# Bluetooth Low Energy in Dense IoT Environments

Albert F. Harris III, Vansh Khanna, Güliz Tuncay, Roy Want, and Robin Kravets

Bluetooth Low Energy was designed as a low-power alternative to classic Bluetooth. However, the use of BLE in dense Internet of Things deployments results in high collision rates and wasted energy. To alleviate some of this contention, we present Opportunistic Listening, an extension to BLE active mode targeting IoT deployments with large numbers of tags and small numbers of scanning devices.


## ABSTRACT

Bluetooth Low Energy was designed as a low-power alternative to classic Bluetooth. However, the use of BLE in dense Internet of Things deployments results in high collision rates and wasted energy. To alleviate some of this contention, we present opportunistic listening, an extension to BLE active mode targeting IoT deployments with large numbers of tags and small numbers of scanning devices. For dense deployments of passive advertising devices, we present the design of Smart LaBLEs (BLE-enabled, electronic, de-centralized hubs), which aggregate multiple advertisements across similar products in a retail environment.


## INTRODUCTION

Just as dramatic improvements in networking technology brought the Internet to the masses, new advances in the miniaturization of computation and lower-power wireless communication are bringing the Internet to "things". This Internet of Things (IoT) is built on the backs of small devices capable of operating for years on a single battery, while transmitting information out into the environment around them. Users with IoT-enabled smartphones navigate through the world, interacting with these devices embedded throughout their environment. While computational devices have rapidly become smaller and cheaper, finding the right wireless technology has been more elusive. No single technology provided sufficiently low-power communication to support extended lifetimes, an effective communication range that enables proximity-based interactions at human scales, and broad deployment on all user devices.

Given the target of ultra-low-power short-range communication, numerous wireless technologies showed potential, including near field communication (NFC), RFID, Bluetooth Classic, and ZigBee. However, each of these technologies had a fatal flaw for IoT deployments. Although available on all new smartphones, NFC is limited to very short distances, approximately 2 inches. Such a limitation does not support the wide variety of applications expected in IoT environments, where, for example, a user may want to interact with a variety of products as they walk around a store. While RFID has a larger range, the cost of an RFID reader currently prohibits it from being deployed on commodity user devices. Although Bluetooth Classic satisfies the low-power communication requirements and is available on most mobile devices, it is unsuited to IoT application demands due to its complex discovery mechanisms. Finally, ZigBee is a promising choice that supports low-power communication that can be specialized to IoT applications. However, ZigBee has not been taken up by the smartphone and mobile device industry, limiting any extensive use or public deployment.

In the face of these limitations, Bluetooth Low Energy (BLE) has emerged to dominate the IoT community [1]. BLE was designed to eliminate the pairing and simplify the complex discovery inherent in Bluetooth Classic, while still supporting short data exchanges [2, 3]. While this satisfied most of the IoT requirements, the fact that BLE could be bundled with Bluetooth Classic chips on smartphones and mobile devices made it the right choice at the right time. Given the explosion of interest in IoT, experts predict large spaces, such as retail stores, will deploy tens, hundreds, or even thousands of BLE tags, all advertising products and services at the same time. Similarly, the number of smartphones and other devices scanning for and interacting with these tags could reach very high numbers. Before more of these BLE devices are deployed, it is essential to understand the impact of this increasing density of advertising and scanning on access to advertised IoT data.

While it is commonly expected that future IoT ecosystems will deploy a large number of tags, the impact of the density of such tags and the corresponding scanning devices was largely ignored in prior work [4–9]. The new demands for IoT-enabled devices makes it clear that contention in the BLE channel will quickly become one of the biggest road blocks to effective, large-scale IoT deployments. After presenting a short tutorial on the BLE channel, we explore two major sources of contention, the first one triggered by the increasing density of scanning devices and the second one simply due to dense IoT deployments of advertising devices.

The first source of contention is rooted in the use of BLE's *active scanning mode*, where scanning devices actively request extra information from particular tags. Although BLE provides this mode to facilitate extended data exchanges, control messages quickly overwhelm the wireless channel even with only a few scanning devices. To reduce such contention, we propose the use of *Opportunistic Listening*, which allows scanning devices to better share the channel by leverag-

Digital Object Identifier: 10.1109/MCOM.2016.1600546CM

*Albert F. Harris III, Vansh Khanna, Güliz Tuncay, and Robin Kravets are with the University of Illinois at Urbana-Champaign; Roy Want is with Google, Inc.*

ing responses from other devices' requests. The second source of contention is triggered when many devices all advertise in the same space, even when using *passive scanning mode*. In dense IoT spaces, significant channel degradation occurs when the number of advertising tags increases [10]. In response, we present the design and implementation of a decentralized aggregation hub for IoT environments called Smart LaBLEs. The name LaBLE is a combination of the word label and BLE, to give a sense of our device's capabilities. A Smart LaBLE dynamically associates with a set of nearby IoT devices advertising similar product information, provides an aggregate advertisement, and silences redundant devices. Next, we present a brief tutorial on BLE, highlighting the problems associated with the current advertising modes in dense networks.

