Assemble, Control, and Test (ACT): A Management Framework for Indoor IoT Systems

Ethan Perry
Department of Computer Science
Colorado School of Mines, Golden, Colorado
Email: eperry1@mymail.mines.edu

Qi Han
Department of Computer Science
Colorado School of Mines, Golden, Colorado
Email: qhan@mines.edu

***Abstract*—Internet of Things (IoT) networks have become increasingly popular in recent years, and while they may be installed in certain environments with relative ease, the systems increase in size, cost, and complexity as they scale to smart buildings. Since these devices do not exist in flat, open areas, but rather exist in buildings where concrete walls and metal structures obstruct device communication ranges, many of the algorithms and systems that work in theory fall short in such real-world scenarios. This research develops a novel relay placement algorithm for IoT system coverage which takes into account the impact of various obstructions on the performance of wireless communication. In addition, this algorithm is incorporated into our IoT network deployment and management framework. We first evaluated our approach in simulation, then tested the system in a real-world scenario where its effectiveness is compared to previous systems and algorithms.**

I. Introduction

It is clear that the world of IoT (Internet of Things) is expanding and only beginning to provide its multitudes of benefits. As a result of the pervasive nature of these systems, many IoT devices are battery operated and have limited energy sources to reduce their cost of installation, maintenance, and increase their portability [1]. However, these devices must continue to deliver important data at all times.

An important application of IoT is the monitoring of smart buildings, which can house thousands of devices. The amount of sensors required for such large scale systems contributes to their immense cost. A method to reduce this cost is to minimize the use of sensor nodes where possible. Such a minimization can be achieved with relay nodes, or IoT devices which contain the same hardware as corresponding sensing nodes but lack the sensors. These relay nodes can carry messages across the network in place of sensor nodes.

A common problem during the installation of relay and sensor based IoT systems regards efficient relay placement, such that relays can facilitate proper communication between all sensor nodes. This can often be a difficult and expensive task, as the incorrect placement of relay nodes can cause malfunctioning sensor networks and challenging problem diagnoses. Thus, utilizing algorithms for effective relay node placement ensures both accurate coverage and a minimization of required relay nodes. Many existing relay placement algorithms [2][3][4][5][6] overlook the imperfect communication ranges of sensor and relay nodes due to obstructions like walls, furniture, and appliances. There is a need for an algorithm which accounts for these obstacles, both to ensure connection quality and also to minimize required relay nodes.

Furthermore, existing IoT systems overlook challenges faced by network deployment in real-world indoor scenarios. Having taken significant time to manually enter coordinates of sensor nodes and somewhat blindly place relay nodes during preliminary testing, it was identified that a visual way of correlating the physical and virtual networks would prove extremely useful. This paper presents Assemble, Control, and Test (ACT), an IoT management framework that incorporates the Obstruction Aware Relay placement (OAR) algorithm to ease the deployment of large scale indoor IoT networks.

II. Related Work

This work is related to two areas of research: large scale IoT network management and relay node placement. As IoT systems have grown in popularity, numerous network architectures and management systems have been proposed. Systems created by [7][8][9][10] all proposed similar features such as real-time topology control, centralized network control, fault management, power management, and a GUI. [10] specifically focused on a REST API and database schema for ideal control over the network in a web interface, aiming their system at facilities management where sensor network gateways connect to an overarching enterprise network. Ku et al. [11] offered a similar system aimed at energy management in smart buildings. Their proposed system is multi-layered, ingesting data in the lowest layer and processing the data as it flows upward towards the user interface. Surprisingly, no systems offer the capability to identify relay or access point placement; rather, they only offer monitoring and management resources.

