Enabling Ubiquitous Interaction with Smart Things

Hanno Wirtz*, Jan Rüth*, Martin Serror*, Torsten Zimmermann, Klaus Wehrle
Chair of Communication and Distributed Systems, RWTH Aachen University
{wirtz, rueth, serror, zimmermann, wehrle}@comsys.rwth-aachen.de
* **Equal Contribution**

***Abstract*—Within the Internet of Things (IoT), Smart Things (STs) promise to permeate all contexts of daily life, offering digital access to their physical functionality. Mobile users then would be able to ubiquitously and spontaneously interact with things they encounter, enabling a wealth of diverse usage scenarios and applications. Currently, however, ST interaction requires a pre-controlled Internet or network connection as well as the prior installation of the ST-specific interaction interface, i.e., smartphone app. Users can thus only interact with known things, in contrast to the vision of spontaneous, ubiquitous discovery and interaction.**

We thus propose STIF (Smart Things Interaction Framework), enabling local wireless discovery of STs spontaneously via Wi-Fi, Bluetooth Low Energy, Visible Light Communication, or Acoustic Communication. STIF allows STs to transmit their interaction interface directly to users and supports interaction based on user input via touch and AR GUIs as well as motion and speech recognition. We implement STIF for Android phones as well as Arduino and Raspberry Pi things and demonstrate the real-life applicability of the supported communication and interaction techniques.

I. INTRODUCTION

As the main building block of the Internet of Things (IoT), *Smart Things (STs)* make their physical functionality digitally accessible. In this, both industry and academia propose a vast diversity of deployment and usage scenarios for STs and envision an adaptability of the actual physical implementation of objects to the respective usage scenario [1]. Leveraging the expected comprehensive dissemination of STs [2], [3], enabling the discovery of and interaction with the functionality of STs could for the first time realize the vision of *ubiquitous computing* formulated by Weiser [4].

In this vision, mobile users spontaneously encounter and interact with STs that permeate all contexts of their daily life. For example, STs may expose the control over smart building functionality, such as heating or lighting, could provide an interface to transportation systems, e.g., to indicate or query a route, or could facilitate authentication and payment. Furthermore, interaction with STs may interconnect shopping scenarios with households and may allow for customizable and transferrable work environments as well as offer communication with otherwise inaccessible objects such as in cargo handling and manufacturing.

As a main requirement for such truly ubiquitous interaction, mobile users need to be able to discover, identify, and interact with arbitrary, unknown, and spontaneously encountered objects. Thereby, users first need to discover and establish a communication link with the respective object, over which they subsequently discover the semantics of the object. Actual interaction between user and object then requires the existence of an interaction interface that presents the object functionality, e.g., adjusting the lighting, to the user and transforms user input into interaction commands that trigger actions at the object. Realizing the mechanisms required for such interaction would then make the functionality and information offered by ubiquitous STs discoverable and accessible for mobile users, within their current context and adapting to its dynamics.

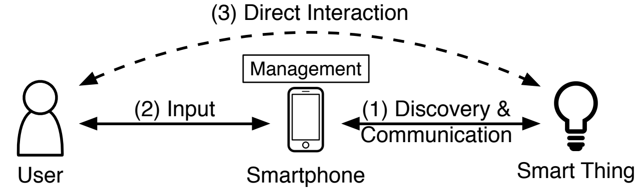

Fig. 1. STIF allows users to discover STs via the communication channels used by the ST and establish communication to obtain the ST interaction interface (1). User input at the smartphone subsequently occurs via the interaction technique designated by the ST and is relayed to the ST (2). STIF thus enables direct interaction within the local context of the user (3) without the requirement of Internet connectivity, global discovery, or pre-installed interaction interfaces.

In stark contrast to this vision, the current design, development, and deployment paradigm of interaction with STs exclusively assume known, pre-configured communication links and pre-installed interaction interfaces, i.e., smartphone apps, per object and application (e.g., [5]–[8]). Specifically, users can currently only discover and communicate with objects over a pre-existing Internet connection or within a shared local network. Actual discovery, communication, and interaction with objects then requires the respective app developed for the respective object and usage scenario.

In reality, however, users may not enjoy continuous Internet connectivity, e.g., underground, indoors, or abroad, in all (mobile) contexts. Furthermore, the sheer number and diversity of STs and usage scenarios prohibit the pre-installation of the appropriate app for all possibly encountered objects. This effectively reduces the set of objects a mobile user *can* interact with to a small subset, namely personal objects and objects to which (network) access and the installation of the app is mediated by a third party. Especially, this paradigm prevents *spontaneous* discovery, communication, and interaction with STs. For example, a user operating a home automation object [7] at his home inherently has no means to discover or

978-1-4673-7331-9/15/$31.00 © 2015 IEEE

interact with another encountered home automation object [8]. This is because objects are, for good reason, not globally discoverable over the Internet and even with local connectivity, the user misses the discovery and interaction interface, i.e., smartphone app, required for this specific object.

We hence propose STIF, as illustrated in Figure 1, enabling users to ubiquitously, autonomously, and spontaneously discover objects, obtain their specific interaction interface, and interact with object functionality exclusively via local wireless communication, mitigating the requirements of Internet connectivity and pre-installed discovery and interaction interfaces. To this end, STIF builds on "traditional" wireless communication channels such as Wi-Fi, Bluetooth Low Energy (BLE), and Near Field Communication (NFC) and additionally incorporates Visible Light Communication (VLC) and Acoustic Communication (AC). Objects then directly provide both their semantics and the definition of their interaction interface to the user and subsequently receive interaction commands triggered by user input into the interaction interface. Our design thereby leverages all input techniques provided by modern smartphones, namely touch and Augmented Reality (AR) Graphical User Interfaces (GUIs) as well as speech input and movement recognition. Building on the set of discovered objects around a user, STIF then additionally offers an intuitive tool for *rule-based* combinations of object interaction. In this, triggering an action at one object triggers a user-defined set of, possibly diverse, actions at further objects. STIF then allows interaction over an arbitrary combination of communication channel and input technique for each encountered object, making the wealth of usage scenarios afforded by the combined diversity of local wireless communication and input techniques accessible to designers of usage scenarios, object implementations, and mobile users.