## Bluetooth Low Energy

BLE was designed to reduce the energy cost for device discovery and the delay for simple data exchanges. In BLE, every *advertising tag* sends an advertising message once every advertising period, which can be configured per tag. Typical advertising periods are between 100 ms and 1 s. Shorter advertising periods can lead to faster data access at the cost of introducing more contention into the shared wireless channel, which can ultimately increase delay.

Since the goal of BLE is to reduce cost and delay, and increase performance of discovery, BLE separates connected communication from advertising by reserving dedicated advertising channels. While using only one channel for advertising would lead to massive contention on that channel, using many channels would lead to the long access delays associated with Classic Bluetooth as well as reduce the available bandwidth for data communication. To balance contention and delay, BLE restricts advertising to **three** advertising channels. Bluetooth does not use these channels for data, and these channels fall between or outside the main frequencies used for IEEE 802.11, allowing better coexistence with WiFi.

BLE tags follow a periodic advertising protocol. At the beginning of every advertising period, each tag transmits its advertising message on each advertising channel. Scanning devices cycle through the channels listening for advertising messages. The behavior of scanning devices is determined by the type of advertising message: passive or active. For both types, the scanning interval is designed to guarantee that a scanning device can receive an advertising message from a tag once every advertising period, assuming no loss or contention.

Since many co-located tags may use the same advertising period size, each tag randomly adds up to 10 μs of jitter to every advertising period to avoid repeated collisions. If two tags have overlapping transmissions in an advertising period, the added jitter significantly reduces the chance of another collision in the subsequent advertising periods. Given the small size of advertising messages, this approach avoids significant collisions in IoT deployments of up to 200 or more co-located tags.

To keep energy consumption low, all tags duty-cycle. Tags are only "on" for the short amount of time taken to send the advertising messages on the advertising channels. Energy is further saved by eliminating carrier sensing from the BLE medium access control (MAC) protocol. This simplification does not come for free, ultimately increasing the potential for collision.

In BLE's simplest form, tags operate in *passive mode*, periodically sending short advertising messages containing a payload of at most 31 bytes. Any BLE scanning device that receives a passive advertising message follows the *passive scanning* rules and does not respond in any way. In dense environments, it is easy to imagine the scores of advertising messages being broadcast by numerous tags throughout an IoT deployment.

For applications to advertise more than 31 bytes of data, the BLE specification [2] introduced an active mode, which allows the transmission of a second 31-byte message. When a tag sends an active advertising message, all scanning devices hearing the message follow the *active scanning* rules to complete the exchange of data. The use of active scanning affects all BLE advertising since the passive advertising messages are now competing for the advertising channels' bandwidth with both the active advertising messages and some additional request/response traffic. It is important to note that, according to the active scanning rules, even user devices uninterested in the extra data complete the request/response handshake. Additionally, the lack of carrier sensing combined with a higher density of tags and/or scanning devices in active mode leads to excessive collisions, the end result being the inability of the scanning devices to hear and so collect data from any BLE tag in a reasonable amount of time.

The lack of carrier sensing combined with a higher density of tags and/or scanning devices in active mode leads to excessive collisions, the end result being the inability of the scanning devices to hear and so collect data from any BLE tag in a reasonable amount of time.

### Passive Scanning

With passive scanning, the simplest mode, each tag periodically sends a passive advertising message (`ADV_NONCONN_IND`) on the three advertising channels. The advertising message can contain up to a 31 byte payload. Although there can be up to 10 ms channel wait time between the transmission of advertising messages, such messages are typically sent back-to-back on the three advertising channels. With such passive scanning, devices simply listen for the advertising messages and there is no direct interaction between the tags and the scanning devices.

### Active Scanning

Although 31 bytes may be sufficient for some applications, others have more data-intensive needs. To support such applications, BLE defines active scanning, where active advertising messages (`ADV_IND`) trigger a three-way handshake. **All** scanning devices receiving an active advertising message respond immediately with a scan request message (`SCAN_REQ`) unicast to the tag. The tag finishes the data exchange with a scan response message (`SCAN_RSP`), which can contain up to an additional 31-byte payload, effectively doubling the available payload space over passive scanning. Scan response messages are broadcast to all potential requesters (as opposed to unicast to the tag that triggered the response).

