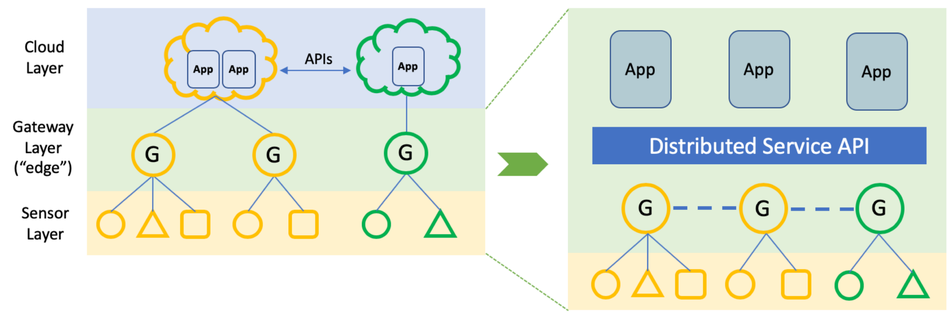
An Architecture for Edge Computing over Underutilized Gateways

Nabeel Nasir
nabeeln@virginia.edu

Bradford Campbell
bradjc@virginia.edu

Motivation

- IoT applications typically follow a cloud-based model with the data and apps on the cloud.
  - Leads to potential privacy issues, higher latency, and siloed ecosystems.
- Moving applications to the edge is a solution, but cloudlets for edge computing are expensive, need to be provisioned and installed, and are not distributed spatially in the deployment.
- **Hypothesis**: Use existing gateway devices in IoT deployments to create a dynamic and distributed, yet standardized, computing platform to enable applications at the edge.
- **Vision**: Robustness similar to cloud computing for the edge by abstracting underlying network and other deployment complexities into a usable platform.



Transition apps from a Cloud-based Internet of Things Network to a distributed network over gateways

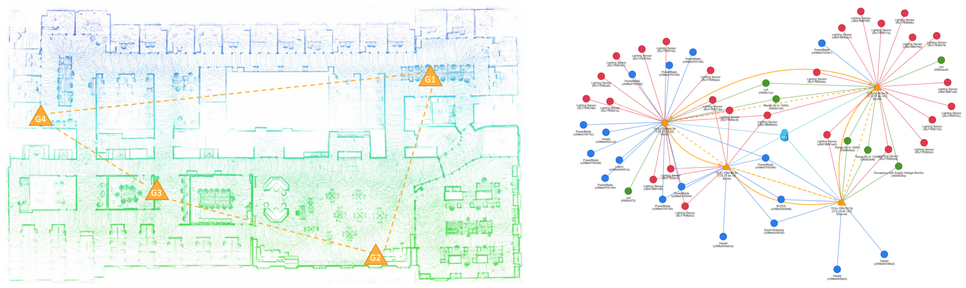
Approach

(1) Gateways discover each other using an out-of-band Bluetooth Low Energy radio. They use a different backhaul network for all higher bandwidth communication.

(2) The current network state (discovered gateways, network reachability, and connected sensors) is abstracted as a **Link Graph**. Each gateway maintains an updated version of this graph.

(3) Applications can use APIs to interact with the gateway network—to disseminate information and query gateways, and to register to sensor streams.

(4) Laptops or smartphones can temporarily connect to the gateway network for service discovery, event notifications, network monitoring & management, and application deployment.



Illustrates the distribution of gateways in our deployment and the resulting link graph visualization

Design Goals

Reduce configuration overhead: Support *plug-and-play* gateway devices for scaling up deployments

Enable distributed application development: Provide a high-level API to facilitate seamless gateway-to-gateway state transfer, network-wide dissemination, and device interaction.

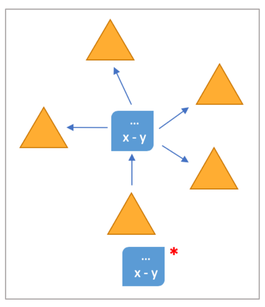
Support for heterogeneous gateways: Support gateway devices with different network interfaces or wireless radios.

No central entity: Should not be reliant on a link to cloud to manage the whole network.

Facilitate service and device discovery for users: Users must be able to discover the smart network and services.

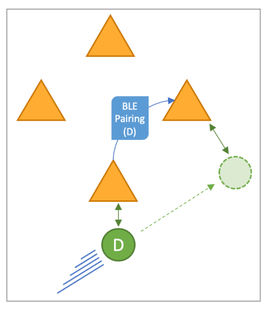
Provisions for network management: With networks likely comprising of numerous devices, wireless protocols, and operating profiles, the architecture must provide facilities for managing the network.