A. An Example Usage Scenario

As a (fictitious) example usage scenario of ubiquitous, wireless interaction with STs envision a user visiting a large-scale, diverse technical exhibition, such as SIGGRAPH. Numerous vendors and exhibitors make a variety of demonstrations, prototypes, and applications accessible via heterogeneous STs, rendering the creation of a single app that contains all accessible functionality as well as the installation of all specific apps infeasible. Instead, using STIF, a user visiting the exhibition spontaneously decides, based on her interests and context, which things she wants to interact with. To this end, objects can encode advertisements for the respective presentation in small (ultra)sound messages that are received by the user's smartphone and that it replies to via AC by encoding the user's interests. Then, the user uses motion gestures that are transmitted via BLE or Wi-Fi to interact with, e.g., a 3D city model, and could actually talk to virtual representations of historical figures through speech recognition input. Last, the user may navigate the exhibition by interacting with the smart navigation system over VLC, using a touchscreen GUI to indicate and transmit her next destination and receiving the best route based on fine-grained localization.

Notably, the presented scenarios can be realized with alternative communication and input sensors, indicating the diversity and flexibility in building ST appliances. For example, navigation may build on Wi-Fi and interaction with the 3D model could benefit from presenting a 3D AR GUI to the user. Furthermore, complex interaction scenarios may become possible in the presence of multiple STs and the combination of their functionality, e.g., in a room-scale exhibition setup. From this, we derive our motivation to offer comprehensive flexibility regarding the combination of input and communication sensors per ST interaction scenario, in order to fully cover the design space of STs appliances, as well as enabling the orchestration of multi-object interaction scenarios within the local context of the user.

B. Structure of this Paper

In the following, we illustrate the current state of the art as well as related approaches in ST interaction (Section II), motivating our design of autonomous and ubiquitous interaction (Section III). We implement STIF for Android smartphones as well as Raspberry Pi and Arduino Yún things and show its real-life applicability along our communication performance and energy efficiency evaluation (Section IV) and a proof-of-concept real-world application scenario (Section V). STIF hence departs from the state of the art by enabling spontaneous and diverse ST interaction, overcoming the limitations of monolithic, Internet-based approaches (Section VI).

II. CURRENT STATE OF THE ART & RELATED WORK

STIF relates to smart IoT object discovery, communication, and interaction approaches. In this section, we discuss existing commercial and academic approaches.

A. Discovery

Existing discovery approaches follow the notion of a interconnected "Web of Things", i.e., a global web representation of STs and their semantics [9], [10]. Search mechanisms based on, e.g., RESTful interfaces and JSON representations then allow semantic 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 semantic object information sufficiently expressive for discovery, e.g., the location and all functionalities of a user's home automation system.

On a local scale, approaches such as *iBeacon* [11] or [12] afford direct discovery of objects by smartphones, e.g., via Bluetooth (BT). However, information is only transferred from the object to the device [11], indicating for example the location in a store, or need to be registered in advance [6], [12]. In contrast, we envision fully bidirectional communication through which the user can interact with the functionality of the object, instead of only receiving information.

B. Communication

"Web of Things" [9], [10] approaches assume an Internet connection both at the thing 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. Moreover, the need for an indirection of relevant information via the Internet somehow contradicts the vision of direct interaction with STs.

[7], [8] 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 deployment of availability of networks nor on users having access to this network.

[13], [14] show the feasibility of alternative techniques like VLC and AC for local communication. In [13], the authors present a VLC system that allows for communication between objects equipped with LEDs and smartphones, using the smartphone's flashlight-LED for sending and the camera for receiving data. The goal of Dhwani [14] is to enable a secure NFC like channel between devices with AC. The proposed setup uses the existing microphones and speakers on smartphones and therefore does not rely on additional hardware. Although the achieved data rates of both presented approaches are comparably low (in the order of several hundred bits per second), they suffice to transmit simple commands between devices.

Within STIF, we aim to facilitate direct communication between the smartphone and the ST. Therefore, we leverage all available communication techniques offered by the smartphone, i.e., 802.11, Bluetooth, NFC as well as VLC and AC.

C. Interaction

Current interaction approaches with STs require pre-installed interfaces and a network connection to communicate with objects [6], [12], [15], inducing the aforementioned drawbacks. In contrast, direct interaction using specialized auxiliary techniques, such as a light beam [16], reduces real-world applicability and does not afford a bidirectional communication channel. STIF enables smartphone-compatible interaction techniques to autonomously control the object, with the object exposing its functionalities to STIF. This self-contained approach allows the direct interaction with devices that are discovered for the first time.

III. DESIGN

Figure 1 provided a high-level overview of our design enabling ubiquitous and direct interaction with STs in our Smart Things Interaction Framework (STIF). Specifically, STIF bootstraps and performs interaction following the sequence ofi) discovering an ST on an arbitrary communication channel, ii) capturing the semantics of said ST, such as location, manufacturer, model or functionality, in order to present the semantics to the user or match them against pre-defined user interests, and iii) obtaining the appropriate interaction interface, that makes the ST functionality accessible to the user, directly from the ST. Thusly representing the ST functionality to the user, STIF iv) captures user input into this interface and transforms it to interaction commands that are v) transmitted to the object, triggering the respective functionality.

By affording this flexibility as well as enabling unmediated discovery and interaction, we argue that STIF meets the requirements of truly spontaneous and ubiquitous interaction with STs. By realizing this design, STIF makes the following three main contributions:

i) Comprehensive incorporation of current and future communication channels of smartphones and STs to support truly flexible **discovery and communication** of and with STs (Section III-A),
ii) Spontaneous, immediate obtainment of the ST's specific interaction interface and representation of the interface to the user in a generic smartphone app with support for a variety of **input** techniques (Section III-B), and
iii) a **management** layer for the combination of communication and input techniques per object as well as the creation and execution of rules for complex multi-object interaction (Section III-C).