The advertise-request-response exchange is very quick, and the cost is limited to the transmission and reception of the scan request and

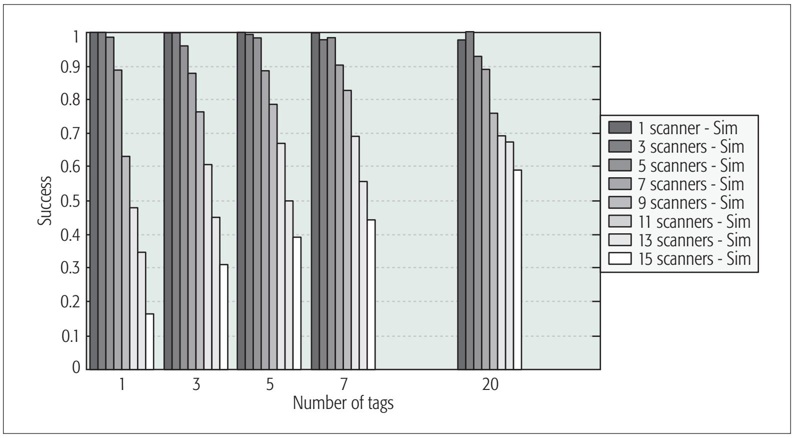


**Figure 1.** 5 s success for active scanning from ns3 simulations.

response messages. However, tags must wait some minimal amount of time after sending an advertising message to see if a scan request message was sent. Although the specification allows for up to 10 ms of channel wait time, tags typically wait only long enough to see if there was an immediate response. Tags respond **only** to the first scan request message in an advertising period on each channel.

Although the scan response message is broadcast, it is important to note that the BLE specification mandates that a scanning device only accepts and processes a scan response message if that scanning device sent a scan request message to that tag on the same advertising channel during that advertising period. All non-requested scan response messages are **dropped**. Thus, any device that intends to send a scan request message and, prior to sending, hears a scan response message will discard the response and proceed in the next period to send its own scan request message, ultimately wasting resources.

With active scanning, the main contention-related problem comes from the behavior of the scanning devices. Similar to the transmission of advertising messages, the transmission of scan request messages does not include any carrier sensing. Essentially, if too many scanning devices receive an active advertising message, the result is a broadcast storm of scan response messages, and most if not all of the requests are lost. Scanning devices could jitter their responses, but this would be of little benefit. First, the addition of any jitter would cause the advertising tag to increase its waiting time, resulting in increased energy consumption since the tags cannot go into low-power mode until they have completed their work for an advertising period. Second, since there is still no carrier sensing, the jitter would have to be slotted or long enough to guarantee no collisions, again resulting in longer waiting times for the tag.

Since scan response messages are broadcast to all potential requesters (as opposed to unicast to the tag that triggered the response), all scanning devices have the ability to hear and successfully receive all scan response messages. However, in a shared wireless network, it is likely that a broadcast storm of scan request messages will result in none of the requests being received correctly by the tag. If no scan request message is received successfully by the advertising tag, no scan response message is triggered.

To reduce the contention from scan request messages, scanning devices implement a backoff mechanism, which unfortunately introduces its own problems. The basic idea for backoff in BLE is that if a requesting device does not get a response, there are likely many other co-located devices and the requesting device backs off in the next period to avoid collisions. Therefore, it is possible for a scanning device to intend to request additional data, but due to collisions, have to back off. During that backoff state, a different scanning device could successfully request the data, but any scan response message will be **ignored** by those scanning devices in backoff.

Additionally, if scanning devices duty-cycle their listening periods, at the start of every listening period, the backoff algorithm is reset. The result is that these devices "forget" about any contention and respond immediately to any requests, ultimately nullifying any potential benefits of the backoff algorithm.

In the end, active scanning quickly becomes ineffective as increasingly more scanning devices are in range of the tags and so respond to the advertising message. Between request collisions and an aggressive backoff for transmitting the scan request messages when contention is detected, many scanning devices are unable to receive the second 31 bytes of data from the desired tag.

## Active Scanning Performance

To understand the impact of the number of scanning devices on successful receipt of extra data requested using BLE active scanning, we first explore the number of scan response messages successfully received by scanning devices. Recall that for a scan response message to be successfully received, a scanning device would first need to receive a tag's active advertising message, and at least one scan request message would need to have been received by the tag (Fig. 2b).

