Opportunistic Interaction in the Challenged Internet of Things

Hanno Wirtz, Jan Rüth, Martin Serror, Jó Ágila Bitsch Link, Klaus Wehrle
Chair of Communication and Distributed Systems (COMSYS)
RWTH Aachen University
{wirtz, rueth, serror, bitsch, wehrle}@comsys.rwth-aachen.de

ABSTRACT

Under intermittent Internet connectivity, enabling interaction between smart objects and mobile users in the Internet of Things (IoT) becomes a challenge. We thus discuss the notion of a "Challenged IoT" and propose Direct Interaction with Smart Challenged Objects (DISCO), enabling objects to define their interaction patterns and interface. Building on the distinct features of Bluetooth Low Energy (BLE), objects then convey their interface directly to mobile users. DISCO mitigates the need for Internet connectivity and pre-installed interfaces, i.e., smartphone apps, of existing approaches and proposes autonomous and local interaction with smart objects as a challenged network scenario. We implement DISCO for Android and iOS smartphones as well as Linux and Arduino objects and illustrate the design space of interaction patterns with Augmented Reality (AR) interaction based on visual object recognition within the tangible interaction sphere of the user. Our system evaluation shows the immediate real-life feasibility and applicability of DISCO on current hardware.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless communication*

Keywords

Opportunistic Interaction; Challenged Internet of Things

1. INTRODUCTION

While *smart objects* begin to permeate our environment, we still lack compelling ways for mobile users to interact with them. Contrary to their name, Internet of Things (IoT) devices are not necessarily continuously *connected* to the Internet, but may only have communication *capabilities*, such as Bluetooth Low Energy (BLE) or IEEE 802.11. The same of course holds true for mobile users. Even if a user device can connect to a smart object, the sheer number and diversity of objects prohibits the user to have the appropriate *interaction interface*, i.e., smartphone app, for all encounter-able objects.

In this paper, we aim to initiate the discussion on the resulting notion of a *Challenged IoT* and propose viable mechanisms for opportunistic local interaction with smart objects. We build on the observation that smart object functionality is largely relevant in the spatial vicinity of an object, e.g., in smart buildings or personal appliances, and hence does not necessarily depend on Internet connectivity. Aside from remotely controlled systems, interaction between mobile users and smart objects occurs *opportunistically* and *locally*, i.e., only when a user wants to access the object's functionality and within her tangible interaction range.

For example, a user can interact with the smart lighting [13] object at home over the Internet or a local network using the pre-installed interface, i.e., smartphone app. However, when she encounters another Internet-connected lighting object [9], she has no means to discover and interact with it. Even if the object is accessible via a local network, she still misses the required interaction interface. Aside from personal objects, this *lack of connectivity and interaction interfaces* is inherent to the envisioned number and diversity of objects that expose functionality, e.g., to control window blinds and lights in smart buildings, within a number of worthwhile scenarios, e.g., underground or in remote or unknown areas.

We propose Direct Interaction with Smart Challenged Objects (DISCO), establishing autonomous and opportunistic interaction with smart objects. We alleviate the need for Internet connectivity through local wireless discovery and interaction over BLE and IEEE 802.11. Accounting for the unfeasibility of pre-installing all possible future interfaces, i.e., apps, we enable smart objects to define and directly communicate the interface required to interact with their functionality. In this, our design caters to the communication standards as well as input and interaction capabilities of smartphones as the prevalent mobile input and interaction device. With our approach, users can receive and visualize the respective interfaces and interact with *all* encountered objects within a *single* generic app, independent of Internet connectivity. Providing a framework for spontaneous, autonomous interaction, DISCO departs from current designs that require centralized discovery and Internet communication [5, 11] and pre-installed, single-purpose interfaces [7, 9, 13].

