WiFiPi: Involuntary Tracking of Visitors at Mass Events

Bram Bonné, Arno Barzan, Peter Quax and Wim Lamotte

Hasselt University - tUL - iMinds
Wetenschapspark 2
3590 Diepenbeek, Belgium
Email: {bram.bonne,arno.barzan,peter.quax,wim.lamotte}@uhasselt.be

***Abstract*—To simulate crowds at mass events, realistic movement data of people is required. Despite their limited capacity for approximating real human mobility, synthetic movement models are traditionally used for this purpose. More realistic simulations can be achieved by using real-life movement data, gathered by observing people in the desired context.**

This paper presents a method for tracking people at mass events without the need for active cooperation by the subjects. The mechanism works by scanning at multiple locations for packets sent out by the Wi-Fi interface on visitors' smartphones, and correlating the data captured at these different locations. The proposed method can be implemented at very low cost on Raspberry Pi computers. This implementation was trialed in two different contexts: a popular music festival and a university campus. The method allows for tracking thousands of people simultaneously, and achieves a higher coverage rate than similar methods for involuntary crowd tracking. Moreover, the coverage rate is expected to increase even further as more people will start using smartphones. The proposed method has many applications in different domains. It also entails privacy implications that must be considered when deploying a similar system.

***Keywords*—Mobile ad hoc networks; mobile communication; privacy; simulation; wireless lan; wireless networks**

I. Introduction

Simulating the movement of crowds at mass events in a realistic manner relies on the ability to precisely tune the simulation parameters to the context of the event. Traditionally, synthetic movement models – such as the random waypoint model or the city section model – have been used for this purpose [1]. Recent studies have shown that using synthetic movement models provides only a limited approximation of real trajectories, and that more realistic simulations can be achieved by using movement data gathered by tracking people within the desired simulation context [2]. A problem often encountered with methods for gathering this type of information is that only a relatively small subset of the population can be tracked at the same time [3].

With the recent growth of smartphone usage, a large percentage of the population now carries a device frequently sending out signals which can be detected. It is estimated that over 50% of U.S. mobile subscribers own a smartphone [4]. This number is only expected to increase over time as more than 1.3 million new Android devices are activated worldwide every day [5]. Modern smartphones have Wi-Fi communication enabled by default, allowing their owners to be detected at any moment by scanning the ether for Wi-Fi packets.

Involuntary tracking provides a significant advantage compared to other techniques for tracking where the subjects need to actively cooperate, either by carrying a specialized tracking device or by actively and willingly sharing information about their location. Systems like the one described in this paper do not require user consent and are therefore capable of tracking a much larger sample set of the population.

This paper makes the following contributions. First, we describe a mechanism for tracking visitors at mass events which makes use of Wi-Fi technology. We explain how this mechanism works at a high level, and continue by discussing its implementation. Next, we describe how the detection mechanism was and is being used in two different contexts to infer movement patterns from visitors. One of these contexts is a three-day international music festival attracting 100 000 visitors every year. The other is a university campus, where we have been tracking students and staff for a period of three months. Section VI provides an overview of possible applications of the tracking method in different domains. In section VII, privacy implications for this technology are discussed. We conclude and present an overview of future work in section VIII.

II. Related work

Several techniques have been proposed over the years to accomplish location determination and to provide the ability to track the movement of objects and people. One of these techniques is the use of wireless sensor networks (WSNs), where tiny motes are attached to objects to achieve tracking abilities [6]. Although a high degree of precision can be obtained, the disadvantages of such a method are clear: the cost of motes is non-negligible, they are not available off-the-shelf and, as such, the number of objects that could be tracked is limited in practice. A WSN system requires active consent and participation from users to carry the motes. Alternatively, the use of tracking applications (sometimes as part of applications providing other functionality) on more common hardware equipped with GPS sensors has been proposed. Although this approach solves the cost issue to some degree, users still need to actively collaborate in the tracking by installing an application and agreeing to be tracked (unless the tracking software is part of malware, a topic which is not discussed

978-1-4673-5828-6/13/$31.00

here). Given the fact that the goal is to design a non-obtrusive solution that requires no active consent from people being tracked, these methods are not applicable.

