SoftBLE: An SDN Framework for BLE-Based IoT Networks

Mehdi Jafarizadeh, Xingzhi Liu, Rong Zheng

Authors

Mehdi Jafarizadeh

Ph.D Candidate Researcher McMaster University (received Ph.D)

5G Developer at Shadobi

Xingzhi Liu

Masters' student researcher at McMaster University

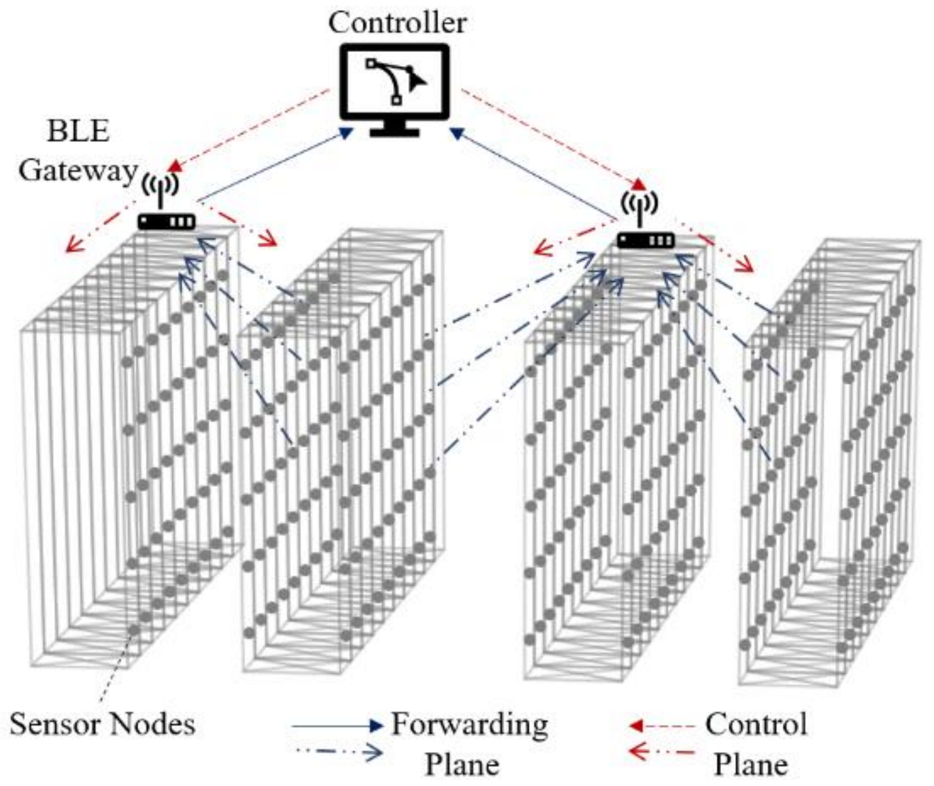
Rong Zheng

Professor in Dept. of Computing and Software at McMaster University

Canda Research Chair in Mobile Computing

Problem

- Dense IoT Device Network are difficult to work with
  - Dropped packets, high congestion, power wasted
- Dense sensor networks shown to have ‘GoodPut’ down $\emptyset(1/n)$ as ‘n’ devices increases
  - ‘GoodPut’ ability to successfully receive sensor data
- Example Problem:
  - Temperature sensors in a server room