Following our approach in the lighting example, the user now discovers an unknown object and obtains its semantics ("lighting") and the graphical interaction interface directly from it. The set of supported interfaces is as diverse as the scope of smart objects and their functionality. While a single "on/off" button suffices to control simple lighting objects, advanced functionality, e.g., adjustable color and intensity, call

for more sophisticated interfaces. To illustrate the underlying design space and support for individual interfaces, DISCO allows interfaces ranging from simple smartphone Graphical User Interfaces (GUIs) to Augmented Reality (AR) interfaces. Finally, DISCO leverages opportunistic networking to disseminate interface information between mobile users based on object semantics, e.g., their location and functionality. A mobile user thus may obtain the interface description of an object prior to encountering it, enabling instant subsequent interaction, e.g., with a newly encountered lighting object.

DISCO enables direct interaction with smart objects through local communication and object-driven interface provision (Section 2). Its communication performance, energy efficiency, and usability (Section 3) make our design immediately applicable, mitigating the requirements of Internet connectivity and pre-installed interfaces (Section 4).

2. DESIGN

Figure 1 provides an overview of our design. Applied to our previous example, Figure 1(a) illustrates the local discovery of an object via IEEE 802.11 or BLE (Section 2.1), establishment of a communication channel, and subsequent obtaining of its lighting system semantics (Section 2.2). Discovery of the interface(s) supported by this object then allows the user to acquire the definition of the smartphone GUI and commands that make up interaction with, e.g., the color and intensity of the lighting (Section 2.3). Subsequently, as shown in Figure 1(b), other users may obtain both object semantics and interface descriptions within opportunistic communication exchanges, mitigating the initial overhead of selecting and obtaining this information in new scenarios, e.g., when newly encountering the lighting system (Section 2.4). Our demo video[1] shows the application and example use cases of our iOS implementation.

2.1 802.11 vs. BLE for Local Communication

Recent work [8,10] proposes BLE as a communication pillar for the IoT. With regard to our design, reasons for choosing BLE over IEEE 802.11 include the following. First and most important, the BLE *network topology* and functional role assignment matches our envisioned scenario. Specifically, peripheral devices, i.e., objects, announce themselves, and central devices, i.e., smartphones, are able to discover and connect to multiple peripherals simultaneously. This affords a means to realize opportunistic and ubiquitous interaction, as a user may discover, communicate, and interact with multiple objects when she is in range. In contrast, realizing this topology within the station (STA) and access point (AP) role distinction of IEEE 802.11 is infeasible. If objects operate as AP, i.e., announce themselves to facilitate discovery, users can only connect to a single object, limiting the interaction scope and imposing iterative interaction. Conversely, objects that operate as STAs need to connect to mobile devices proactively and cannot account for user consent or interests.

Second, associating to an IEEE 802.11 network, at either the object or the mobile device, induces a substantial *time overhead* prior to communication. In a ubiquitous and mobile interaction scenario, this overhead impedes instant discovery and interaction and as a result degrades the perceived usability. In contrast, BLE promises significantly lower time overheads. We quantify this overhead in Section 3.2.
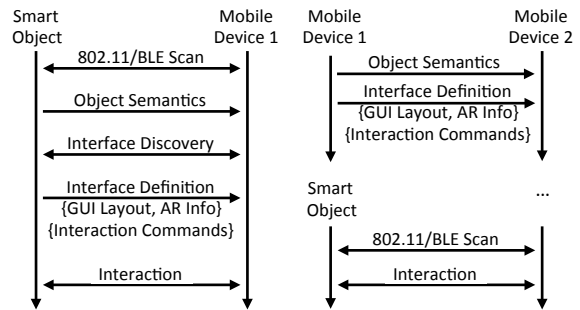
(a) Direct interaction.   (b) Opportunistic exchange.

Figure 1: DISCO enables mobile users to discover objects and their functionality and semantics locally, e.g., via IEEE 802.11 or BLE. Objects define and directly provide their interface for interaction (left). Users exchange interface information opportunistically based on object semantics, e.g., the location.

Third, BLE mandates the implementation of Generic Attribute Profiles (GATTs) and their transport over the Attribute Protocol (ATT), facilitating standardized *discovery of object semantics* and functionalities [2]. 802.11 does not specify such a mechanism but requires an additional discovery protocol to be executed after the network association.

