The Internet of Things Has a Gateway Problem

Thomas Zachariah, Noah Klugman, Bradford Campbell,
Joshua Adkins, Neal Jackson, and Prabal Dutta
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI 48109
{tzachari,nklugman,bradjc,adkinsjd,nealjack,prabal}@umich.edu

ABSTRACT

The vision of an Internet of Things (IoT) has captured the imagination of the world and raised billions of dollars, all before we stopped to deeply consider how all these *Things* should connect to the *Internet*. The current state-of-the-art requires application-layer gateways both in software and hardware that provide application-specific connectivity to IoT devices. In much the same way that it would be difficult to imagine requiring a new web browser for each website, it is hard to imagine our current approach to IoT connectivity scaling to support the IoT vision. The IoT gateway problem exists in part because today's gateways conflate network connectivity, in-network processing, and user interface functions. We believe that disentangling these functions would improve the connectivity potential for IoT devices. To realize the broader vision, we propose an architecture that leverages the increasingly ubiquitous presence of Bluetooth Low Energy radios to connect IoT peripherals to the Internet. In much the same way that WiFi access points revolutionized laptop utility, we envision that a worldwide deployment of IoT gateways could revolutionize application-agnostic connectivity, thus breaking free from the stove-piped architectures now taking hold. In this paper, we present our proposed architecture, show example applications enabled by it, and explore research challenges in its implementation and deployment.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Design, Documentation, Management, Performance, Standardization

Keywords

Internet of Things, Gateway, Mobile Phones, Bluetooth Low Energy, Sensor Networks, Low-Powered Devices

HotMobile'15, February 12–13, 2015, Santa Fe, New Mexico, USA.
ACM 978-1-4503-3391-7/15/02
http://dx.doi.org/10.1145/2699343.2699344.

1. INTRODUCTION

Mobile computers, including laptops, tablets, and smartphones, have experienced unparalleled success due in no small part to an abundance of wireless connectivity. Widespread Wi-Fi and cellular networks provide universal and transparent access to the Internet and cloud-powered applications. This has driven the success of mobile computing. The coming wave of tiny, embedded, low-power, wireless, mobile, and wearable devices, however, does not currently enjoy the same level of ubiquitous and universal access to the Internet. Due to battery constraints and lifetime considerations, these devices tend to rely on low-power wireless communications like Bluetooth Low Energy (BLE) instead of more well-connected, but also more power intensive, Wi-Fi and 3G/4G cellular radios, despite their increasing ubiquity. To connect to the Internet, these devices require an *application layer gateway*—a system capable of translating data from the low-power link to the Internet at large. However, current implementations of these low-power links do not provide an *Internet gateway*, but rather, as Figure 1 depicts, a narrow connection to a device-specific application that must be installed on a smartphone or laptop. Opening a new webpage on a laptop does not require a new application on the Wi-Fi router, but connecting a new IoT device does require a new smartphone app, a new laptop dongle, or a new basestation device, as Figure 2 illustrates.

From smartwatches that interoperate with only a small subset of smartphones to wearable health monitors that cease communicating when their paired phone dies, it is clear that the Internet of Things (IoT) has a gateway problem. While the global network of well-connected smartphones provides a promising foundation for ubiquitous, low-power, last-inch networking, the current siloed, segmented, and application-specific approach to wireless connectivity is hampering the growth potential of this emerging device class.

Addressing this problem requires a new networking architecture for low-power wireless devices that better leverages the opportunities provided by the worldwide network of smartphones. Such an architecture would need to provide convenient and transparent access to the Internet for low-power devices while offering data integrity, security, throughput, and lifetime for the phone and device.