Our work assumes single-tier networks, where sensor and relay nodes have the same radios. In [2], Cheng et al. present a 3-approximate algorithm and a 2.5-approximate algorithm, both with $O(n^3)$ complexity, and both targeted at single-tiered networks. Lloyd and Xue [3] present a 7-approximate single-tiered algorithm using a minimum spanning tree (MST) based algorithm with complexity $O(nlg(n) + |MCST(x, r, R)|)$, where $|MCST(x, r, R)|$ is the cardinality of the MST. No experiments or simulations were performed in either investigation. Similar single-tier relay placement algorithms are also presented in [4] and [5], where experiments and simulations

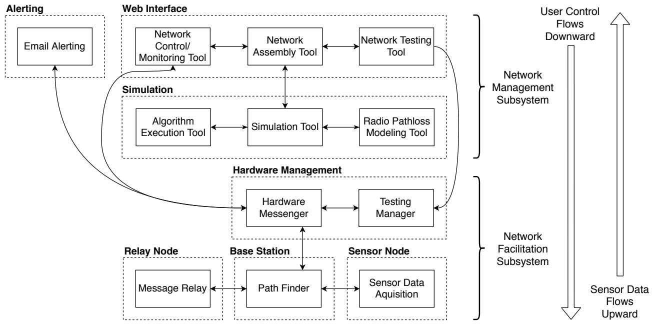


Fig. 1. ACT software architecture, demonstrating the overarching *network management* and *network facilitation* subsystems.

were performed. All of these investigations assumed a constant communication range for both the sensor and relay nodes.

Steiner and minimum spanning trees were popular choices amongst a majority of relay node placement algorithms. One feature which was not studied in the aforementioned algorithms is obstacle awareness — meaning that these algorithms, while being applicable to open spaces with no obstacles or terrain, may not produce satisfactory results indoors.

III. ACT Framework

ACT provides rich resources to network administrators for creating and thoroughly testing networks before managing and monitoring them. These IoT networks are comprised of sensor nodes, relay nodes, and base stations, which work together to provide effective indoor monitoring. ACT consists of two subsystems. The *Network Facilitation Subsystem* is responsible for maintaining communication between the base station and all nodes, while the *Network Management Subsystem* offers control and monitoring of the system from a web interface. The two subsystems, though ultimately very different, are intertwined in order to support overarching system functions. A holistic view of the system architecture can be seen in Fig. 1.

A. Network Facilitation Subsystem

In order to keep the network alive and communicating, the network facilitation subsystem runs on all nodes and the base station. It serves several critical functions, such as the assignment of node ID numbers, the facilitation of messaging across the network, and the recovery of lost connections.

Every node in the network, excluding the base station, has both a local and a parent ID number. For any given node, its parent ID number corresponds to the next upstream node before the base station (ancestor of all nodes) while the local ID represents the address the node is listening for messages on. The assignment of ID numbers can occur in two different ways. During provisioning of nodes, local and parent IDs can be labeled within their uploaded code. Alternatively, IDs can be assigned "over-the-air," where a base station communicates a new parent and local ID to installed nodes, allowing the network topology to be dynamic.

Network nodes have no understanding of the rest of the network — they only read sensors, and interpret, process, and send messages. Accordingly, messages indicate to nodes when the message is not destined for them and should be forwarded to another node in the network. When a node receives this message, whose payload contains an array of ID's representing the message's path, the node strips the next ID from the array and passes the message to the node listening on that ID. This allows all network path processing to be handled on the base station, which is faster and more capable of processing message pathways than sensor or relay nodes.

There are two types of connection recovery. The first, being similar to the logical link sub-layer in traditional networking, helps affirm that a message is successfully transmitted between two nodes. If the sender does not receive an acknowledgement from the receiver, it will resend its message until too many failed transmissions have occurred. The second is similar to TCP in traditional networking. Upon a message being sent across the network, the corresponding node will expect an acknowledgement and will continue to send messages until one is received. It is only after several attempts that the sender will stop and consider the message delivery a failure.

B. Network Management Subsystem

The network management subsystem provides an interface for controlling and monitoring the system. Its four main functions are network assembly, control, monitoring, and testing. The web interface provides a three-step setup to tune a network and to affirm that it functions properly. The setup is called ACT, which stands for Assemble, Control, and Test.

Assemble. Inputting the topology of a network by node location is a time consuming task. Accordingly, the network assembly tool was created to aid in designing networks before they are physically installed. The assembly tool features drag and drop functionality to place nodes and obstacles around an interactive panel, as seen in Fig. 2. Users can perform the following functions to achieve their desired network:

- Add nodes at the origin or coordinate location.
- Freehand nodes by double clicking on the panel.
- Change the scale and dimensions of the panel.
- Drag and drop to move nodes.
- Set a node as a base station node.
- Remove a node from the network.
- Drag to widen or shorten the range of a node.
- Run a network through a relay-placement algorithm.
- Generate a minimum spanning tree for the network.