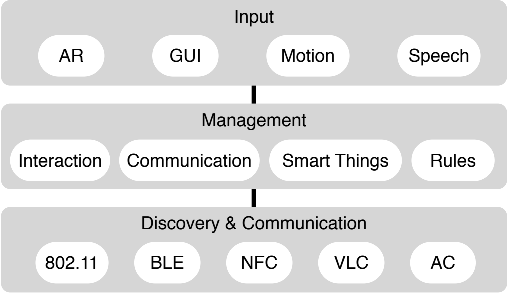

Fig. 2. Layered STIF functionality: *Discovery and communication* of and with Smart Things over a specific communication channel is matched to a specific *input* technique in the *management* layer. Rules hold a set of interaction commands to be transmitted to a set of STs over designated communication channels upon the firing of a trigger event.

Each contribution is thereby reflected in a distinct layer within the Smart Things Interaction Framework (STIF), as illustrated in Figure 2. We present the design details of each aspect in the following sections and Section IV-A offers more technical details of our implementation. Furthermore, we provide a notion of the required instrumentation (overhead) of STs in our design (Section III-D).

A. Communication

In striving for comprehensive coverage of viable communication mechanisms, we take an optimistic approach. Namely, we address all available communication mechanisms available in smartphones today and furthermore account for advances in Visible Light Communication (VLC) and Acoustic Communication (AC) by including both mechanisms as we expect them to make the transition from academic [17], [18] to real-world implementations. Hence, STIF incorporates and offers support for communication over Wi-Fi, BLE, NFC, VLC, and AC. Each communication technique, i.e., smartphone sensor, is thereby registered within STIF and may be set to discover ST in the local context continuously, periodically, or triggered by the user.

Wi-Fi offers an attractive communication channel with high throughput and large communication ranges and therefore makes a substantial interaction scope around a mobile user accessible. In this, the 802.11 ad-hoc mode seems to meet our requirements for spontaneous, mobile ST discovery and communication. However, this mode is not supported by typical smartphones [19], [20]. We thus build on the 802.11 infrastructure mode in STIF and leverage the tethering capabilities of current smartphones, i.e., create an 802.11 network with a pre-defined or uniquely prefixed SSID (e.g., "`stif_ssid`") at the smartphone. STIF thereby enables multiple STs to connect to a smartphone and to proactively provide their semantics. A mobile user is then able to obtain the interaction interface of selected or all STs, either by active selection or automatically for defined semantics of interest.

Alternatively, STs could operate an 802.11 network. Mobile users then discover STs and would need to iteratively connect to each ST in order to obtain its semantics and, if desired, the interaction interface. While STIF also supports this topology, we currently prefer the topology outlined above because, by providing the network, unmodified smartphones can communicate and interact with multiple STs in parallel[1].

In contrast to the topological issues of Wi-Fi, **Bluetooth Low Energy (BLE)** [22] offers a perfect topology for the envisioned usage of STIF. Specifically, STs operate as peripherals and in this function make themselves discoverable for smartphones, operating as centrals. Then a central can, in parallel, initiate an association to multiple peripherals, discover their semantics, obtain their interfaces, and communicate interaction commands. STs therefore announce a human-readable identifier that indicates STIF functionality and make their semantics and interaction commands accessible over Generic Attribute Profiles (GATTs). The reduced communication range of BLE, in comparison to up to 100 m in Wi-Fi, thereby could afford a subjectively more tangible interaction range around the user.

Wi-Fi and BLE thereby represent the "traditional" communication channels that lend themselves to mobile and ubiquitous interaction, due to the offered range and throughput. In light of the diversity of application scenarios for Smart Things, we envision additional use casesi) for which lower data rates ($\leq$ 1 kB/s) suffice or ii) that benefit from a more direct communication between ST and mobile user, i.e., her smartphone. Examples for such use cases are the distribution of map material and object semantics to users entering a room or building via Near Field Communication (NFC), fine-grained localization via Visible Light Communication (VLC), or background information pushing over (ultra)sound Acoustic Communication (AC).

STIF hence includes **NFC** in peer-to-peer mode as a communication channel. Discovery of STs thereby requires close proximity of the ST and the smartphone, either by placing the ST in a sufficiently confined location that the user passes, e.g., a door frame, or via a visual marker that indicates an appropriate placement of the smartphone. STs then transfer their semantics and possibly further information, interaction interfaces, and interaction commands. In the aforementioned example of providing map material to users, the user may interact with the ST to query for details about locations and entities on the map or for navigation.

VLC, in contrast, offers a continuous communication channel that most notably requires a line of sight (LOS) connection. Current smartphones already include Light-Emitting Diodes (LEDs), the potential of which was demonstrated in [17], hinting at the possibility of consumer-grade, mobile VLC using smartphones. STs or smartphones then periodically announce their presence via "beacon" messages and engage in communication upon reception of an appropriate request by the smartphone or response by the ST. Selection of the ST to communicate and interact with would then intuitively occur by pointing the smartphone LED towards the ST. VLC thereby offers a spatially directed communication channel that is observable and controllable for the user.

In contrast, **AC** offers a pervasive communication channel that is similar to Wi-Fi and BLE regarding arbitrary spatial sender-receiver positioning. However, AC may function without the requirement of a wireless network association prior to sending and receiving and therefore offers a true background channel. STs can leverage this feature to disseminate static, self-contained information, such as wireless advertisements, to mobile users passing by, with advertisements triggering a smartphone alert if they match the user interests. Conversely, mobile users may use AC to announce their presence or interests. Notably, current smartphones are equipped with microphones and a loudspeaker, providing a basis for AC. The resulting communication is, however, audible (and annoying) for humans and is thus not usable as a ubiquitous communication channel in real-life. Including ultrasound capabilities to the speaker and microphones thereby is a small effort and could in the future enable numerous AC-based applications, e.g., [23], in addition to interaction with ST.

