

Demo Abstract: An Architecture for Edge Computing over Underutilized Gateways

Nabeel Nasir
University of Virginia
nabeeln@virginia.edu

Bradford Campbell
University of Virginia
bradjc@virginia.edu

ABSTRACT

Internet of Things applications typically run on the cloud and away from the end devices, leading to potential privacy and security risks, lower latency, and reduced reliability. Such deployments commonly use gateways to aggregate and send device data to the cloud. We hypothesize that these gateways are underutilized and can perform more than just packet forwarding. This work is an attempt to build a distributed platform over such gateways, with an objective to push applications to the edge of the network to overcome the shortcomings of the cloud. Our solution enables heterogeneous gateways to discover each other, and provides programming interfaces for developers to run applications on the platform without having to deal with the underlying network and device complexities. We showcase two applications that use our APIs, and also demonstrate how end users can interact with the platform via their personal computers (smartphones, laptops, etc.).

CCS CONCEPTS

• **Networks** → **Network design principles; Sensor networks; Network management; Network monitoring.**

KEYWORDS

Edge Computing, Internet of Things, Distributed Applications

ACM Reference Format:

Nabeel Nasir and Bradford Campbell. 2019. Demo Abstract: An Architecture for Edge Computing over Underutilized Gateways. In *The 17th ACM Conference on Embedded Networked Sensor Systems (SenSys '19)*, November 10–13, 2019, New York, NY, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3356250.3361934>

1 INTRODUCTION

Sensor deployments typically follow a star topology with gateways acting as data aggregators that push data to the cloud. Such a cloud model is preferred by developers for better access to computing resources, better support for writing applications, and ease of managing devices centrally. However, data in the cloud is vulnerable to privacy and security risks, and running applications away from the data source could lead to delayed response times. Also, applications could be forced to work on condensed data due to bandwidth costs, and availability and quality of service are dependent on the Internet

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 '19, November 10–13, 2019, New York, NY, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6950-3/19/11...\$15.00

<https://doi.org/10.1145/3356250.3361934>

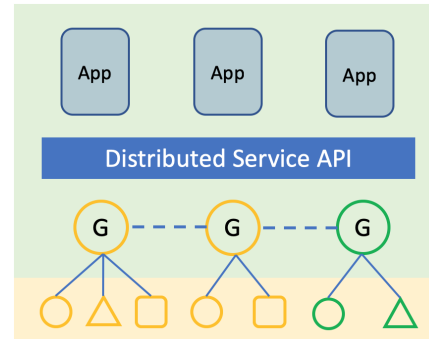


Figure 1: Our solution is to have gateways that discover each other, form a gateway network, and provide a platform which abstracts the underlying hardware and network complexities to support distributed applications.

connection. However, convenience and practicality trumps these gaps, mostly due to the non-existence of a better model. Edge computing is a step to address these gaps and is considered desirable for applications that require in situ processing for fast response times. Current implementations of edge computing are based on a cloudlet model [2], where a machine sits at the edge of the network, that handles functions like running time-critical applications, and condensing data that is pushed to the cloud. However, cloudlets are prone to central point of failure, and are generally expensive. To bridge these gaps, we envision edge computing in a more cost-effective way with existing gateway infrastructure. We believe that the gateways used in sensor deployments are underutilized, and can handle more computations rather than just being aggregators for cloud data. Our hypothesis is to link together gateways under the purview of a single ownership, and create a distributed computing platform. We draw inspiration from cloud computing, where comprehensive software stacks and high-level abstractions made the cloud into an indispensable computing environment. We hope to bring such robustness and usability to an edge computing paradigm by abstracting the underlying hardware and network complexities.

However, edge computing over such gateways presents certain challenges in comparison to cloud computing. First, the gateways are distributed through space as they need to be near the devices they support. Next, the gateways may use a diverse set of backhaul networks depending on connectivity available at the deployment site. Also, deployments are likely to be highly dynamic, as new devices are added, gateways are relocated, and deployment scenarios change. Lastly, the gateways are unlikely to be homogeneous, due to the wide range of wireless protocols and prevalence of vendor customization. Creating a cohesive platform will require managing

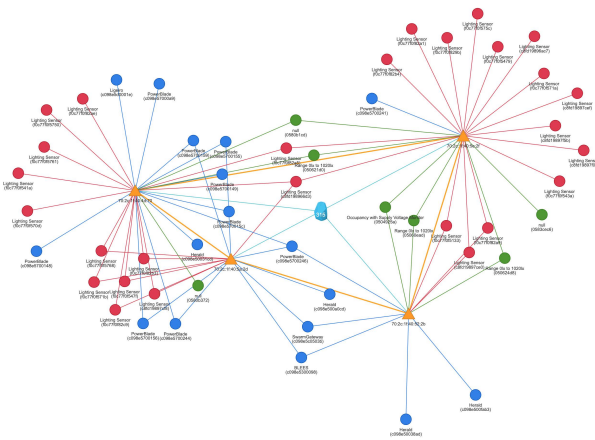


Figure 2: The link graph provides a live visualization of the deployment. It shows the which gateways are discoverable to each other and also which sensors are connected to each of the gateways. Yellow triangles represent gateways and all other vertices represent sensors. Connections between devices are represented by edges.