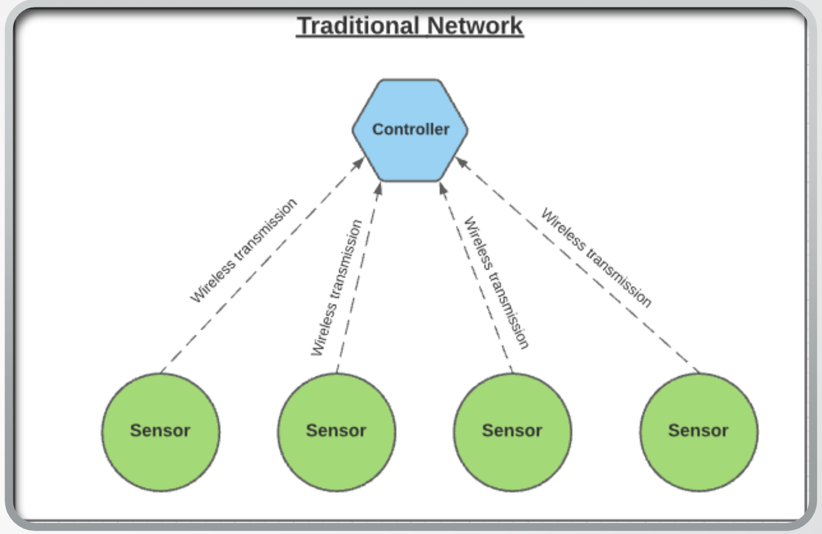
Goal

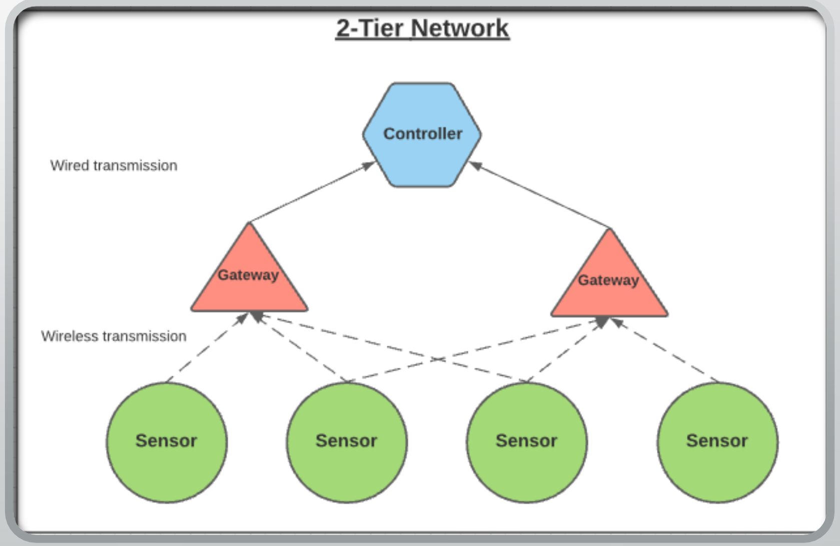
Increase Packet Reception Rate (PRR)

Decrease Power Consumption (PWR)

Hierarchical Networks (2-Tier Networks)

- Benefits:
  - Reduce traffic to controller
  - Decrease latency with gateway closer to sensor and wired connection to controller
- Problems:
  - Without customizability, lacks the ability to scale with more devices and handle dynamic changes in traffic





SoftBLE Solution

- What if we can control and customize BLE traffic AT RUN TIME to create an optimal network?
- A Software Designed Network (SDN) that provides controllability for BLE Based 2-Tier Networks
- **Contributions:**
  1. An SDN designed as an overlay on a two-tier network forwarding plane
  2. Two orchestration algorithms for optimized scanning parameters on the gateway and advertising parameters on the sensor

Brief BLE Background

- Bluetooth Low Energy is a wireless personal area network that is similar to Bluetooth, but emphasizes low energy consumption while sacrificing some bandwidth and connectivity features

BLE Background

- Light-weight and power efficient
- Uses 40 possible channels with only **3 for advertising**
- Has 5 possible states, but SoftBLE only utilizes 3 of them
  - Connected and Initiated are too power hungry