Versichele et al. [3] developed a method for tracking people at mass events which uses Bluetooth signals sent out by mobile phones to detect a person's location. While similar to the approach described in this paper, this detection method requires phones to have their Bluetooth functionality set to 'discoverable', a feature which is disabled on modern smartphones for security reasons [7], [8]. Because of this limitation, Bluetooth tracking can only be used to track older generation cell phones, resulting in a coverage rate of around 8% of the population. We expect this coverage rate to further decrease as more people switch from older cell phones to smartphones. Our approach on the other hand requires only control signals sent out as part of the 802.11 protocol, which are required for Wi-Fi communication to function properly. This tracking method is future proof because unlike Bluetooth, Wi-Fi is enabled by default on modern smartphones. Furthermore, our method does not depend on a smartphone being put into a discoverable mode, nor does it require the smartphone to be actually connected to a wireless network. The tracking method of Versichele et al. can be used complementary to our own method to allow for tracking visitors using both Bluetooth and Wi-Fi signals. Using this combination, both older cell phones and smartphones can be tracked, which may provide a coverage rate close to the sum of the individual coverage rates for both tracking methods.

Other work has been done in the area of using Wi-Fi communication to obtain information (besides location) about smartphone users. Cunche et al. have used passive Wi-Fi monitoring [9] to derive social links between smartphone owners. Moreover, Rose et al. [10] have shown that SSIDs found in 802.11 probe requests can be used to produce a list of locations a smartphone user has visited. The described technique works by looking up the broadcasted SSIDs in the WiGLE wardriving database [11]. This database contains the locations of different wireless networks all over the world, submitted by users, and identified by their SSIDs. Our approach differs from the work done by Rose et al. in that it allows for tracking users without the presence of an infrastructure of access points (with specific SSIDS). Rose's solution also is unable to infer the time at which a device was at a certain location. Similarly, Becker et al. [12] have described how cellular telephone networks can be used to study human mobility on a large scale. This, too, measures mobility on a much larger, less granular scale than the technique described in this paper.

Network simulations use either synthetic movement data or data gathered from real-life crowd tracking, with a preference for the latter [2], [1]. The CRAWDAD database [13] is a resource containing wireless trace data from many contributing parties. Methods for acquiring this data vary from equipping people with sensors to using network traces from access points which are under the control of researchers. We believe that our method can provide substantial benefits for people willing to extend the CRAWDAD database, by allowing for tracking of visitors without requiring active cooperation.

III. Detection mechanism

In this section, we lay out the structure of the mechanism used for tracking smartphone owners. We first describe which elements of the 802.11 standard enable us to detect devices on a single location, and continue with an overview of how we use these detection techniques to track a device across different locations.

A. Detecting a device

In order to be able to detect a particular device that enters a detector's range, the detector needs to scan the ether for packets that have the following characteristics:

- In order to be able uniquely identify a device, the captured packets should contain the hardware (MAC) address of the device's Wi-Fi interface.
- To make sure that all devices in range are detected, the packets must be sent out regularly by the device.

The IEEE 802.11 standard [14] describes three types of packets that possess these properties:

- *Probe Requests* are used by Wi-Fi devices to scan for known access points. When a Wi-Fi device is not connected to an access point, it will send out these requests at regular intervals to search for access points it has been connected to in the past, which are remembered by the device. Moreover, our tests show that modern smartphones by Samsung, Apple, HTC and other popular brands also send out probe requests regularly even when connected to an access point, presumably to find access points with a stronger signal or with a higher priority than the currently connected access point.
- An *Association Request* is sent by a Wi-Fi device when it wants to connect to a certain access point. These requests contain information about the client (e.g. supported data rates) and request the access point to allocate resources for serving this particular client.
- A *Reassociation Request* is sent when a Wi-Fi device wants to connect to another access point on the same network. This is the case for example when the Wi-Fi device is roaming and has detected an access point with a stronger signal serving the same network. Reassociation requests can even be triggered by any station present on the network. Indeed, by sending out a *disassociation frame*, a station is able to request all associated clients to disconnect from the network, effectively forcing them to reassociate afterwards. Moreover, disassociation frames can be easily forged by a third party [15], causing every connected device to reassociate.

