Scalable Control System for Bluetooth Mobile Devices

Giorgio Corbellini[§], Lukas Kuster[§†], Thomas R. Gross[†]

[§]Disney Research
8006 Zurich, Switzerland

[†]Dep. of Computer Science
ETH Zurich, Switzerland

***Abstract*—Bluetooth (BT) devices are usually grouped in a star topology in small clusters called personal area networks (PANs). The population of BT PANs is limited to 8 devices, but setups in entertainment theme parks or interactive installations might want to employ a larger number of mobile devices. In those scenarios, only off-the-shelf devices, without any hardware modifications, can be accepted, so a PAN must be extended to become a low-cost cellular network for consumer products. We explore the design of such a cellular network of BT nodes. The paper discusses the practical aspects of a wireless control system and reports the experience obtained from implementing a proof-of-concept system. The prototype implementation is based on BT dongles, which act as cellular base stations, and Sphero robotic balls, which are low-cost consumer mobile robots. This wireless control system allows the robots to be controlled by a single device; it is modular and scalable and offers handover and localization services typical of common cellular networks.**

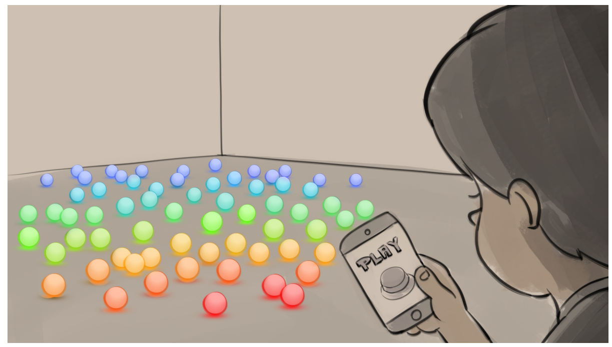

Fig. 1: Concept art (© Disney): Interactive and intuitive installation to control large number of consumer products with a single device.

I. Introduction

Personal area networks (PANs) of consumer devices are used in numerous settings. Nodes (members) of a PAN can be smartphones, smart watches, smart wristbands, other wearable accessories, or consumer robots. PANs connect these devices to one another and may also provide connectivity to local-area or wide-area networks. Commercial PANs that use low-power radio frequency (RF) such as Bluetooth to exchange data and communicate are limited to few devices. Nevertheless, sometimes it is necessary or desirable to connect more members than are directly supported by a PAN, *e.g.*, to control a large number of devices using their native RF technologies. The scenario of controlling a large network of devices is particularly appealing in case of mobility within large areas while maintaining the constraints imposed by an industry-standard PAN: no modifications of the device hardware, and operation within the power budget/level of off-the-shelf (consumer) devices.

Controlling a large number of mobile consumer devices requires flexible and scalable protocols to enable reliable communication. Such protocols must abstract from the limitations imposed by the RF standards to cope with the variable density of the network. We consider a control system for a large network of Bluetooth Classic (BT) devices as a representative of the class of such scenarios. The control system implements concepts from cellular network systems for low-cost consumer devices that have limited power communication range, slow mobility, and limited power processing units on board.

The control system builds multiple localized PANs and tracks the mobility of mobile devices. The instantaneous PAN density is kept below a set threshold of fairness, therefore, the control system can enforce handover of one or more devices from a PAN to another one.

We implemented a proof-of-concept system to control a network of *Sphero* nodes [8] and use a testbed of 20 nodes to experiment with a localization protocol based on RSSI estimates of multiple BT master devices.

II. Background

Bluetooth Classic is a wireless technology for PANs over short distances [2]. The members of a PAN form a piconet, and a BT PAN can connect up to 8 devices with a star topology with one master and up to 7 slaves that communicate using time-multiplexing. Thus, all nodes in the same piconet share the available bandwidth, so the average throughput of a piconet with a few nodes is higher than the average throughput of a large piconet. To allow nodes to join a piconet, the master initiates an *inquiry* phase (looking for available slaves) and then enters a *paging* phase (connecting to a slave node by sending a page frame).

BT Classic was not designed to handle large networks of devices [11]. Networks with more than 8 active devices can be organized into a scatternet [2] that interconnects nodes that belong to different piconets using some border nodes (connected to multiple piconets). But the protocols for scatternet formation, routing, and maintenance [4], [9] are not ideal for mobile scenarios as the network topology continuously varies, so different approach is needed.