Upon discovery and initiation of communication with a ST, STIF envisions the ST to transmit the definition of the interaction interface the ST wants the user to use. We detail the different definitions and instantiations of interaction interfaces as well as the requirements towards the underlying communication channels in the following section.

B. Input

We target smartphones as interaction devices for mobile users. STIF hence caters to input techniques, for the manipulation of ST functionality, that are appropriate for the use with a smartphone. Namely, STIF supports traditional touchscreen GUIs, Augmented Reality (AR) GUIs, motion recognition, and speech recognition interaction interfaces. In order to represent such a diversity of interface definitions from a ST, STIF consists of a generic launching point app that receives interface definition and realizes the actual interface according to the obtained definition.

Specifically, an interface definition always consists of a number of *elements* that are to be included and that may have some initial state, e.g., a slider setting of 50 % defining the initial setting of a light bulb's brightness. Each element is associated to an *action* that is to be triggered upon activation of the element, e.g., sending a command to the ST. Notably, the generic STIF app does not need to know or understand the specific functionality of each element or the overall interface.

Touchscreen GUIs thereby are the natural interface type for smartphone-based interaction. A ST defines the interface by indicating the set of, for example, buttons, switches, and text fields that make up the GUI. Note that it is not necessary to extensively describe the visual appearance and layout of each element. Instead, the ST only indicates a predefined type indicator that STIF then realizes with the default GUI element of the respective operating system, in order to save communication overhead.

STIF furthermore supports **Augmented Reality (AR) GUIs** to provide a visual and tangible connection of interactions with

---

[1] While wireless network virtualization [21] allows multiple associations in parallel, current smartphones do not support this technique and we expect STs to be better customizable in this regard.

ST which touchscreen GUIs can not afford, as illustrated in [6]. In addition to the GUI elements, STs therefore transmit the Computer Vision (CV) material required to detect and localize the ST as well overlay it with the interface elements in the camera view. Input then again occurs over touchscreen elements. AR GUIs thereby offer an augmentation of ST without or with limited physical interfaces, such as monuments, buildings, or intentionally simple appliances [6].

An exciting venue for input recognition on smartphones is **motion or gesture recognition** because it does not require a conscious effort but can be performed, e.g., without looking at the smartphone. A ST may thus associate a set of gestures, e.g., "up/down/left/right", "push/pull", or "right/left rotation" with interaction commands and again only needs to provide gesture type indicators to the smartphone. STIF then recognizes the respective gestures via the gyroscope and accelerometer. Notably, at the cost of higher complexity, gestures can be combined or repeated to construct hierarchical, sequential, or continuous interaction patterns.

Last, STIF offers speech recognition input. STs thereby store a very restricted dictionary of input commands together with their acoustic representation and provide them to the smartphone as the interface definition. STIF then performs statistical speech recognition to match spoken words to input commands and transmits the command (ID) to the ST. Notably, reducing the input dictionary *at the object* mitigates the requirement of complex speech recognition of all possible spoken inputs.

In general, we strive for reduced communication overhead in the interface provision in order to account for the throughput characteristics of NFC, VLC, and AC. Still, provision of an AR GUI is impractical over these communication channels, while all other interface types are viable.

C. Management

As illustrated in Figure 2, STIF employs a management layer that allows the flexible orchestration of communication channels and input techniques, as spontaneously discovered for each ST. To this end, the management layer stores and manages specific ST as well as their respective **input–communication combination**. Furthermore, it controls the communication sensors available on the smartphones and serves as an abstraction layer and uniform ingress/egress point for all communication with STs.

Surpassing traditional interaction with STs via Internet services using specific, isolated apps, local interaction with STIF makes *all* locally available ST accessible for interaction. Obtaining the interaction interface of each ST thereby allows fully autonomous control over the sequence and combination of ST interaction. STIF leverages this control and enables users to devise complex **interaction rules** that combine and chain interaction with multiple STs, similar to Internet-based chaining of services in *if this then that (IFTTT)* [24]. Each rule thereby consists of the included STs as well as a *trigger* event and multiple *actions*, i.e., interaction commands, triggered by this event. We thereby envision users to instrument whole rooms or scenarios according to their preferences and routines or to increase the efficiency of reoccurring interaction events.

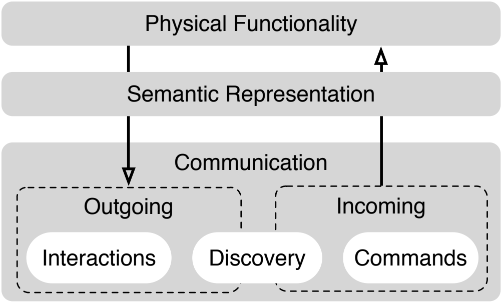

Fig. 3. Layered design of STs in STIF: The physical functionality and semantics of STs are made accessible via a bidirectional, wireless communication channel.

D. Smart Things Instrumentation

In order to enable interaction between users and arbitrary STs, the latter must implement the following functionalities: i) a wireless communication channel for announcement beacons and the reception of user input commands, and ii) provision of semantics that describe the ST's physical functionality and how the user may trigger them. Figure 3 shows how these functionalities are realized by STs in our design. Each ST has certain physical functionalities, e.g., switching lights on/off, that are made discoverable to smartphones using a semantic representation such as JSON. The communication layer transmits these semantics, as well as interaction interface definitions, to smartphones and accepts user input in the form of pre-defined interaction commands. All received interaction commands are then processed by the communication layer and handed to the underlying functionality to trigger the corresponding physical action.

IV. TECHNICAL EVALUATION

In this section we evaluate STIF with regards toa) the communication performance and robustness when transmitting ST interfaces and commands (Section IV-C), and b) the energy efficiency of our approach with respect to the various communication channels (Section IV-D). To allow to put our results into perspective we start by briefly sketching our Android (Section IV-A) and ST implementations (Section IV-B) first.