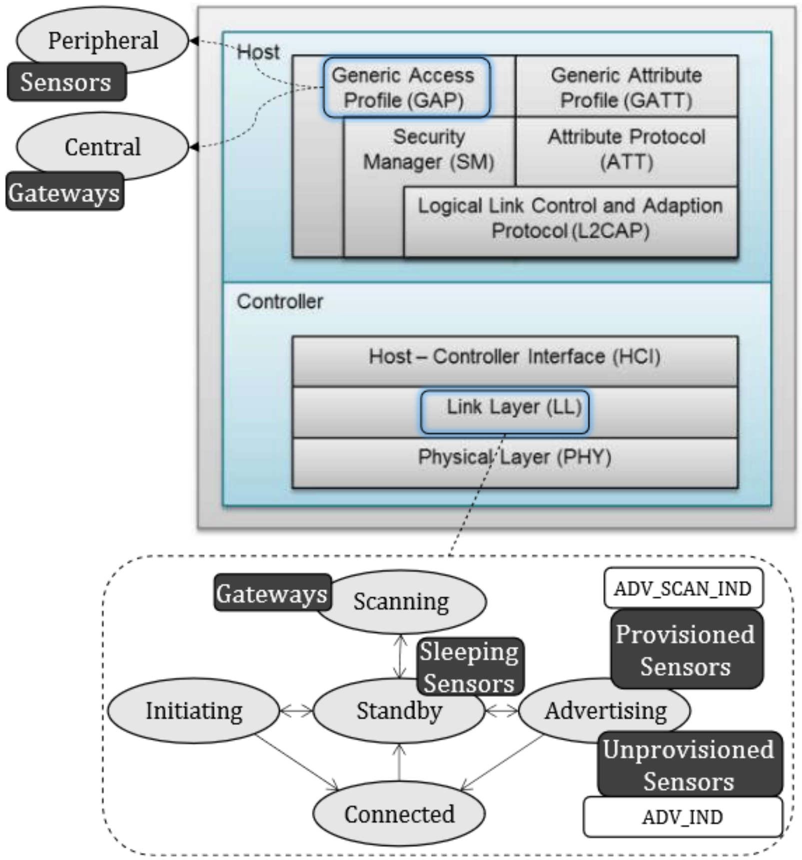


Figure 2: BLE protocol stack.

BLE Old Way

Sensors:

- Advertise for the "advertising interval" time + a "random delay"
- Can be configured via the following:
  - Advertising Channel Map
  - Advertising interval
  - TX power level
  - BLE Address (specifically Random Private Non-Resolvable) (Settable by the application)

Scanners:

- Scan on a channel map for a period of time called the scanning window
- Scan for a scanning interval then changes to next address on channel map
- Can be a passive scanner (scanning close to source)
- Can be an active scanner (accepts scan requests and connection requests form sensor)
- Can be configured to only scan a certain list of addresses

SDN – Software Designed Netowork

- SDN = Networking approach that uses software-based controllers with an emphasis on dynamic programmability for efficient networks
  1. **End User**: Sensor Nodes (generating data)
  2. **Flows**: Data ‘flows’ coming from sensors
  3. **SDN Switches**: BLE Gateways that forward flows
  4. **SDN Controller**: Central computer connected to gateways

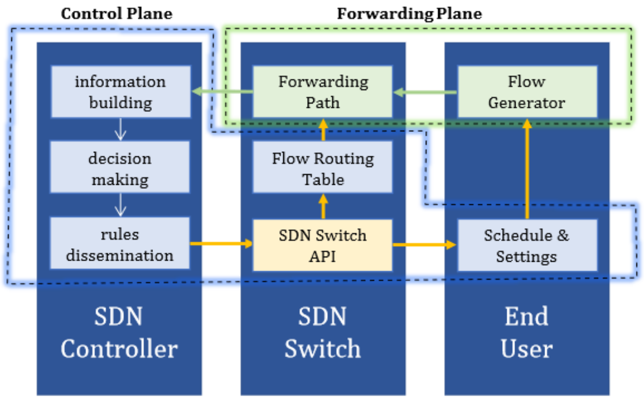


Figure 4: The components of SDN framework in a SoftBLE.

Provisioning of an End User (Sensor)

- Sensors start un-provisioned and continually send out provision requests
  - *'Please accept my data and let me into the network'*
- After a certain number of requests, the controllers configures the sensor and allows it in the network
  - *'Okay come on in, but here is your schedule and how you should act'*
- Upon provisioning an observation matrix is created

Sensor-Gateway Observation Matrix

Generated during provisioning

Many to Many matrix defining the RSS (Received Signal Strength) from the sensor to each gateway

$$O = \begin{bmatrix} o_{11} & \cdots & o_{1N} \\ \vdots & o_{ij} & \vdots \\ o_{M1} & \cdots & o_{MN} \end{bmatrix}, o_{ij} = \begin{cases} 1 & \text{if } \max\left(rss_j^i\right) > P_{sen} \\ 0 & \text{if } \max\left(rss_j^i\right) < P_{sen} \end{cases}. \quad (1)$$