Fourth, BLE is designed for continuous *energy efficient operation*. Note that, in this paper, we assume objects to be connected to a power supply, e.g., in smart environments [9, 13]. However, mobile devices expend battery resources during discovery and communication. We quantify the energy gains of BLE in Section 3.3.

Last, the comparably small communication range of BLE defines a *tangible interaction sphere* around the user. While a soft and subjective aspect, we argue that interaction within $\leq 15\,\mathrm{m}$, instead of up to $100\,\mathrm{m}$ in IEEE 802.11, increases the usability, immediateness, and palpability.

2.2 Communication

In DISCO, smart objects serve as peripherals and announce a descriptive, human-readable identifier along with a predefined service identifier, that indicates DISCO functionality, in their BLE advertisement records. Mobile devices operate as BLE centrals and discover objects by scanning for this service identifier. Subsequently, a mobile device connects to each discovered object. Note that the Bluetooth standard [2] does not define a maximum number of peripherals a central can connect to while real-world experiences[2] report a limit of around 20 peripherals. In human interaction with objects, this suffices for a viable interaction scope, e.g., when controlling the lights in a room.

When connecting to an object, the mobile device discovers different types of information about the object via BLE characteristics. First, a specified characteristic offers descriptive object semantics, such as the location and vendor type. We use this information to present a list of discovered objects to the user or when exchanging object information opportunistically (Section 2.4). Second, a set of additional characteristics indicates the interaction patterns supported by an object. We discuss possible interaction patterns in the next section.
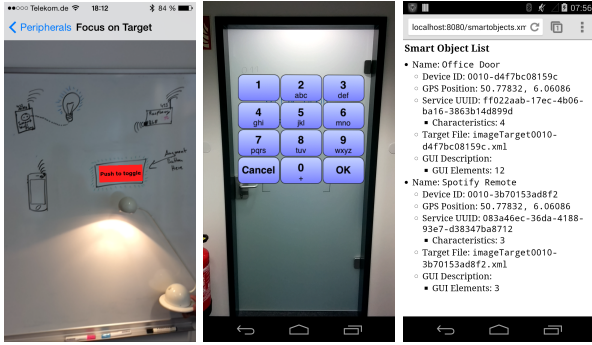
(a) Light source. (b) Pin pad GUI. (c) Object list.

Figure 2: Light source control (left), unlocking a door via a pin pad GUI (middle), object semantics for selection in opportunistic dissemination (right).

To obtain the respective interaction interface, a mobile device subscribes to the respective characteristics, similar to downloading an app tailored to the semantics and functionality of each object. Doing this, we enable objects to define and provide their interface and allow users to interact with all spontaneously discovered objects, mitigating the need for pre-installed apps and interfaces.

2.3 Interaction

Depending on the object functionality, different interaction interfaces are appropriate. For example, a light switch only incorporates "on/off" functionality, while more sophisticated systems [9, 13] allow setting the color and intensity of lighting. As such, an interface to interact with a light switch only requires a single GUI switch, while comprehensive interaction with a lighting system would require both a color and an intensity selector. The diversity of object functionalities coupled with the set of input sensors supported by smartphones, e.g., touch, sound, and movement, thereby hint at the heterogeneity and range of possible interfaces.

In its simplest form, the object defines a set of GUI elements in a list view, e.g., switches, input fields, or a "send" button. Upon user input via one of the elements, the associated command is sent to the respective BLE characteristic, triggering the respective object functionality.

In this paper, we also explore more sophisticated interaction methods that follow recent results from Human Computer Interaction (HCI) [7]. Specifically, we enable the visual recognition of objects in the camera view via Computer Vision (CV) image descriptors and augment the object with an AR GUI that visualizes and intuitively makes the interactive object functionality accessible. In [7], the authors focus on the HCI aspect and rely on an intermediate server, an existing IEEE 802.11 network, and pre-defined interfaces, i.e., GUIs. We propose to extent this interaction pattern to ubiquitous applicability, without any pre-defined infrastructure or interfaces. Thereby, we add a tangible notion to the interaction with smart objects and support visual identification of the actual object the user is interacting with.

