of smart services in cities and decentralized areas, incorporates a comprehensive range of IoT components such as sensors, RFID devices, and mobile terminals. It employs a two-level simulation approach: a coarse simulation for general behaviours and interactions and a more detailed simulation for specific aspects like wireless communications, utilizing tools like OMNeT++. This simulator supports various protocols crucial for realistic IoT scenarios, such as epidemic dissemination protocols, and integrates tools like MASON and SUMO for modelling urban systems and intelligent traffic control. While not explicitly mentioning a flooding mechanism for Service IoT (SIoT) Agents, its flexible architecture suggests compatibility with various communication mechanisms. It addresses scalability challenges through adaptable programming frameworks, ensuring compatibility with a broad range of development tools. It is capable of modeling complex scenarios, like a smart market, illustrating its potential to significantly contribute to smart city services and strategies.
- Ref. [31] introduces a 3D virtual environment simulator for IoT-based smart house systems, enabling controlled testing and evaluation. The simulator features virtual sensors replicating real-world conditions and an autonomous agent generator to simulate human-like behavior in the smart house. It is a cost-effective tool for testing and evaluating IoT-based smart house systems, enhancing design and issue identification. It features a wide array of virtual sensors and an autonomous agent generator to accurately replicate real-world conditions and human-like behavior in smart houses. This simulator supports a broad range of IoT components, such as motion detection and temperature sensors, providing a comprehensive tool for architects and engineers to configure smart house layouts intuitively via a GUI-

based interface. Its innovative motivation-based behavior planning method for generating autonomous agents ensures realistic interactions within smart environments, suggesting its potential use in larger urban planning projects. Although it does not specify communication protocols, its design implies compatibility with common IoT technologies. The GUI facilitates easy design and testing of complex systems, indicating scalability and offering insights into optimal sensor placement and smart house dynamics. The development environment is flexible, hinting at compatibility with various IoT tools and mainstream programming environments, enabling thorough testing and refinement of smart house systems before real-world implementation. However, further improvement in virtual sensor accuracy through machine learning is suggested.

2.4. Fog Simulators

Fog computing is an extension of cloud computing that brings computation closer to the data source which are IoT devices. Fog simulators help in modeling and evaluating the performance of fog nodes, their interaction with cloud and edge devices, and the overall system efficiency. The main features here are hybrid modeling (combining features of cloud and IoT), latency evaluation, local data processing, etc. They are designed to handle a mix of large-scale cloud resources and numerous edge devices. They are used to evaluate data offloading strategies, fog node placement, and hybrid cloud-fog architectures. Simulators of fog computing deal with the complexity of integrating both cloud and edge components, ensuring seamless data flow and processing. Some of the important and widely used fog simulators are: FogTorch ^[32], EmuFog ^[33], EdgeCloudSim ^[16], FogNetSim++ ^[34], iFogSim ^[35], and YAFS ^[36]. Listed below are a few examples of fog simulators:

- MobIoTSim ^[37] is a mobile IoT device simulator designed to facilitate the learning, testing, and development of IoT applications more efficiently. It can emulate IoT devices, generate real-time sensor data, and respond to messages using popular IoT protocols and data formats. Users can create IoT environment simulations with custom settings and connect the simulator to cloud gateways, like IBM Bluemix, for device management and notifications. However, MobIoTSim may not fully replicate real IoT device behavior and lacks support for simulating network errors, recording and replaying cases, and connecting real devices. It also does not explicitly simulate device mobility. These additional features could enhance its realism and utility for IoT testing.
- iFogSim ^[35] is an academic toolkit focused on simulating resource management policies in IoT, Edge, and Fog computing environments. It evaluates policy impact on latency, energy use, and network congestion metrics. This toolkit can model complex fog environments, aiding the assessment of real-time IoT applications. iFogSim comprises four key components: application, network, resource models, and a simulation engine. It includes sample policies for reference but does not support mobility. This resource is valuable for academic research into IoT, Edge, and Fog computing, particularly regarding resource management policy assessment. However, iFogSim does not support mobility.