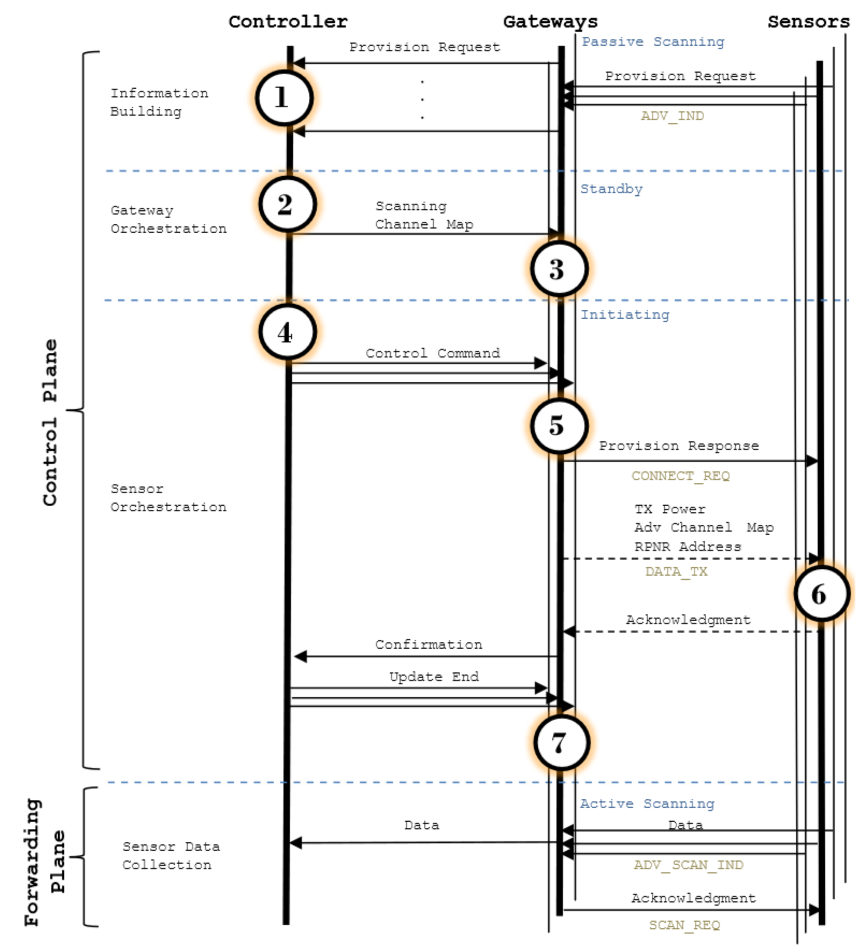
$$\forall i \in \{1,..,N\}, j \in \{1,..,M\}$$

What is configured in SoftBLE?

Channel Map of Sensors:	TX Power Level:	Advertise Address:	Channel Map of Gateway:	Whitelist of Gateway:
• What channels a sensor will broadcast to	• How powerful will the sensor broadcast	• What private address a BLE device will have when broadcasting	• What channels a gateway will listen to	• What 8 addresses will the Gateway listen to on that channel

Control Plane

- Sits on top of the forwarding plane
- Only job is to provision new sensors OR re-provision disconnected sensors
- Three Jobs
    1. **Information Building**: *'How strong is this sensor, what is our current sensor network'* (think observation matrix)
    2. **Gateway Orchestration**: *'Which channels are each gateway scanning and what is open'*
    3. **Sensor Orchestration**: *'Okay SENSOR you should broadcast this strong on this channel for this long and GATEWAY listen on this channel'*



Gateway Orchestration

- Gateway's only scan ONE channel, deciding what channel that should be is gateway orchestration
- Assign a Gateway channel such that the neighboring gateway has the least number of interfering sensors
- Max-Min Optimization Heuristic applied here to determine best channel:
  - Max: Find the gateway that has the max common sensors on the three channels
  - Min: Find the channel on that has the least number of common sensors and choose that one