Our approach uses BLE, common on modern smartphones, as the primary link between low-power peripherals and capable smartphones. In contrast to the application-specific design of device-phone interactions, however, we envision an open, two-prong gateway model. First, we envision that any BLE device could leverage any smartphone as a temporary IP router and act as a normal IP end host. Second, any phone could proxy a Bluetooth profile to the cloud on behalf of a device. The former allows for a high degree of flexibility while the latter may be better suited to the power and processing constraints of the device. Both can be implemented as part of an independent app or OS service on the phone.

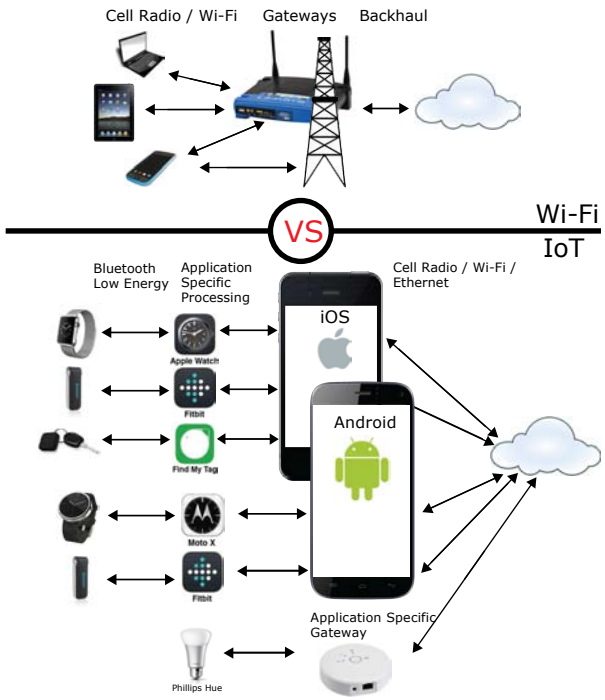

Figure 1: The IoT Gateway Problem. Currently, a separate physical router or smartphone application must be provided in order to enable gateway services for each type of IoT device deployed. This contrasts with any mobile computer's ability to connect to the Internet via a single Wi-Fi router.

Current applications cannot be entirely replaced by transparent gateways, however. The asymmetry in capabilities between smartphones and peripherals leads to some application-specific functionality, like location information or user interfaces, being handled by the phone. To support some such usage scenarios, we propose to extend the architecture to allow devices to request certain services from the paired smartphone, such as the phone's location or the current time. Services like these may be critical to the application but difficult for a cost and energy constrained peripheral device to acquire on its own. This suggests a possible new role for the smart phone—as an opportunistic context server for nearby devices.

A worldwide collection of Internet-connected smartphones provides an unprecedented opportunity to provide last-inch connectivity for the billions of IoT devices expected to emerge in the next few years, crucially, without requiring each phone to load every application-specific gateway app. A simpler (than IPv6) approach might be to provide a generic BLE gateway and a set of common services. Such a network could also provide Internet access to stationary sensors tasked with monitoring homes, offices, cities, or other areas. Instead of requiring nodes to form mesh networks to relay data back to a few Internet-connected gateways, each node could piggyback on passing smartphones to offload or receive data. Indeed we are witnessing siloed versions of such approaches from Fitbit [8] and Tile [23].

This network architecture—of shared access using untrusted, crowd-sourced gateways—raises many questions concerning usability, availability, incentives, security, privacy, and deployability. In this paper, we identify some of the key issues and begin to explore them, with the goal of raising awareness and generating discussion about both the opportunities and challenges.

2. APPLICATIONS

To motivate the need for a well-defined, cross-platform architecture for connecting low-power devices and sensors to the Internet, we describe several applications that are enabled or improved by our proposed gateway architecture.

2.1 Ambient Data Collection

Sensors installed in buildings, homes, cities, remote environments, and other locations can provide invaluable streams of data for monitoring, control, analysis, and prediction applications. Retrieving data from each device, however, is often challenging due to sensor power constraints, poor wireless connectivity, or expensive data links. One solution that has been extensively studied is to mesh-network sensors to allow data packets to hop through the network, but this often fails in areas with poor RF characteristics, and the demands of packet forwarding take a substantial toll on sensor lifetime.