In our implementation, we incorporate the *Qualcomm Vuforia* [14] framework for recognition and GUI augmentation and extend it by the ability to spontaneously receive GUI interface descriptions via BLE. In this, we support recognition using both visual markers and image descriptors. Hence, objects that support AR-based interaction define BLE char-

acteristics that provide visual marker, image descriptor, and GUI information to subscribed devices, respectively. Specifically, the image descriptor characteristics convey expressive visual characteristics of the object for recognition and the GUI characteristic describe the GUI elements as well as their placement relative to the visual center of the object. GUI input events on the smartphone display are then written via defined commands to the respective BLE characteristic for interpretation by the object. Figures 2(a) and 2(b) show use cases of controlling a light source via an AR button and unlocking a smart door using an AR pin pad.

2.4 Opportunistic Exchange of Object and Interface Information

We envision the dissemination of object and interface information to motivate an opportunistic networking use case, i.e., to exchange said information, such as GUI and visual descriptors, between mobile users for two reasons. First, while BLE fits well our communication scenario, it supports only a relatively small bandwidth [2, 4]. In contrast, opportunistic networking can exploit higher bandwidth channels, e.g., IEEE 802.11 or Bluetooth EDR. Second, pre-fetching of interface information lowers the time between a user encountering an object and the user interacting with it.

DISCO hence encapsulates received object and interface information in JSON objects and exposes them for selection based on object semantics, e.g., the object location, name, and functionality. Figure 2(c) shows a screenshot of a generic list presentation obtainable by users within opportunistic contacts. While not addressed in this paper, numerous approaches exist that facilitate the required transport and addressing of information [6] as well as establishing opportunistic networking [18].

3. EVALUATION

We evaluate DISCO with regard to a) the communication performance and robustness of transmitting object and interface information using BLE, b) the spontaneity and usability of interaction as afforded by BLE in comparison to 802.11, c) the energy consumption in comparison to local 802.11 networking and traditional Internet-based interaction over carrier networks, and d) the usability of our proposed AR GUI interface on current smartphones. Our results thereby highlight the real-world applicability of our design. To the best of our knowledge, we are the first to a) provide a comprehensive BLE evaluation on commodity devices, complementing the evaluation of isolated radios and fixed parameters [4], and b) propose BLE for challenged networking.

We implemented the mobile device functionality of DISCO for both Nexus 5 Android and iPhone 5S iOS mobile devices and represent smart objects with Raspberry Pi and Arduino Yún platforms. The smartphones natively support BLE, we extend the Pi and Yún platforms with Inateck Bluetooth 4.0 USB adapters and modify both the kernel-space `bluez`[3] stack as well as the user-space `bleno`[4] stack for peripheral functionality. On both the Android and iOS devices, we implement an app that receives interface information from objects in a defined grammar and provides the indicated interface to the user.
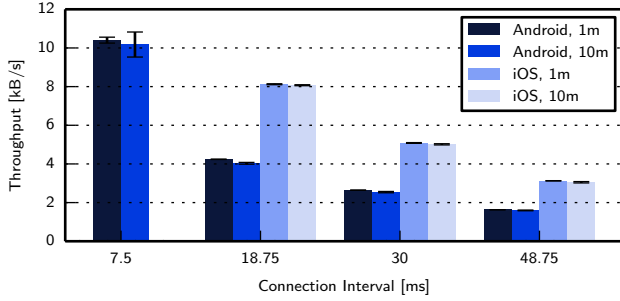
---

[3] http://www.bluez.org/
[4] https://github.com/COMSYS/bleno

Figure 3: BLE throughput to Android and iOS devices over distance and connection intervals.

3.1 Communication Performance

In order to assess the feasibility of transmitting object and interface data, we measure the possible throughput over the distance between an object and Android and iOS mobile devices. In this, we are interested in whether BLE affords bulk data transmissions prior to low-volume transmissions of interaction commands. Especially, we evaluate a) the isolated single-direction throughput, b) the performance of concurrent transmissions from multiple objects, and c) the robustness of competing transmissions. In all evaluation scenarios, we transmit 100 kB from the object to the mobile device and repeat each measurement 50 times. For comparison, CV descriptors currently amount to 50 kB–100 kB, while interaction commands and GUI elements require a few bytes.