Sensor Orchestration

- Sensors can advertise on any of the three channels at varying transmission power levels
- But:
  - More channels, more traffic, lower PRR
  - Higher Tx, more power, more traffic, lower PRR
- Sensor orchestration is attempting to find that happy medium

$$E[PRR]_s = 1 - (1 - \hat{prr}_s)^R.$$

$$\begin{aligned} E[PWR]_s &= \sum_{r=1}^{R} \frac{1}{\delta} \cdot E_s^{adv} \cdot E[PRR]_s (1 - E[PRR]_s)^{(r-1)} \\ E_s^{adv} &= P_s^{tx} \cdot \left\| C_s^S \right\| \cdot \left(\frac{|DATA|}{\mu} + P_{ifs} \right) \\ &\quad + P_{rx} \cdot \left(\frac{|SR|}{\mu} \right) + P_{ifs} + P_s^{tx} \cdot \left(\frac{|SS|}{\mu} \right) \\ P_s^{tx} &= P_{tx} \cdot 10^{(TX_s/10)}, \end{aligned} \tag{13}$$

Sensor Orchestration Factors

- Expected PRR:
  - Broken down into a Per Gateway PRR
  - Solving for this, want higher PRR
- Interference counter:
  - Number of potentially colliding sensors
- Traffic Load:
  - Sum of transmissions in a region
- Expected Number of Retransmissions:
  - Estimated number of times data must be re-transmitted
- Expected Power:
  - Solving for this, want lower power

Sensor Orchestration

- The final algorithm for sensor orechestration is a simple nested for-loop running in O(c) time
- First loop through all possible channel combinations (37,38,39), (37,38) etc.
  - For each channel combination loop through all 13 power configurations
    - Check for the lowest E[pwr] and highest E[prr]

Algorithm 2: TX power optimization on the sensors

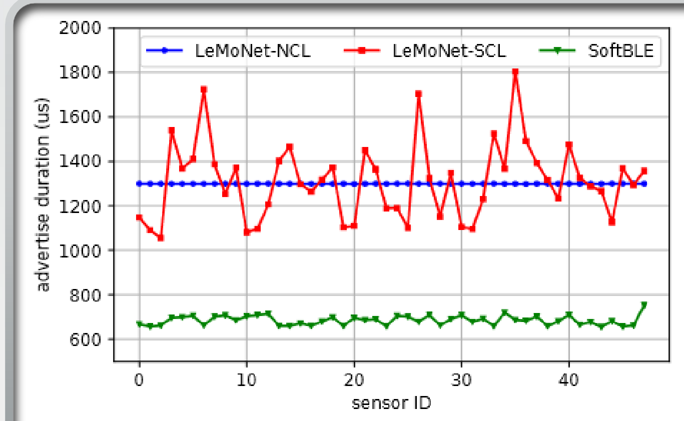
```
input  : sensor ID (s), observation matrix (O), RSS of provision
         requests received from s (rss_s)
output : assigned advertising channel map (C_s^S) and TX power
         (TX_s^S) to sensor s
PTX ← {−21, −18, −15, −12, −9, −6, −3, 0, 1, 2, 3, 4, 5}
bestC ← {37, 38, 39};
bestP ← 5;
bestPWR ← ∞;
for C ← Subsets of {37,38,39} do
    for p ← 1 to 13 do
        TX ← PTX[p];
        Estimate E[PRR_s] based on C, TX, rss_s, O using (5);
        Estimate E[PWR_s] based on E[PRR_s] using (13);
        if E[PRR_s] > T and E[PWR_s] < bestPWR then
            bestC ← C;
            bestP ← PTX[p];
            bestPWR ← E[PWR_s];
C_s^S ← bestC;
TX_s ← bestP;
```