A. Smartphone Implementation

We implemented our framework for the Android operating system. We use LG Nexus 5 smartphones running a stock Android 4.4 image as representatives for typical mid to high-end consumer smartphones. For the realization of the different communication channels we try to leverage standard Android software components where possible. We thus use Android's BLE functionality and the standard GATT service.

Our Wi-Fi implementation makes use of Android's tethering capabilities providing soft access point (AP) functionality using a fixed predefined SSID such that STs in range can scan the wireless medium and connect to the smartphone. Within the network, we make use of the standard TCP/IP stack for transport of object semantics and interaction commands.

For VLC, we make use of the smartphone's LED flash and the smartphone's ambient light sensor. We implement an On/Off keying together with a Manchester code, thus in theory allowing for flicker-free communication. Unfortunately, both components are currently not efficiently steerable using the Android OS as also noted in [13], i.e., the OS induces severe delays through scheduling and abstraction. In our implementation, one VLC data frame offers space for 40 byte of payload and the receiver has to reply with an Acknowledgment (ACK). A retransmission occurs after 7.5 ms when no ACK was received.

In addition to VLC, we establish an acoustic link between a ST and smartphone by using the loudspeaker and microphone to send and record simple frequency modulated data. We use 17 frequencies in the band between 9.5 kHz and 15.5 kHz to modulate a data frame. While this frequency range is still audible to humans, it is beyond the typical human voice spectrum but still in the operation range of typical smartphone hardware.

As some of the communication channels offer only low data rates, we encode data using gzip compressed MessagePack[2], a space efficient binary serialization, thus saving significant time during transmission in comparison to simple encoding schemes like JSON.

For AR GUIs, we use the Qualcomm Vuforia framework [25] to augment ST in the smartphone's camera view. Speech commands are recognized using Pocketsphinx [26], a handheld-optimized version of the popular CMU Sphinx.

B. *Smart Thing Implementation*

To implement ST prototypes we use the Raspberry Pi and Arduino Yún platform. For BLE we attached off-the-shelf Inateck Bluetooth 4.0 USB dongles to the platforms and modified the Linux kernel `bluez`-stack[3] and user-space `bleno`-stack[4] to allow for greater tuning potential of Bluetooth connection parameters from user-space. We leverage the `OpenWrt` Linux capabilities of the Yún for Wi-Fi communication. For VLC, we attached an LED to one of the I/O pins and added a Light Dependent Resistor (LDR) to the on-board analog to digital converter. Using these platforms and techniques we are able to use the same data encodings as on the smartphone. In addition, we are currently implementing the acoustic channel on the Arduino Yún.

C. *Communication Performance*

To see whether our communication channels afford the required bandwidth and resilience to transport ST semantics and interaction commands, we evaluate the throughput and the Packet Delivery Ratio (PDR) under varying conditions. Our evaluation will hint at possible use cases of each communication channel while acknowledging typical payload sizes. These range from 10 to 20 byte for interaction commands, 100 byte for normal GUIs with touch or motion, 200 byte for speech interfaces, and up to 100 kB for complex AR interfaces. Note that we omit a discussion of Wi-Fi, as we do not adapt any mechanism and thus do not contribute novel results. We therefore focus on BLE, VLC, and AC.

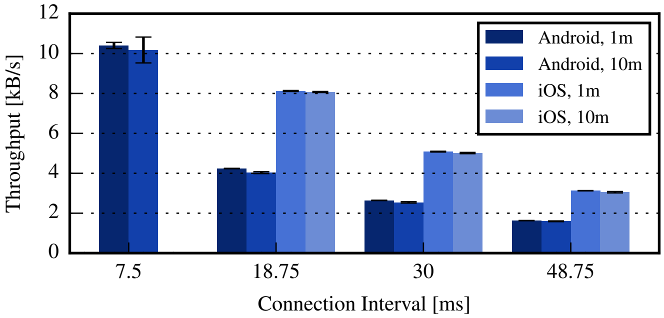

Fig. 4. BLE throughput to Android and iOS devices over distance and connection intervals.

1) Bluetooth Low Energy: The Bluetooth specification [22] as well as an isolated evaluation in [27] 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 contrast, our first measurement results did not surpass a throughput of 1.6 kB/s on Android and 3.1 kB/s on iOS. We found that Android and iOS use a different *default* connection interval, essentially defining intervals in which data may be transmitted, and a different `Maximum Transmission Unit (MTU)`. Thus, the combination of the number of frames send by the respective Bluetooth stack, the MTU, and connection interval define how efficiently the medium can be occupied. We observe that Android uses the BLE default MTU of 20 byte while iOS tries to negotiate an MTU of 132 byte. In addition, Android uses a connection interval of 48.75 ms while iOS defaults to 30 ms.

We modified the Linux bluez stack to gain control over the connection interval parameter from user-space. This allows the ST to increase the throughput on demand. Figure 4 shows the average throughput from 50 measurements transmitting 100 kB between ST and Android/iOS while varying the connection interval and distance. We reach a maximum throughput of 10.5 kB/s (Android) and 8 kB/s (iOS). Note that iOS rejects connection intervals $\leq$ 18.75 ms. Furthermore, we observe that the distance between sender and receiver has only a marginal influence on the performance in our indoor measurement setup.

From these results and the ST's capability to switch the transmission speed on demand we derive that the use of BLE is feasible to transport even complicated AR interfaces in short time. In addition, the stable throughput over greater distance indicates that BLE is also suitable to preload object semantics in the background, i.e., while the user is moving. For an in-depth discussion on BLE performance please refer to our previous work in [28].

2) Visible Light Communication: To asses possible fields of application we evaluate our proof-of-concept VLC prototype in terms of throughput and resilience. However, as our Android software gives only limited control over the LED and ambient light sensor, we evaluate our system using two Arduino boards. Nevertheless, we expect that future Android versions offer greater control over the LED and ambient light sensor thus making it feasible to implement VLC on smartphones.