In contrast, our BLE gateway architecture would leverage the smartphones that people already carry to collect data from installed sensors. As an example, consider scientists seeking to measure temperature and relative humidity in a forest by deploying sensors. Rather than requiring a cellular data plan for each sensor or the scientists to visit each node periodically, we imagine a system where hikers traveling on well-defined trails can provide connectivity for these sensors. As a hiker walks by a sensor, the sensor will attempt to use the hiker's mobile phone as a gateway. Because the sensors conform to a common architecture, a hiker would not need to download any software to connect to the sensors. The phone, which may be disconnected from a data network, could hold the data for some time before forwarding it. Hikers may be interested in being a courier for the data because of its scientific nature [2], or because the scientists will compensate them [14].

This method of data retrieval can extend to other applications as well. Sensors installed in buildings, particularly older buildings with challenging RF characteristics, could use the daily occupants of that building to relay their data. In this case, the occupants may be incentivized by obtaining controls for temperature and lighting on their smartphones in exchange for forwarding sensor data.

2.2 Cross Platform Connectivity

Some newer wearable devices are limited by the model of smartphone to which they are capable of connecting. For example, the upcoming Apple Watch will only be able to pair with a recent iOS device to obtain network connectivity. Other smartwatches, like those from Motorola and Samsung, follow a similar model even though they all use BLE communication. This closed, siloed approach is detrimental to the growth and usefulness of this class of devices.

With an open gateway architecture, any smartwatch could ask any smartphone it encounters to agree to act as a gateway. The phone could then provide a connection for any low-bandwidth Internet applications running on the device. Certain applications which are highly user specific, such as notifications on the smartphone, may still require a specific smartphone or app running on the phone.

2.3 Masking Smartphone Failures

Requiring a BLE peripheral or wearable device to link to exactly one smartphone inserts an unnecessary failure point for these devices. If the paired smartphone is not present or is discharged, the otherwise functional tethered device loses its ability to send or receive data. An open gateway model would allow devices to use any nearby smartphones to forward or receive data. In certain situations, such as when using a fitness monitor at the gym or after a smartphone's battery has depleted, it would be preferable not to lose functionality because a specific phone is unavailable, as many do today.

(a) IoT Devices (b) IoT Gateways (c) BLE Apps on a smartphone

Figure 2: Currently, each of the peripherals in (a) requires its own gateway as shown in (b) and/or an application like those shown on the smartphone in (c) in order to function. Each gateway in (b) and each application in (c) does not support more than a single type of peripheral. Each gateway in (b) connects directly to the Internet through either a computer, Wi-Fi, or wired Ethernet connection.

3. PROPOSED ARCHITECTURE

To provide Internet connectivity for resource-constrained devices, we propose a smartphone-centric approach. Smartphones can act as a useful gateway due to their near-constant Internet connection, mobility, and ubiquity, but they also dictate what wireless protocol IoT devices must use based on what is commonly available on the phones. Although Wi-Fi is ubiquitous in many parts of the world, and is presently implemented in many IoT devices, its large power requirements make it unsuitable for low-power applications. While some low-power links, like IEEE 802.15.4, provide features that would be useful in this regime, their lack of smartphone support make them unattractive. Instead, we argue that Bluetooth Low Energy (BLE) is the most promising protocol for connecting IoT devices. Its widespread deployment in smartphones and suitably low-power draw make it an attractive solution.

BLE is a link-based, point-to-point protocol between two devices, one in peripheral (slave) mode and the other in central (master) mode. In our architecture, the smartphone remains in central mode while all IoT devices behave as peripherals. Peripheral nodes transmit periodic beacons, termed advertisement packets, to notify nearby central nodes of their presence. Once a central device hears an advertisement, it can establish a connection between the two devices to transfer information. This connection process is standardized by the BLE specification. How and which information is transferred between the device and smartphone is specific to each application, however. To allow the phone to behave as a generic gateway, our architecture focuses on the specification of two general and reusable approaches to transferring data that many applications could use. An overview of the architecture is shown in Figure 3.