III. WIRELESS CONTROL SYSTEM USING BT

With the goal of providing fairness and low-latency to each node, a multiple-piconet system must optimize the number of BT masters and slaves. Given a network with a fixed number of mobile nodes, the nodes of the same BT piconet equally split the available piconet bandwidth. Therefore, to maximize the throughput per node, the number of nodes per piconet should be minimized. Such a setup leads to higher number of active piconets. On the other hand, all piconets are uncoordinated and do independent frequency hopping, thus, they can interfere with one another. Therefore, the usage of fewer dongles reduces the risk of interference.

We empirically tested various scenarios of BT interfering piconets to find the best tradeoff between the number of dongles (piconet masters) and slaves. The dongles are all connected to the same computer via an USB hub, and in every test the BT slaves form a circle with radius of 50 cm around them.

	1S1D	10S10D	10S5D	10S2D
Sat.	9.007	10.26	11.34	24.7
10 Hz	8.837	8.762	10.01	25.88
20 Hz	8.676	9.464	16.95	25
50 Hz	8.445	10.93	13.73	26.29

TABLE I: Average round trip time of ping messages in piconets of different sizes (D=Dongles, S=Slaves) for different ping rates: Saturation mode, 10 Hz, 20 Hz, and 50 Hz.

The investigation consisted of measuring the quality of the network in terms of average latency of BT frames. The latency, expressed by the round trip time (RTT), measures the time that elapses from the transmission of a ping frame to the reception of the relative ACK frame.

	1S1D	7S1D	14S2D	20S3D
Sat.	9.007	27.73	28.37	28.17
10 Hz	8.837	32.13	31.57	30.01
20 Hz	8.767	29.43	28.04	27.26

TABLE II: Average round trip time of ping messages for different number of piconets for different ping rates: Saturation mode, 10 Hz, and 20 Hz. The presence of additional piconets does not affect the average RTT.

We use two ping modes: a periodic mode and a *saturation* mode, where the system waits until the response of the previous message is received before the next ping message is sent to the slave (that is a sphero robot).

Table I shows the average RTT of 1000 ping messages. The length of every ping request is 23 bytes and the ping response is 22 bytes long. If only one Sphero device is connected to the system we measure an average round trip time of about 9 ms, independently of the rate or mode we send frames. This corresponds to a maximum rate of about 110 Hz. Tables I and II show that the number of nodes in a piconet affects the RTT because transmission opportunities are multiplexed over time.

Table II shows that piconets do not influence each other thanks to the frequency-hopping procedure. The interference of multiple piconets is negligible for our purpose even if all piconets are saturated.

Empirical tests show that piconets with seven slaves can send ping frames with a maximum rate of about 35 Hz. Higher rates result in increasing frame buffers. Therefore, the RTT also increases over time.

In conclusion, we identify a threshold of about 30 Hz as maximum data rate within a piconet with arbitrary size, nevertheless, for faster systems minimizing the number of nodes per BT master is a design choice that keeps the latency low.

IV. SPHERO CONTROL SYSTEM

a) The Sphero Robot: The testbed is based on Sphero mobile robots as BT slaves [8]. A Sphero robotic ball is a small spheric robot with BT connectivity, an accelerometer, a gyroscope and LEDs. A Sphero device keeps track of the rotations of the motors, and this feature allows the device to estimate its position relatively to its starting position.

A smartphone or a computer can connect to the Sphero device and send commands or requests compliant with the Sphero API [7]. A Sphero robot may be controlled by (1) direct commands sent and immediately executed by the robot and (2) macros, which are short sequences of commands that can be loaded into the memory and triggered by direct commands. The Sphero control system presented in this paper uses *spheron* SDK, which is a framework for JavaScript applications that runs on the Node.js platform.

b) Platform: The control system (on a Linux machine with Ubuntu 14.04) uses the BlueZ BT protocol stack, which supports a up to 16 dongles (serving a max of 112 simultaneous slaves).

c) Software Architecture: The *SpheroController* provides the user with functions to discover new Sphero devices and to bind them to a specific dongle and can automatically scan and connect all the Sphero

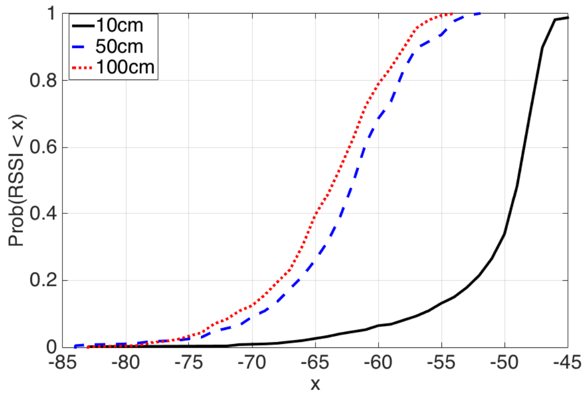

Fig. 2: Measurement of RSSI using the inquiry method.