Figure 5 shows the throughput and PDR of our VLC system under different lighting conditions while increasing the distance between sender and receiver. We repeat each measurement 50 times while transmitting 1 kB for each distance

---

[2] http://msgpack.org/

[3] http://www.bluez.org

[4] https://github.com/COMSYS/bleno

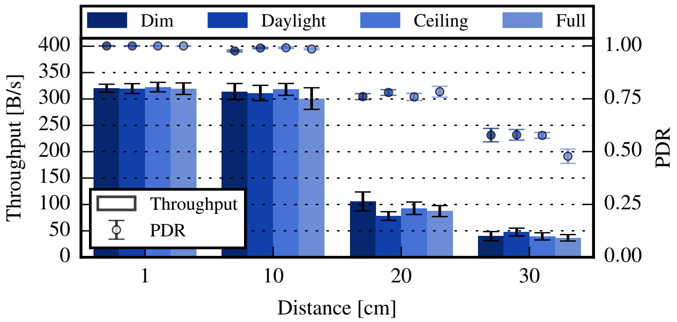

Fig. 5. Barplot showing VLC throughput and average PDR between two Arduinos over varying distances and under different ambient lighting conditions.

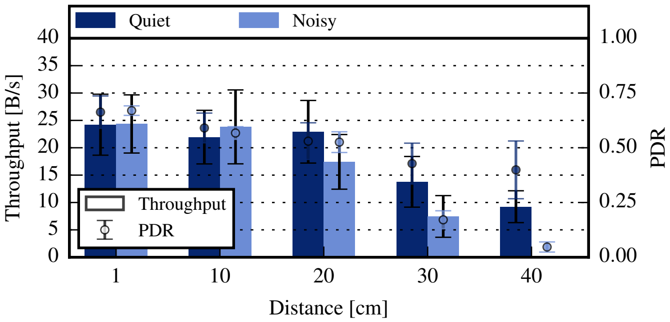

Fig. 6. Barplot showing AC throughput and average PDR between two Android smartphones over varying distances and noise levels.

step. The throughput measurement uses ACKs while the PDR measurements do not. We measure data rates slightly above 300 B/s for distances of 1 and 10 cm. At distances of 20 and 30 cm, the throughput degrades heavily down to 75-100 B/s. This performance drop can be explained by the drop in the PDR down to 50-80 %. Even though the drop does not seem very high, it influences not only the data frames sent but also the ACKs in reverse direction, thus accounting for the severe drop. While the measured data rates do not allow the transmission of complex AR interfaces, small payloads, e.g., touchscreen GUIs or interaction commands, are easily transmittable. Thus, commands triggered in a preloaded interface can be transmitted via VLC to the ST by approaching it and initiating the transmission.

VLC offers a great potential within our framework and with the interaction with STs in general. LEDs are already ubiquitously deployed in many devices, typically for status indication. These LEDs could be used for sending and receiving as described in [17] thus interfacing STs seamlessly using VLC. We feel that with more sophisticated VLC approaches, e.g., as in [13], and error coding, better distances and throughput can be achieved thus making VLC applicable for transmissions over greater distances and thus easier interactions.

3) Acoustic Communication: To evaluate how well AC supports our design and which application scenarios are feasible, we mount a measurement in which two smartphones are placed on top of a desk while varying their distance. In addition, we compare the performance in a quiet environment and under the influence of white noise, measuring about 67 dB SPL at the smartphones. To measure the PDR we transmit 100 packets containing 40 byte of payload, in contrast to the throughput measurement in which we transmit 256 byte payload chunks in frames containing 40 byte of payload. For the throughput, we again aim for a reliable transmission thus the receiver ACKs the data.

Figure 6 shows the results of 30 repetitions in each measurement setup. In general we observe a performance degradation when we increase the distance between sender and receiver. Even when putting the devices immediately next to each other, the PDR only barely exceeds 70 %. The PDR in the quiet and noisy setting is comparable up to 30 cm, at 30 cm the performance drops critically. The PDR even decreases to only 5 % in the noisy environment at a distance of 40 cm.

The throughput shows similar trends. For distances up to and including 10 cm, on average we measure no more than 25 B/s in both scenarios. Again, with a falling PDR, the throughput degrades to below 10 B/s in the quiet case, no successful transmission is possible in the noisy setting at a distance of 40 cm.

Altogether, we observe high standard deviations in all measurements. We feel that current smartphone loudspeakers and microphones are not designed to operate above the typical spectrum of human voice, thus giving substandard results. We chose these high frequencies to reduce the disturbance on humans hearing. Eventually, we would prefer to use ultrasound to make the communication inaudible. Therefore, efforts to use ultrasound for localization on smartphones as for example in [23] would thus also be beneficial to us. Nevertheless, even with these low data rates, AC is a valuable addition to our framework, as it allows us to send small updates and commands to STs while not keeping a LOS requirement. Thus AC could for example be used when a user performs a motion gesture and needs to transmit the associated action to the ST. With AC she would not need to place or point her device to a dedicated spot as it would be required with VLC or NFC.

D. Energy Efficiency

Our framework targets mobile smartphone users, therefore the power consumption introduced by our framework is a critical key component. To account for the impact on the battery life we measure the energy consumption introduced by our communication techniques. We measure BLE, Wi-Fi, AC, VLC, and for comparison, 3G power consumption. BLE, Wi-Fi, and 3G are measured on our Android smartphone while we use an Arduino Yún for VLC and a Raspberry Pi for another comparative BLE measurement. To asses the power consumption we measure the voltage drop over a shunt resistor in series to a constant power source, connected to the respective device, using a `Tektronix TDS 2024B` oscilloscope. As the voltage drop is proportional to the current drawn by the device, we can calculate the power consumption.

