PhD Forum Abstract: Untangling the Cloud from Edge Computing for IoT

Nabeel Nasir
University of Virginia
nabeeln@virginia.edu

ABSTRACT

Edge computing improves latency and scalability in IoT by shifting applications from cloud to edge servers. However, it requires a control plane in the cloud, which hinders reliability and scalability, is impractical for remote deployments, incurs considerable costs, and offers limited privacy controls. We propose reusing existing IoT gateways for compute, providing resource elasticity using heterogeneous edge hardware, and following a user-driven privacy model. We envision edge computing that is cost-effective, scalable, and privacy-aware, without the dependence on cloud. Our results show that gateways can be used to execute a significant subset of edge applications offering better scalability and responsiveness.

CCS CONCEPTS

• **Networks** → **Cyber-physical networks**.

KEYWORDS

Edge Computing, Internet of Things, Cloud Computing, Privacy

ACM Reference Format:
Nabeel Nasir. 2021. PhD Forum Abstract: Untangling the Cloud from Edge Computing for IoT. In *The 19th ACM Conference on Embedded Networked Sensor Systems (SenSys'21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3485730.3492900

1 INTRODUCTION

Edge computing is touted as a solution to handle the massive influx of data in the Internet of Things (IoT). It prescribes executing applications closer to devices, consequently reducing latency, minimizing bandwidth, and improving privacy by operating on premises. The current model of edge computing has devices, IoT gateways, edge servers, and the cloud, with all control and configuration handled at the cloud. Having the control plane at the cloud allows for easier management, and enables executing most tasks at the edge but allows heavier tasks to be offloaded to the cloud.

However, having the control plane away from devices, and in the cloud, introduces several issues. First, it is impractical for remote deployments without a stable Internet connection or in secure deployments which do not prefer a cloud endpoint. Second, it reduces reliability and incurs considerable cloud and bandwidth costs. Third, it hinders scalability due to configuration overhead at the cloud. Fourth, current platforms incorrectly consider all data from a deployment to belong to a single user, leading to limited privacy controls for users. All these factors indicate that current edge computing platforms are designed top-down from the cloud to the devices, rather than the other way around.

Instead, what is needed is an edge computing platform designed with a focus on the devices and users, and shifting the control plane closest to the devices. We propose three key ideas to make this shift possible. First, instead of relying on expensive edge servers, use a distributed network of IoT gateways which are closest to the devices, and execute applications on this network without the support of cloud. Second, to improve application performance and offer some form of resource elasticity offered by the cloud, support special purpose edge hardware like GPUs, secure enclaves, etc. Third, allow users to have better privacy control on their data by means of privacy policies that operate one hop from devices.

The primary contribution of this work is in providing an alternative approach to enable edge computing for IoT without depending on the cloud or requiring expensive edge servers, as described in Figure 1. It would also allow leveraging specialized edge computing hardware to be used in a more meaningful way, rather than being used in smaller disjoint projects. It would also enable users in IoT environments to be more cognizant about how their data is being used and provide much needed control over their data.

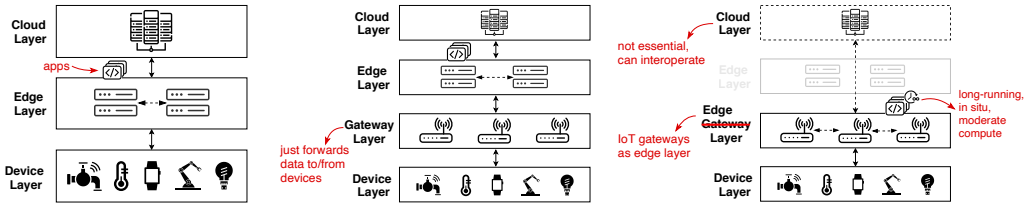
2 RELATED WORK

Edge Computing Platforms: Platforms like AWS IoT Greengrass [3] can be used to execute applications in the edge network. However, they assume a limited architecture with all data and applications available on a central edge node, provide no access control of device data to applications, require a cloud connection for configuration and application deployment, and has significant configuration overhead and lack of simple application abstractions. Our work avoids requiring a centralized node and reuses available gateways to execute applications, enforces access control by restricting the data applications have access to, operates without Internet connectivity, and provides convenient abstractions for setting up gateways and devices to reduce development burden.

Task Mapping and Edge Heterogeneity: [4] provides a task mapping algorithm for an edge server to map tasks on edge gateways. However, this framework requires a priori information on worst case execution times and deadlines of the tasks, which are difficult to obtain in real world use cases. In comparison, our optimization only operates on the task's execution requirements, and the gateways' resource usage information.