Users can choose between different algorithms depending on the environmental conditions. While a user might choose a traditional relay-placement algorithm when designing a topology for an open space, they would be recommended to use the OAR algorithm presented in this paper for indoor scenarios, which was specifically designed for such settings.

Control. If a user decides they have finished building their network of interest (relay nodes included), they can "set nodes for control," populating the control panel.

The control panel offers the ability to monitor and directly interact with the network from a centralized location. Upon setting the nodes for control, the selected topology will also

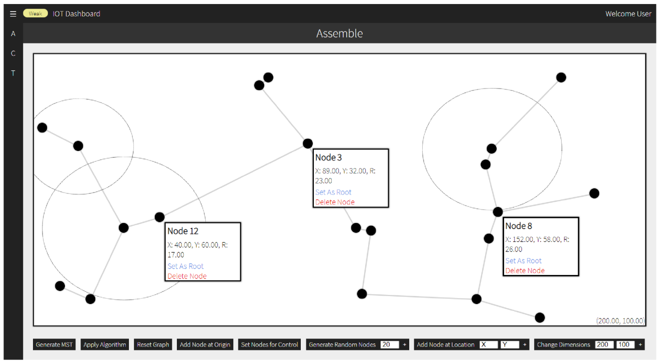


Fig. 2. A screenshot of a user interacting with the Assembly Panel by changing node ranges via dragging and opening an options menu via clicking.

be updated in the hardware services and ready to use with the physical network. At this point, the network can be set up as it is displayed in the assembly panel. Thus, a digital copy of the network exists on the web interface, requiring less direct monitoring and maintenance over the physical network, and being especially efficient with real-world IoT systems which contain hundreds of devices for environmental monitoring.

To update network status, a "refresh" button within the control panel can be selected. Assuming correct network setup, each node will change its status from "offline" to "online" as the base station establishes connection, and the IoT dashboard will change its overall status from "unhealthy" to "healthy." It will remain this status unless a node is disconnected.

The control panel features numerous functions:

- Direct messaging a numeric code to a node.
- Refreshing a node's connection.
- Deleting the node from the list.
- Testing a node by sending a series of test packets.
- Reassigning the role of a node.

Besides having an understanding of whether a node is online or offline, users can set up custom notifications and alerts on the control panel. With nodes reporting numeric codes when certain environmental events occur (for example, the temperature rising or a person passing by), status indicators can be set up for when events are sent to the base station.

Test. After configuring network control, users can test their network for reliability by customizing the number of packets to be sent to each node, the selection of nodes to be included in a network test, and the number of iterations the test should attempt. Test packets are by default formatted to 32 bytes, though users can customize the data sent to each node.

Network Alerting is a small yet critical part of the management interface. Users can set up email accounts to receive alerts from the system based on sensor data.

Network Simulations are used to support the relay placement algorithm designed for indoor scenarios. They operate the server-side of the management dashboard assembly panel. In ACT, relay algorithms proposed by Lloyd et al. [3], Cheng et al. [2], and the OAR algorithm are implemented. The simulation tool is flexible in supporting varying user groups, as it is able to process many randomly generated networks from the command line, or just a single assembled network from the front-end interface.

IV. Relay Placement Algorithm

Several factors are taken into consideration when placing relay nodes for indoor IoT networks:

- Sensor nodes will be placed regardless of surroundings.
- Sensor nodes will be placed strategically (discreetly in the corners of rooms and hallways).
- Sensor and relay nodes have the same range.
- Sensor nodes can also act as relays.

Before designing our relay placement algorithm, we investigate the impact of indoor environments on radio propagation.

A. Radio Propagation in Indoor Scenarios

The log-distance path loss model for radio propagation has been widely adopted in previous work. As opposed to log-normal shadowing, which can be used to compensate for overlooked clutter, we explicitly consider the obstacles between two radios rather than setting radio ranges to be equal.

Each obstruction's material has a unique impact on signal power. In buildings, these attenuating materials comprise walls and structures near the ceiling, where nodes are placed. Accordingly, the attenuation of each wall must be known to find an optimal arrangement of relay nodes. To measure this effect, we implemented a range-testing system on top of ACT.

