On the way to distribute Compute Continuum Urban applications by deploying Clusters of Drones

Giuseppe Tricomi*‡, Luca D'Agati‡, Michele Arena‡,
Francesco Longo‡, Giovanni Merlino‡, Antonio Puliafito‡, Stefano Silvestri*
**ICAR-CNR: Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche Italiano*, Napoli, Italy
{giuseppe.tricomi, stefano.silvestri}@icar.cnr.it
‡*Università di Messina, Dipartimento di Ingegneria* Messina, Italy
{giuseppe.tricomi, luca.dagati, michele.arena, francesco.longo, giovanni.merlino, antonio.puliafito}@unime.it

2024 34th International Telecommunication Networks and Applications Conference (ITNAC) | 979-8-3503-7911-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ITNAC62915.2024.10815156

***Abstract*—Nowadays, Smart Cities leverage cutting-edge technologies to improve quality of life, optimize sustainability, and streamline urban infrastructure. Among the key technologies revolutionizing urban planning and infrastructure development, drones and the IoT have established themselves as true catalysts of change. The following steps in urban environment evolution go through the Compute Continuum paradigm adoption, where distributed applications seamlessly integrate into the urban fabric, extending computing power from the cloud to edge devices such as drones. These quickly deployable drones assist in traffic management, car parking, crowd monitoring and control, weather assessments, security, and emergency responses. This is possible thanks to the exploitation of drones' bird's-eye view, enabling urban planners and stakeholders to collect precise data, conduct inspections, and perform assessments with unprecedented efficiency. On the other hand, IoT sensors and devices provide real-time insights into various aspects of urban life, from traffic patterns and air quality to waste management and energy consumption. The use of drones in Smart Cities is not limited to simple data collection; they are versatile tools with numerous applications, particularly in the areas of surveying, mapping, and inspections, by the acquisition of data from their sensors that may be processed in near Fog devices belonging to Smart City Compute Continuum infrastructure. This work presents a study describing architectural insights, application deployment and management of drones in cluster acting as sensing elements.**

***Index Terms*—Compute Continuum, Computing Continuum, Smart City, UAV, S4T, I/Ocloud**

I. Introduction

In the modern era, Smart Cities leverage cutting-edge technologies to enhance the quality of life, optimize sustainability, and streamline urban infrastructure. Drivers of this transformation are the IoT and UAV exploitation in urban applications, which become true catalysts of diffusion of devices and revolutionizing urban planning and infrastructure development.

This work is partially supported by "Project SoBigData.it receives funding from European Union – 675 NextGenerationEU – National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, 676 PNRR) – Project: "SoBigData.it – Strengthening the Italian RI for Social Mining and Big Data Analytics" 677 – Prot. IR0000013 – Avviso n. 3264 del 28/12/2021." Moreover, this work was partially supported also by project SERICS (PE00000014) under the PNRR MUR program funded by the EU - NGEU.

Nowadays, researchers' works focus on the advancement of the urban environment through the adoption of the Compute Continuum (CC) paradigm, where distributed applications seamlessly integrate into the urban fabric, extending computing power from the cloud to edge devices such as drones. These quickly deployable drones assist in traffic management, car parking, crowd monitoring and control, weather assessments, security, and emergency responses, even without the necessities of physical installation realized in urban spaces. This is made possible by exploiting drones' bird's-eye view, which enables urban planners and stakeholders to collect precise data, conduct inspections, and perform assessments with unprecedented efficiency. On the other hand, IoT sensors and devices provide real-time insights into various aspects of urban life, from traffic patterns and air quality to waste management and energy consumption.

The use of drones in Smart Cities is not limited to simple data collection; they are versatile tools with numerous applications, particularly in the areas of surveying, mapping, and inspections. By acquiring data from their sensors, drones can process this information in nearby Fog devices, which are part of the Smart City CC infrastructure. This integration allows for real-time analysis and decision-making, significantly enhancing the responsiveness and efficiency of urban management systems. Furthermore, drones can also be employed in disaster management, providing rapid aerial assessments of affected areas, identifying hazards, and assisting in search and rescue operations (e.g., to find missing persons or to follow criminals running away from police, to mention a few).

The data collected by drones is often processed in real-time or near-real-time, thanks to the integration with Fog and Edge computing systems. These systems bridge the gap between cloud computing and local devices, providing low-latency processing capabilities and enabling prompt actions based on the gathered data.