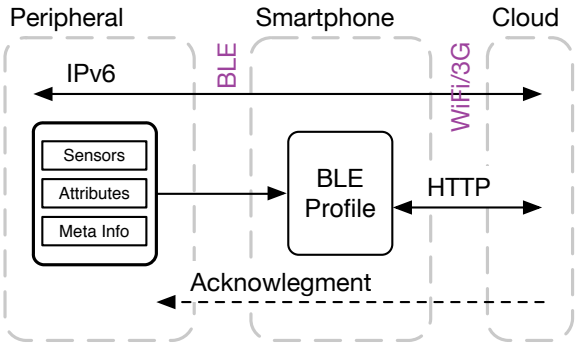

Figure 3: Proposed Architecture. Our approach consists of two data transmission mechanisms: (1) via IPv6, using the smartphone as a temporary IPv6 router and treating the peripheral as an IP-connected end host, and (2) via proxy, using the smartphone to forward the peripheral's BLE profile to the cloud.

3.1 IPv6 Routing

The first data transport mechanism between BLE peripherals and smartphones in our architecture is a raw IPv6 packet transfer over BLE. This would allow each IoT device to behave as any other IP end host and to take advantage of the flexibility of working at the network layer. The peripheral must be capable of running an IPv6 stack, which is feasible as demonstrated by the IPv6 stacks running on sensor motes [11, 20]. The phone must act as an IPv6 router between its Internet connection and the peripheral. The mechanisms for building this IP network on a BLE link are currently being formalized by the IETF and BLE SIG [5, 15, 16].

The primary challenge to using this data transport is the complexity of communicating at the IP layer. All resource-constrained peripherals should not be expected to support a full IP stack. Further, this class of sensor can benefit from offloading work to a more capable device. While the flexibility of providing an IP layer is extremely beneficial for supporting a wide variety of applications, we propose an additional data transport that offers less flexibility but is better optimized for immediate use with the BLE specification and contemporary IoT device applications.

3.2 BLE Profile Proxy

The second data transport mechanism operates by using the smartphone gateway as a proxy for the information contained in the BLE data structures on the peripheral. At a high level, the gateway relays the services, characteristics, and attributes shared with it from the BLE peripheral to a remote server. This more naturally aligns with existing BLE devices, as the data organization between the peripheral and central node in existing, application-specific BLE interactions does not fundamentally change.

3.2.1 Gateway Configuration

To support this proxy architecture, IoT peripherals must extend the data they send to the phone with meta information that dictates how the phone should proxy the BLE profile data. This configuration meta information will be contained in the peripheral's broadcasted advertisements, to which the gateways will have access without requiring a connection with the peripheral.

Data Flow. As part of the meta information advertised, the peripheral must indicate data flow parameters like the content, type, destination and rate of the data to be forwarded. We imagine that, once received on the gateway, the data would be bundled and sent as an HTTP POST request to the specified destination.

Reliability. When connecting to an unpredictable gateway for an unknown amount of time, particularly in a mobile environment, the peripheral faces a challenge to know if its data were successfully transmitted to the intended destination. To address this, a peripheral can specify the level of reliability it would like the phone to try to achieve. This reliability setting is analogous to the transport layer selection in other networking applications. The highest two levels of reliability allow for peripherals to request that a gateway device provide immediate connectivity (level 1) or eventual connectivity (level 2). Both of these levels provide the peripheral with some form of acknowledgment from the end recipient. This supports near real-time and retryable applications, and is analogous to the delivery guarantees provided by TCP. The second two levels of reliability require that the gateway either makes a best effort to forward at a later time (level 3) or a best effort to forward immediately regardless of Internet connection state (level 4). These last two specify that the peripheral is not requesting an acknowledgment. This supports near real-time and delay-tolerant applications, and is analogous to UDP.