Although successful interaction with a BLE tag must be measured by the receipt of the advertising message and the scan response message, merely looking at the number of losses is not the correct metric for an IoT environment. Instead, many IoT applications operate at human scale and can tolerate some delay. In other words, a loss might not be noticed by a user, as long as the data is ultimately received within a certain window of time. Therefore, a more realistic metric for success is reception within a given timeframe. In our experiments, we use a 1 s advertising period. Based on the general expectation of a user's attention [11], we present results for a 5 s success window, meaning that a response message has been successfully delivered if at least one of the tag's scan response messages is received by a scanning device within 5 s of the initial request. To evaluate the 5 s success metric, success is calculated per scanning device for every 5 s window starting at the first second across the entire experiment. For comparison, we also present results for the absolute number of successful request-response interactions, which we call *total success*. For both total success and 5 s success, the results represent the average across all scanning devices in that experiment.

### Active Scanning: Simulation Analysis

To evaluate the impact of density on BLE active scanning, we ran a set of simulations with 1, 3, 5, 7, 9, 11, 13, and 15 scanning devices listening for 1, 3, 5, 7, 9, 20, and 50 advertising tags in active mode. Since there is no current implementation of BLE in ns-3 [12], we adapted the existing lr-WPAN module, which simulates IEEE 802.15.4 in the same frequency band as BLE by constraining the channel to a 1 MHz band, removing carrier sensing from the physical layer and implementing the BLE advertising, backoff, and handshake algorithms in the MAC layer. Since the three advertising channels (37, 38, and 39) are orthogonal, these channels were simulated as three distinct physical channels with no interference. All protocol parameters were taken from the BLE specification.

Although total success is under 85 percent for 1 and 3 scanning devices, 5 s success is almost 100 percent (Figs. 1 and 2). Essentially, allowing the scanning device to wait can compensate for many of the losses. For five scanning devices, success decreases, but stays within an acceptable range. The total success hovers around 50 percent, but the 5 s success starts to dip below 95 percent. However, as the number of scanning devices increases to 9, 5 s success drops below 70 percent.

This rapid drop in success is a reflection of the choice of three advertising channels. Although the scanning devices are not synchronized in any way, the use of three advertising channels allows for limited load balancing across the channels for the request-response exchange. With up to three scanning devices, there should be limited impact. Essentially, there is a high probability that one scanning device may be scanning alone on a given channel, so there are no collisions. With five or six scanning devices, there may still be channels with only one scanning device sending a request. However, by the time nine scanning devices are scanning, there is excessive contention, and the success rate drops dramatically. The real impact of this contention and backoff can be seen in Fig. 2, which shows the actual success for all advertising messages, not waiting for the 5 s window.

As the number of active scanning devices increases, success drops significantly. Interestingly, success is significantly worse for fewer tags. Essentially, if there is only one tag, success depends on the receipt of data from that one tag. For more tags, devices may be trying to receive data from the same devices at the same time, so success increases. However, for more than 10 scanning devices, success never goes beyond 60–70 percent.

### Active Scanning: Experimental Analysis

To validate our simulations, we ran a set of experiments with 1, 3, 5, and 9 Nexus 5 phones scanning for 1, 3, 5, 7, and 20 tags in active mode. We used a combination of off-the-shelf Estimote beacons pre-configured with advertising periods of 950 ms and NRF Smart Beacon Kits configured to match the 950 ms advertising period. Experiments were run for 10 min across each combination of phones and tags. All devices, tags, and phones were put into a Faraday cage to eliminate any interference from external wireless sources.