None of the three types of packets described above are sent out continuously. We define a *detection round* to be the minimum time interval for which we can be reasonably sure that a device sends out at least one of these three types of packets. The ideal duration of a detection round was empirically determined to be 130 seconds: each of the tested Wi-Fi devices (including multiple Android devices, notebooks,

an iPhone, and an iPad) sends out a probe request at least once every two minutes, regardless of whether it was connected to a wireless network.

We refrain from using a mechanism which analyzes every possible type of 802.11 packet because capturing and processing all packets would introduce a great computational strain on the detectors, while adding little benefit. Indeed, processing every large data packet at the user space level instead of dropping it at the level of the network interface could very easily overload low-cost, low-power detectors. Furthermore, to ensure completely transparent and non-obtrusive operation, our mechanism does not use disassociation frames to force reassociation of devices.

Channel hopping techniques could be used to capture packets on all different 802.11 channels. While channel hopping would allow a detector to detect (re)association requests on all different channels, the fact that the radio is tuned into a single frequency band for only a short period at a time has a negative impact on the number of complete probe requests detected (probe requests are sent on all channels). Empirical tests have shown that the cost of channel hopping does not outweigh its advantages (i.e. fewer devices are detected in total) and it is therefore not used in the proposed solution.

B. Tracking the location of a device

Using the device detection method described above, a system which tracks the movement of smartphone users can be created by dispersing multiple detectors over the coverage area. The location of a specific device – and thus, its user – can then be determined by keeping track of the different times at which each detector detected a packet originating from the device's MAC address.

An optimal tracking setup considers the placement of the detectors as well as the range of the antennas to cover an area that is as large as possible. The latter can be tuned by opting for directional (beam-type) or omni-directional antennas with a specific gain factor. Detection ranges of individual detectors are allowed to overlap: both detectors could then be used to establish a more precise location of the detected device.

It is a requirement that the clocks on the different detectors are at least loosely synchronized, in order to be able to correlate data from the detectors afterwards. Alternatively, the detectors could have their logging information sent to a server immediately. The server could then be held responsible for correctly synchronizing the data from different detectors.

By correlating data from different detectors over time, a path can be established for every visitor. By taking into account physical properties of the tracked area, such as blocked paths and distance between detectors, more granular paths can be inferred. Moreover, in case of only one entrance and/or exit, it can be determined when a visitor enters or leaves the tracked area.

Additionally, to further increase location determination precision, the RSSI value – which indicates the received signal strength for a received 802.11 packet – could in theory be used. From this value, an estimation could be made on the distance between the detector and the device sending out the packet. However, empirical tests have shown that the RSSI value is of little use in crowded environments containing a high amount of electronic devices and people due to severe fluctuations and noise in the data sets. Because of these environmental factors, the RSSI value is currently not used in the detection mechanism. Rather, the overlapping range of the individual detectors provides a similar, but more consistent result.

Fig. 1. The Wi-Fi detector, consisting of: (a) a Raspberry Pi, (b) a Wi-Fi dongle, (c) an external long-range antenna, (d) a USB hub, (e) a Nokia N78 phone, and (f) a heartbeat LED.

IV. Implementation

The detection mechanism as described above was implemented on a Raspberry Pi computer. The Raspberry Pi was the hardware of choice because of its low power requirements (3.5W), its small size, and its very low cost (under US$35). A USB hub was attached to the Raspberry Pi for power and expansion, and a Wi-Fi dongle supporting monitor mode was used for capturing packets. An external antenna was attached to the Wi-Fi dongle to increase the detection radius. Either a cell phone or an ethernet cable was used for network communication, depending on the facilities at the tracking site[1]. Lastly, an LED was connected to the Raspberry Pi's GPIO pins to provide a status indicator, displaying whether or not the detector was a) scanning, b) connected, and c) aware of the correct local time[2]. The result can be seen in Figure 1.

The Raspberry Pi was running Raspbian, a Debian-based GNU/Linux distribution specifically tailored for use with the Raspberry Pi. The detection software was implemented in Python, making use of the `scapy` packet capturing and manipulation library [16]. This library was chosen because it allows for filtering of packets at the kernel driver level by making use of the Berkeley Packet Filter (`bpf`), while still allowing for easy interaction with captured packets at the user space level. This combination ensures that the scanner software

[1]Note that a network connection is not required for the detector to function correctly. A network connection was used to display a real-time overview of the gathered data.