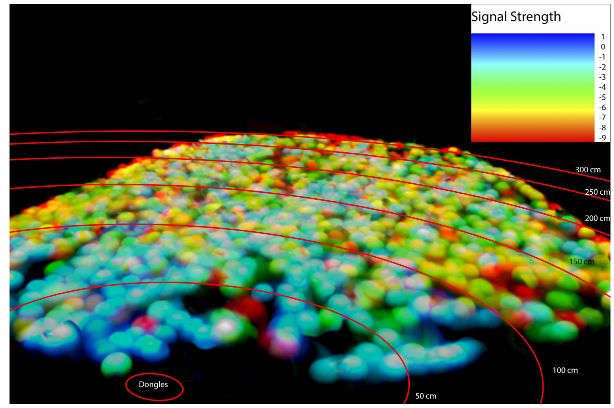

Fig. 3: A long exposure picture of moving Sphero devices that change their color according to the signal strength. The values in the legend follow the the golden receive power range scale. Blue corresponds to a strong signal whereas red indicates a weak signal.

devices found during the scan. Spheros are connected to the closest dongle. For each physical BT dongle, the software maintains a list of all Sphero objects that are bound to it [5].

d) Ranging Using RSSI: The Received Signal Strength Indicator (RSSI) indicates the signal strength of a radio signal. BT defines two alternative ways to estimate the RSSI of a master-slave radio link.

The first way can be used to measure the RSSI of an established BT connection. In connected state, an HCI (host controller interface) command of the BlueZ BT stack reports RSSI values. The result of the command is an integer value within *the golden receive power range* [10]. The second way is to use the inquiry scan method and is preferable to the HCI method because multiple masters can start an inquiry process at the same time to estimate the distance of all BT devices in the vicinity.

Figure 2 shows the distribution of RSSI values measured during the tests (about 500 values per each of the three distances) using the inquiry method. The RSSI values measured with the inquiry method shown in Figure 2 are Gaussian distributed.

Sphero robots can be programmed to light up in a color that depends on the RSSI level; Figure 3 illustrates the strength of the signal of 10 moving Spheros in an indoor lab scenario in the presence of other interfering devices operating in the of 2.4 GHz band. Every measurement averages 8 consecutive values obtained with the HCI method. While measuring the RSSI of a given location, every Sphero rotates on the z-axis to smooth the effect of the non-omnidirectional antenna pattern. In fact, we empirically observed that the relative orientation of dongles and Spheros has an impact on the RSSI value. In the figure, a blue color indicates a strong signal, whereas a red color indicates a weak signal. In this figure, the effect of multipath communication is visible which affects the measurements and subsequently leads to erroneous estimates.

e) Mapping (Smoothed Out) RSSI to Distance: The measured RSSI level is affected by multipath and the radiation pattern of the receiving antenna [6]. To range the Sphero devices, we empirically derived a log distance path loss model based on RSSI measurements. We carried on the measurements in a noisy environment with several RF systems operating at 2.4 GHz. We measured the RSSI level at different distances from a BT dongle.

f) Handing over Connection of Mobile Objects: A cellular BT network larger than 8 nodes uses multiple dongles as base stations. As in regular cellular networks, the dongles are static and located at specific positions so that their radio coverage areas partially overlap, so that mobile devices may connect to the nearest dongle and disconnect from it when the signal level degrades.

In contrast to other radio systems, BT does not natively support any handover mechanism among piconets [1]. Thus, we designed our own.

Since a BT slave can only be connected to one master, seamless handover is not an option; the connection must be torn down and then re-established with the new nearest dongle. As soon as the RSSI level of a connection drops below a given threshold, a Sphero device should disconnect from the BT dongle, scan the environment for a new closest BT dongle, and then connect to this dongle if there is enough capacity available on the new dongle. To detect the closest dongle, every disconnected Sphero device estimates its distance with respect to all available dongles.

Figure 4 shows a snapshot of mobile scenario with nine Sphero devices. The snapshot is taken in a lab space and shows the situation right after an inquiry scan. In the figure there are three dongles with different color IDs. The Sphero devices are moving toward a new location but are still connected to the dongles that

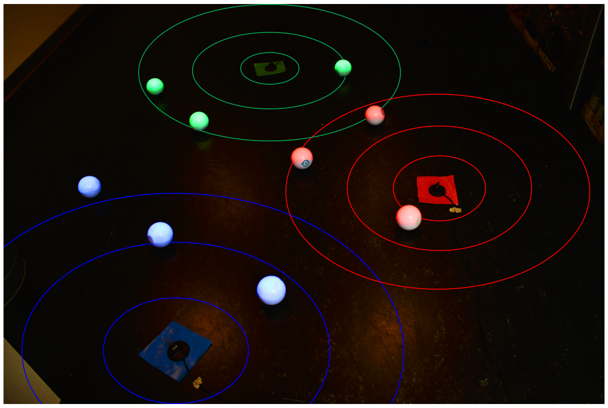

Fig. 4: Sphero devices connect to the dongle that has the strongest signal. The color of the Sphero indicates the dongle it is connected to.

measured the strongest signal.

g) Localization: Localization uses an arbitrary number of static dongles (with known location) to estimate the position of the Sphero devices. Per each Sphero device, the system trilaterates the estimated distances of the three closest dongles using the path loss model we derived.