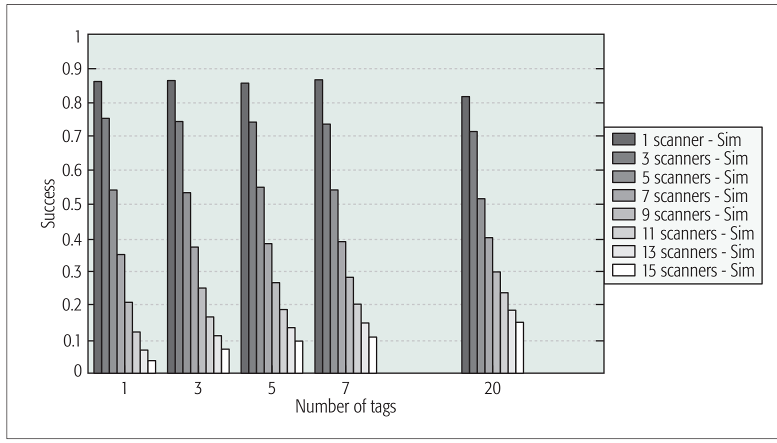


**Figure 2.** Total success for active scanning from ns3 simulations.

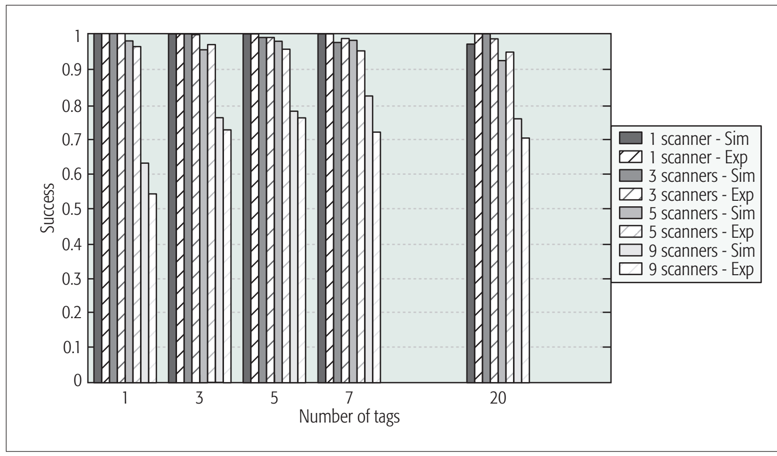


**Figure 3.** Comparison of simulation and experimental results for 5 s success for active scanning.

As expected, the experimental results follow the same trends as the results from our simulations (dark bars represent the simulation results, and the neighboring striped bars represent the comparative experimental data) (Fig. 3). When more than 5 scanning devices are added into the environment, both the total and 5 s success metrics decrease significantly beyond any acceptable success rate. Although the simulations predict slightly better performance, both sets of experiments have similar trends. This is due to the fact that our simulations do not capture all real systems' costs and delays due to buffering or interrupts. However, the similar trends in performance degradation validate our simulation results.

## Opportunistic Listening

As is clear from our evaluation in the previous section, contention and backoff result in decreased success with active scanning. This is true for even a limited number of scanning devices, especially given a large number of tags. One cause of this decrease is that during backoff, scanning devices cannot receive scan response messages that are in response to other scanning devices' scan requests. Even if a scanning device in backoff receives a scan response message, the message is discarded because that device did not send a scan request message to that tag.

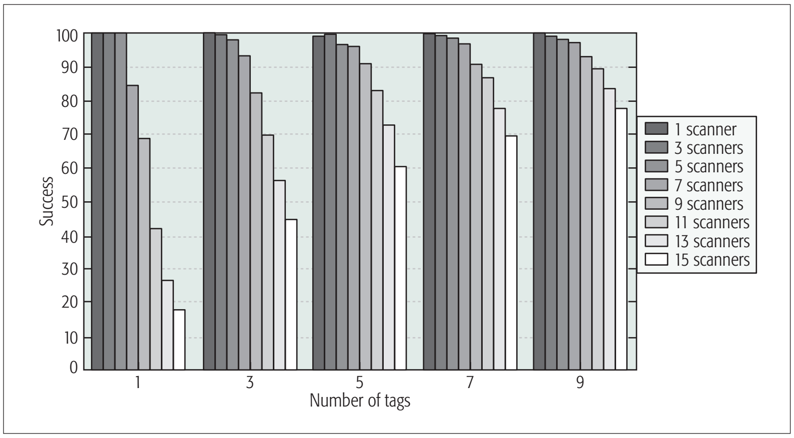


**Figure 4.** 5 s success for active scanning with Opportunistic Listening from ns3 simulations.

In response, we present **Opportunistic Listening**, which extends the capabilities of the scanning device to accept these non-requested response messages if the device is currently in backoff mode for that advertiser on that channel. Essentially, as long as one scanning device sends a scan request message, all scanning devices that have backed off can successfully receive and process the resulting scan response message. Successful opportunistic receptions do not impact the backoff algorithm, which still follows the BLE specification, and thus helps reduce contention from broadcast storms. It is important to note that Opportunistic Listening only applies to scanning devices that could have sent a scan request message if they were not in a backoff state. All other unsolicited scan response messages are ignored as per the BLE specification to reduce energy consumption from processing such messages.

To evaluate the impact of Opportunistic Listening, we modified our ns3 simulations, since support for Opportunistic Listening on real devices would require changes to the proprietary firmware. As expected, Opportunistic Listening always improves the successful reception of response messages over standard BLE when more scanning devices are present and when more tags are advertising (Fig. 4). Given the cooperative nature of Opportunistic Listening, the performance gains increase as more scanning devices are present. For example, for 9 tags and 5 scanning devices, Opportunistic Listening increases the 5 s success from 86 to 97 percent. For 9 tags and 15 scanning devices, Opportunistic Listening impressively increases the 5 s success from 48 to 78 percent.