Gateway Services. Peripherals may wish to ask the gateway to append information on their behalf to the outgoing data. For example, information about the location of the peripheral or the current global time may be difficult for an IoT device to obtain, but straightforward for a smartphone. Therefore, the gateway smartphone should provide a suite of services that can append information to the data from the peripheral, similar to the IPv6 options framework. Implemented generically, the services subsystem could be extended to other data augmentation applications and possibly offloaded to "cloudlet-style" computational services.

User Incentivization. A major hurdle in adopting this architecture is incentivizing smartphone owners to allow their devices to behave as gateways. If such schemes are created, peripherals must be able to communicate to a potential gateway that it supports a particular incentive system. The gateway would then be able to decline forwarding for that peripheral or later retrieve its compensation. Unilateral system support, like Apple Pay, could also help.

Data. Peripherals may wish to use the remaining bits of the advertisement packet to broadcast small amounts of data. This data could then be forwarded by the gateway using the meta information without forming a BLE connection with the peripheral.

3.3 Gateway Administration

Gateway owners should be able to configure how and to what extent their smartphone is utilized as a gateway. The gateway configuration settings allow owners to cap the data rate and choose which data augmentation services and incentive programs to support. Additionally, a gateway will maintain a whitelist and blacklist to enable fine-grained access control.

3.4 Application-Specific Apps

Our proposed architecture is not intended to replace all peripheral-specific apps on a smartphone. Some apps utilize or display data that is collected by the peripheral. These apps should be designed primarily to display information from the backend cloud service, and should, instead of implementing a custom siloed gateway for the peripheral, allow all forwarding data requirements to be handled by the gateway service on any nearby smartphone.

3.5 Universal Gateway

This architecture is designed to ensure that a peripheral device is not restricted to using one specific smartphone in order to connect to the Internet. That is, any peripheral should have the opportunity to connect through any smartphone, creating a universal gateway out of every smartphone.

4. RESEARCH QUESTIONS

Our proposed architecture raises many technical challenges, including determining the terms of the relationship between an arbitrary peripheral and gateway, where in the software stack the architecture should be implemented, how the gateway should forward data on behalf of a peripheral, and how to incentivize users to enable their devices as gateways. In this section we more deeply discuss many of the research questions that must be explored to realize our open gateway proposal.

4.1 Data Flow Mechanisms and Policies

Our proposed architecture requires smartphones to forward data on behalf of connected peripherals, but the specifics of this are not solidified. How should the smartphone accomplish this? Which data should it send? Should there be a data size limit? More interestingly, how does the phone relay data back to the device? If the peripheral disconnects or moves away, how should the gateway respond? Is it responsible for storing the data and attempting to forward the data later? Should the gateways notify the remote server that its response was never heard?

Forwarding data from the peripheral to the cloud is not the only avenue for data movement. One particularly important example is remote device updates. If a bug or vulnerability is discovered in a particular device, patching a device without replacing it is critical. How does the update patch get to the peripheral? Must it query for updates? If the peripheral is located such that nearby gateways do not have an Internet connection, can mobile gateways store a set of updates and apply them at a later time?

Different peripherals may have different data integrity needs. At what point can a peripheral be sure its data reached the end server? How long should a peripheral store data locally? How does a server acknowledge receipt of a range of data? On the gateway side, if the gateway does not immediately have an Internet connection, how long should it hold the data? If different gateways offload data from the same sensor, how should reordering or deduplication be handled?

Certain peripheral-gateway interactions may be fleeting. The peripheral may not know for how long it will be able to transfer data to or from the gateway. Should the peripheral be optimistic and retry later if the gateway moved away too soon? Or should the peripheral try to evaluate the bandwidth of the link by progressively increasing the amount of data it transfers?