V. Discussion

This paper describes a localization system for mobile devices using BT. The system is scalable for several reasons: (1) Dongles use inquiry scans to range the Sphero devices. The duration of the inquiry process does not depend on the number of discovered devices. (2) The current system can handle up to 112 Sphero devices because of a limitation of the Linux BlueZ stack.

The system is also modular because of the way its architecture is designed. SpheroExt modules can be extended with other modules that use different versions of BT or different physical layers to communicate.

This work focuses on observing the interference within a single piconet and among different piconets showing how local BT dynamics affect the design choices of a cellular network in terms of handover strategies.

The system presented in this paper focuses on BT classic technology, which is characterized by a slow pairing process and limitations in terms of piconet size. The low energy version of BT (BLE) saves energy by duty cycling the radio and this version supports mesh networking [3], which in practice results in larger piconets. Although BLE is more efficient than classic BT and can support more devices, the pairing process is still slow, and the control system would also rely on the slow inquiry process instead of the HCI method.

VI. Conclusions

The paper describes a wireless control system for Bluetooth (BT) devices as a proof-of-concept for networking a larger number of BT devices than can be connected in a Personal Area Network (PAN piconet). The control agent is a server that connects to a mesh network of dongles; this system has been demonstrated for up to 20 Sphero robots.

An important practical aspect is the interference of nodes (Sphero devices) within the same piconet on one another and also the interference among different piconets. Our measurement results confirm the suggestion based on theory that in large networks it is a good design choice to minimize the number of slaves per piconet to increase the piconet fairness. There are many scenarios that can employ a large number of BT devices. The approach discussed here shows that such networks are possible without any modification of the hardware of the off-the-shelf devices. Working with un-modified consumer products is crucial if we want to support the designers of applications for mobile devices, and it is encouraging so see that even a low-cost system like Bluetooth Classic allows building larger mesh networks.

References

[1] E. Amos. Mobile Communication System, May 1972. US Patent 3,663,762.

[2] Bluetooth SIG. Bluetooth Adopter Specifications. https://www.bluetooth.org/en-us/specification/adopted-specifications. [Accessed 21 July 2015].

[3] Bluetooth SIG. Bluetooth Core Specification 4.2. Available at https://www.bluetooth.org/en-us/specification/adopted-specifications, 2-Dec-2014.

[4] Chih-Min Yu and Yin-Bin Yu. Reconfigurable Algorithm for Bluetooth Sensor Networks. *Sensors Journal, IEEE*, 14(10):3506–3507, Oct 2014.

[5] G. Corbellini, L. Kuster, and T. R. Gross. Scalable control system for bluetooth mobile devices. Technical report, Disney Research Zurich and ETH, December 2015.

[6] E. Elnahrawy, X. Li, and R. P. Martin. The Limits of Localization Using Signal Strength: A Comparative Study. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 406–414. IEEE, 2004.

[7] Orbotix. Sphero API Documentation. http://orbotixinc.github.io/Sphero-Docs/docs/sphero-api/index.html. [Accessed 21 July 2015].

[8] Orbotix. Sphero Robot. http://www.sphero.com/sphero. [Accessed 20 July 2015].

[9] S. Sharafeddine, I. Al-Kassem, and Z. Dawy. A Scatternet Formation Algorithm for Bluetooth Networks with a Non-uniform Distribution of Devices. *Journal of Network and Computer Applications*, 35(2):644 – 656, 2012. Simulation and Testbeds.

[10] F. Subhan, H. Hasbullah, A. Rozyyev, and S. Bakhsh. Analysis of Bluetooth Signal Parameters for Indoor Positioning Systems. In *Computer Information Science (ICCIS), 2012 International Conference on*, volume 2, pages 784–789, June 2012.

[11] E. Vergetis, R. Guerin, S. Sarkar, and J. Rank. Can Bluetooth Succeed as a Large-scale Ad Hoc Networking Technology? *Selected Areas in Communications, IEEE Journal on*, 23(3):644–656, March 2005.