However, as can be seen, as the number of scanning devices increases beyond 7, the success rate begins to fall off. In fact, as the number of scanning devices increases over 10 for a single tag, the success rate falls below 50 percent even with Opportunistic Listening. The ultimate problem is that too many devices are scanning for and requesting data from the tags in active mode, resulting in excessive channel contention and collisions. Therefore, for environments where a large number of scanning devices is expected, such as a retail environment, further optimization is required. As such, we next present the design and implementation of Smart LaBLEs, decentralized aggregation points for IoT objects.

## The Smart LaBLE System

Smart LaBLEs (BLE enabled product or shelf labels) are designed to alleviate contention in environments where multiple objects advertise the same information, for example, a retail environment where each product advertises its presence to shoppers for purchasing and to the store for managing product stocking. However, the techniques presented here generalize to any environment where one device can act as an advertising proxy for others. When used in a retail store, Smart LaBLEs automatically configure their associated displays to show product information for the product that is shelved nearest to them. This allows products to be moved on shelves without the need to manually update any signage. Additionally, such automatic configuration allows the Smart LaBLEs to display dynamic information such as the number of products of a certain type remaining on a shelf. As long as the distance between the Smart LaBLE and the nearest product is less than twice the distance between products of different types, each Smart LaBLE can use minimum instantaneous received signal strength indication (RSSI) values. Although channel characteristics change over time, values received within a 1 s window are sufficient to accurately determine the nearest product [10]. To further take into account non-uniform shelving, Smart LaBLEs use average RSSI values over a short window to smooth out anomalous positioning of the tags on the product shelf [10]. For initial Smart LaBLE tests, tags were placed at the bottom center of each product (Fig. 5).

The Smart LaBLEs prototype uses Nordic Semiconductor BLE Dongles and Arduino Uno devices with attached color LCD displays (Fig. 6). Each Smart LaBLE is attached to a central laptop for data collection to generate the results presented in this section. Essentially, the BLE dongle attached to each Smart LaBLE functions as a passive scanning device, listening for advertising messages from any products in its reception area.

Each product has a tag with a MAC address utilized to encode the product identification. The BLE MAC address is 6 bytes. The first 4 bytes of the address encode the product identification (in the prototype, the flavor of Gatorade™), and the last two bytes encode a unique identifier for the product of that type. The unique identifier allows the system to track which specific products have been sold (e.g., to determine if the most recently shelved products were sold first). The 31-byte payload is reserved for other product information, including cost, description, and so on.

Each Smart LaBLE listens for all tags within its reception range and records the product type, unique id, and RSSI for the advertising messages. The Smart LaBLE also records a timestamp representative of the message receive time. The Smart LaBLE system does *not* require time synchronization between the tags themselves. Only the time window around which messages are received impacts the accuracy of the system [10].

The Smart LaBLEs in the prototype system are attached to color LCDs. Once a Smart LaBLE determines the product for which it should display information (i.e., different flavors, and hence colors, of Gatorade), the Smart LaBLE changes its display color to match that of the product, and displays a product identification and the number of products of the same type that are on the shelf. Thus, each label displays information for the product nearest to the label itself. If different products are mixed within a column behind each label, the front-most product information is displayed until that product is purchased. However, the system maintains information about the total number of products of each type near each label. Given that the Smart LaBLE system has access to the total number of products near a label, other metrics could be used to decide what information to display. For example, a Smart LaBLE could display product information related to the product nearest the label in greatest numbers.

Since Smart LaBLEs were designed to alleviate contention, the choice of advertising period significantly impacts the performance and potential benefits of the system. As seen in the previous sections, if the advertising window is very short, channel contention can cause the loss of advertising messages, ultimately *increasing* the amount of time it takes to successfully receive advertising messages from each of the products on the shelf. Additionally, sending more frequent advertising messages consumes more energy and drains the battery of the tags more rapidly; batteries that are essentially impossible to change. However, overextending the advertising period results in inaccurate estimates of which product is closest to the Smart LaBLE. As our analysis shows, RSSI comparisons can only produce sufficient estimates if the advertising messages from all of the tags are received within a reasonably short window. Thus, if the advertising period is too long, the accuracy of the estimates suffers, potentially causing the Smart LaBLEs to display the wrong product information.