If a gateway provides local processing, it could greatly reduce the latency and increase the reliability of returning processed data to the peripheral. This model has been demonstrated by cloudlets, which provide local computation before the cloud to mobile phones seeking to offload to the cloud [19]. If a gateway provides local processing, how should its computing services be structured and made available? Does processing on the gateway place too high of a cost on the gateway owner?

Smartphones incur a cost when acting as a gateway for a peripheral in both battery life and data communication costs. How should gateways choose when to forward data? How does the architecture ensure that peripherals are not communication starved if the gateways are selective?

4.2 Implementation Considerations

Details of how to implement our proposed architecture remain to be explored. A major question is where on smartphones the gateway logic should reside. Must the gateway functionality be in the operating system layer? Is a user-installable app sufficient? What are the limitations of the BLE APIs available in the commercial smartphone platforms? Would changes or enhancements to the available BLE APIs facilitate gateway development?

4.3 Privacy and Security

Leveraging a wide body of smartphones as gateways raises numerous privacy issues. It is conceivable that a peripheral owner can localize a gateway owner by receiving data through that gateway from peripherals at known locations. Conversely, a peripheral moving through a collection of colluding gateways could be localized. Both may be examples of privacy violations. What techniques for anonymization of forwarded data could be used to mitigate these anti-privacy effects?

Smartphones could log all traffic they forward. What encryption utilities are best suited for constrained peripherals to prevent the smartphone from snooping? Should peripherals have long-standing, symmetric key trust relationships with the cloud to facilitate encrypted communications?

4.4 User Incentives

Incentivizing users to allow their smartphones to act as gateways is critical to realizing our architecture. We imagine a scheme in which the owner of the data forwarded through a gateway rewards the owner of that gateway for the connectivity they provided. How do the data owner and gateway device agree on the transaction cost? How do data owners protect against users abusing the system?

Incentive systems for participation in crowd-sourced sensing projects have often been proposed and have seldom been implemented. Is there a solution for the gateway application? What types of incentives are most compelling? What are the risks of potential abuse for such incentivization schemes and how can they be addressed? What if an application cannot afford to incentivize users? Would incentive tiers be an effective solution?

4.5 Trustworthiness of Gateways

Allowing anyone to operate a gateway opens the possibility for gateways to be untrustworthy or actively malicious. How can peripherals detect and blacklist bad gateways, or vice versa? How can smartphones efficiently blacklist and whitelist? Is it possible for gateways to build and demonstrate reputation-based trust? How can the negative effects of bad gateways be mitigated? How can peripherals ensure their data are successfully relayed in the face of malicious gateways?

4.6 Permanent Gateways

Smartphones carried by people are not the only potential gateways for low-power peripherals. What role do dedicated gateway hardware devices play? Can they be transparently added to the network when smartphones are not sufficient? Can laptops or other computers be compatible gateways? Should dedicated gateways be identified as they may be able to transfer more data? Should there be classes of gateways in general? What is the potential of adding BLE radios to Wi-Fi routers in the future?

4.7 Industry Adoption

To realize a ubiquitous universal gateway, IoT device manufactures, app developers, and service providers need to agree upon a standardized architecture.

Each player has an incentive to support a universal gateway. IoT device manufactures would experience an increased ability to connect their devices to the Internet. A similar increase in connectivity provided by widespread Wi-Fi networks opened up large markets in laptop and tablet computing. Service providers would experience an increase in data usage and app developers would no longer have the responsibility of implementing the gateway portion of their IoT application.

Conversely, each player may be reluctant to support a universal gateway because of the financial benefits they currently receive from customers buying into a proprietary ecosystem. What entity should coordinate support for a universal gateway? What services should a universal IoT gateway offer to be attractive to industry?