[2]Since Raspberry Pi's do not have a real-time clock (RTC), the time for the Raspberry Pi is reset at boot time. For providing the Raspberry Pi with the correct time, we used `ntp` when a network connection was available. For situations in which no network connection was available, a custom-made Android application was created which could set the time remotely via broadcast Wi-Fi packets.

is as lightweight and speedy as possible, while still being able to extract useful information from packets on-line, at the scanner itself. Because of these optimizations, the detectors are able to process more than 4 000 detected Wi-Fi nodes per detection round in real time.

A list of detected devices was sent to a central server after every detection round in order to both provide a failsafe logging mechanism and to be able to gather statistics in real time. These real-time statistics include information such as the crowd density at different detectors, information about visitors' devices (manufacturer, broadcasted SSIDs), and spatio-temporal information about the visitors. An example of the dashboard displaying part of this information can be found in Figure 2.

V. Experimental setup

Two experiments were performed using the detector software. The first experiment consisted of placing the detectors at a three-day international music festival. For the second experiment, detectors were placed at the university campus, tracking visitors for over three months.

A. Pukkelpop 2012

Pukkelpop is a Belgian music festival attracting 100 000 visitors[3] every year. The 2012 edition spanned three days, from August 16th to August 18th. During these days, fifteen detectors were placed at strategic locations, ranging from the 8 different stages to important passageways. Three detectors contained a cell phone for real-time monitoring. The total area of coverage was about 400m by 500m.

Combining the data from all fifteen detectors, a total of 137 899 unique devices (MAC addresses) were detected. These detections also include some devices not belonging to festival visitors. For example, some detections might have resulted from devices in passing cars or from devices that are part of the fixed infrastructure. For this reason, we included in the final dataset only those devices which were detected at at least two different locations, at different moments in time. Filtering out the devices conforming to this requirement, we find a total of 29 296 detected devices, giving us a relatively good estimate of the number of people carrying a Wi-Fi-enabled device at the festival (29.3%). The 29 296 devices account for a total of 307 256 data points, spread out over the three days of the festival.

It must be noted that the previous numbers establish a lower bound, not only because it is expected that smartphone usage will increase over time, but also because technical difficulties occurred during the first experiment. Indeed, during this experiment, power failures were common, and both the detector software and the Raspberry Pi still suffered from childhood diseases. These problems caused the detectors to sometimes malfunction, hindering them from detecting some devices. Another reason that these numbers establish a lower bound is our requirement that devices should be detected at at least two different locations. Because of this, rather stationary visitors are not part of the results.

[3]The Pukkelpop 2012 festival attracted 100 000 unique visitors over the course of three days (number was provided to us by the Pukkelpop organization).

B. University campus

To demonstrate the versatility of the system, detectors were also placed at the Hasselt University (Universiteit Hasselt) campus. Besides being an indoor location, this scenario also differed from the one above in the fact that more long term monitoring (3 months+) was required and the coverage area was smaller with more overlap between detectors to increase accuracy. The main campus building of Hasselt University typically has around 3 200 daily visitors, consisting of students, staff and other guests.

For this experiment, four detectors, all of them provided with a network connection, were placed both at the main building itself and at the Expertisecentrum Digitale Media (EDM, Hasselt University's multimedia research center). Two of the detectors were placed relatively close to each other. This way, it was possible to cross-check data from those detectors, allowing us to verify that devices were either tracked by both detectors or not detected at all. Over a period of three months, 16 486 devices passed within the range of at least one detector. In total, the devices were detected a total number of 4 486 310 times.

On average, 1 383.4 unique devices are detected at the main campus per working day. Assuming that every visitor carries exactly one switched on Wi-Fi enabled device[4], it can be concluded that the detectors provide a coverage rate of around 40% for this scenario.

VI. Applications

Using low-cost hardware for tracking people offers a wide variety of applications, of which we give some examples in this section. We emphasize that this is only a small subset of many possible use cases.

