Designing an Energy-Efficient Cloud-to-Things Continuum for Real-Time Mobile Applications

Marco Pettorali, Francesca Righetti, Carlo Vallati, and Giuseppe Anastasi

Dept. of Information Engineering, University of Pisa, Pisa, Italy, {name.surname}@unipi.it

***Abstract*—The proliferation of IoT devices and the increasing demand for real-time applications have driven a shift from traditional Cloud computing to Edge computing, giving rise to the Cloud-to-Things Continuum (C2TC). Many real-time IoT applications involve Mobile Nodes (MNs), such as autonomous vehicles and mobile robots, and require low-latency and high reliability. Optimal allocation of network and computing resources in the C2TC is crucial to meet various application requirements while ensuring energy-efficient allocation of resources. In this paper, we present a tool for the optimal allocation resources in the C2TC to support real-time mobile applications.**

***Index Terms*—Cloud-to-Things continuum, real-time applications, node mobility, optimal resource allocation.**

I. Introduction

The rapid proliferation of IoT devices and the increasing demand for real-time applications across various domains, including smart cities, smart transportation, industrial automation, and smart healthcare, have driven a shift from traditional Cloud computing to Edge computing. Edge computing introduces intermediate processing layers between IoT devices (Things) and the Cloud, forming what is known as the *Cloud-to-Things Continuum (C2TC)*. This continuum enables more efficient resource management by optimizing both computing and communication resources to meet diverse Quality of Service (QoS) requirements.

Unlike conventional IoT applications, real-time IoT applications demand low-latency and high-reliability communication, where data must be transmitted and processed within strict deadlines. Another key challenge in real-time IoT systems is the presence of Mobile Nodes (MNs), such as autonomous vehicles, mobile robots, and wearable devices, to ensure seamless operation of processes, e.g., in an industrial or smart city environment. These nodes introduce additional complexity due to their mobility patterns. MNs typically rely on wireless connectivity, which, while offering flexibility and cost efficiency, also introduces variability in network performance. Finally, problems related with energy efficiency and sustainability are becoming more and more relevant. Not only MNs may be energy constrained, but it is necessary also to minimize the energy consumption along the entire C2TC, especially at Fog/Edge nodes.

Many existing resource allocation frameworks primarily aim to minimize execution time or balance computational load, but they often neglect application-specific constraints such as maximum tolerable delay or minimum reliability thresholds. Other works do consider latency but typically rely on average delay metrics and lack probabilistic models that capture variability and worst-case behavior.

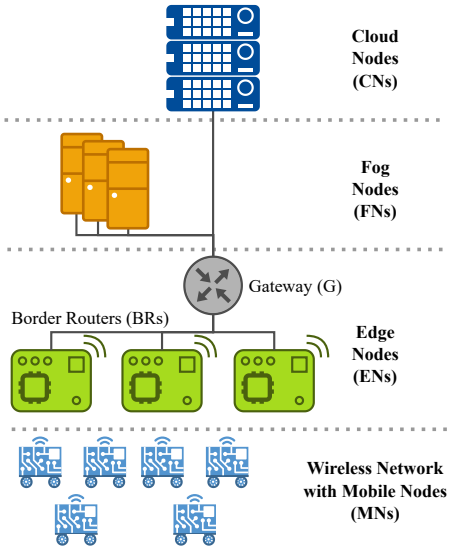

Fig. 1: Reference architecture of the C2TC system

To address these limitations, we have proposed *J-NECORA (Joint NEtwork and COmputing Resource Allocation)* [1], an analytical framework for C2TC-based systems with both static and mobile nodes. *J-NECORA* jointly allocates computing and communication resources across the C2TC, ensuring that application-specific constraints on delay and reliability are satisfied through the use of probabilistic delay models. When no feasible solution exists, it returns a *best-effort* allocation that minimizes delay and maximizes reliability, thus enabling the coexistence of applications with diverse requirements. We are currently working on extending J-NECORA to integrate different communication technologies (such as TSCH, WiFi, and 5G) and energy efficient strategies to minimize the energy consumption of edge/fog nodes.

II. System Model

Fig. 1 shows the system architecture, which is distributed across multiple layers and consists of Mobile Nodes (MNs), Edge Nodes (ENs), comprising a set of Border Routers (BRs) and a Gateway (G) enabling communication between BRs, Fog Nodes (FNs) located between the BRs and the Cloud, and Cloud Nodes (CNs). We assume that MNs lack computing capabilities; therefore, all application processing is carried out at a BR, FN, or CN.

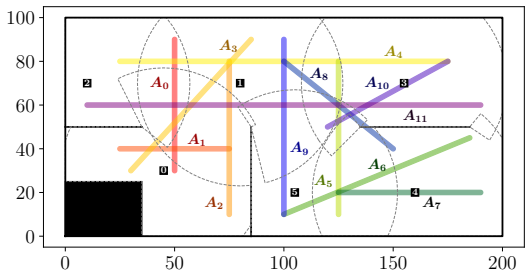

Fig. 2: Reference scenario with 12 processes and 6 BRs

MNs move within an *Area of Interest (AoI)*, where they collect and transmit data to BRs via wireless links (*WL*). BRs communicate with G through a wired backbone (*BB*), which in turn connects to FNs and CNs via wired links (*FL*, *CL*). Each application A_p is characterized by: (i) process p to be executed on a host h; (ii) message generation period T_p; (iii) maximum tolerable delay d_p; minimum required on-time ratio r_p; (iv) number of MNs m_p associated with the application.