3.1.1 Single Transmission Throughput

The specification [2] as well as an isolated evaluation [4] report a maximum theoretical throughput between 29.6 kB/s and 33.8 kB/s and an actual achieved throughput of 7.31 kB/s. In this, the main parameters influencing the possible throughput are the *connection interval*, i.e., the minimum interval between communication steps, and the *Maximum Transmission Unit (MTU)*, i.e., the maximum payload in each BLE frame. In [4], the parameters were set to 7.5 ms and 20 bytes.

In our evaluation, we strive to increase the throughput from the object to the mobile device and assume that the object is configurable while no control over the mobile devices is possible, reflecting a ubiquitous uncoordinated scenario. In this, both the Android and iOS stack do not allow configuration of BLE parameters, preventing a mobile-side configuration of transmissions. However, we observe that the iOS device negotiates an MTU configuration of 132 Bytes with the object, while the Android device adheres to the default 20 Byte MTU. We further observe that both the Android and iOS allow objects, via our modified `bluez` stack, to define the connection interval, providing a means to increase the possible throughput. Figure 3 hence shows the BLE throughput between an object and iOS as well as Android devices over their distance as well as the configured connection interval.

Decreasing the connection interval from the standard configuration of 48.75 ms (Android) and 30 ms (iOS) allows us to increase the throughput to up to 10.5 kB/s (Android) and 8 kB/s (iOS). The iOS device rejected connection intervals $\leq$ 18.75 ms. From these results, we derive the feasibility of transmitting object semantics and interfaces within a few seconds, depending on their actual size. We observe no significant impact of the measured distances on the throughput.
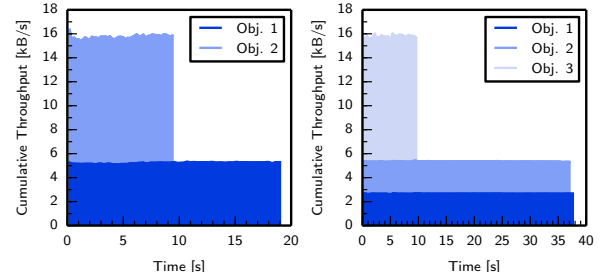
(a) Two sending objects. (b) Three sending objects.

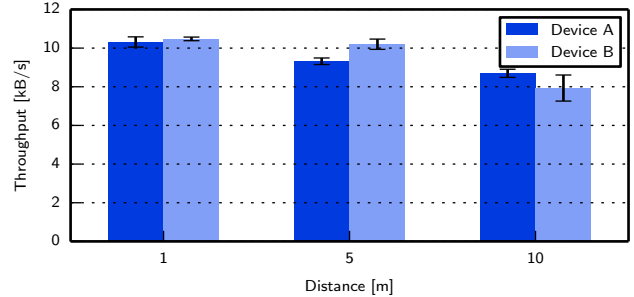Figure 4: Cumulative concurrent throughput.

Figure 5: Throughput of competing transmissions.

3.1.2 Concurrent Transmission Throughput

We envision users obtaining data from multiple objects simultaneously and thus evaluate the performance of concurrent transmissions from multiple objects to a single Android device. Hence, Figures 4(a) and 4(b) show the contribution of each sending object to the overall *cumulative throughput* as received at the mobile device over time. We observe that the cumulative throughput significantly exceeds the throughput of single transmissions, achieving up to 15.46 kB/s when receiving from two objects and up to 15.38 kB/s from three objects. Communication with multiple objects thus occurs simultaneously instead of sequentially. In this, one of the objects achieves the maximum throughput measured in the previous evaluation (10.31 kB/s in Figure 4(a); 10.16 kB/s in Figure 4(b)), while all other transmissions share the remaining capacity (5.14 kB/s in Figure 4(a); 2.62 kB/s and 2.69 kB/s in Figure 4(b)). Unfortunately, these slower transmissions do not reclaim the capacity vacated after the high-speed transmission finishes, as indicated by the subsequent lower cumulative throughput in both scenarios. We are currently investigating means of adapting the remaining transmissions.