References

1. Salama, M.; Elkhatib, Y.; Blair, G. IoTNetSim: A modelling and simulation platform for end-to-end IoT services and networking. In Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing, Auckland, New Zealand, 2–9 December 2019; pp. 251–261.
2. Therrien, V.; Mellah, H.; Boutin, V.; Sansò, B. A large-scale simulator for NB-IoT. *IEEE Access* 2022, 10, 68231–68239.
3. Kasnesis, P.; Toumanidis, L.; Kogias, D.; Patrikakis, C.Z.; Venieris, I.S. Assist: An agent-based sIoT simulator. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 353–358.
4. Bruschi, N.; Haugou, G.; Tagliavini, G.; Conti, F.; Benini, L.; Rossi, D. GVSoc: A highly configurable, fast and accurate full-platform simulator for RISC-V based IoT processors. In Proceedings of the 2021 IEEE 39th International Conference on Computer Design (ICCD), Storrs, CT, USA, 24–27 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 409–416.
5. Ta, D.T.; Khawam, K.; Lahoud, S.; Adjih, C.; Martin, S. LoRa-MAB: A flexible simulator for decentralized learning resource allocation in IoT networks. In Proceedings of the 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), Paris, France, 11–13 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 55–62.
6. Lopes, M.M.; Higashino, W.A.; Capretz, M.A.; Bittencourt, L.F. Myifogsim: A simulator for virtual machine migration in fog computing. In Proceedings of the 10th International Conference on Utility and Cloud Computing, Austin, TX, USA, 5–8 December 2017; pp. 47–52.

7. Li, X.; Huang, Q.; Wu, D. Distributed large-scale co-simulation for IoT-aided smart grid control. *IEEE Access* 2017, 5, 19951–19960.
8. Brambilla, G.; Picone, M.; Cirani, S.; Amoretti, M.; Zanichelli, F. A simulation platform for large-scale internet of things scenarios in urban environments. In *Proceedings of the First International Conference on IoT in Urban Space*, Rome, Italy, 27–28 October 2014; pp. 50–55.
9. Brundu, F.G.; Patti, E.; Osello, A.; Del Giudice, M.; Rapetti, N.; Krylovskiy, A.; Jahn, M.; Verda, V.; Guelpa, E.; Rietto, L.; et al. IoT software infrastructure for energy management and simulation in smart cities. *IEEE Trans. Ind. Inform.* 2016, 13, 832–840.
10. Jung, T.; Shah, P.; Weyrich, M. Dynamic co-simulation of internet-of-things-components using a multi-agent-system. *Procedia Cirp* 2018, 72, 874–879.
11. Ergun, K.; Yu, X.; Nagesh, N.; Cherkasova, L.; Mercati, P.; Ayoub, R.; Rosing, T. RelloT: Reliability simulator for IoT networks. In *Internet of Things-ICIOT 2020, Proceedings of the 5th International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, 18–20 September 2020*; Proceedings 5; Springer: Cham, Switzerland, 2020; pp. 63–81.
12. Melia, S.; Nasabeh, S.; Lujan-Mora, S.; Cachero, C. MoSloT: Modeling and simulating IoT healthcare-monitoring systems for people with disabilities. *Int. J. Environ. Res. Public Health* 2021, 18, 6357.
13. Bosmans, S.; Mercelis, S.; Denil, J.; Hellinckx, P. Testing IoT systems using a hybrid simulation based testing approach. *Computing* 2019, 101, 857–872.
14. Barriga, J.A.; Clemente, P.J.; Sosa-Sánchez, E.; Prieto, Á.E. SimulateloT: Domain Specific Language to design, code generation and execute IoT simulation environments. *IEEE Access* 2021, 9, 92531–92552.
15. Barriga, J.A.; Clemente, P.J.; Hernández, J.; Pérez-Toledano, M.A. SimulateloT-FIWARE: Domain specific language to design, code generation and execute IoT simulation environments on FIWARE. *IEEE Access* 2022, 10, 7800–7822.
16. Sonmez, C.; Oztogve, A.; Ersoy, C. Edgecloudsim: An environment for performance evaluation of edge computing systems. *Trans. Emerg. Telecommun. Technol.* 2018, 29, e3493.
17. Zeng, X.; Garg, S.K.; Strazdins, P.; Jayaraman, P.P.; Georgakopoulos, D.; Ranjan, R. IOTSim: A simulator for analysing IoT applications. *J. Syst. Archit.* 2017, 72, 93–107.
18. Barriga, J.A.; Clemente, P.J.; Pérez-Toledano, M.A.; Jurado-Málaga, E.; Hernández, J. Design, code generation and simulation of IoT environments with mobility devices by using model-driven development: SimulateloT-Mobile. *Pervasive Mob. Comput.* 2023, 89, 101751.
19. Firouzabadi, A.D.; Mellah, H.; Manzanilla, O.; Khalvandi, R.; Therrien, V.; Boutin, V.; Sansò, B. PloT: A Performance IoT Simulation System for a Large-Scale City-Wide Assessment. *IEEE Access* 2023, 11, 56273–56286.
20. García-Magariño, I.; Lacuesta, R.; Lloret, J. ABS-SmartComAgri: An agent-based simulator of smart communication protocols in wireless sensor networks for debugging in precision agriculture. *Sensors* 2018, 18, 998.
21. Lee, H.H.; Kwon, J.H.; Kim, E.J. FS-IIoTSim: A flexible and scalable simulation framework for performance evaluation of industrial Internet of things systems. *J. Supercomput.* 2018, 74, 4385–4402.
22. Alwasel, K.; Jha, D.N.; Habeeb, F.; Demirbag, U.; Rana, O.; Baker, T.; Dustdar, S.; Villari, M.; James, P.; Solaiman, E.; et al. IoTSim-Osmosis: A framework for modeling and simulating IoT applications over an edge-cloud continuum. *J. Syst. Archit.* 2021, 116, 101956.
23. Szydló, T.; Szabala, A.; Kordiumov, N.; Siuzdak, K.; Wolski, L.; Alwasel, K.; Habeeb, F.; Ranjan, R. IoTSim-Osmosis-RES: Towards autonomic renewable energy-aware osmotic computing. *Softw. Pract. Exp.* 2022, 52, 1698–1716.
24. Almutairi, R.; Bergami, G.; Morgan, G.; Gillgallon, R. Platform for Energy Efficiency Monitoring Electrical Vehicle in Real World Traffic Simulation. In *Proceedings of the 2023 IEEE 25th Conference on Business Informatics (CBI)*, Prague, Czech Republic, 21–23 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–8.
25. Sotiriadis, S.; Bessis, N.; Asimakopoulou, E.; Mustafee, N. Towards simulating the internet of things. In *Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops*, Victoria, BC, Canada, 13–16 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 444–448.
26. Bounceur, A.; Clavier, L.; Combeau, P.; Marc, O.; Vauzelle, R.; Masserann, A.; Soler, J.; Euler, R.; Alwajeeh, T.; Devendra, V.; et al. CupCarbon: A new platform for the design, simulation and 2D/3D visualization of radio propagation and interferences in IoT networks. In *Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 12–15 January 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.