To solve the problem of contention, once a Smart LaBLE detects the nearest product, it signals all devices on similar products to increase their advertising message period, reducing the frequency of advertising messages received from products the Smart LaBLE has already seen. The Nordic Tags have the ability to have their parameters changed via over-the-air signals, easily facilitating this function. For our prototype, during Smart LaBLE auto-configuration, the tags on each product are set to have an advertising period of 100 ms. One possible simplification could be achieved by having a longer advertising period even during the auto-configuration stage. Essentially, given that products are frequently stocked during times when stores are not busy, it is possible that taking longer to configure the Smart LaBLEs would not be a problem. In this case, there would be no need to utilize BLE radios capable of over-the-air configuration. This could have the benefit of making the tags cheaper to manufacture as well as more energy-efficient.

## Conclusions

BLE has the potential to revolutionize IoT technology and enable the adoption of IoT in a wide range of application scenarios. However, the blind use of BLE, especially in densely deployed environments, will quickly hinder any realization of this vision.

We have presented an introduction to issues related to the primary radio technology used in IoT environments today: Bluetooth Low Energy.

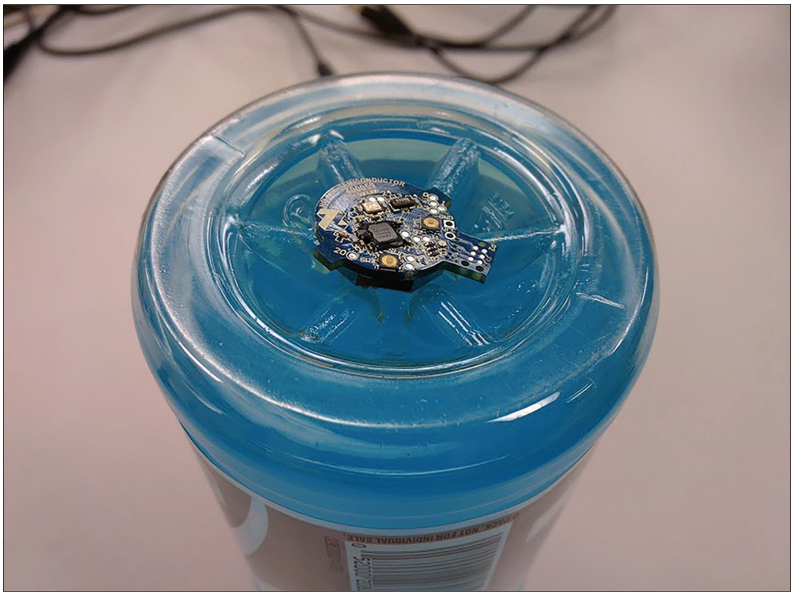

**Figure 5.** Tagged product.

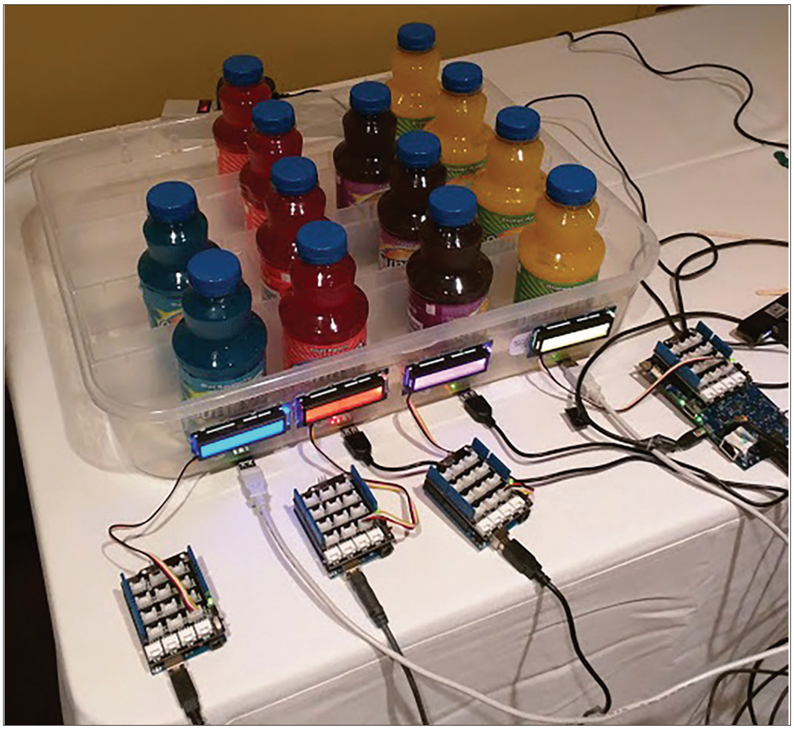

**Figure 6.** Smart LaBLE system.

Essentially, deploying BLE radios in dense environments can lead to contention problems that rapidly degrade network performance. Our evaluation of BLE in diverse environments shows that for applications requiring the transmission of more than 31 bytes of data, active scanning is not feasible if multiple devices are scanning in the same area.