A. Link and Node Model

Each network link is modeled using a delay probability distribution $\gamma_{p,b}^{link}(t)$, where $link$ indicates the type of link (*WL*, *BB*, *FL*, or *CL*). These delay distributions can be obtained analytically, through direct measurements, or by using available datasets. Given these distributions, the end-to-end communication delay distribution $\gamma_{p,h}^{\text{COM}}(t)$ for a message generated by an MN and transmitted to a host h, can be computed by convolving the distributions of the links along the path from the MN to the host.

To model the processing time distribution, we assume that processes are allocated to hosts with sufficient resources to meet their requirements. We introduce the CPU share $\lambda \in [0, 1]$ to represent the CPU fraction allocated to a process on a host, and we model the execution time probability distribution $\gamma_{p,h}^{\text{EXE}}(t, \lambda)$ as a function of λ. In addition, packets can experience queueing delays before being processed. The queuing time is modeled as a $G/G/s$ queue, where s represents the number of CPU cores available on the host. Assuming $s = 1$, we compute the probability distribution of the queueing delay $\gamma_{p,h,m}^{\text{QUE}}(t)$ using the approach in [2].

Assuming independence among the delay components, the total end-to-end delay $\gamma_{p,h}(t, \lambda, m)$ for an application A_p with m MNs, running process p on host h with CPU share λ, is given by the convolution of the communication, queuing, and execution delays:

$$\gamma_{p,h}(t, \lambda, m) = \gamma_{p,h}^{COM}(t) * \gamma_{p,h,m}^{QUE}(t) * \gamma_{p,h}^{EXE}(t, \lambda) \quad (1)$$

III. Performance Evaluation

J-NECORA allocates network and computing resources in the C2TC to satisfy the application requirements. Given the set of requirements, J-NECORA solves an optimization problem to choose the best host for each application, ensuring that the end-to-end delay meets the maximum tolerable delay d_p with a probability of at least r_p [1].

To evaluate its capabilities, we tested J-NECORA in a simulated IoT scenario composed of 6 BRs, 1 FN, 1 CN, and 12 applications, using TSCH as the communication protocol, similarly to [1]. Fig. 1 illustrates the BR deployment and MN AoIs used in the experiment. Applications have different QoS requirements, as summarized in Table I. Three configurations are considered, differing in the distribution of computing power, as detailed in Table II. The FN and CN have fixed capacities of 8 GHz and 19.2 GHz, respectively, and are omitted from the table for the sake of space. For each scenario, J-NECORA computed the optimal allocations, reported in Table III, and the maximum number of supported MNs per application, shown in Table IV. Applications A_9, A_{10}, and A_{11} are omitted for brevity, as they were allocated to the CN in all scenarios with up to 13 MNs each.

TABLE I: Requirements of the applications

r_p	d_p 50 ms	75 ms	100 ms	500 ms
0.995	A_0	A_3	A_6	A_9
0.99	A_1	A_4	A_7	A_{10}
0.95	A_2	A_5	A_8	A_{11}

TABLE II: Capacity of edge nodes (GHz)

	BR0	BR1	BR2	BR3	BR4	BR5	Total
HOM	3	3	3	3	3	3	18
HET1	5	5	3	3	1	1	18
HET2	4	4	3	3	2	2	18

TABLE III: Process allocation table

	BR0	BR1	BR2	BR3	BR4	BR5
HOM	A_0	A_1	A_4	A_5, A_6	A_7	A_8
HET1	A_0, A_1	A_2, A_5	A_6	A_7	A_8	
HET2	A_0	A_1	A_5, A_6	A_7	A_2	A_8

TABLE IV: Max number of MNs supported

	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
HOM	1	2	12	13	6	1	1	13	13
HET1	1	1	2	13	13	4	11	12	1
HET2	13	13	1	13	13	1	1	12	4

IV. Conclusions and Future Works

Analytical tools like J-NECORA enable informed decision-making during the design phase of IoT systems based on C2TC, improving performance and ensuring compliance with real-time application constraints. We are currently working on extending J-NECORA to integrate (i) communication technologies beyond TSCH (such as WiFi and 5G) and (ii) energy efficient strategies to minimize the energy consumption of edge/fog nodes. While J-NECORA can be used in the design phase, in [3], we have proposed an online algorithm for the allocation of resources in C2TC at run-time to cope with reconfigurable systems.

References

[1] M. Pettorali, F. Righetti, C. Vallati, S. K. Das, and G. Anastasi, "J-necora: A framework for optimal resource allocation in cloud-edge-things continuum for industrial applications with mobile nodes," *IEEE Internet of Things Journal*, 2025.

[2] M. Ackroyd, "Computing the Waiting Time Distribution for the G/G/1 Queue by Signal Processing Methods," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 52–58, 1980.

[3] M. Pettorali, F. Righetti, C. Vallati, S. K. Das, and G. Anastasi, "Dynamic Resource Allocation in Cloud-to-Things Continuum for Real-Time IoT Applications," in *IEEE Int. Conf. on Smart Computing (SMARTCOMP 2025)*, 2025.