5. RELATED WORK

IPv6 in BLE. The Core Bluetooth Specifications 4.1 and 4.2 contain descriptions of a new scheme that allows peripherals access to established dedicated channels in the L2CAP layer for communication over IPv6 [5, 6]. Additionally, the Internet Engineering Task Force (IETF) has prepared two draft documents that further demonstrate the push toward enabling IPv6 over BLE. The first of these documents describes techniques in 6LoWPAN that allow for IPv6 transport over BLE [15]. The second document describes a new Bluetooth Internet Protocol Support Profile which will be responsible for configuring the BLE connection and handling the data flow for IPv6 transactions [16]. The work of the IETF helps define the next steps toward connecting BLE devices to the Internet. This, along with promising implementations of some of these ideas [24], demonstrates progress toward solving the problem of how BLE-connected IoT devices can access the Internet. Still, simply enabling IPv6 connectivity alone falls short of the full set of possibilities of a true IoT gateway. But, these activities further validate the need for IPv6 routing in the emerging Internet of Things.

Delay Tolerant Networking. The use of mobile phones as gateways leads to challenges stemming from the lack of continuous network connectivity. Mobile wireless ad hoc networks allow for the continuation of previously disrupted communication when the mobile node is in range of the network. This type of routing has been demonstrated in many projects involving delay tolerant networking [9, 18]. We consider work that describes the tradeoffs of delay tolerant networks in energy, latency, and storage while moving forward in the design of our architecture [21].

Data Muling. Many projects demonstrate that data mules, mobile surrogates such as smartphone gateways that can transport data between two hosts that would otherwise be unable to communicate with one another, can provide connectivity for sensor networks [7, 13]. Additionally, data muling over Bluetooth on human-carried mobile phones has been shown to provide a reliable network, even for remote sensor deployments [17].

Existing Services for IoT Devices. Over the past couple of years, a number of companies have announced services promoting the connection of smart products. Thread, for instance, is described as a home-based mesh network capable of connecting hundreds of products within a house and enabling online control via a border router connection to Wi-Fi [22]. Helium is a platform developed for metropolitan-sized networks of low-powered connected devices using a modified 802.15.4 protocol and IPv6 addressing, but optimized for very low data transfer [10]. The AllSeen Alliance is a group of consumer brands promoting mainstream adoption of an interoperable and universal software framework for the Internet of Things based on the AllJoyn open source project [1]. Apple's Homekit is a framework available to approved application developers in iOS 8 that enables communication and control of connected products in the house which meet Apple's technical specification [3]. Apple has also introduced iBeacon, a low-powered and low-cost BLE-based proximity solution that specifies the public transmission of unique application-specific identifying information in a BLE peripheral's advertisements for which iPhones specifically scan as a background operation [4]. Detection of an application's known peripheral identifier on an iPhone can prompt various actions, enabling location-based advertisements and rough indoor navigation.

Static Gateway Solutions. One common solution for providing connectivity to IoT devices is to bundle each type of device with a custom hardware gateway. This approach leads to both a longer time to market for the manufacturer and an explosion of hardware gateways for users with many IoT devices. Intel, McAfee, and Wind River have collaborated to provide a service to help IoT manufactures design dedicated IoT hardware gateways [12]. Although this project provides tools for building gateways that can support any specific IoT application, it remains siloed and relies on the massive deployment of a new hardware ecosystem.

6. CONCLUSION

We propose a general-purpose IoT gateway on modern smartphones as a software service that provides universal and ubiquitous Internet access to BLE-connected IoT devices. This provides a scalable alternative to the narrow, application-specific gateway structure hampering the development and growth of IoT networks today. Our proposed approach utilizes the smartphone as both an IPv6 router for less resource-constrained endpoints (allowing IoT devices to communicate as IP-connected hosts) and as a BLE proxy (relaying profile data from the IoT device to the cloud).