27. Österlind, F. A Sensor Network Simulator for the Contiki OS; Swedish Institute of Computer Science: Stockholm, Sweden, 2006.
28. Savaglio, C.; Fortino, G. A simulation-driven methodology for IoT data mining based on edge computing. *ACM Trans. Internet Technol. (TOIT)* 2021, 21, 1–22.
29. Cárdenas, R.; Arroba, P.; Blanco, R.; Malagón, P.; Risco-Martín, J.L.; Moya, J.M. Mercury: A modeling, simulation, and optimization framework for data stream-oriented IoT applications. *Simul. Model. Pract. Theory* 2020, 101, 102037.
30. D'Angelo, G.; Ferretti, S.; Ghini, V. Multi-level simulation of internet of things on smart territories. *Simul. Model. Pract. Theory* 2017, 73, 3–21.
31. Lee, W.; Cho, S.; Chu, P.; Vu, H.; Helal, S.; Song, W.; Jeong, Y.S.; Cho, K. Automatic agent generation for IoT-based smart house simulator. *Neurocomputing* 2016, 209, 14–24.
32. Brogi, A.; Forti, S. QoS-aware deployment of IoT applications through the fog. *IEEE Internet Things J.* 2017, 4, 1185–1192.
33. Mayer, R.; Graser, L.; Gupta, H.; Saurez, E.; Ramachandran, U. Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures. In *Proceedings of the 2017 IEEE Fog World Congress (FWC)*, Santa Clara, CA, USA, 30 October–1 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
34. Qayyum, T.; Malik, A.W.; Khattak, M.A.K.; Khalid, O.; Khan, S.U. FogNetSim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access* 2018, 6, 63570–63583.
35. Gupta, H.; Vahid Dastjerdi, A.; Ghosh, S.K.; Buyya, R. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Softw. Pract. Exp.* 2017, 47, 1275–1296.
36. Lera, I.; Guerrero, C.; Juiz, C. YAFS: A simulator for IoT scenarios in fog computing. *IEEE Access* 2019, 7, 91745–91758.
37. Pflanzner, T.; Kertész, A.; Spinnewyn, B.; Latré, S. MobloTSim: Towards a mobile IoT device simulator. In *Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Vienna, Austria, 22–24 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 21–27.

Retrieved from <https://encyclopedia.pub/entry/history/show/126427>