A. Real-time crowd management

An interesting application for organizers of mass events is to use the real-time gathered data for crowd control. Similar to the dashboard (pictured in Figure 2) that was used for visualizing crowd data during our experiments in real time, one could visualize the real-time data in a way that shows the flows of people moving, and provides additional information on crowd density compared to the maximum capacity in a particular location. Moreover, a visualisation of real-time data could prove to be vital in an evacuation scenario where the goal is to get people to move to – or away from – a certain location as fast as possible.

Furthermore, the data can be used after-the-fact to gather some interesting statistics about visitor behavior at music festivals, for example:

- Which artists or stages are most popular and which artists attract a similar audience?
- Which (unforeseen) crowd movements happen on the terrain over the duration of the festival (due to unplanned events or scheduling issues between the stages)?

[4]Note that a visitor may be carrying more than one switched on Wi-Fi enabled device (notebook, smartphone, tablet), or that he/she may not be carrying a smartphone at all.

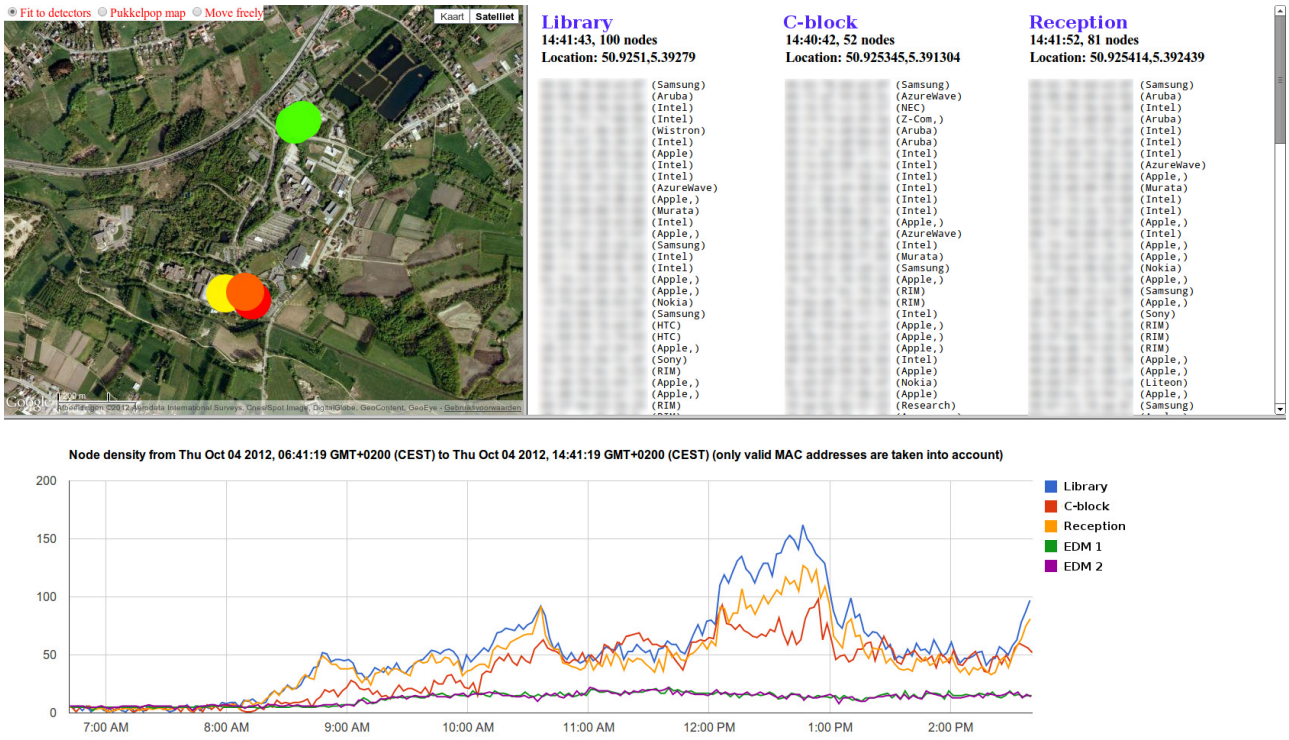

Fig. 2. The dashboard showing information from the second scenario (university campus). Left: a heatmap showing the density around the different detectors; right: the MAC addresses of the currently detected devices (blurred out), together with the device manufacturers; bottom: a timeline showing the number of detected devices per detector over time.