As we begin to explore this architecture, we hope to determine the feasibility and scalability of our proposed approach—standard gateways and peripheral services—and of our methods for ensuring reliability, security, and incentives. If successfully implemented on the global smartphone infrastructure, our architecture could expedite the growth of a global, highly-connected, robust Internet of Things in a cost-effective and convenient manner. However, even if our vision of *any* IoT device connecting to *any* smartphone proves too radical a departure from the status quo, the basic ideas could still be deployed in more constrained administrative domains, like a home, office, or university campus. This approach would provide most of the benefits we seek while relaxing the more challenging aspects of security, privacy, and trust in the network, opening the door to a post-MANET for the post-mobile era.

7. ACKNOWLEDGMENTS

We wish to thank our shepherd, Matt Welsh, and the anonymous reviewers for their detailed comments and feedback. This work was supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP), a Semiconductor Research Corporation program sponsored by MARCO and DARPA. This material is based upon work partially supported by the National Science Foundation under grants CNS-1111541, CNS-1239031 and CNS-1350967, and generous gifts from Intel, Qualcomm, and Texas Instruments.

8. REFERENCES

[1] AllSeen Alliance. Open source IoT to advance the Internet of Everything. https://allseenalliance.org.

[2] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. SETI@Home: An experiment in public-resource computing. *Commun. ACM*, 45(11):56–61, Nov. 2002.

[3] Apple Inc. HomeKit. http://developer.apple.com/homekit.

[4] Apple Inc. iBeacon for developers. http://developer.apple.com/ibeacon.

[5] Bluetooth Special Interest Group. Bluetooth core specification 4.1. https://www.bluetooth.org/en-us/specification/adopted-specifications, 2013.

[6] Bluetooth Special Interest Group. Bluetooth core specification 4.2. https://www.bluetooth.org/en-us/specification/adopted-specifications, 2014.

[7] A. Chakrabarti, A. Sabharwal, and B. Aazhang. Using predictable observer mobility for power efficient design of sensor networks. pages 129–145. ISPN, 2003.

[8] Fitbit Inc. Fitbit. http://fitbit.com.

[9] M. Grossglauser and M. Vetterli. Locating nodes with EASE: Last encounter routing in ad hoc networks through mobility diffusion. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1954–1964 vol.3, March 2003.

[10] Helium Systems Inc. Helium. http://helium.co.

[11] J. W. Hui and D. E. Culler. IP is dead, long live IP for wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 15–28, 2008.

[12] Intel Corporation. Developing solutions for the Internet of Things White Paper. http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/developing-solutions-for-iot.pdf, 2014.

[13] S. Jain, R. C. Shah, W. Brunette, G. Borrello, and S. Roy. Exploiting mobility of energy efficient data collection in wireless sensor networks. pages 327–339. Mobile Networks and Applications, 2006.

[14] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

[15] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez. Transmission of IPv6 packets over Bluetooth low energy "draft-ietf-6lo-btle-03", 2014.

[16] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez. Transmission of IPv6 packets over Bluetooth low energy "draft-ietf-6lo-btle-04", 2014.

[17] U. Park and J. Heidemann. Data muling with mobile phones for sensornets. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 162–175, New York, NY, USA, 2011. ACM.

[18] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, Feb 1999.

[19] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based cloudlets in mobile computing. *Pervasive Computing, IEEE*, 8(4):14–23, Oct 2009.

[20] T. Savolainen and M. Xi. IPv6 over Bluetooth low-energy prototype. In *Aalto University Workshop on Wireless Sensor Systems, Aalto, Finland*, 2012.

[21] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, WDTN '05, pages 260–267, New York, NY, USA, 2005. ACM.

[22] Thread Group. Thread. http://threadgroup.org.

[23] Tile Inc. Tile. http://thetileapp.com.

[24] H. Wang, M. Xi, J. Liu, and C. Chen. Transmitting IPv6 packets over Bluetooth low energy based on BlueZ. In *Advanced Communication Technology (ICACT), 2013 15th International Conference on*, pages 72–77, Jan 2013.