Connectivity-optimal Shortest Paths Using Crowdsourced Data

Tim Hultman, Abdeldjalil Boudjadar, Mikael Asplund
Department of Computer and Information Science
Linköping University, Sweden
timhu652@student.liu.se, abdeldjalil.boudjadar@liu.se, mikael.asplund@liu.se

***Abstract*—With the increasing dependency of ubiquitous connectivity for applications ranging from multimedia entertainment to intelligent transportation systems, having good signal coverage becomes vital. Therefore, route planners and navigation systems should take into account not only the physical distance, but also the characteristics and availability of the cellular network on the potential routes. In this paper we present a route planning tool that finds the connectivity-aware shortest paths based on crowdsourced data from OpenStreetMap and OpenSignal. The tool calculates optimal paths and allows physical distance to be traded against signal quality. The evaluation shows that a 15% increase of the physical path length can achieve an 8.7dBm improvement of worst-case signal strength.**

I. Introduction

The cloud-oriented paradigm has fundamentally changed the way computer systems are organised. From initially dominated by enterprise applications, it is now entering the embedded and mobile domain. Ericsson's recent Connected Vehicle Cloud initiative is just one of many current examples of this trend. Having connectivity anywhere, anytime allows new smarter algorithms for distributed control as well as reducing installation and maintenance costs. However, a consequence of this transformation is that bad network connectivity is not just a nuisance, but can result in real degradation of service to users.

In this paper we consider how mobile entities in an urban environment can plan their physical routes by taking into account the network connectivity along different routes in addition to the physical route length. Consider for example an ambulance where a doctor can perform pre-hospital care through telemedicine [9]. In such a scenario it can make sense to take a slightly longer route if good connectivity can be guaranteed.

This raises two interesting questions. First, whether there are reasonable alternative routes in an urban environment that provides significantly better signal strength without being excessively long. Second, if there are available data sources with sufficient resolution and quality on which to base such decisions. We answer both of these questions in the affirmative with the help of a route planning tool that uses online crowdsourced data repositories to calculate connectivity-aware shortest paths.

The Connectivity-Aware Route Planner (CARP) tool integrates map data from OpenStreetMap and signal strength data from OpenSignal into a unified framework. The tool currently supports three routing policies, classical shortest path, a bounded degradation policy that finds routes where the period of reduced signal strength is bounded, and a threshold policy that ensures that the signal strength never goes below a given threshold.

We evaluate the impact of these routing policies on the physical route length as well as the resulting received signal strength along the chosen paths. The evaluation also provides insights into the usefulness of OpenSignal as a data source as well as a preliminary validation with a proof-of-concept physical experiment.

In summary the contributions of this paper are threefold, (1) the Connectivity-Aware Route Planner (CARP), that integrates two crowd-assisted data sources to provide connectivity-aware routes (2) an assessment of the usefulness of crowdsourced signal data for predicting connection quality in a medium-sized city and (3) an evaluation of our approach that demonstrates that it is indeed possible to trade slightly longer path lengths for real improvements in signal strength.

II. System Overview

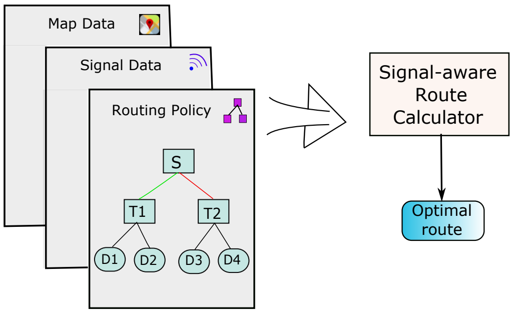


Fig. 1: Workflow of route planner.

This section introduces the main functionality of our system application. In order to calculate the shortest signal-aware path between a source and destination, three different types of input are needed: map data, signal data and routing policies. The map data describes the physical world geographically, and is turned into a graph representation. Signal data allows us to enhance the map structure, such that for each piece of the structure there is signal information. Finally, a given routing policy that can combine the physical attributes of the map with the signal information to calculate a shortest path with an optimal connectivity (network signal quality).