In summary, this work presents an all-encompassing study describing architectural insights, application deployment, and management strategies for running applications based on drone exploitation in Smart Cities. It highlights the transformative potential of drones and IoT in creating more efficient, sustainable, and livable urban environments. As Smart Cities continue to evolve, the synergy between drones, IoT, and the CC will undoubtedly play a central role in shaping the future of urban living.

The rest of the paper is structured as follows: Section II reviews existing work on network virtualization and IoT; Section III details the proposed architecture and system design; Section IV describes the main workflows enabling the deployment; and Section V concludes the paper and outlines potential directions for future research.

II. Related Works

Recent advancements in IoT Edge/Fog computing and UAV technologies have significantly influenced the development of intelligent systems for smart cities. This section highlights key contributions that align with the objectives of this work. Merlino et al. [1] propose an extension of OpenStack to support Function-as-a-Service (FaaS), improving IoT service performance by bringing computational resources closer to data sources. Their work emphasizes the flexibility and efficiency of FaaS, particularly in urban applications, aligning with our focus on drone and IoT integration within the Smart City CC. Benomar et al. [2] and Tricomi et al. [3] explore the management of IoT infrastructure in smart city environments, using OpenStack and IoTronic to enhance real-time data processing and scalability at the edge. These studies underscore the importance of infrastructure management and data analytics for optimizing urban applications, supporting our framework for decision-making in Smart Cities. D'Agati et al. [4] present a framework combining UAV autopilot systems with cloud computing, improving UAV operations through advanced mission planning and real-time analytics. This integration is crucial for enhancing UAV performance in autonomous operations, a core component of our proposed architecture. Serrano et al. [5] propose a robust fingerprinting technique for audio signal recognition that could inspire methods for efficiently processing sensor data in smart city environments. Finally, LEACH (Low Energy Adaptive Clustering Hierarchy) has been extensively utilized to optimize energy consumption in wireless sensor networks (WSNs) [6]. Recent works [7], [8] have adapted LEACH for edge computing to distribute computational load efficiently, inspiring our approach to energy management in drone networks. These references collectively establish a foundation for our work, which focuses on optimizing UAV and IoT operations within the CC paradigm.

III. Architecture

The CC architecture of this work extends the one used in the preliminary attempts to set up and manage a small fleet of UAVs made in [4], where through the mediation of Neutron overlay facility a Fog node was able to control two drones coordinating their operations.

Stack4Things (S4T) [9] and I/Ocloud [10] represent the cornerstone of our infrastructure aiming to manage the Edge at the same time as a whole or with a fine granularity. S4T offers advanced configuration options, behavior customizability, advanced networking features and devices fleet management [1]–[3]. Moreover, in I/Ocloud, the Edge, which is mostly IoT devices, may expose their facilities (sensing and actuation) via a Virtual Node connected to the physical device. This way, heterogeneous IoT environments can be managed, and these devices can be facilitated in integrating into larger Cloud-based applications.

The proposed architecture integrates unmanned aerial vehicle (UAV) technology with cloud computing. This innovative approach streamlines vehicle management by employing a companion board that runs the Lightning-Rod daemon. This daemon facilitates interaction with the S4T cloud agent, enhancing communication between vehicles and simplifying overall system control. An all encompassing overview of the proposed architecture is shown in Figure 1, where the three dimensions of computational paradigm belonging to the CC are exploited by different components involved in the proposed system.

Cloud is exploited in the preliminary resources management and assignment, both in terms of hardware resources management than to coordination of the components at the application level.

Fog and Edge are mostly involved in executing applications relying on UAVs. Fog hosts software components that support and coordinate the fleet of devices acting on the surface of an urban environment's sector.

The architecture will be described in the next sub-sections according to the interaction perspectives.

A. Cloud to Edge perspective

Cloud owns the computational resources to orchestrate the Edge devices by performing optimal selection, assigning the devices (Fog and Edge) to a fleet, and connecting all devices with an overlay network. This is enabled by the exploitation of facilities offered by S4T and Neutron (as depicted in the upper part of Figure 1) which are responsible for implementing the requests received by the "Cloud Application Orchestrator". After the fleet is selected A Fog node is used to interact with a vehicle; its objective is to host a "Cloud Fog Coordinator" and a QGroundControl station to send the coordinates of a mission[1] to a vehicle.