We used NRF-24L01+ radios in our experiments. The transmission power is -18 dBm, antennae gain is 3 dB, and minimum receiving sensitivity is -82 dBm. These values hold for NRF-24L01+ radios in their minimum power mode, representing real-world devices saving as much power as possible.

First, a log-distance path loss model was tuned for the main floor of the house used as the real-world test scenario. To tune this model, multiple pairs of radios were placed in different locations of the house with a clear line of sight between each pair. After placing one radio in a stationary position, the other radio was slowly moved away until the majority of packets were dropped between the two radios. After testing five pairs of radios, the average range was calculated to be 32 feet. Using a reference distance of one meter (i.e., 3.28084 feet), a path loss exponent of $\gamma = 2.9898$ was calculated.

The same process was repeated with radios placed so that a wall separated them. Wall materials were classified into four categories. Thin doors, glass windows, and other minor obstructions were categorized into the "thin barrier" type. Walls comprised of gypsum boards were grouped into the gypsum wall category. Thick barriers consist of a gypsum wall plus another obstruction such as wooden cabinets or a substantial wall decoration like tile. Lastly, double layer clay-brick walls were grouped into the brick wall category. The average range of radios accounting for these material obstructions are shown in Table I.

Based on the indoor path-loss model and the attenuation of walls, it can be estimated whether two radios form a reliable connection. Given the significant difference in attenuation

TABLE I
RADIO COMMUNICATION RANGE THROUGH DIFFERENT MATERIALS

Obstruction Type	Average Radio Range	Power Attenuation
Open Space	32 ft	
Thin Barrier	25 ft	3.2054 dB
Gypsum Wall	19 ft	6.7688 dB
Thick Barrier	15.75 ft	9.2046 dB
Brick Wall	3 ft	30.7375 dB

between each building material, there is strong motivation to design an algorithm that takes advantage of these findings.

B. Obstruction Aware Relay (OAR) Algorithm

All symbols used in the OAR description are summarized in Table II. Rather than prioritizing minimum length of edge connections as many existing algorithms do, the OAR algorithm minimizes the power attenuation between radios in a multi-step process. During **Step 1**, an edge is formed between every node. Using this set of edges, a minimum spanning tree prioritizing lower *node weakness* ($w(e)$) is created (see Table II). This is a metric used to represent the number of relays that must be placed; prioritizing edges with lower weakness will require fewer relays. In **Step 2**, relays are equidistantly placed between two sensor nodes to bridge a connection (line 7). This does not always result in an efficient bridge, however. Equidistantly placed relays may overestimate the required relays if one sensor node is placed behind a wall, for example. Therefore, after bridging the sensor nodes, a check is performed to eliminate unnecessary bridging relays (lines 10-25).

While Kruskal's Greedy Algorithm in **Step 1** has a complexity $O(n_r^2 log(n_r))$ for n_r^2 total edges, this does not include the summation of obstruction attenuation, which consists of a linear scan through n_o obstructions to find intersections. Thus, this step has a complexity $O(n_o n_r^2 log(n_o n_r))$. **Step 2** of the algorithm loops through the $n_r - 1$ edges of the minimum spanning tree. Lines 10 through 25, where unnecessary relays are removed from an edge, yields a worst case complexity of the longest edge $max(E)$ divided by the range of the smallest radio $min(R)$ for a $O(n_o n_r^2 log(n_o n_r) + n_r(max(E)/min(R)))$.

V. Evaluation of OAR via Simulation

The simulation tool in ACT was written for the dual purpose of testing the relay placement algorithm and supporting the web interface. This tool can be run from the command line, taking parameters of trial numbers, obstacles, sensor nodes, and testing space dimensions, and returning the number of relay nodes, number of walls crossed by edges between nodes, and overall power loss due to walls and obstructions. For the purposes of this experiment, a simulation model was created with regard to real-world parameters. The virtual testing space was set up with 20 nodes and 30 walls/obstacles in a 100 by 100 foot area. The obstacles were randomly generated and given material properties mimicking the attenuating properties of brick and gypsum walls, and thick and thin barriers.

