Decentralized Offloading Decision-Making of Control Computations

Emil Sundström

Abstract— **For large-scale systems involving many devices that offload control computations to edge servers, efficient resource allocation strategies are essential. In this extended abstract, I advocate for decentralized resource allocation solutions, motivated by their simplicity, scalability, and trustworthiness. The focus is particularly on applications involving search-and-rescue operations using unmanned aerial vehicles (UAVs), where many autonomous devices operate simultaneously over large areas. The key contribution of the work is the development of a decentralized decision-making policy that scales effectively as the system grows.**

I. Introduction

In recent years, advanced control algorithms have shown great promise across many applications. However, these algorithms often require lots of computing resources, making them challenging to use on devices with limited processing power and battery life.

Future telecommunication platforms are expected to include built-in computing capabilities, enabling devices to offload demanding computations to the 6G network infrastructure [1]. This capability allows simpler devices to become ”smart” in some sense, by enabling usage of computational resources with low latency.

Implementing large-scale computation offloading presents several practical challenges, especially developing algorithms suitable for both local and remote computing. From a network infrastructure perspective, managing resources efficiently is crucial because mobile devices frequently connect and disconnect, requesting or releasing computational resources. For successful real-world implementation, solutions must be scalable and simple.

In this extended abstract, I present the part of my research where the resource allocation problem for large distributed and decentralized systems have been in focus. I propose a straightforward method that allows individual devices to decide when and where computations should be offloaded. The focus is specifically on control tasks that typically require low data transfer but strict timing requirements. The approach is suitable for various mobile systems, including autonomous vehicles, robots, and drones, with a particular focus on unmanned aerial vehicles (UAVs).

A primary motivation for using decentralized decision-making is trust. Trust includes several important considerations, including data security, reliability, and transparency of computational tasks. Developers and manufacturers must have absolute confidence that sensitive data and critical computations will be handled securely, reliably, and efficiently. Centralized offloading methods, while beneficial in some scenarios, can raise concerns about data privacy, potential misuse of information, and reliance on infrastructure operated by third parties. By adopting a decentralized approach, devices keep control over their data and where their computational tasks are transferred, significantly reducing these risks. Furthermore, decentralized methods align with principles of robustness and resilience, ensuring continued functionality even in scenarios where network conditions vary or infrastructure encounters disruptions.

Since local computation remains an available fallback if the connection to the distributed resources is interrupted, allowing each device to autonomously determine its offloading strategy enhances both trust and practicality, aligning closely with real-world expectations and user preferences. Hence, decentralized decision-making is relevant when considering offloading of control computations.

The main contributions of this research include:

- A resource allocation method that guarantees performance quality and scales well with the number of devices.
- A clearly defined model specifically tailored for offloading control tasks.
- Addressing the problem of efficiently allocating resources for repetitive control tasks on a large scale.

Previous research has extensively studied offloading methods. Early work, such as that by Kumar et al. [2], tracked the evolution from basic feasibility studies to more sophisticated infrastructure and decision-making processes. Practical methods for smartphones have been presented by Cuervo et al. [3] and Chun et al. [4]. Vehicle-related offloading has been discussed by Luo et al. [5], Gala et al. [6], and Lumpp et al. [7], focusing primarily on integration challenges rather than large-scale scalability, which is our primary focus.

Resource allocation from a network provider's point of view has been studied by Chen et al. [8] and Jošilo and Dán [9]. Our approach differs by focusing on application-driven, decentralized decision-making rather than centralized control. Game theory has also been explored for offloading scenarios [10]–[12]. However, existing methods often use complex network models unnecessary for typical control tasks. The simplified approach from my research contributes to ongoing discussions about how to best organize computational resources in future network systems.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors are members of the ELLIIT Strategic Research Area at Lund University.