In response, we have presented an extension to BLE: Opportunistic Listening. Opportunistic Listening outperforms active scanning in dense IoT environments. However, when the number of scanner devices is high, even Opportunistic Listening's performance degrades. To address the problem of dense use of passive advertising, we have presented the design of a decentralized aggregation point targeted at retail IoT spaces called Smart LaBLEs.

## References


[1] J. Niemine *et al.*, "Networking Solutions for Connecting Bluetooth Low Energy Enabled Machines to the Internet of Things," *IEEE Network*, vol. 28, no. 6, Nov. 2014.

[2] "Bluetooth 4.2 Core Specification," tech. rep., Bluetooth Sig., 2014.

[3] R. Want, B. Schilit, and D. Laskowski, "Bluetooth LE Finds Its Niche," *IEEE Pervasive Computing*, vol. 12, no. 4, 2013, pp. 12–16.

[4] R. Faragher and R. Harle, "An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications," *Proc. 27th Int'l. Tech. Meeting of the Satellite Division of the Institute of Navigation*, 2014.

[5] M. Siekkinen *et al.*, "How Low Energy Is Bluetooth Low Energy? Comparative Measurements with Zigbee/802.15.4," *2012 IEEE Wireless Commun. and Networking Conf. Wksps.,* 2012, pp. 232–37.

[6] "Bluetooth Smart Technology: Powering the Internet of Things"; http://www.bluetooth.com/pages/bluetooth-smart, aspx, accessed June 10, 2016."

[7] R. Lea and M. Blackstock, "City Hub: A Cloud-Based IoT Platform for Smart Cities," *IEEE 6th Int'l. Conf. Cloud Computing Tech. and Science*, 2014, pp. 799–804.

[8] A. Kwiecie *et al.*, "Reliability of Bluetooth Smart Technology for Indoor Localization System," *Computer Networks*, Springer, 2015, pp. 444–54.

[9] A. Al-Fuqaha *et al.*, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 4, 2015, pp. 2347–76.

[10] A. F. Harris III *et al.*, "Smart LaBLEs: Proximity, Autoconfiguration, and a Constant Supply of Gatorade," *1st IEEE/ACM Symp. Edge Computing*, 2016.

[11] A. Oulasvirta *et al.*, "Interaction in 4-Second Bursts: The Fragmented Nature of Attentional Resources in Mobile HCI," *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 2005, pp. 919–28.

[12] ns3 network simulator, http://www.isi.edu/nsnam/ns/, accessed on June 10, 2016.


## Biographies


Albert F. Harris III (aharris@illinois.edu) is a research scientist in the Department of Computer Science at the University of Illinois. His research focuses on sensing in real world environments, with a current emphasis on the integration of sensing devices into IoT ecosystems. Dr. Harris received his Ph.D. in computer science in 2006 and his J.D. in 2012. During his time in industry, he worked in a digital crimes unit.

Vansh Khanna (vkhanna2@illinois.edu) was an undergraduate researcher working with Prof. Kravets in the Department of Computer Science at the University of Illinois.

Güliz Tuncay (tuncay2@illinois.edu) is a Ph.D. student in computer science at the University of Illinois at Urbana-Champaign. Her research focuses on mobile security, mobile computing, and the Internet of Things. She received her M.Sc. in computer engineering from the University of Florida in 2013 and her B.Sc. in computer engineering from the Middle East Technical University in 2010. More information can be found at http://tuncay2.web.engr.illinois.edu.

Roy Want [F] (roywant@google.com) received his doctorate from Cambridge University, England, in 1988, and is currently a research scientist at Google. His research interests include mobile and ubiquitous computing. To date, he has authored or co-authored more than 75 publications, with 90+ issued patents in this area, and holds the grade of ACM Fellow. For more information about his academic and industrial achievements, see http://www.roywant.com/cs/.

Robin Kravets (rhk@illinois.edu) is a professor at the University of Illinois at Urbana-Champaign and the head of the Mobius group, which researches communication issues in all types of networks that are challenged by mobility, including IoT ecosystems, wireless LANs, sensor networks, vehicular networks, mobile social networks, and personal area networks. Her research focuses on solutions that enable effective power management, connectivity management, data transport, congestion management, location management, routing, and privacy. More information can be found at http://mobius.cs.illinois.edu.