Two methods for planning routes that take signal strength into account are evaluated. They are both based on Dijkstra's shortest-path algorithm for graphs. For the purpose of this presentation we will consider the graph to be constructed of vertices and annotated edges. The annotations provide information about physical distance and signal quality. There are some further practical considerations made in the actual implementation, but which are not presented here for reasons of brevity.

A. *Threshold-based Route Planning*

The threshold-based route planning policy results in paths where the signal strength never go below a certain threshold value. This is achieved by modifying the way that Dijkstra's algorithm finds the closest unvisited vertex. Whenever the signal strength data along the edge to a neighbouring vertex is below the given threshold, the corresponding edge is marked as unusable and is therefore not considered by the algorithm.

For a given threshold parameter (i.e., a bound on the worst acceptable signal strength) this algorithm provides the optimal path by virtue of running Dijkstra's algorithm on a subgraph of the original graph. Therefore, this algorithm is pareto-optimal with respect to pathlength and worst-case signal strength.

B. *Bounded Degradation Route Planning*

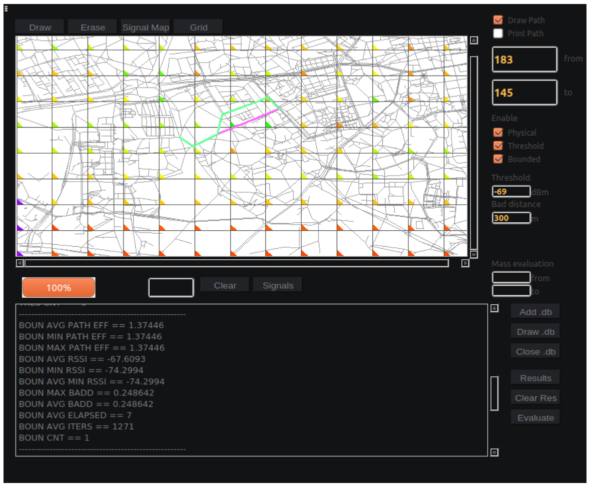
Bounded degradation route planning attempts to find routes where the stretch of weak signal along the route is bounded. That is, an application requiring network coverage and loses it temporarily may build up a backlog until it re-establishes a signal. As long as the time (captured as a distance in our application) is sufficiently low, going below a given threshold is acceptable. This is achieved by keeping track of the distance travelled with bad connectivity.

Much like threshold-based routing, whenever the cost to reach a neighbouring vertex is evaluated. If the signal data of the edge is compared below a threshold we calculate the total distance that is considered below this level, called *bad distance*. If the bad distance is sufficiently short, the edge is still valid. Once a good signal area is reached the bad distance variable is reset and the route is once again allowed to cross a bounded stretch of bad connectivity. This problem has similarities to forbidden subpaths in graphs [7], with the distinction that there is no set of known forbidden paths. Instead, a forbidden path is detected whenever the bad distance crosses the given threshold.

Vertices that are considered non-reachable due to a too long bad distance cannot be discarded, since the algorithm can potentially update the bad distance information by finding a longer path with better connectivity. These vertices are tentatively put aside, but can be re-included in the set of unvisited nodes if their bad distance variable is updated.

III. Connectivity-Aware Route Planner (CARP)

In this section, we outline how the CARP tool acquires map data and signal data, as well as a description of the combination of such data and the final internal representation of such a combination in terms of a graph that can be explored using Dijkstra's algorithm. By building a graph from the map data, provided by OpenStreetMap[1], and combining it with a grid of boxes containing signal data from OpenSignal[2], it provides a testbed for route planning policies that take signal strength into account. Logged data can be plotted and processed for statistics regarding distance, average signal strength, signal maxima, and other characteristics. Figure 2 shows a screenshot of our signal-aware route planning tool.



