Pathsim: Optimizing Cellphone Contact Tracing using Realistic Path Simulation

Computer Science and Information Systems Master’s Thesis Proposal

Michael Mark

CSIS Master’s Thesis

Saginaw Valley State University

December 15, 2020

**Abstract**

In this thesis, I propose Pathsim. *a carrier-based privacy preserving contact tracing method based on the cellphone data collected from the walking path of two users*. Pathsim is a novel contact tracing idea that uses the existing cellphone network data (also known as MRO data) from cellphone carriers to formulate a method showing user proximity by exploiting the characteristics of the wireless signal. Based on our preliminary findings, the cellular log data between two cellphones can be used to estimate the proximity of the cellphones with high accuracy over 95% of the time. Pathsim is built on this solid foundation and considers realistic walking paths, to further enhance and predict user proximity based on their predicted pathing. It is important to note that our current method does not require knowledge of the actual user location to work. It instead compares signal vectors at different locations and computes a similarity index that is used to estimate the distance between the locations. Thus, Pathsim will preserve user privacy, which is becoming a major concern in current contact tracing efforts.

**Background**

Contact tracing has taken an even important role with the onset of the recent COVID-19 pandemic. Early efforts at tackling contact tracing has taken many forms. A few notable mentions are:

1. Contact tracing using cellphone GPS data, credit card transactions, CCTV cameras. This was used by the Government in South Korea [1] to determine the route taken by an infected individual to alert other users who may have been in close proximity with the individual. Obviously, methods like these can be seen as privacy intrusive.
2. Contact Tracing based on Bluetooth: A number of Bluetooth based contact tracing methods [2] have also emerged during the current pandemic. All of these suffer from the same set of drawbacks. They require the user to install an app on their phones, open the app, allow it to turn on their Bluetooth module which can then discover other users. The major issue with these methods is that they have not been widely adopted by users and are susceptible to well-known network attacks outlined in [3]
3. A third category of contact tracing methods is labelled as social media contact tracing [4], which uses user data social networks like check-ins, geo-tagged photos to identify peers who may have been in close contact with the infected user. Privacy issues aside these methods require the user and all of their peers to be active on social networks, which may not be a reasonable assumption.

Other categories of contact tracing include contact tracing using Wi-Fi MAC addresses, which is useful as an indoor option, and using wearable devices to track proximity to other devices in the same area. Pathsim is different from all of the above methods since it does not rely on the user location, does not require any user intervention, and will likely be more reliable than most of the methods described above.

**Theoretical Foundation**

The primary work that Pathsim builds upon is called Vecsim [5], a novel method that estimates the proximity of 2 locations by looking at the cellphone log data. In this section I provide a brief background on Vecsim, since most of the theoretical foundations are shared by both methods.

In an LTE network, cellphone carriers use towers to cover service areas. These towers typically mount 3 base stations that can be identified with a unique identifier called PCI. Each base station continuously logs cellphone data from connected devices, or user equipment (UEs), every 5.12 seconds and logs them into a file called the MRO (maintenance, repair, operations) log data file. Vecsim uses the MRO data which contains information like

* *Timing Advance (Tadv):* This represents the upper limit of the length of the signal from the base station to the user. This value is proportioned between 0-10 with a granularity proportion of 78 meters.
* *Angle of Arrival (AoA):* Also denoted by , this is the angle from which a signal is received from the top of the tower, which is a real value between [0, 2]. This can be visualized as the angle of the signal from the cellphone observed at the base station.
* *Reference Signal Received Power (RSRP):* This value is a measure of the signal strength between the UE and the connected base station. The cellphone also reports the RSRP values for neighboring base stations in the vicinity.

Vecsim uses the above information to estimate the distance between the signal vectors at 2 locations. It is important to note that Vecsim does not focus on localization, that is, finding the exact positions of the UEs, but rather, focuses on the relative distance between 2 UEs, which is a much easier problem to solve. The distance estimate between any two points can be visualized as follows.

Diagram

Description automatically generated

Figure 1: Vecsim initial estimate

From Figure 1 above, we can derive an initial estimate between two locations as:

where,

: the distance between the two users

: *Tadv* values reported by the two users

: Angle of Arrival values reported by the two users.

: 78 meters, the granularity proportion of the MRO file given by the base station

While this is a very simple and accurate estimate, the issue with this initial estimate is that the above formula assumes that the users are in Line of Sight with the tower (Figure 1). The distance estimate will not be accurate in cases where the signal reaches the UE using a reflected or diffracted path. Still, it serves as a good starting point for Vecsim.

Vecsim further utilizes this initial estimate by empirically defining two probability density functions to estimate the maximum likelihood based on the distances. It does this by choosing 8 distinct distance values 25m, 50m, 75m, 100m … 200m and then computes the maximum likelihood score for each of these distances. The maximum likelihood is based on the signal strength, or RSRP value from the MRO data as well as the initial estimate obtained from above. The result is a similarity score between 1-8, representing the distance estimate between 2 locations, with 1 representing 25m and 8 representing 200m.