3.1.3 Robustness of Competing Transmissions

Last, we evaluate the robustness of competing transmissions, e.g., when multiple users interact with multiple, distinct objects in parallel. Specifically, the impact of competing transmissions on the frequency hopping scheme in each network determines the performance and usability of DISCO in populated scenarios. Figure 5 hence shows the throughput over distance as measured at two Android devices that receive from different objects. Throughput slightly decreases with increasing distances, contrary to the results obtained in Section 3.1.1. Since we perform all measurements in an indoor office environment, we attribute this to increased interference from 802.11 networks. Still, both transmissions achieve very viable throughput results.
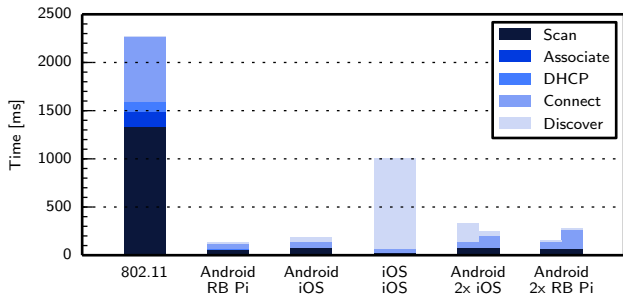
Figure 6: Time overhead of 802.11 and BLE.

3.2 Interaction Usability

The usability of opportunistic interaction is impeded if the discovery and connection with objects induces a substantial time overhead. We thus evaluate the time overhead of establishing communication between object and mobile device in 802.11 and BLE. Figure 6 shows the cumulative averages of the connection steps over 30 runs for an Android device associating to an 802.11 AP as well as between BLE devices. In the figure, the upper device denotes the central, the lower name denotes the peripheral, and "2x" indicates a central connecting to two peripherals in parallel. The latter setting indicates the degree to which connections can be parallelized and adjacently shows both connection timings. Note that the "Associate" and "DHCP" steps are exclusive to 802.11 as the "Discover" step is to BLE.

The results highlight the suitability of BLE for the envisioned usage scenario, as the combined connection steps in BLE require between 6 % (Android/Raspberry Pi) and 44.1 % (iOS/iOS) of the time required for an 802.11 association. This variation shows the interplay of specific implementations. Furthermore, the results indicate a poor parallelization of connection establishments ("2x" results). Interaction in 802.11 would require iteratively expending the time overhead shown in Figure 6 for each object AP.

3.3 Energy Efficiency

We envision users, i.e., their mobile devices, to continuously discover and communicate with objects, making the energy consumption of DISCO crucial for its real-world applicability. In order to assess this consumption and to compare it with local 802.11 networking and IoT interaction over carrier networks, we measure the energy consumption of the respective steps in our design. Note that measurement hardware such as in [18] is not available to us, we follow an equivalent setup as described in [16]. We replace the battery with an adjustable DC power supply and measure the current drain via a Tektronix TDS 2024B oscilloscope over a series-connected shunt. Because we are unable to access the iPhone 5S battery, our measurements only show the consumption of Android devices and Raspberry Pi objects. We measured the consumption 5 times for 100 s with the display turned off and no running applications. Figure 7 shows the average and standard deviation of the results on top of the system standby consumption.

In general, BLE consumes significantly less energy than 802.11 or 3G, at the expense of lower data rates and communication ranges. While higher data rates and shorter resulting transmission durations amortize the higher consumption in 802.11 for large payloads, its challenging topology definition and the small payloads in interaction render 802.11 highly
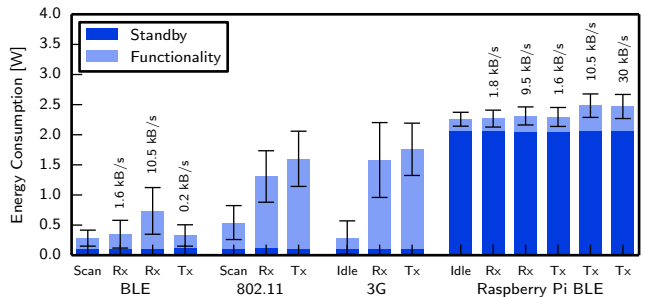