Algorithm 1 Obstruction Aware Relay (OAR) Algorithm

1: **Step 1:**
2: **for** each r in R **do**
3: Form a unique edge with every other radio r and put that into the set E for a total of n_r^2 edges.
4: **end for**
5: Generate a minimum spanning tree (MST) M with Kruskal's greedy algorithm using each edge's weakness $w(e)$ as it's priority — a lower weakness is favored.
6: **Step 2:**
7: **for** each e in M **do**
8: **if** $c(e)$ is $false$ **then**
9: Remove the edge from M and create $floor(d(e)/r(e,\gamma))$ new radios placed equidistantly between the radios comprising the edge e, forming a set of edges called E_0.
10: **for** each e_0 in E_0 **do**
11: Let p_1 be the first radio of e_0.
12: Let p_2 be the second radio of e_0.
13: **for** each edge e_1 connected to p_2 **do**
14: Let p_3 be the radio connected to p_2 by e_1.
15: Let e_2 be an edge connecting p_1 and p_3.
16: **if** $c(e_2)$ is $true$ **then**
17: Remove e_0 from E_0 and e_1 from E_0.
18: Remove the relay radio p_2.
19: Insert e_2 into E_0.
20: Let p_2 equal p_3.
21: **else**
22: Break the loop.
23: **end if**
24: **end for**
25: **end for**
26: Insert each e in E_0 into M.
27: **end if**
28: **end for** ▷ Algorithm Output: Optimal Set of Edges M.

We have chosen to compare OAR against two existing algorithms that are designed for single-tiered networks: the $O(n^3)$, 3-approximate algorithm identified in [2], and the $O(nlg(n) + |MCST(x,r,R)|)$, 7-approximate algorithm from [3]. Neither has considered obstructions between nodes.

Since the two baseline algorithms assume constant range for every radio, the simulation took a "weakest link" approach, making the range of all radios in the simulation equal to the range of the weakest radio link. Only this way could there be a guarantee that every radio in the network would end up connected after running the algorithm. Table III shows the results of 10,000 simulated trials. The simulation results indicate that OAR significantly outperforms the other two algorithms, as it uses many fewer relays and avoids more power attenuation due to walls and obstructions.

VI. ACT Performance Evaluation via Testbed

To evaluate our system in a more realistic environment, we set up an IoT system for building monitoring. We used Arduino Nanos in combination with the NRF24L01+ (NRF) radios. While sensor nodes consist of a microcontroller, wireless radio, and a sensor setup with temperature, humidity, and smoke detection, relay nodes only consist of a microcontroller and a wireless radio. This means that sensor and relay nodes have the same communication capabilities, but that relays

TABLE II
SYMBOLS USED IN OAR

Variable	Description
R	The set of radios which can transmit and receive messages from one another.
r	A single radio.
n_o	The number of obstacles.
E	A set of edges, each of which connects two radios together.
e	A particular edge in the set.
$a(e)$	The sum of the attenuation of all obstacles obstructing the edge formed between two radios.
$d(e)$	The distance between the two radios which comprise the edge e.
γ	The log-path loss constant tuned to an indoor environment.
$r(e, \gamma)$	The range of the radios in an edge e accounting for the cumulative attenuation of intersecting obstructions.
$c(e)$	A boolean representing if an edge is *truly connected*, true if the two radios of the edge encompass each other in their range.
$w(e)$	The weakness of an edge, equal to the length $d(e)$ of an edge, divided by the range of the radios of the edge $r(e, \gamma)$

TABLE III
RECOMMENDED RELAYS AND POWER LOSS FOR SIMULATED ALGORITHMS

Algorithm	Average no. of Relays	Total Lost Power
OAR	9.08 Relays	135.7 dB
[2]	87.77 Relays	326.29 dB
[3]	124.45 Relays	407.11 dB

cannot sense environmental data. Base station Intel NUC master computers are connected via USB to two or more Arduino Nano devices, each with their own NRF radio.

A. Physical Experiment Setup

In such a way that reflects the purpose of this research, being a practical model for real-world, indoor environments, the main floor of a house was used as a test setting. Attenuating obstacles such as brick walls, tile surfaces, appliances, gypsum wall, and wooden doors are present in this setting.

