Peer provided cell-like networks built out of thin air

Andrés Arcia-Moret, Zafar Gilani, Arjuna Sathiaseelan, Jon Crowcroft
Computer Laboratory
University of Cambridge
Cambridge, Cambridgeshire CB3 0FD, UK
Email: firstname.lastname@cl.cam.ac.uk

Abstract—The success of Wi-Fi technology as an efficient and low-cost last-mile access solution has enabled massive spontaneous deployments generating storms of beacons all across the globe. Emerging location systems are using these beacons to observe mobility patterns of people through portable or wearable devices and offer use-cases that can help solve critical problems in the developing world. In this paper, we design and develop a novel prototype to organise these spontaneous deployments of Access Points into what we call virtual cells (vcells). We compute virtual cells from a list of Access Points collected from different active scans for a geographical region. We argue that virtual cells can be encoded using Bloom filters to implement the location process. Lastly, we present two illustrative use-cases to showcase the suitability and challenges of the technique.

I. INTRODUCTION

Due to recent advances in technology that allow mobile devices to find geolocation fairly accurately, there has been tremendous growth of interest in developing and exploiting location-based services. These services enable users to share their activities and happenings in real-time, since they are integrated with social networking applications. Popular examples include Facebook Places, Foursquare and LevelUp.

In this paper, we present an initial work that proposes to use the simple active scanning process in Wi-Fi to organise spontaneous deployments of Access Points (APs) in urban areas into what we call virtual cells (vcells). A vcell corresponds to a collection of lists of APs produced by progressive scanning on the move. Moreover, each vcell is computed per geographical region, typically small in size or area. The size of the vcell varies and depends on the speed of collection and the particular device performing the scanning. There are several reasons responsible for this variation such as the antenna, the radio, the transmission conditions or the full scanning time (i.e., total time spent on actively receiving beacons [6]). We discuss a mechanism that uses Bloom filters (BF) [18] to represent vcells. This mechanism is intended to locate conveniently mobile devices given a Bloom filter containing very few or approximate fingerprints1. Our intention is to rely on a preliminary process of collection of the ground-truth for geolocation by using a GPS. Then we build a compact and fast database indexed by BFs that efficiently maps vcells into locations.

Various commercial enterprises offer many widely accessible geolocation services. For example, Google’s Android API uses the location from a GPS or the network service provider to determine device’s approximate location. Similarly, iOS API uses geotagged locations of nearby cellular towers or Wi-Fi hotspots, along with its crowdsourced database of Wi-Fi hotspots, to determine device’s approximate location. Commercial Web services such as Google Maps Geolocation API use MAC addresses and many other metrics (e.g., IP, age, channel, signal strength, etc.) to provide a location. Our proposed technique uses fingerprints of well-known locations of Wi-Fi hotspots to approximate a device location to a reference point or a landmark. Different from mentioned services, our proposal is intended to be open source, self-contained and crowd-sourced service.

The novelty of our technique consists of (1) grouping APs found in spontaneous deployments into cell-like network topology to assist a location service, (2) whereas other location prediction techniques use passive scanning, uncontrolled active scanning, or full connection information from Layer 3 [1], [2], [3], [4], [5], vcells are built from controlled Layer 2 active scanning considering the dynamics described in [6], (3) vcells can be duly encoded into BFs for facilitating the location retrieval, (4) while both GPS and other services mentioned above can only work outdoors, vcells can also offer a location service for indoors.

Potential vcell use-cases are listed below.

Need for localisation amidst disasters. Internet connectivity during extraordinary circumstances cannot be trusted. However, we may have islands of Wi-Fi APs that can be utilised by a stand-alone application to help people locate guidance maps, safe areas, rendezvous points, etc.

Human tagged locations. Naming and description of the addresses are not formalised and may be subject to whims of domineering authorities in developing regions. They may also be under-reported on common services such as Google Maps, either because of lack of a business case or feasibility to deploy Google street view cars. It is also normal to have nameless streets, short-cuts, by-passes, thus making vcells a mean to have an ad-hoc common agreement based on a mutual consensus of the crowds for approximate location.

Mitigate the lack of cellular coverage. With the proliferation of IoT devices and community networks that provide WiFi access, it becomes convenient to deploy APs that may just act as location anchors. They are low-cost, energy-efficient (compared to a GPS solution), and can even be deployed with

1 A fingerprint is a list of access points collected after a single scan.
a small sustainable source of energy (e.g. solar panels).

The rest of the paper is divided into following sections. Section II describes three main components of the prototype system. We evaluate the vcell algorithms in two illustrative journeys and propose the applicability of Bloom filters for compact representation in Section III. Section IV describes related work in this field. Finally, we conclude the paper in Section V.

II. SYSTEM DESIGN

The system prototype considers three main components: the mobile application, the vcell forming algorithms and the Bloom filter based location. To create vcells, the mobile device collects topology fingerprints and progressively aggregates scans based on a commonality metric.