- How stationary are visitors? How much time do they actually spend on the festival site?

This gathering of statistics does not only apply to music festivals, but also to places like shopping centers. There, the data could be used for example to monitor the shops in which customers spend most of their time, or to oversee queueing times at the cash registers.

B. *Mobility models for simulations*

Opportunistic, multi-hop networks deal with using ad hoc communication to provide network connectivity among different devices in a local area. A possible application for this technology is a mass event, where conventional cellular networks are likely to become overloaded due to the high amount of visitors.

In simulating opportunistic networks, it is essential that the simulation runs are performed using realistic movement patterns [2], [1]. Because of this, real mobility data of visitors at a mass event can provide invaluable information for creating a realistic simulation. The dataset acquired at the Pukkelpop festival was converted to two different types of mobility traces: one that could be used by the ONE opportunistic network simulator [17], and one that could be directly used in simulations run by either ns-2 or ns-3 [18]. We discuss this further in section VIII.

C. *Ubiquitous computing*

In the domain of Ubiquitous Computing, a low-cost Wi-Fi detector could be used to infer which people are currently present within a certain room, and to tune the atmosphere accordingly. A visitor to a room can have preferences for certain types of lighting or genres of music. Furthermore, user interfaces can be tweaked to accommodate a user's preferences, or the user could automatically be logged in to certain services. It must be noted that the maximum detection interval of 130 seconds may be too high in such a scenario. In this case, techniques such as disassociation requests (see section III-A) could be used to speed up the detection of a device as a wireless infrastructure using access points is likely to be present.

A Wi-Fi detector could also be used to save energy in rooms where no one is present. Assuming that every visitor carries a smartphone, if it is detected that all devices that were previously present have now left the room, lights and other appliances could be turned off automatically.

Lastly, as described by Rose et al. [10], information from probe requests can be used to infer past location data from users, allowing for user profiling. This user profiling could aid for example in automatic language selection, choosing the user's language based on the locations over the world he/she has visited most often.

VII. Privacy implications

Clearly, tracking people via their smartphones brings about some privacy implications. The most obvious one is that when the MAC address of a person's smartphone is known, it is easy to reconstruct the complete path this person has travelled.

Because MAC address information needs to be shared among different detectors for tracking purposes, it is not

possible to solve this problem simply by associating a unique identifier to every MAC address in each individual detector.

A naive solution might consist of creating a one-way hash of every MAC, in order to obfuscate the MAC addresses, while still making sure that the identifier would remain the same over different detectors. It would then however still be possible to track a certain MAC address by calculating its one-way hash. Thus, care must be taken that the data is anonymized in some other way (e.g. by associating a random number with every MAC address) *after* it has been combined from all individual detectors.

However, even if the MAC address of a person's smartphone is not known, it is still possible to derive information from the captured Wi-Fi probe requests alone. For example:

- The list of known SSIDs is available as part of the probe request. From this list we can derive other networks the user has connected to, which may include e.g. the SSID of the home network, the SSID of places visited, or even the user's personal name (as part of the SSID of the user's home network or mobile hotspot).
- The manufacturer of a person's smartphone, which can be derived from the first part of a smartphone's MAC address, can be used to identify that person. To illustrate how, consider the scenario where the visiting times for a specific person at certain places are known. The list of devices detected at that time can then be reduced to a list of devices matching that person's smartphone manufacturer, which makes it easy to derive the MAC address of the smartphone.

Moreover, de-anonymization of the dataset is possible when some other information is known. Indeed, relationship graphs in social networks can be compared to devices travelling together amongst different detectors in order to infer real-world identities from the mobility dataset. This is analogous to previous work done by Narayanan et al., wherein original identities are derived from anonymized social network graphs [19].

VIII. Conclusion and future work

We have shown that tracking of visitors at mass events can be achieved at a very low cost and – more importantly – unobtrusively and without requiring active cooperation. The proposed method can easily be tailored to suit various contexts of use, demonstrated by the two scenarios presented. A number of possible applications have been discussed, along with some privacy implications that should be kept in mind when using the proposed solution.