First, we establish a baseline for the standby power consumption of each device by turning all communication interfaces off and disabling the screen if present. Then, we measure the average power consumption 5 times over 100 s. Subsequently, we measure each communication interface by turning it on and performing a predefined task. Again, we repeat the measurement 5 times and measure 100 s each. The difference of the first and the second measurement thus denotes the average power consumption of the respective

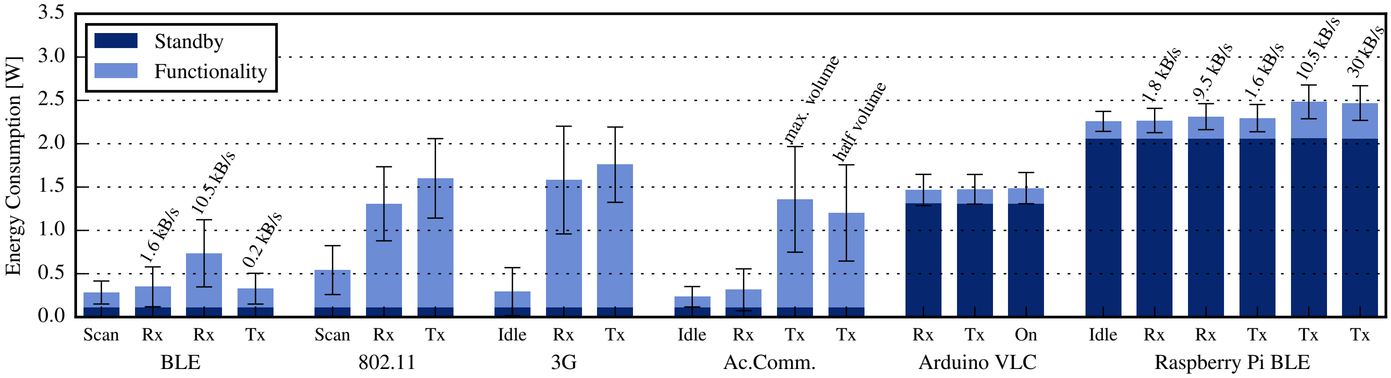

Fig. 7. Power consumption of communication technologies used in STIF.

communication technique. Figure 7 depicts the results from all measurements.

The power consumption of BLE supports our envisioned continuous sensing approach. Scanning for STs (cf. Figure 7 BLE Scan) consumes comparably much power as an idle 3G interface (cf. Figure 7 3G Idle), thus attesting the applicability for a day long usage. In contrast to scanning with Wi-Fi (cf. Figure 7 802.11 Scan), BLE consumes significant less energy and, due to its protocol design, already offers semantic discovery. Receiving data with BLE (cf. Figure 7 BLE Rx) can be tuned to the requirements, i.e., how much data needs to be transmitted, thereby also tuning the energy requirements. Transmitting data to a ST (cf. Figure 7 BLE Tx), e.g., interaction commands, shows the same adaptability. In this, we do not require a high bandwidth as these commands are typically only a few bytes in size, thus affording low energy consumption. Preferring Wi-Fi over BLE is only useful when a high throughput link, i.e., a lot of data needs to be transported, or a very long range is required. Wi-Fi and BLE can however coexist and be activated according to the current usage scenario.

In addition to measuring the consumption on the smartphone we also measured BLE on the Raspberry Pi as an example for a ST. Most notably the standby consumption of the Pi is significantly higher. This is not necessarily a problem as STs can be connected to a power outlet. Overall, BLE adds a similar overhead to the baseline as it does for the smartphone, most notably the consumption does not rise as dramatic for faster reception rates (cf. Figure 7 Raspberry Pi Rx) as it does in the smartphone case. The same is true for transmission (cf. Figure 7 Raspberry Pi Tx), which are independent of the speed as well, thus again attesting the feasibility to increase transmission speed as required.

The power consumption of AC shows the biggest variations between sending (cf. Figure 7 Ac. Comm. Tx) and receiving (cf. Figure 7 Ac. Comm. Rx) of over 1 W. Even when halving the output volume the power consumption does not drop significantly. However, as we envision that AC will mostly be used for short commands, the transmit power consumption is not as critical as passive listening for updates. Indeed, idle listening (cf. Figure 7 Ac. Comm. Idle) and receiving, i.e., having to actually compute on top of the received data, has no significant impact. These findings support the envisioned usage scenario in which we will only sporadically send data but listen for announcements in the background.

The VLC measurements were performed on the Arduino Yún. All three measurements receive, transmit, and constantly on (cf. Figure 7 Arduino VLC Rx/Tx/On) show a very similar and low energy drain. During reception the Arduino does only perform analog to digital conversion to read the voltage drop over the LDR while during transmission only an LED needs to be switched using Manchester code, both do not pose very high energy requirements. We cannot even spot a difference between having the LED constantly on or doing high frequent flickering. Thus, frequent and infrequent usage of VLC does not impose any restriction on the applicability. In addition, VLC is a promising building block in our framework as we do not pose high energy and component demands on the STs, we just require an LED which a lot of devices already have for status indication.

Accounting for the energy consumption of prominent examples for ST interaction is difficult, for example "Web of Things" applications involve, apart from the smartphone a lot of different entities, e.g., routers, servers, and the STs that one would need to consider. Nevertheless, for the smartphone itself, it already poses high energy demands. The smartphone would typically communicate either via Wi-Fi or a mobile broadband uplink, e.g., like 3G. When looking at the power consumption for Wi-Fi and 3G, cf. Figure 7 802.11/3G, we see the highest demands from all measured interfaces. In addition, approaches like the "Web of Things" raise further problems like discovery of nearby things, to solve these we need additional energy e.g., to use GPS or other localization techniques.

V. APPLICABILITY

We demonstrate the applicability of STIF in a video[5] by showing two exemplary use cases with different user to ST interactions.

The first use case illustrates the combination of different input and communication techniques to control ceiling lighting in an office. After the user discovered the lighting control, i.e., the ST, and received its semantics via BLE, she is able to interact with the ST in four possible ways. These are then presented in the GUI of the STIF app. When she selects *Touch* for interaction, the framework shows a switch, which was generated based on the received semantics. After switching the light on or off, the respective command is sent to the ST

---

[5] http://www.comsys.rwth-aachen.de/short/secon15-stif/

using VLC. If she selects *AR* for interaction, STIF recognizes the marker of the ST and augments the camera view with a virtual user interface. In this case, the respective commands are transmitted via BLE. Another option for interaction is *Speech*. When selected, the STIF app shows the user what she has to say in order to issue a command. Again, here we use VLC for communication between the smartphone and the ST. Finally, she could also select *Motion* to interact with the ST by using gestures. Therefore, STIF displays the gesture and the respective command with animations, which have also been derived from the semantics. After recognizing a valid gesture, STIF transmits the resulting command via BLE.