Applications



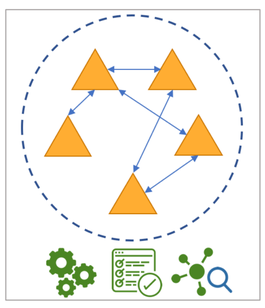
State Dissemination

Gateway devices that have state information that needs to be synchronized with other gateways.
E.g.: Sharing EnOcean sensor mappings



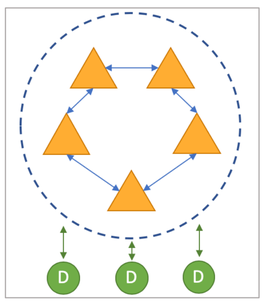
Hop-by-Hop Transfers

Applications where state may need to follow devices in a building-scale or larger network.
E.g.: SeamBlue, DTLS encryption



Global Oversight

Monitoring connected gateways and devices, bootstrapping common configuration, sharing specific configuration etc.



Central Platform

Applications that maintain shared state across gateways and synchronize updates.
E.g.: Federated Machine Learning without Cloud support

Implementation

Deployment details

- Gateway hardware: Samsung ARTIK 530s (Quad-core ARM Cortex-A9)
- Our deployment has four gateways and 274 sensors (BLE, RF)
- Sensors used: temperature sensors, estimote beacons, door sensors, lighting sensors, air quality sensors, power monitors, occupancy sensors

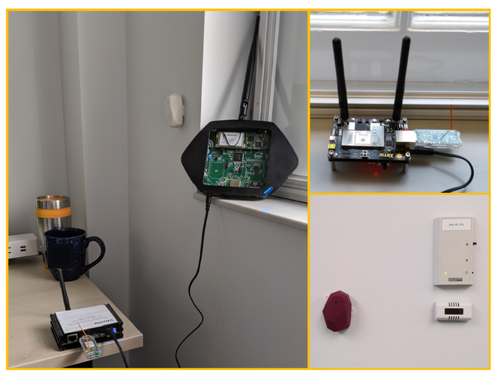
Application Development

Service API (to interact with the gateway network)

- disseminate (recipients, tag, data)
- query (recipients, tag, replyTag, data, callback)
- subscribeToSensorData (sensorID, callback)

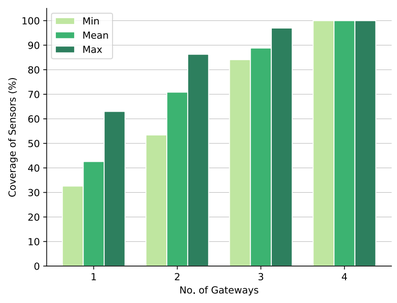
Gateway API (to interact with individual gateways)

- /neighboringGateways
- /connectedSensors
- /linkGraph

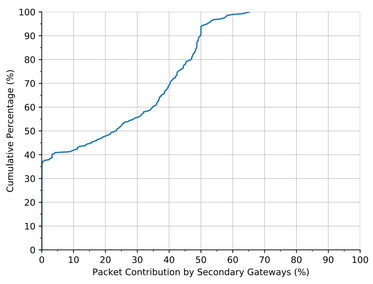


Gateway and sensor devices deployed in our lab

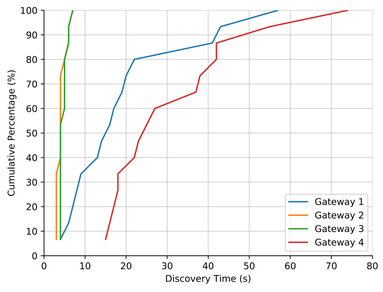
Evaluation



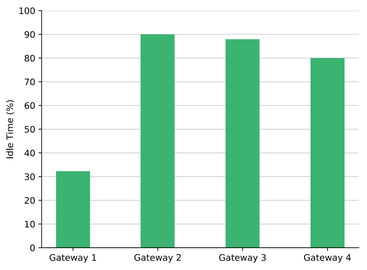
Building scale deployments require multiple gateways
Even a strategically placed single gateway only covers 62% of deployed sensors, and at least 3 gateways are required to provide 90% coverage.



Multiple gateways needed to receive sensor data without loss
CDF for the packet contribution by secondary gateways for all sensors in our deployment. 50% of the sensors have secondary gateway contributions of more than 25%.



Gateway discovery time is within acceptable limits
Overall, the discovery time is <50 seconds 90% of the time for all gateways for 15 trials.



Gateways are underutilized and have CPU cycles to spare
Idle percentage as reported by the iostat utility across an average of 27 days. Most gateways spend the vast majority of their time idle, with one gateway running additional web data scraping applications.