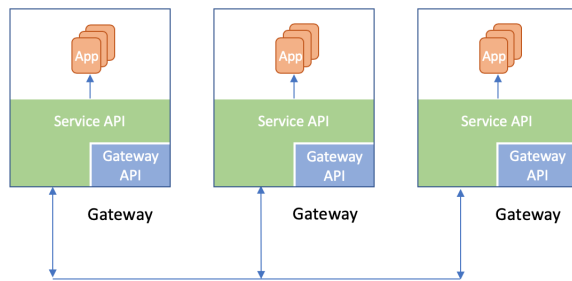


Figure 3: Shows how the Gateway API and Service API are structured to provide abstractions for apps that run on gateways. Gateway API allows apps to query locally available information on the gateway where it resides. Service API instances residing on all gateways work together to enable distributed application development. Gateway API forms the foundation for the Service API.

these issues with IoT deployments. We propose an automatically discovering gateway network and a set of application programming interfaces for developers to interact with this network as a single entity, by abstracting the underlying gateway and network complexities. An overview of this architecture is described in Figure 1. We also introduce auxiliary devices to enable network management and application deployment. We believe that our approach will help replace costly, ad-hoc approaches for edge computing with a more cost-effective distributed model that reuses existing hardware.

2 DESIGN OVERVIEW

We present the workflow of the system and detail the components that are involved in the workflow. The first step for creating a distributed platform is to address the heterogeneity of gateways.

We require these gateways to have a common radio (Bluetooth Low Energy (BLE) is used in our current implementation) to discover each other using advertisements. In addition, these gateways have a backhaul network interface (WiFi, Ethernet, etc.) which is used for all further communication among the gateways. Once gateways are discovered, we use a live deployment graph called a link graph to depict the relationships of which gateways are discoverable to each other, and which sensors are connected to each of these gateways. Figure 2 shows the link graph for our sensor deployment.

The next step is to provide developers with abstractions to run applications over this platform. For this, we have the Gateway and Service APIs. The Gateway API provides a limited interface to query the specifics of a gateway. This includes details like neighboring gateways, connected sensors, and other infrastructure details like the network interface in use, and network statistics. The Service API, built on top of the Gateway API, provides useful abstractions to interact with the gateway network as a single computing platform, irrespective of where the application is deployed or how the underlying network topology is. Figure 3 explains how the Gateway and Service APIs are structured on the gateway network. Service API lets application developers interact with the distributed platform to disseminate data to other gateways, query gateways, and register for sensor streams. Lastly, we allow smartphones, laptops, or other devices that use BLE to connect to the gateway network. This allows users to deploy applications, manage gateways or devices, or interact with the gateway network to discover services and devices that are nearby.

3 DEMONSTRATION

Our demonstration consists of three parts. First, we showcase the link graph of auto-configuring gateway and device deployments. This will be a live view of the gateways and devices that we deploy at the venue. Second, we demonstrate example applications that showcase the API support for interacting with the devices in the network. Specifically, we will demonstrate deploying an app which consumes streams of sensor data from the devices in the network. We will also demonstrate how EnOcean energy harvesting sensors [1] that are discovered by a gateway can share this discovery knowledge to all other gateways in the network, allowing for improved sensor mobility. Also, we demonstrate connecting personal computers (laptops, mobile phones) as auxiliary devices to the gateway network for network oversight and code deployment. Participants can use their smartphones to discover devices that are part of the network, and be notified when sensor values satisfy certain specified conditions. They can also use their laptops to deploy applications that run on the distributed platform.

4 ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation under grant CNS-1823325.

REFERENCES

- [1] EnOcean GmbH. 2019. Energy Harvesting Wireless Sensor Solutions from EnOcean. (2019). Retrieved April 1, 2019 from <https://www.enocean.com/>
- [2] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai. 2014. Cloudlets: at the leading edge of mobile-cloud convergence. In *6th International Conference on Mobile Computing, Applications and Services*. 1–9. <https://doi.org/10.4108/icst.mobica.2014.257757>