Department of Automatic Control, Lund University, Lund, Sweden {first name.last name}@control.lth.se

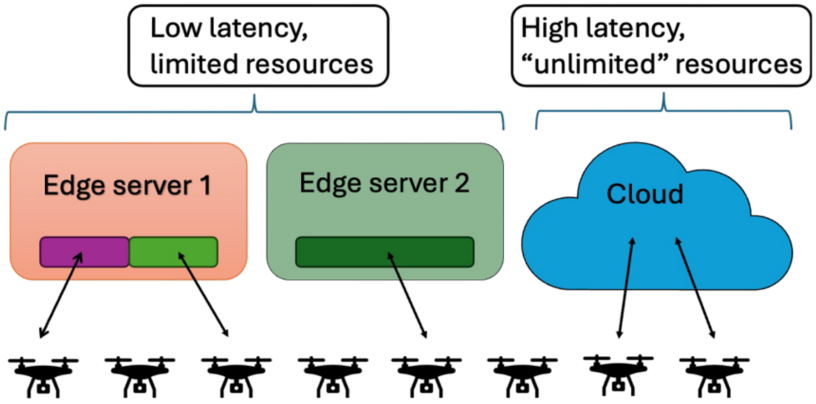

Fig. 1. The figure sketches the decentralized large-scale offloading problem. The devices are sketched as UAVs and the servers modelled as edge servers and cloud servers. Edge servers are assumed to have limited computational capacity, while the cloud servers have limited capacity, but to an unlimited number of devices.

II. Approach

This section briefly describes the proposed distributed resource allocation approach. The methods presented here utilize game theory as the primary framework. Although rigorous mathematical analysis is essential for fully understanding these methods, this extended abstract provides a concise overview without extensive mathematical details.

Figure 1 illustrates the core concept of the approach. Devices are distributed across a wide geographical area, each with unique computational demands and varying network connectivity. These devices interact with two types of servers: smaller edge servers, which offer limited computing capacities with lower latencies, and larger cloud servers, which provide a specific amount of computational resources to each connected device regardless of the total number of connections. Typically, cloud servers are physically located further away compared to edge servers. Allowing devices to

Fig. 2. The figure sketches the considerations that the local offloading manager of the devices must take for the system to converge to a resource allocation policy that scales well with the number of devices.

independently determine when and where to offload computational tasks shifts the resource allocation problem into a question of valuation. Each device evaluates various metrics, including financial costs, resource availability, and network connectivity, to make informed offloading decisions (see Figure 2). An essential requirement for this decentralized approach is that the overall system must remain scalable and quickly converge to an effective solution even as the number of participating devices increases. For this purpose, the field of game theory provides a rich set of tools to examine both convergence and scalability properties of distributed decision-making processes, which have been extensively utilized in my research on the resource allocation problem.

References

[1] A. Karapantelakis *et al.*, “Co-creating a cyber-physical world,” Ericsson Research, Tech. Rep., 2024, available at Ericssons’s Website, access date: March 17, 2025.

[2] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,” *Mobile Networks and Applications*, vol. 18, pp. 129–140, 2012.

[3] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “Maui: making smartphones last longer with code offload,” in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, 2010, pp. 49–62.

[4] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: elastic execution between mobile device and cloud,” in *Proceedings of the Sixth Conference on Computer Systems*, 2011, pp. 301–314.

[5] C. Luo, J. Nightingale, E. Asemota, and C. Grecos, “A uav-cloud system for disaster sensing applications,” in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 2015, pp. 1–5.

[6] G. Gala, G. Fohler, P. Tummeltshammer, S. Resch, and R. Hametner, “Rt-cloud: Virtualization technologies and cloud computing for railway use-case,” in *2021 IEEE 24th International Symposium on Real-Time Distributed Computing (ISORC)*. IEEE, 2021, pp. 105–113.

[7] F. Lumpp, M. Panato, F. Fummi, and N. Bombieri, “A container-based design methodology for robotic applications on kubernetes edge-cloud architectures,” in *2021 Forum on Specification & Design Languages (FDL)*. IEEE, 2021, pp. 01–08.

[8] M.-H. Chen, B. Liang, and M. Dong, “Multi-user multi-task offloading and resource allocation in mobile cloud systems,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6790–6805, 2018.

[9] S. Jošilo and G. Dán, “Joint wireless and edge computing resource management with dynamic network slice selection,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1865–1878, 2022.

[10] M.-A. Messous, H. Sedjelmaci, N. Houari, and S.-M. Senouci, “Computation offloading game for an uav network in mobile edge computing,” in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[11] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, “A game theoretic resource allocation for overall energy minimization in mobile cloud computing system,” in *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design*, 2012, pp. 279–284.

[12] X. Chen, “Decentralized computation offloading game for mobile cloud computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2014.