Performance Analysis

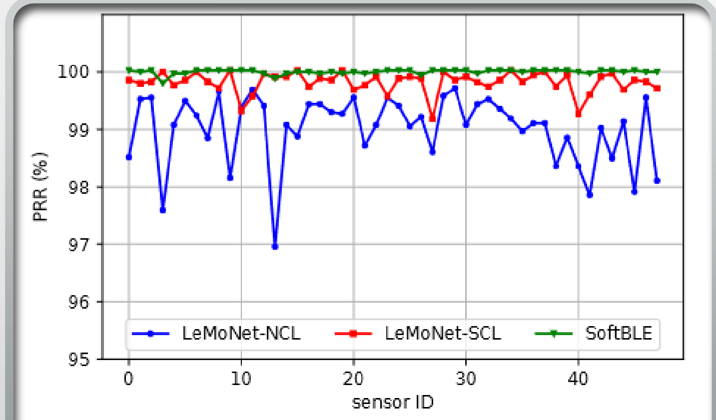
- Compared against LEMoNet
  - A static tiered network developed by this same research group
- LEMoNET vs. SoftBLE
  - In SoftBLE gateways only respond to sensors on their whitelist
  - RSS's are extracted directly from device and not estimated in SoftBLE
  - No NCL mode in SoftBLE
  - TX Power in Soft BLE is variable
- Run in two modes for LEMoNET:
  - Normal Connectionless Mode:
    - "Here is my data, take it if you want it"
  - Scannable Connectionless Mode:
    - "Here is my data, I will keep resending N times until someone sends a scan request confirmation that it was received"
- 48 sensors, 2 Gateways

Results

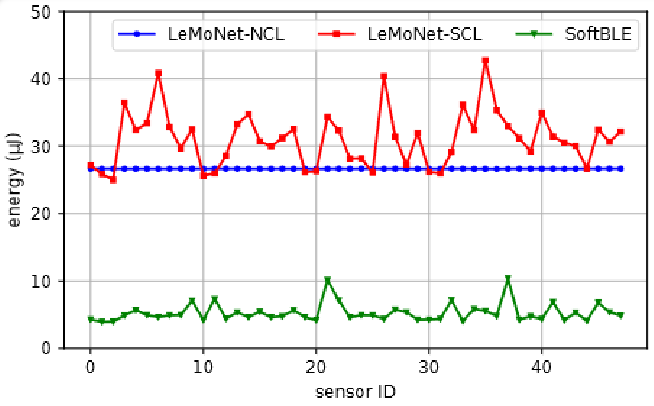
- Almost all sensors chose a single channel based on a higher RSS, but one device had a bad signal to both gateways and chose to broadcast on both channels
- PRR 99.9%
- PWR down 70%