The "***Cloud Application Orchestrator***" oversees the entire system. This entity manages one or multiple Fog Node Controllers, each one coordinating a specified cluster. This allows for scalable and coordinated control of UAV fleets without latency, bandwidth, and privacy issues. The "Cloud Application Orchestrator" interacts with the S4T cloud through the S4T's API, thanks to which it can communicate with the fog node (in particular, the Cloud Fog Coordinator) to instruct it about the fleet of UAVs deployment driven by using QGroudControl.

[1]Mission is the name used in QGroundControl to indicate a destination assigned to a vehicle.

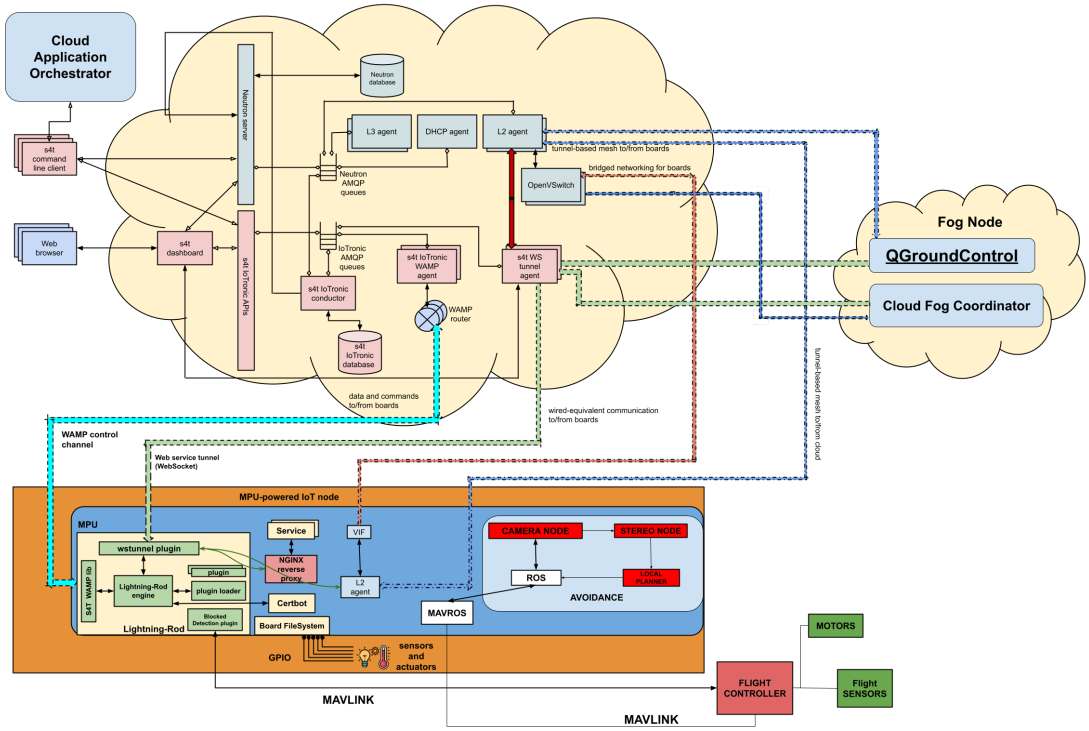

Fig. 1. Architectural overview showing the integration of all elements afferent to Cloud-Fog-Edge parts of the infrastructure. This picture highlights the integration of PX4-Avoidance with I/Ocloud [4] those work together to satisfy the Cloud Application Orchestrator request.

B. Fog to Edge perspective

As previously stated, the Fog node hosts a component called "***Cloud Fog Coordinator***," which coordinates the fleet and supports the devices in their application-specific activities. Cloud Fog Coordinator begins its work by forwarding the command received by the "Cloud Application Orchestrator" to the fleet of drones, identifying a special drone in the fleet (or a subset of drones) can act as ***Communication Gateway Drone*** (CGWD) and assigning all other UAVs to a subsection of the monitored area. Moreover, the "Cloud Fog Coordinator" controls each ***Worker UAV*** during its operations through a mission assignment performed by the QGroundControl component.