In the second use case, we show the interaction between a user and multiple STs. Therefore, we extend the scenario from the first use case by adding two additional STs, i.e., the office's blinds and a coffee machine. After the user has discovered all three STs and received their semantics, she is able to combine multiple actions in a single rule. Such rules allow her to facilitate her daily routines. In order to do so, she adds a customized rule via the GUI. As a trigger, she selects the *Speech* based control of the ceiling light. Then she defines two actions connected to this trigger, i.e., opening the office blinds and turning on the coffee machine. When she now enters her office for the first time in the morning and switches on the light through voice activation, the blinds open and rise and the coffee machine makes a coffee.

VI. CONCLUSION

In this paper, we embed the vision of ubiquitous computing into the emerging scenario of the IoT, in which widely deployed Smart Things offer ubiquitous interaction with physical functionality. We propose STIF, addressing the infeasibility of comprehensive pre-defined connectivity with and global discovery or STs as well as the impossible task of pre-installing specific interaction interfaces, i.e., smartphone apps, for all STs encountered in the future. STIF alleviates these challenges in affording spontaneous, local wireless discovery of STs, their semantics and, most importantly, their specific interaction interfaces in a flexible and extensible framework.

We implement STIF for Arduino and Raspberry Pi STs and Android smartphones and incorporate the Wi-Fi, BLE, NFC communication capabilities of smartphones as well as take the first steps towards smartphone-based Visible Light Communication and Acoustic Communication. We demonstrate the viability of spontaneous, ubiquitous interaction in STIF and highlight the performance and energy efficiency of the respective discovery and communication channels. Our implementation of a combined office usage scenario furthermore highlights the real-life feasibility and possibilities of the proposed communication and interaction mechanisms as well as local, rule-based scenario orchestration in STIF. Future work will add additional communication sensors such as ultrasound and further explore VLC by using more sophisticated and controllable LEDs, such as in [17].

ACKNOWLEDGMENT

This work has been funded by the German Research Foundation (DFG) in the Collaborative Research Center (SFB) 1053 "MAKI – Multi-Mechanism-Adaptation for the Future Internet".

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Comput. Netw.*, vol. 54, no. 15, 2010.

[2] Cisco White Paper, "The Internet of Things – How the Next Evolution of the Internet Is Changing Everything," [Online] http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, 2011.

[3] Ericsson White Paper, "More than 50 billion connected devices," [Online] http://www.ericsson.com/openarticle/mwc-connected-devices_1686565587_c, 2011.

[4] M. Weiser, "The Computer for the 21st Century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, 1999.

[5] N. Blocks, "Next Generation Control of Your Environment," http://ninjablocks.com/.

[6] V. Heun, S. Kasahara, and P. Maes, "Smarter Objects: Using AR Technology to Program Physical Objects and Their Interactions," in *CHI EA*, 2013.

[7] Philips, "Hue - Wireless Lighting," http://meethue.com/.

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

[9] S. Mayer and D. Guinard, "An Extensible Discovery Service for Smart Things," in *WoT Workshop*, 2011.

[10] D. Guinard, V. Trifa, and E. Wilde, "A resource oriented architecture for the Web of Things," in *IoT*, 2010.

[11] Apple, "iOS: Understanding iBeacon," http://support.apple.com/kb/HT6048.

[12] N. Davies, A. Friday, P. Newman, S. Rutlidge, and O. Storz, "Using Bluetooth Device Names to Support Interaction in Smart Environments," in *MobiSys*, 2009.

[13] G. Corbellini, K. Aksit, S. Schmid, S. Mangold, and T. Gross, "Connecting Networks of Toys and Smartphones with Visible Light Communication," *Communications Magazine, IEEE*, vol. 52, no. 7, 2014.

[14] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: Secure Peer-to-peer Acoustic NFC," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, 2013.

[15] F. Kawsar, E. Rukzio, and G. Kortuem, "An Explorative Comparison of Magic Lens and Personal Projection for Interacting with Smart Objects," in *MobileHCI*, 2010.

[16] D. Schmidt, D. Molyneaux, and X. Cao, "PICOntrol: Using a Handheld Projector for Direct Control of Physical Devices Through Visible Light," in *UIST*, 2012.

[17] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross, "LED-to-LED Visible Light Communication Networks," in *MobiHoc*, 2013.

[18] B. Zhang, Q. Zhan, S. Chen, M. Li, K. Ren, C. Wang, and D. Ma, "Enabling Keyless Secure Acoustic Communication for Smartphones," *Internet of Things Journal, IEEE*, vol. 1, no. 1, 2014.

[19] Google, "Ad-Hoc Support in Android," [Online] http://code.google.com/p/android/issues/detail?id=82.

[20] iOS Developer Library, "Document Transfer Strategies," [Online] http://developer.apple.com/library/ios/technotes/tn2152/_index.html.

[21] R. Chandra, P. Bahl, and P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card," in *INFOCOM*, 2004.

[22] Bluetooth Special Interest Group (SIG), "Bluetooth Core Specification Version 4.1."

[23] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling Fine-grained Indoor Localization via Smartphone," in *MobiSys*, 2013.

[24] IFTTT, "IFTTT: Put the internet to work for you." https://ifttt.com/wtf.

[25] Qualcomm, "Vuforia Framework," http://www.vuforia.com/.

[26] D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky, "Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices," in *ICASSP*, 2006.

[27] C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," *Sensors*, vol. 12, 2012.

[28] H. Wirtz, J. Rüth, M. Serror, J. A. Bitsch Link, and K. Wehrle, "Opportunistic Interaction in the Challenged Internet of Things," in *CHANTS*, 2014.