☑ Draw Path
☐ Print Path
183 from
145 to
Enable
☑ Physical
☑ Threshold
☑ Bounded
Threshold
-69 dBm
Bad distance
300 m

```
--------------------------------------------------
BOUN AVG PATH EFF == 1.37446
BOUN MIN PATH EFF == 1.37446
BOUN MAX PATH EFF == 1.37446
BOUN AVG RSSI == -67.6093
BOUN MIN RSSI == -74.2994
BOUN AVG MIN RSSI == -74.2994
BOUN MAX BADD == 0.248642
BOUN AVG BADD == 0.248642
BOUN AVG ELAPSED == 7
BOUN AVG ITERS == 1271
BOUN CNT == 1
--------------------------------------------------
```

Fig. 2: Screenshot of application running on Ubuntu.

A. *Building a Graph Representation*

OpenStreetMap (OSM) is a collaborative project aimed at providing an editable world map. The data is collected by volunteers and provided for free via various APIs, as listed on the OSM website. By using any of the provided APIs, map data can be obtained. OSM map data (as provided by Overpass API), consists of *nodes*, *ways*, and *relations*. For the purpose of finding physical routes, we are only interested in the road network part of the map (which can be inferred from the set of nodes and ways).

Thus, the map data is parsed and transformed into a graph representation, where every intersection (and cul-de-sac) is represented by a vertex, and roads between intersections are represented by edges. We have a dedicated data structure for the edges which mean they can be split up in portions, allowing variable signal strength assignments to portions of edges. Edge objects allow us to represent a changing signal along the distance of the road.

B. *Building a Signal Map*

OpenSignal is a company that provides an interactive map of signal strength, cell tower locations, and a number of other features. The underlying data is collected by individuals using the OpenSignal smartphone application. OpenSignal provides

[1] http://www.openstreetmap.org

[2] http://opensignal.com/

APIs to request data. An API key is required, and request rates must be observed. No data can be stored, and thus must be requested each time the application starts (CARP complies to this restriction).

The API used for this application is called NetworkStats API, and returns various network parameters, among which is Received Signal Strength Indication (RSSI). The API requires a centre point a square size for which data is provided. The CARP tool aquires a sequence of such squares according to a grid pattern with the square size as a configurable parameter.

C. Combining Graph with Signal Data

Once a graph has been built, and the signal data has been requested and parsed, these two data sets are combined. This is done by iterating through the list of edges (i.e. roads). Every edge consists of a number of locations, as described by the map data. Each of these individual locations is then assigned signal data based on the closest centre point. To avoid data duplication, only a reference to the closest centre point is saved. Since this distance check is performed for each location, any given edge may reference several centre points. This means that the signal strength may be varying as the edge is travelled, which is closer to a real-world representation than an edge with a single averaged value for signal strength.

IV. Evaluation

The evaluation is presented in three parts. First, we investigate the impact of different grid granularities when requesting signal data. Second, we show how the three implemented routing policies perform with regard to path efficiency and achieved signal strength. Finally, we describe the result of a physical validation experiment.

In all experiments, a map of the south-west part of the city of Linköping was used. The size of the area was 3765 by 2624 meters.

A. Assessment of Signal Map

Recall from Section III-B that in order to be able to request data from OpenSignal, one of the parameters that must be given is a box size. Choosing this box size has major implications on the properties of the signal map data. In order to decide this value there are several considerations that must be adhered to. Geometry rules dictate that as the side of a square decreases linearly, the number of squares (and thus API requests) increase quadratically. Because we have request rates that must be observed, this is a major limiting factor.

Secondly, the total number of RSSI samples reported by the OpenSignal API varies considerably for different box sizes. Figure 3 shows how the total number of signal strength samples reported from OpenSignal varies with the box size for fixed area. The expectation would be that no matter the box size, the *total number of samples* remain constant for the entire area. However, the graph shows that the total number of samples for the entire map is *several times larger* for 100m boxes than for 400m boxes. We can find no reasonable explanation other than that samples from surrounding areas are in some way taken into account by OpenSignal when estimating the signal strength in a requested area.