After the initialization phases, the UAV Cluster patrols a predetermined number of areas, exchanging messages with each other (only those part of the same Cluster) and capturing images. These images are then transmitted to the "Cloud Fog Coordinator", which analyzes the data and determines the next operational steps. At the end of a mission, the UAV remains on standby, awaiting further instructions from the 'Cloud Fog Coordinator", the same happens when the whole area is patrolled.

Further tasks performed by the "Cloud Fog Coordinator" are:

- Receive and process (or re-transmit, depending on the kind of elaboration request) the UAV's images and data sensed captured by the UAVs to produce feedback for the main application running on the Cloud facilities,
- Receive instructions from the application running on the cloud aiming to redeploy the Cluster, change the "Cloud Fog Coordinator" configuration or duties, or stop the operation before completion.

C. Edge to Edge perspective

The most important parts of this work concern the communication and interaction occurring among the Edge devices. The devices can exchange messages about energy monitoring, Cluster configuration and management, and obviously, the application data perceived.

Edge devices optimize the energy consumed in data transmission by reducing the transmission distances and transferring the data via the CGWD mediation, working in analogy with the LEACH approach for distributing the energy consumption for ML tasks that is presented in section II. The drone fleet is composed of a small set of CGWDs (1 active and others inactive) and a series of Worker UAVs; the inactive CGWDs work as Worker UAVs until their intervention is requested by the Fog node. Commonly the CGWD stays stationary in a position that reduces the energy spent by members to transmit data to it, and the Fog node which is designed to receive data (e.g., images), process them and if needed send commands to the fleet.

IV. Cluster Setup and Management in application context

One of the most interesting aspects of this work is the dynamic approach to setting up the Cluster of Edge devices to be connected with the Computing Continuum infrastructure.

The procedure for setting up the Edge device Cluster is shown in Fig. 2, which refers to the setup of a Cluster of UAVs.

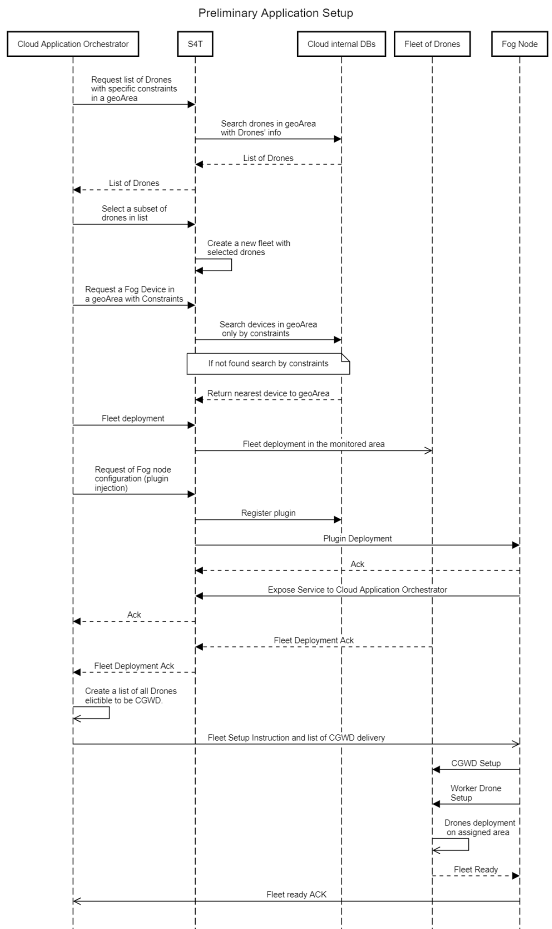

Fig. 2. Sequence Diagram of preliminary operation made to set up a Cluster of UAV serving applications in an area.

The driver of this procedure is the "Cloud Application Orchestrator", and it interacts with S4T to identify the Edge devices (i.e., drones, UAVs) and the Fog node in the area (or near to) under analysis. The selection uses geo-shapes to represent the area of interest, enabling the retrieval of the devices present in the region and the closest location. This is very useful for use along with functionality aiming to make the selection of devices satisfy specific constraints, as shown in the sequence diagram of Fig. 2 for Drones and Fog Node.

As a result of this, the selection process produces a new fleet of Edge devices that has to be deployed in the area after the selection of all devices involved is completed; fleet deployment is a time (and battery)-consuming operation, so to optimize time during the deployment of drones the Fog device is configured by exploiting the S4T functionalities of plugin injection that enables the configuration of the device with a portion of code customized for the application duties. This way, the Fog node becomes a controller for the Cluster running near the device (this solves several issues such as latency and privacy, and furthermore, it reduces the attack surface, to mention a few). When the Fog node is ready to begin its cluster coordinator role, another feature of S4T is used to enable interaction with the Cloud-hosted applications, to expose the services hosted by the controller in the Fog node.