General Design. To collect our datasets we used Hunter\(^2\), an Android mobile phone application that scans WLAN topologies. We used this application to store Wi-Fi AP data and use those data points to map APs into vcells to predict places of mobility and interaction.

The core of the application is composed of helping modules that support the vcell calculation, namely the GPS location (for providing the ground truth), Wi-Fi scanning module and path approximation algorithm (to adjust the geo-position based on existent maps). From a user’s perspective, a mobile can collect the surrounding WLAN topology to interact conveniently with a central service (e.g., a Topology Manager, see [19], [20] for further information).

By default, the scanning module is meant for two main purposes: topology discovery and keeping alive a session between a mobile and its serving AP. In our particular case, the scanning function can be conveniently and continuously invoked (and with different invocation frequencies) to obtain different perspectives of the surrounding topology. Since the full-scanning time varies on every mobile device, we suppose a central service that helps to build vcells (we discuss a similar service in [19], [20]).

vcell algorithm. We cluster or group APs by AP detection frequency. The vcell formation algorithm is defined in Algorithm 1. The cell size, i.e., how many scans form a cell, is controlled by the cell condition (CC). The CC is an adjustable commonality metric that defines the quantity of overlapping APs within a vcell. So, if the intersection between the current scan and the tentative vcell (curr_cell) satisfies the CC, then the scan will be part of the curr_cell. Essentially, the unique() function ignores duplicates between curr_cell and scan. If the CC is not satisfied then the curr_cell is appended to the vcellList, and a new vcell will get formed during next iterations. The algorithm terminates when all vcells are defined. Note that one AP can be a part of many cells, thus forming intersection areas.

Algorithm 1 vcell formation

```
Require: APs scan logs (scanList)

vcellList ← vcell from first scan of scanList
curr_vcell ← vcellList.pop()
for each scan in scanList do
  if (scan ∩ curr_vcell) ≥ CC * len(curr_vcell) then
    curr_vcell ← unique(curr_vcell ∪ scan)
  else
    vcellList.append(curr_vcell)
    curr_vcell ← scan
  end if
end for
```

Algorithm 2 Define overlap between vcells

```
Require: vcellList
for i in vcellList do
  for j in vcellList do
    if (vcellList[i] ∩ vcellList[j]) is bounded by OC then
      overlapList ← (vcellList[i] ∩ vcellList[j])
    end if
  end for
  vcellList[i].append(overlapList)
end for
```

Bloom filter assisted device location discovery. Two requirements are justifying this component: (1) simplicity in the implementation of a location discovery service, and (2) a CPU and energy efficient search of a fingerprint in the vcell database. A BF is an ideal data structure to cover these requirements because a BF can be quickly built out of a small number of scans, then matched up with an established vcell. However, false positives (FP) are an inherent characteristic of BFs. In our particular case, during a journey when vcells have already been formed, an FP can be detected if the fingerprint starkly differs from previous vcell locations. These include abrupt jumps in location or insensible fingerprint information. Hence, we can proceed to correct such instances. According to the probability equation of false positives \( p = \left(1 - \frac{1}{m}\right)^k \), a BF

---

\(^2\)Hunter App is an Android application developed at the University of Los Andes by Andrés Arcia-Moret and Jose Marquez, available at goo.gl/6QHUIR
We have performed one collection in the downtown of Mérida
city in Venezuela, at walking speed (5 kph) in which we have
found 1826 APs for a 5 km journey. And a similar one in the
Alexandras Avenue in Athens, Greece at running speed (10
kph) in which we have found 687 APs for 15 km journey. Both
measurements were performed during June 2015 and using
high full-scanning times. Collected lists are conveniently passed
to the central service by doing delay-tolerant uploads as in [9].
As expected, the distance between two different consecutive
scans is more spaced when the speed of collection is higher.

Fig. 1 presents different cell conditions produced by Algo-

rithm 1 for the journey in Athens. One possible application for
different \(v_{cells}\) sizes corresponds to the refined location. As the
condition gets more relaxed, for example as depicted in
Fig. 1 (left), cells become bigger, and it is easier to find certain
fingerprint inside the cell. This rough approximation could be
refined through other less relaxed conditions, such as depicted
in Fig. 1 (right). Varying the precision of the cells could have
multiple uses, a location of a mobile could be more precise
when using smaller cells. Actually, from the collected datasets
one can see that small cells (for both speeds of collection)
approximate the location of a fingerprint to a few dozens
of meters, much like a GPS position.

On top of the previous application, we could infer the
mobility pattern of an individual device. Depending on the
needs we could approximately indicate whether a mobile is
going through a particular sequence of cells, or (using more
relaxed conditions) the mobile is indeed positioned within
a large \(v_{cell}\). This information is useful in use-cases such
as estimating the dynamics of disease propagation, being
especially helpful in developing regions.