Figure 7: Energy consumption of local and cellular communication for the Nexus 5 device and RB Pi.

expensive in the challenged IoT setting. Similar, discovery and communication exclusively over 3G carrier networks induces a high energy consumption. In contrast, the low energy consumption of object discovery ("Scan") and reasonably fast communication (10.5 kB/s TX) in BLE make it especially suitable for continuous discovery and interaction in ubiquitous scenarios. Indeed, in the isolated measured setting, scanning for objects via BLE consumes 0.28 W compared to 0.54 W for 802.11 scans. Disregarding GUI operation and interaction, continuous BLE scans for objects affords a life time of 30 h for the Nexus 5 battery (8.44 Wh) while continuous 802.11 scans only afford 15.6 h. Last, the Raspberry Pi object shows a comparable energy consumption. Note that we are unable to increase Android-side TX throughput (cf. Section 3.1.1) and that the energy consumption of 30 kB/s TX was measured between two Raspberry Pi as a proof-of-concept throughput maximization.

3.4 GUI Performance

	1 Obj.	2 Obj.	3 Obj.	4 Obj.	5 Obj.
Nexus 5	29.87 fps	29.86 fps	29.71 fps	29.76 fps	29.57 fps
iPhone 5S	30.01 fps	30.02 fps	30.02 fps	30.01 fps	30.00 fps

Table 1: Nexus 5 and iPhone 5S FPS when recognizing and augmenting increasing numbers of objects.

AR-based interaction with smart objects requires mobile devices to recognize, track, and augment the respective number of objects in the camera view. In this, object recognition entails a computational matching of the received CV image descriptors against visual object features. Hence, we evaluate the computational overhead and real-world applicability of interacting with multiple discovered objects simultaneously. To this end, we instruct the app to recognize an increasing number of objects and augment them with 3D objects. As a performance metric, we measure the frames per second (FPS) delivered by our app over a duration of 100 s.

Both the Nexus 5 and the iPhone 5S experience no discernible decrease of the possible frame rate of 30 FPS over any number of objects. Furthermore, we find that the complexity of the augmented GUI does not influence this performance, however we only use an un-animated GUI. We regard this result, also reported by existing evaluations[5], as surprisingly positive and attribute it to the quality of implementation in the *Vuforia* [14] framework. From these results, we assess the immediate real-world applicability of AR-based interaction interfaces that allow the visualization of object functionality and provide a notion of tangibility [7].

---

[5] https://developer.vuforia.com/forum/qcar-api/frame-rate-my-application&sort=2 (second to last post)

4. RELATED WORK

DISCO is related to smart object discovery in the IoT, communication, and interaction approaches. In this section, we discuss existing commercial and academic approaches.

4.1 Discovery

Existing approaches follow the notion of a "Web of Things", i.e., a global web representation of smart objects and their semantics [5,11]. Search mechanisms, e.g., based on RESTful interfaces and JSON representations, then allow discovery within the managed set of objects. We believe that such mechanisms are feasible for dedicated objects but cannot include the estimated billions of objects. Especially, inclusion in such a service induces the privacy risk of disclosing object semantics sufficiently expressive for discovery, e.g., the location and functionality of a home automation system.

On a local scale, approaches such as *iBeacon* [1], [3] afford direct discovery of objects by smartphones, e.g., via BLE. However, information is only transferred from the object to the device [1], indicating for example the location in a store, or need to be registered in advance [3,7]. In contrast, we envision fully bidirectional communication through which the user can interact with the functionality of the object.

4.2 Communication

"Web of Things" [5, 11] approaches assume an Internet connection both at the object and the mobile user. We argue that a multitude of worthwhile scenarios does not meet this assumption, e.g., underground, abroad, and in remote areas. Also, connectivity does not directly yield object discovery.

[9,13] also enable interaction through integration of objects into existing 802.11 networks. While we share the motivation of local communication, we strive for autonomous interaction that neither depends on the availability of networks nor on users having access to this network.