This is possible because S4T can create an overlay network based on Neutron virtualization facilities among the application's Cloud-hosted component and the Fog devices used as cluster coordinators (one or more, depending on the application's needs and then on the number of clusters deployed).

At this point, the "Cloud Fog Coordinator" is directly reachable by the "Cloud Application Orchestrator" which may invoke the services related to CGWD setup and Worker Drone setup to update the fleet with a simple invocation of one of the services exposed by the "Cloud Fog Coordinator". When the setup is completed, the fleet is deployed and ready to be activated, an acknowledgment is sent to the "Cloud Application Orchestrator", and the fleet starts the task assigned.

During application runtime, the Cluster operates as an independent entity, even though the main application operating at the Cloud level may interact with the Cluster through the "Cloud Fog Coordinator" mediation. Indeed, the "Cloud Fog Coordinator" communicates with the drones to guide them during their missions.

From a high-level point of view, the QGroundController running in the Fog node begins the communication that assigns a new mission to the drone when the previous mission is completed, as reported in Section III-B. This is the only communication between a worker node and the "Cloud Fog Coordinator" in normal conditions. This is because all the battery-consuming communication (e.g., data perceived transmission) is designed to be realized in communication 1-to-1 between the worker drone and the CGWD, as shown in Fig. 3. Indeed, as depicted in the upper part of Fig. 3, the communication between the Fog node and the Worker drone exploits the classical WSN routing mechanism, where information is routed from and to the Edge nodes through the CGWD.

Furthermore, there exist two exceptions shown in the two opt blocks of Fig. 3: I) *Mission assignment to Worker Drone* and II) *Definition of New CGWD*.

The former is used to communicate with a drone that partially loses its capability to communicate with the infrastructure. In this case, QGroundControl may exploit the low-energy communication channels throughout CGWD by requesting a mission forward to a Worker Drone. The tentative of "mission forwarding" may fail; in this case, advice is sent to QGroundControl, marking the Worker Drone as unreachable.

The latter represents the case in which the CGWD becomes unreachable (e.g., crashes, malfunctions, or simply the battery is gone). The "Cloud Fog Coordinator" may identify this condition with a direct message from CGWD or after three failed updates of the CGWD status. When it happens, a new CGWD is activated from the list of inactive CGWDs (prepared during the Cluster setup phase) by sending an "Activation message" that has to be followed by an "Activation