Privacy and Control of User Data in IoT: Our approach would be similar to the privacy policies defined in [2], but have more fine

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SenSys'21, November 15–17, 2021, Coimbra, Portugal
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9097-2/21/11.
https://doi.org/10.1145/3485730.3492900



(a) Edge Computing Architectures

(b) Actual implementations use gateways

(c) Shift applications to gateways

Figure 1: a) Edge computing generally follows a layered architecture with apps executing on edge servers. b) Actual deployments require IoT gateways to bridge communication with devices. c) We reuse gateways to also execute applications.

grained time and condition based policies, and would be enforced on a distributed set of gateways rather than on a central edge server.

3 PROPOSED APPROACH

There are three key pieces required to achieve the proposed goals which are described below.

Distributed gateway network for edge computing: An infrastructure which uses inexpensive IoT gateways to provide edge computing needs to be built first. It requires a middleware that runs on all of the IoT gateways, to form a network with minimal overhead, support interactions with IoT devices, provide an API for applications to interact with devices with minimal code, and operate on limited gateway resources by optimizing deployment decisions.

Leveraging heterogeneous gateways: The platform needs to support IoT gateways with varying capabilities (GPU support, high storage etc.). The idea is to match application requirements collected at deployment time, and pick a gateway with capabilities most suited for it. Capabilities and resource usages of gateways need to be monitored by the platform on a real time basis. We plan to identify an optimal deployment strategy to schedule applications based on these parameters: i) app's requirements: device needs, specialized resource needs (GPU etc.), execution length (long running vs. short task), ii) gateway resource usage: CPU, memory, specialized resource availability, iii) overall resource utilization.

User-driven Privacy Policies: The platform should allow or restrict access of devices to be used by applications during specified time ranges. There are challenges in designing the policy itself, in associating devices to the affected users, and in enforcing the policy. In the initial prototype, we plan to specify time based policies similar to how times are specified in cron jobs. Enforcing the policy would require filtering data streams emanating from each gateway based on rules specified in the policy.

4 PRELIMINARY RESULTS

We have designed a distributed network of IoT gateways which enables applications to interact with devices connected to the network. The platform supports user modules to interact with devices, and an API to register devices, deliver data to the platform, and accept control messages. The platform masks underlying complexities to applications, giving an illusion that all devices are available on a single gateway, simplifying interactions with devices on different gateways. It also minimizes configuration overhead by using wireless radio discovery among gateways. Developers can connect their laptops temporarily to the platform to deploy applications, and manage the network. Deployed applications are received by a scheduler, a distributed service which picks the ideal gateway to execute the application based on resource usage of gateways, proximity to devices required by the application, and other preferences. We also identified a set of applications suitable for the platform, including if-this-then-that type applications, sensing and actuating in large-scale deployments, and inherently distributed applications.

We have implemented this design using multiple Raspberry Pi 4B [1] boards deployed in a floor of our building, supporting around 250 IoT devices. We have conducted several experiments to ascertain the best network topology for a platform of IoT gateways, and found that a distributed architecture improves CPU and memory utilization, and reduces network traffic and application latency. We also compared our work with AWS IoT Greengrass, and showed a 10x and 2.5x improvements in latency and network traffic respectively, indicating better scalability and responsiveness.

5 CONCLUSION

In this work, we propose to leverage heterogeneous IoT gateways and distribute workload on them to achieve edge computing that is cost-effective, scalable, general-purpose, and privacy-aware, without the drawbacks of cloud dependency. A prototype infrastructure has already been built which works with a specific set of applications, and will be improved further by incorporating special purpose edge hardware and user-driven privacy policies.

6 BIOGRAPHY

Nabeel Nasir is a PhD candidate advised by Prof. Bradford Campbell at the University of Virginia. His interests are in Edge Computing and Distributed Systems. He has worked at Adobe Systems and EnLite Research. He expects to submit his dissertation in 2023.

REFERENCES

[1] The Raspberry Pi Foundation. 2021. *Raspberry Pi 4.* Retrieved Mar 20, 2021 from https://www.raspberrypi.org/products/raspberry-pi-4-model-b

[2] Primal Pappachan et al. 2017. Towards privacy-aware smart buildings: Capturing, communicating, and enforcing privacy policies and preferences. In *IEEE 37th International Conference on Distributed Computing Systems Workshops.*

[3] Amazon Web Services. 2020. *AWS IoT Greengrass.* Retrieved October 1, 2020 from https://aws.amazon.com/greengrass/

[4] Daniel (Yue) Zhang, Tahmid Rashid, Xukun Li, Nathan Vance, and Dong Wang. 2019. HeteroEdge: Taming the Heterogeneity of Edge Computing System in Social Sensing *(IoTDI '19).* ACM, New York, NY, USA. http://doi.acm.org/10.1145/3302505.3310067