Vecsim uses one additional feature to tackle the discontinuous nature of the signal. It was observed that there can be locations in cities with many buildings, where Vecsim reports a large distance estimate even when the actual locations are quite close. For example, if a person is emerging from a small alley onto a major roadway, the signal shows a large change even though the two locations are actually very close to each other. To tackle this Vecsim creates a database of Discontinuity Pairs, called the Discontinuity Pairs Database (DPD), which will keep track of such locations. While this may seem challenging, it is important to note that the cellphone data is being collected from millions of people walking in a city every day. By simply logging these walks, it is possible to find most discontinuous pairs in a given area. In the Vecsim study, over 90% of DPD pairs were learned after 500,000 paths logged [5].

**Pathsim**

Vecsim already provides good results for a contact tracing method that preserves user privacy and does not require any additional actions from the user. However, it can be improved. Vecsim only relies on the signal vector at 2 locations to estimate the distance. Due to the discontinuous nature of the wireless signal, this still leads to some inaccuracies. In this thesis, I propose Pathsim, a similarity measurement based on the signal vectors between 2 paths generated by two users. The primary advantage of Pathsim over its predecessor is that Pathsim can better handle the discontinuity of the signal, by computing the distance estimates between multiple points from two paths. Furthermore, while Vecsim simply computes the distance estimate between 2 points, Pathsim can also estimate how long the 2 users may have been in close proximity given the signal vectors from their respective paths. This is crucial to contact tracing methods and will mitigate the bulk of the false positive cases that arise from running Vecsim. It will also reduce the time to learn unnecessary DPD pairs, so it will have less of a database storage and strain on the server.

It is important to note that Pathsim will still not require user intervention or the actual locations of users. Therefore, user privacy is still protected. Pathsim will use realistic walking paths to determine the range and duration of proximity between an infected individual and other users. I believe Pathsim will be achieve higher accuracy when determining proximity between two users and lower the overall percentage of false positives.

**Preliminary Work**

The first step towards developing Pathsim has been to generate random walk simulations in our generated city model, shown in Figure 3. The city model is generated as matrix of 0s and 1s where 1s represent the roadways with cellphone coverage, while 0s represents the indoor buildings without cellphone coverage. We note that indoor localization using LTE log data is a much harder problem to solve and limit our study to outdoor environments only.

Diagram

Description automatically generated

Figure 2: MATLAB plot of example shortest path

A ray tracing program (Wireless Insite) was used to generate the signal field from 13 towers placed at random locations in the area.

While there are several methods to generate paths from random walks or the shortest path between two points, we found that the paths generated from these methods are not representative of actual walking paths. For example, users may tend to hug the side of a building, representing sidewalks, and cross the street at intersections only when walking from one location to another. Therefore, we refined our random walk algorithm to consider these constraints when simulating such paths. The result can be seen in Figure 4, where the outer green band shows a more realistic path that may be taken by a user as opposed to the shortest path.

A picture containing qr code

Description automatically generated

Figure 4: Shortest path vs Realistic Path

In collaboration with other peers working on this project, we have created a method to generate realistic walking paths and can generate a very large number of paths (i.e., more than 10,000) very quickly to use in our Pathsim algorithm.

**Limitations of Pathsim**

From our work, we have stressed to reduce the time it takes to generate the number of random paths. However, we have only been able to reduce the time to a few minutes to generate 10,000 paths. As time goes by, we do believe that the code will become simpler and this time will be further reduced. Although this is a processing issue that may slow down the system, it will only happen on the server side and not the client side.

Another limitation that Pathsim may encounter is the fact that MRO data is only sent out every 5.12 seconds. Therefore, currently the shortest time to predict a user’s path might be this timing since there are no other ways to predict someone’s path just from a single point. Of course, as we have more log data regarding a signal, the more accurate the prediction will be. However, this is something more for our team to think about and to work around. We do foresee that this front-end processing will reduce the error and provide a higher accuracy in finding user proximity, which is the goal in enhancing Vecsim.

**Conclusion**

In conclusion, I propose Pathsim, a contact tracing method based on LTE log data that is collected by cellphone carriers. Pathsim will use features of the signal vector like *AoA*, *Tadv* and *RSRP* to estimate the proximity between two paths. If successful, this will be a novel method that preserves user privacy, as it does not depend on the user location, and can be used with widespread adoption as it does not require any user involvement. Furthermore, by using realistic path simulations, Pathsim will be evaluated in real-world conditions.

**References**

[1] COVID-19 National Emergency Response Center, K. C. f. D. C. Case Management Team, and Prevention., “Contact transmission of covid-19 in south korea novel investigation techniques for tracing contacts.,” 2020.

[2] Google, Apple, “Privacy-preserving contact tracing.” https://www.apple.com/ covid19/contacttracing/, 2020.

[3] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha, “A survey of covid-19 contact tracing apps,” IEEE Access, 2020.

[4] K. Zhang, X. Liang, J. Ni, K. Yang, and X. Shen, “Exploiting social network to enhance human-to-human infection analysis without privacy leakage,” IEEE Transactions on Dependable and Secure Computing, 2018.

[5] Avishek Mukherjee, Yaoguang Zhong, Zhenghao Zhang, Tingting Zhao, and Jinfeng Zhang. 2020. Vecsim: Carrier-based, Privacy-Preserving Cellphone Contact Tracing. In Proceedings of the 18th ACM Symposium on Mobility Management and Wireless Access (MobiWac '20). Association for Computing Machinery, New York, NY, USA, 47–55. DOI: https://doi.org/10.1145/3416012.3424627