(c) Advertise duration per duty cycle



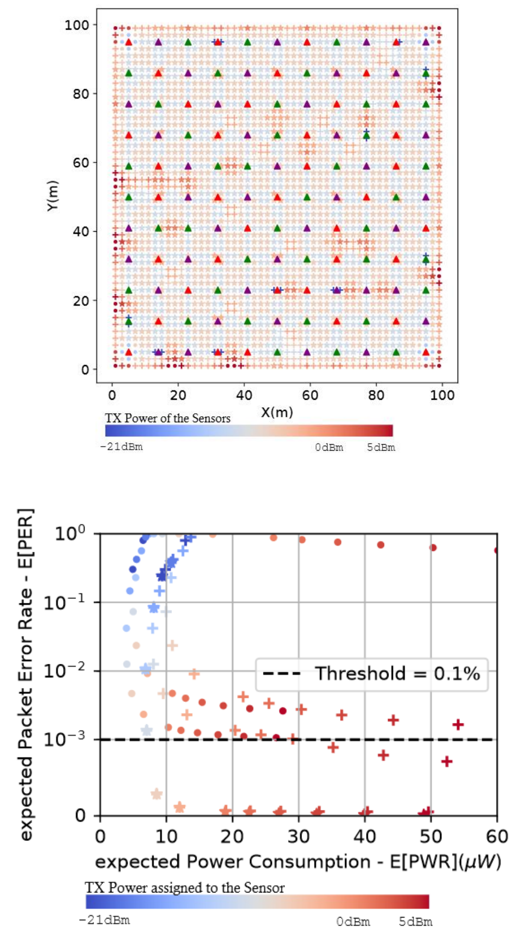
(b) Packet Reception Rate



(d) Mean energy consumption for TX per duty cycle

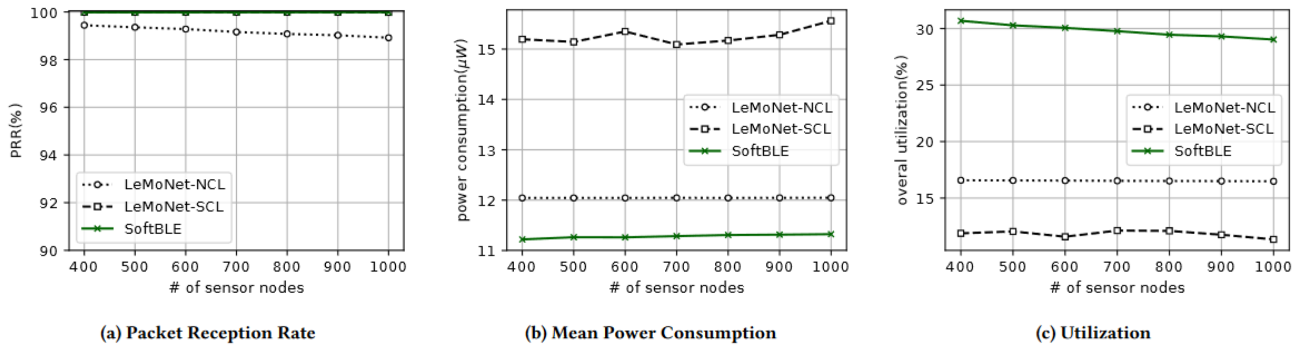
Simulation Study

- They also simulated a much larger network to test scalability:
  - Performance at Scale: How do 2500 sensors perform in network
  - Parameter Study: How do devices get configured in large networks
- See tradeoff in power and PER
- Notice how center sensors broadcast on 3 channels at low power and edge sensors broadcast on one channel at high power



Scalability + Duty Cycling Device

- SoftBLE scales very well
- Changing Duty Cycle (or how often data is pushed out) is able to be handled well by SoftBLE



igure 11: Effects of the number of sensors on the performance of SoftBLE. The sensors are deployed randomly in a $50m \times 5($
rea. Duty cycles of all sensors are set to 5s.

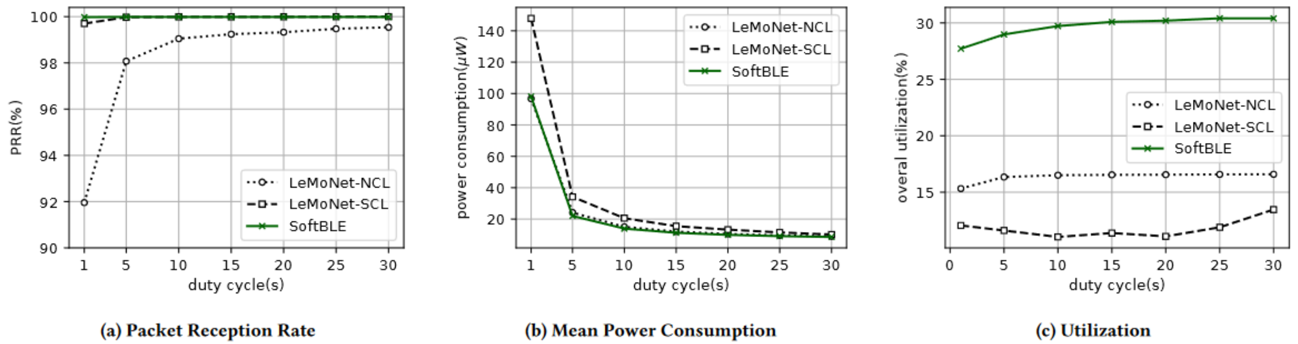


Figure 12: Effects of duty cycle on the performance of SoftBLE. 600 sensors are deployed randomly in a $50m \times 50m$ area.

Questions

- What if they worked on parameter tuning the advertising interval?
  - Could this lead to more efficiency or a lower PRR?
- How do you scale Gateways, since they are limited to 8 sensors per?
- Is it really necessary to choose the min-max algorithm or should gateways just choose channels that are not like their neighbor?