It would be interesting to compare the proposed method to actual GPS mobility traces collected by having visitors actively cooperate. From this, both the accuracy of the detection method and the value of interpolation between multiple detectors could be inferred.

The data acquired at the Pukkelpop festival is part of a project in which we aim to develop a smartphone application which allows opportunistic (ad hoc) communication at mass events such as music festivals as an add-on to infrastructure-based networks. The gathered data is currently being used in network simulation experiments, in which the optimal opportunistic routing protocol for use at mass events is determined.

Acknowledgements

We would like to thank the Pukkelpop organization for allowing us to perform experiments during the festival. Thanks also go out to the UGent CartoGIS research group.

References

[1] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[2] N. Aschenbruck, A. Munjal, and T. Camp, "Trace-based mobility modeling for multi-hop wireless networks," *Computer Communications*, vol. 34, no. 6, pp. 704–714, May 2011.

[3] M. Versichele, T. Neutens, M. Delafontaine, and N. Van de Weghe, "The use of Bluetooth for analysing spatiotemporal dynamics of human movement at mass events: A case study of the Ghent Festivities," *Applied Geography*, vol. 32, no. 2, pp. 208–220, 2012.

[4] Nielsen, "Americas New Mobile Majority: a Look at Smartphone Owners in the U.S." 2012. [Online]. Available: http://blog.nielsen.com/nielsenwire/online_mobile/who-owns-smartphones-in-the-us/

[5] L. Spradlin, "Eric Schmidt Reveals New Activation Numbers," 2012. [Online]. Available: http://www.androidpolice.com/2012/09/05/eric-schmidt-reveals-some-new-activation-numbers-1-3-million-android-devices-activated-each-day/

[6] H. T. Kung and D. Vlah, "Efficient location tracking using sensor networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, 2003, pp. 1954 –1961 vol.3.

[7] Android Open Source project, "Bluetooth - Android Developers," 2013. [Online]. Available: http://developer.android.com/guide/topics/connectivity/bluetooth.html#EnablingDiscoverability

[8] Apple Inc., "iOS: Third-party Bluetooth headsets, headphones and keyboards," 2012. [Online]. Available: http://support.apple.com/kb/ht1664

[9] M. Cunche and R. Boreli, "I know who you will meet this evening! Linking wireless devices using Wi-Fi probe requests," *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–9, Jun. 2012.

[10] I. Rose and M. Welsh, "Mapping the urban wireless landscape with Argos," *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10*, p. 323, 2010.

[11] "WiGLE: Wireless Geographic Logging Engine," 2013. [Online]. Available: http://wigle.net

[12] R. Becker, R. Cáceres, K. Hanson, S. Isaacman, J. M. Loh, M. Martonosi, J. Rowland, S. Urbanek, A. Varshavsky, and C. Volinsky, "Human mobility characterization from cellular network data," *Commun. ACM*, vol. 56, no. 1, pp. 74–82, 2013.

[13] J. Yeo, D. Kotz, and T. Henderson, "CRAWDAD: A Community Resource for Archiving Wireless Data at Dartmouth," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 21–22, Apr. 2006.

[14] IEEE Standards Association, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," in *IEEE Standard for Information technology*. New York: IEEE Computer Society, 2012, vol. 2012, no. March, ch. 11.

[15] C. He and J. C. Mitchell, "Security Analysis and Improvements for IEEE 802.11i," in *The 12th Annual Network and Distributed System Security Symposium (NDSS'05)*, Stanford, 2005, pp. 90–110.

[16] P. Biondi, "Network packet forgery with Scapy," Talk at PacSec, 2005.

[17] A. Keränen, J. Ott, T. Kärkkäinen, and A. Ker, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. New York, NY, USA: ICST, 2009.

[18] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, "ns-3 project goals," in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, ser. WNS2 '06. New York, NY, USA: ACM, 2006.

[19] A. Narayanan and V. Shmatikov, "De-anonymizing Social Networks," in *30th IEEE Symposium on Security and Privacy*, Berkeley, CA, 2009.