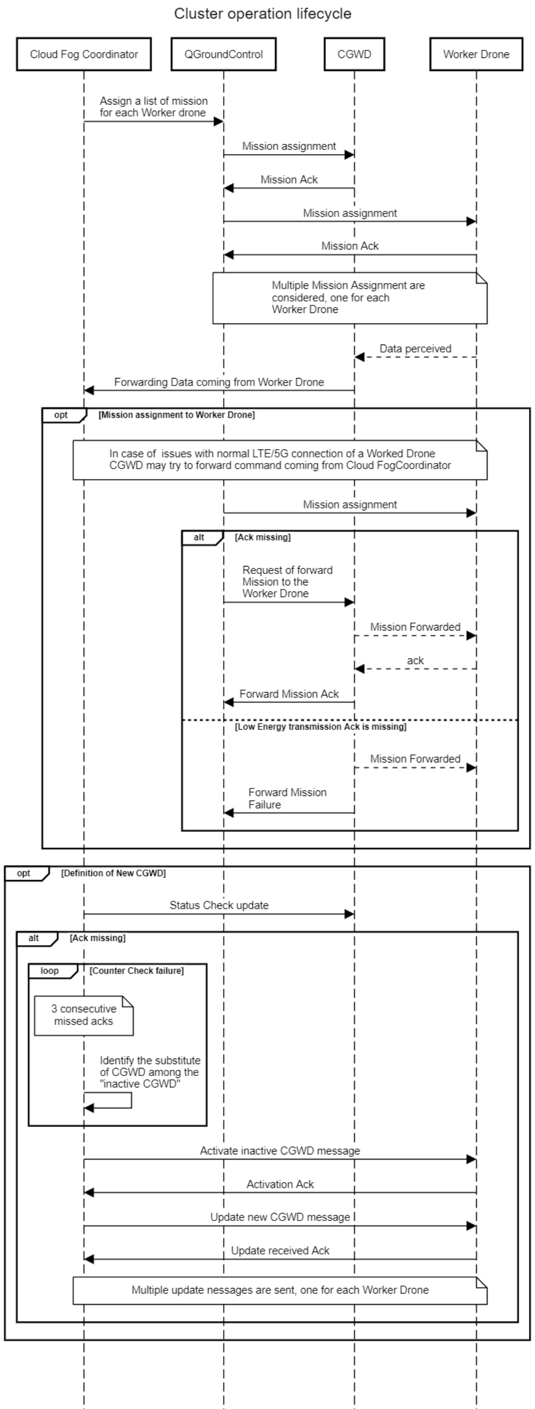

Fig. 3. Sequence Diagram of Cluster lifecycle most common communication.

Ack." When the new CGWD is activated, the "Cloud Fog Coordinator" uses the "update messages" to inform all worker drones of the new CGWD presence.

V. Conclusion and Future Works

This work presents an architecture enabling the exploitation of urban-scale (or, more generally, wide-area-scale) applications relying on flexible and fast-to-deploy cluster of drones sensing capabilities. Furthermore, this work represents the basis of a new era for Smart Cities, which until now have suffered from the huge cost of installation and the difficulties of continuous maintenance, updating, and upgrading the Edge devices deployed in a public environment to gather information. The solution relies on open-source tools that enable the owner to customize it according to environmental needs.

Future work will focus on a real environment (e.g., an open field), performed with tests of the deployment and robustness of a cluster of drones deployed to different atmospheric conditions and for different applications. Future steps in this research include the concepts of realizing an overlay network among devices working under dynamic scenarios where mesh networks and multi-hop communication strategies serve the UAVs involved in the cluster.

Acknowledgment

We thank Simona Sada and Giuseppe Trerotola for the administrative and technical support provided.

References

[1] G. Merlino, G. Tricomi, L. D'agati, Z. Benomar, F. Longo, and A. Puliafito, "Faas for iot: Evolving serverless towards deviceless in i/oclouds," *Future Generation Computer Systems*, vol. 154, pp. 189–205, 2024.

[2] Z. Benomar, L. D'Agati, F. Longo, G. Merlino, and A. Puliafito, "Managed elk deployments at the edge with openstack and iotronic: an italian smart city case study," in *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2022, pp. 287–292.

[3] G. Tricomi, M. Giacobbe, I. Ficili, N. Peditto, and A. Puliafito, "Smart city as cooperating smart areas: On the way of symbiotic cyber–physical systems environment," *Sensors*, vol. 24, no. 10, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/10/3108

[4] L. D'Agati, F. Longo, G. Merlino, A. Puliafito, and G. Tricomi, "Enhancing uav operational efficiency through cloud computing and autopilot system integration," in *2024 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2024, pp. 85–92.

[5] S. Serrano, M. A. B. Sahbudin, C. Chaouch, and M. Scarpa, "A new fingerprint definition for effective song recognition," *Pattern Recognition Letters*, vol. 160, pp. 135–141, 2022.

[6] R. Herrero, *Fundamentals of IoT Communication Technologies*. Springer International Publishing, 2021.

[7] M. Ashwini and N. Rakesh, "Enhancement and performance analysis of leach algorithm in iot," in *2017 International Conference on Inventive Systems and Control (ICISC)*. IEEE, 2017, pp. 1–5.

[8] F. Ramadhan, R. Munadi, and I. Istikmal, "Modified combined leach and pegasis routing protocol for energy efficiency in iot network," in *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*. IEEE, 2021, pp. 389–394.

[9] F. Longo, D. Bruneo, S. Distefano, G. Merlino, and A. Puliafito, "Stack4things: An openstack-based framework for iot," in *2015 3rd International Conference on Future Internet of Things and Cloud*. IEEE, 2015, pp. 204–211.

[10] D. Bruneo, S. Distefano, F. Longo, G. Merlino, and A. Puliafito, "I/ocloud: Adding an iot dimension to cloud infrastructures," *Computer*, vol. 51, no. 1, pp. 57–65, 2018.