Last, autonomous wireless communication using RFID tags in objects presents a viable mechanism for future designs but is incompatible with current smartphones. In contrast, Near Field Communication (NFC) is supported by smart phones but requires a distance of less than 10 cm.

4.3 Interaction

Current interaction approaches with smart objects require pre-installed interfaces and a network connection to communicate with objects [3, 7, 12], inducing the aforementioned drawbacks. In contrast, direct interaction using auxiliary techniques, such as a light beam [17], reduces real-world applicability and does not afford a bidirectional communication channel. DISCO enables smartphone-compatible interaction techniques to autonomously control object functionality.

5. CONCLUSION

In this paper, we strive to initiate the discussion of the Challenged Internet of Things (IoT), with regard to both connectivity and interaction interfaces. Specifically, we acknowledge the fact that intermittent Internet connectivity and the dependence on pre-installed interfaces prohibits ubiquitous interaction with smart objects. We propose local, opportunistic communication and interaction in DISCO, enabling smart objects to define and provide their interaction interfaces immediately to users. We illustrate the resulting design space in the provision of AR interfaces and GUIs.

We implemented DISCO for Android and iOS smartphones as well as embedded object platforms and demonstrated our design at MobiSys'14 [15]. Our evaluation highlights the immediate real-life applicability of DISCO and shows the benefits of building on BLE for the discovery and establishment of a communication channel. Especially, BLE saves device resources while enabling continuous discovery of smart objects and provides viable throughput performance for the transmission of object and interface information.

Future work targets a comprehensive analysis of application scenarios and requirements within the Challenged IoT. Furthermore, we strive to include further input, such as movement and sound, as well as communication sensors, for example (visible) light and sound, into our extensible design.

6. REFERENCES

[1] Apple. iOS: Understanding iBeacon. http://support.apple.com/kb/HT6048.

[2] Bluetooth Special Interest Group (SIG). Bluetooth Core Specification Version 4.1. https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=282159.

[3] N. Davies, A. Friday, P. Newman, S. Rutlidge, and O. Storz. Using bluetooth device names to support interaction in smart environments. In *MobiSys*, 2009.

[4] C. Gomez, J. Oller, and J. Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12, 2012.

[5] D. Guinard, V. Trifa, and E. Wilde. A resource oriented architecture for the web of things. In *IoT*, 2010.

[6] O. R. Helgason, E. A. Yavuz, S. T. Kouyoumdjieva, L. Pajevic, and G. Karlsson. A mobile peer-to-peer system for opportunistic content-centric networking. In *MobiHeld*, 2010.

[7] V. Heun, S. Kasahara, and P. Maes. Smarter objects: Using ar technology to program physical objects and their interactions. In *CHI EA*, 2013.

[8] Joe Decuir. Bluetooth 4.0: Low Energy. http://chapters.comsoc.org/vancouver/BTLER3.pdf.

[9] L. Labs. LIFX - The Smart WiFi Light Bulb. http://lifx.co/.

[10] Lee Badman. Bluetooth 4.1 Aims For Internet of Things. http://www.networkcomputing.com/a/d-id/1234594.

[11] S. Mayer and D. Guinard. An extensible discovery service for smart things. In *WoT Workshop*, 2011.

[12] D. Molyneaux and H. Gellersen. Projected interfaces: Enabling serendipitous interaction with smart tangible objects. In *TEI*, 2009.

[13] Philips. Hue - Wireless Lighting. http://meethue.com/.

[14] Qualcomm. Vuforia Framework. http://www.vuforia.com/.

[15] J. Rüth, H. Wirtz, and K. Wehrle. Demo: Ubiquitous interaction with smart objects. In *MobiSys*, 2014.

[16] A. Sampson. Measuring Smartphone Energy on a Budget. https://homes.cs.washington.edu/~asampson/blog/powermeasurement.html.

[17] D. Schmidt, D. Molyneaux, and X. Cao. Picontrol: Using a handheld projector for direct control of physical devices through visible light. In *UIST*, 2012.

[18] S. Trifunovic, A. Picu, T. Hossmann, and K. A. Hummel. Slicing the battery pie: Fair and efficient energy usage in device-to-device communication via role switching. In *CHANTS*, 2013.