Different types of environmental monitoring require significantly different node arrangements. For example, measuring the temperature of different rooms likely requires fewer nodes than setting up several motion detectors in each room. To consider various application scenarios, two network densities were chosen for testing: a higher density ten-node setup and a five-node setup. Additionally, while some networks require strategic arrangements of nodes (i.e., setups where nodes are placed discreetly in corners of rooms), other networks can simply maintain grid-like arrangements, where each node monitors a roughly equal-sized area. Both of these network arrangements were taken into consideration with the two densities of network nodes, totaling four network arrangements (an example is seen in Fig. 3).

Using the network assembly tool in ACT with measurements of obstruction attenuation for the main-floor test setting, relay node placements were generated with the OAR algorithm. For comparison, relay placements were also generated using the algorithms developed by [2] and [3]. These algorithms considered the range of each radio as 32 feet — the range in the indoor open space environment as measured in our radio propagation studies. However, since these algorithms prioritize minimum distance between each radio, neither algorithm recommended any relay nodes for this particular network topology. They both simply recommended the minimum spanning tree connecting all nodes. Since these algorithms recommended identical network topology, their testing was grouped together into the "non-obstruction aware" group.

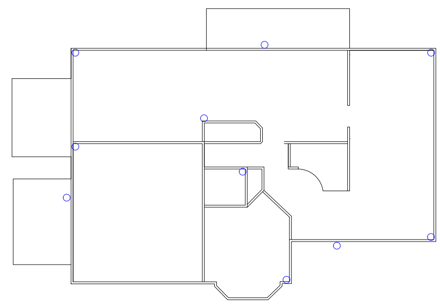


Fig. 3. An example of a strategic network with ten nodes. Nodes are placed inside and under external structures (porches and sheds).

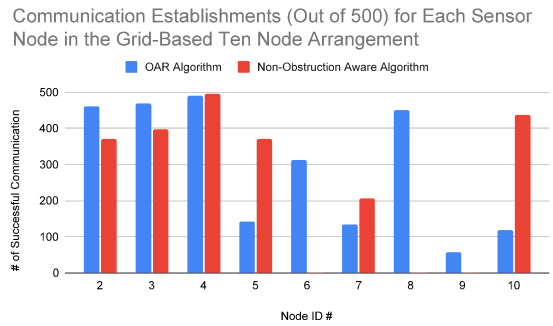
To test each network topology with each algorithm group, a "communication establishment", or the process of a message traveling all the way from the web interface to a sensor node and back, was made with every node 500 times. This tested our system and each of its individual components end-to-end.

B. Physical Experiment Results

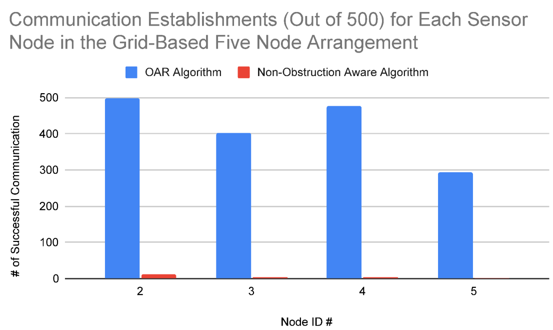
Considering overall reliability in each tested network, the OAR algorithm outperformed the non-obstruction aware algorithms (Fig. 4). Similarly to the simulation results, the OAR algorithm tended to avoid walls with higher attenuation, such as brick walls and thick barriers, while the non-obstruction aware algorithms generally intersected more walls of all kinds, accumulating much higher power attenuation (Tables IV). These results were particularly obvious in the grid-based arrangement of five nodes, as the non-obstruction aware algorithms barely managed to transmit any data across the network where the OAR algorithm maintained the majority of the communication, as seen in Fig. 4b. This was also true in the strategic arrangement of ten nodes (Fig. 4c).