Fig. 2 illustrates the case for a cell overlapping. Cell
overlaps is an indication of the continued coverage provided
by the spontaneous deployment within the urban area. We
have tested both relaxed and more strict cell conditions, and
results have been similar. There are always overlapping scans
whenever a cell condition is relaxed. For the case shown in
Fig. 2, we have set the cell condition to 0.30. The black
markers show a common subset of scans between two neighbor
cells indicating a transition when walking from one cell to the
next. The common subset of scans reports an overlap condition
between 20% and less than 30% of the quantity of APs shared
among the two cells.

There are a few inherent shortcomings of the system that
should come to light. The scanning process during data collec-
tion is challenged by various obstacles, depending their shape
and size. Typically obstacles include people, vehicles, and
buildings. All these elements compose particular environments
that alter scanning results. We have also observed that, in more
open spaces, i.e., having fewer obstacles, \(v_{cell}\) sizes tend to be
bigger for a fixed cell condition. Moreover, at a higher speed of
collection (but at the same scanning frequency), \(v_{cells}\) tend to
be less dense, i.e., it results in a fewer APs per fingerprint.
However, for our test cases, Algorithm 1 always builds a
continuum of \(v_{cells}\).

IV. RELATED WORK

Mobility has been widely studied using geo-localised cel-

lular connectivity data for a variety of different reasons [10],
[11], [12]. While cellular data covers macro regions and is
extremely rich in spatiotemporal information, Wi-Fi data is
confined to micro-regions and can provide better insights into
problem spaces that cannot be addressed using cellular data.

In [1] experiments recording up to 6 months of human mo-
bility data have been conducted, with high temporal resolution,
finding a strong correlation between simple Wi-Fi scanning
traces and human location. Their results encourage the use
of Wi-Fi positioning through simple scanning wardriving due
to the inferred high presence (on crowded Wi-Fi places) of
humans in their sample. Although there exist other studies
about Wi-Fi wardriving and location prediction systems [3],
[2], they do not address any mechanism to convert scans
systematically into meaningful positions.

Location prediction through Wi-Fi network has been studied
from many perspectives, but to the best of our knowledge,
one of them address the problem of clustering APs into
\(v_{cells}\) as we discuss in this article. [4] uses Wi-Fi AP location
and characteristics along with GSM to predict the speed of
the mobile device. They have studied the duration of the
strongest Wi-Fi access point perceived. In their particular case, authors demonstrate that different degrees of mobility (dwelling, walking or driving) can be inferred through the collection and post-processing of Wi-Fi and GSM beacons. [5] uses Layer-3 location prediction by processing Wi-Fi traces. They present an analysis based on Markov models on users location from extensive access point interactions. They define a single location of a user as an established connection (i.e., scanning, authentication, and association, and later disconnection) to a single AP during a certain period. They rely on user attachment and detachment traces on the confined network from which authors have complete access to the extensive logs.

Traceability through Wi-Fi technology has been investigated from several end-point perspectives. In [13], the authors propose a cooperative system called Argos capable of distributively collect Wi-Fi traffic in an urban area. A group of geographically distributed sniffers is instructed to be aware of a desired client unique address, and after the required traffic is collected, all intervening sniffers pass the information to a central server which in turn infers about the mobility patterns. Finally, there has been a considerable amount of literature claiming the use of the received signal strength (RSS) as the primary indication of the position of a mobile [14], [15], [16]. However, these approaches refer to a physical layer procedure based on propagation models to infer the approximative location of a particular device. There is a broad range of variations of this location prediction mechanism and it is tough to follow a single one that fits all devices. Moreover, the RSS depends entirely on the receiving antenna and radio so that the result can vary within two different mobiles in the same location. Furthermore, to improve the estimation of the position, generally large amount of data is required, thus increasing the energy consumption [17].

V. CONCLUSION

We have introduced a novel technique to organise APs found in spontaneous deployments into what we define as *vcells*. Different from existing Wi-Fi location techniques, *vcells* are built from progressive Layer 2 active scanning. We have explained *vcells* and their properties such as their shapes, associated properties and proposed a way of defining and exploiting overlapping of *vcells*. For this, we have collected data from deployments in urban areas, and we have illustrated *vcells* emulating as typical vermiform shapes considering different mobile reception conditions.

We have discussed a novel geolocation system for mobile devices using Bloom filters to represent *vcells* and to provide a simple position retrieval method. We have argued that BF representation could be potentially be exploited for scenarios in which the abstraction of mobility patterns is useful, thus making *vcells* relevant for localisation amidst disasters, human tagged locations, and mitigating the lack of cellular coverage.

ACKNOWLEDGMENT

This work was partially funded by the Internet Society, the EU Metrics project (Grant EC607728), and the H2020 RIFE project (Grant 644663). We also would like to thank José Angel Márquez for *vcells* representation.

REFERENCES