VII. CONCLUSION

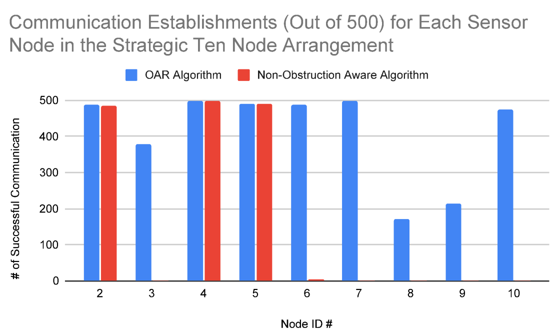
The ACT framework and OAR algorithm are designed to resolve a major issue relating to deployment and management



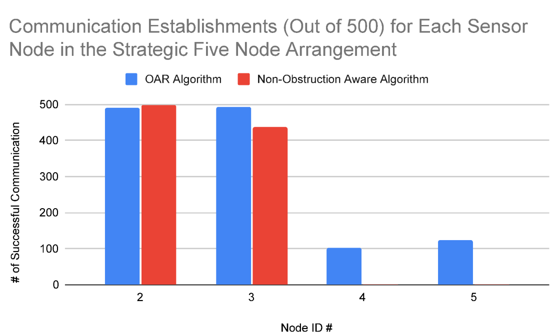
(a) The grid-based ten node network arrangement.



(b) The grid-based five node network arrangement.



(c) The strategic ten node network arrangement.



(d) The strategic five node network arrangement.

Fig. 4. Graphs displaying total communication establishments (Out of 500) for each sensor network.

TABLE IV
TOTAL OBSTRUCTION POWER ATTENUATION FOR EACH TESTED SCENARIO

Network Type	**OAR Algorithm**	**Non-obstruction aware Algorithms**
Grid-based Ten Node	33.844 dB	204.1756 dB
Grid-based Five Node	20.3064 dB	44.2751 dB
Strategic Ten Node	100.8613 dB	138.3676 dB
Strategic Five Node	40.6128 dB	54.1504 dB

of indoor IoT networks, i.e., relay node placement taking into account the walls and other obstructions. To simplify the management of the network, we developed the ACT framework to assist researchers, network administrators, and technicians alike. In simulations, the OAR algorithm significantly outperformed existing algorithms for single-tiered IoT networks in terms of the number of relay nodes it recommended and the power attenuation it avoided. Additionally, the ACT framework, tested in a end-to-end manner, had significantly better performance with the OAR algorithm as opposed to the non-obstruction aware relay algorithms. The reduction of power loss due to wall and obstacle avoidance prevents the unnecessary additions of relay nodes in indoor networks, also thereby preventing excess network power consumption, installation and maintenance costs, and network setup failures. In the hope of achieving a new optimum, pervasive walls and obstructions must be considered as sensor networks and IoT systems become the new normal for modern buildings.

REFERENCES

[1] J. Henkel, S. Pagani, H. Amrouch, L. Bauer, and F. Samie, "Ultra-low power and dependability for iot devices (invited paper for iot technologies)," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 954–959.

[2] X. Cheng, D. Du, L. Wang, and b. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, June 2008.

[3] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 134–138, Jan 2007.

[4] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1399–1411, Aug 2012.

[5] C. Ma, W. Liang, M. Zheng, and H. Sharif, "A connectivity-aware approximation algorithm for relay node placement in wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 515–528, Jan 2016.

[6] H. Zeng and Z. Kang, "Relay node placement to restore connectivity in wireless sensor networks," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, May 2017, pp. 301–305.

[7] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-wsn: Software-defined wsn management system for iot applications," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2074–2081, Sep. 2018.

[8] Y. Moon, J. Lee, and S. Park, "Sensor network node management and implementation," in *2008 10th International Conference on Advanced Communication Technology*, vol. 2, Feb 2008, pp. 1321–1324.

[9] X. He, Y. Ma, and M. Mizutani, "Efficient multiple gateway system for wsn management in bems," in *2012 Ninth International Conference on Networked Sensing (INSS)*, June 2012, pp. 1–2.

[10] A. Malatras, A. Asgari, and T. BaugÉ, "Web enabled wireless sensor networks for facilities management," *IEEE Systems Journal*, vol. 2, no. 4, pp. 500–512, Dec 2008.

[11] T. Ku, W. Park, and H. Choi, "Iot energy management platform for microgrid," in *2017 IEEE 7th International Conference on Power and Energy Systems (ICPES)*, Nov 2017, pp. 106–110.