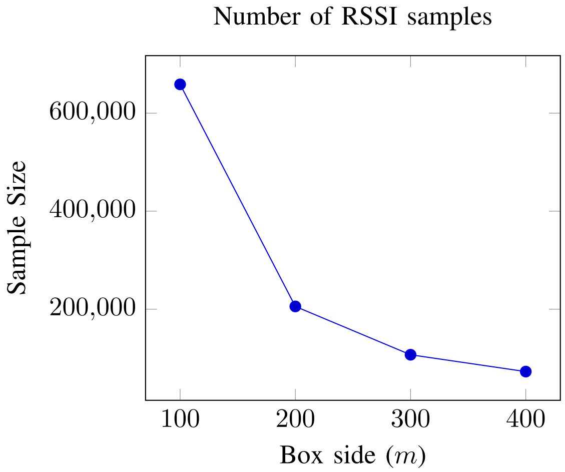


Fig. 3: Total amount of sample sizes for the entire map segment.

Figure 4 shows another reason for not choosing too small box sizes. The Y-axis shows the percentage of boxes for which no RSSI values are reported from the API. With so many of the boxes lacking information, it would be necessary to try to interpolate these values from surrounding areas. That is, box sizes below 300 meters does not seem to increase accuracy.

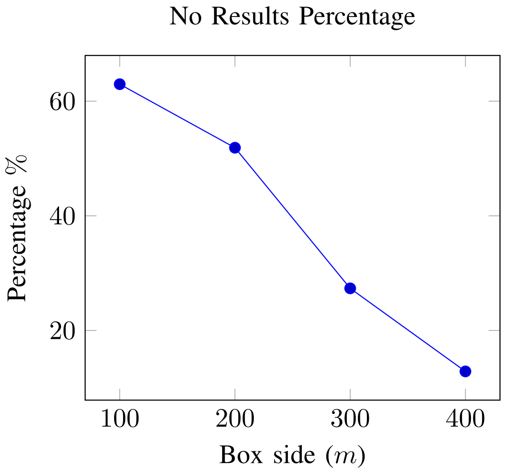


Fig. 4: Percentage of boxes returning no results.

Based on these findings CARP uses boxes with 300m sides as the default. The typical setup time for such a box size is 3 minutes, although this is possible due to an agreement with OpenSignal where the minute limit has been increased 8 times. Without such an agreement, the setup time for the same map then becomes approximately 24 minutes, and uses 6% of the monthly limit (number of total requests permitted by OpenSignal).

B. Routing Policies

To compare the different routing policies we selected 5 points of interest in the Linköping area (Linköping's Golf Club, the cinema, hospital emergency entrance, one close to downtown, one close to the university). The points where chosen to be reasonably spread out on the map and with sufficient distance between themselves. The 5 points results in 10 pairs between which the three routing policies were used to find routes.

Few different metrics can be applied when comparing different routes. In this paper we are focusing on average RSSI, minimum RSSI, and path efficiency. The average RSSI provides an all-round value that is easy to understand, however it provides little information regarding e.g. the length of the path through poor signal areas, which may impact an end-user application. The minimum RSSI is used in order to highlight the local signal minima for each path, which is practical to avoid poor signal areas. In combination with path efficiency, we attempt to show where the strengths and weaknesses for each routing policy lay, and what the cost can be for taking a detour. There are more network QoS metrics that could be used when evaluating routes (and planning them), such as hand-off dropping probabilities [11] and reducing hand-overs when moving between network cells, however the data as available from OpenSignal is limited in these areas and are thus omitted.

The average signal strength was also calculated for each of the routes in order to compare how well they scored. Threshold-based routing is expected to have the highest minimum for RSSI, and typically the highest average as well, as it will take detours (although the shortest possible detour) in order to maintain a certain value of signal strength. Threshold-based route planning is therefore also expected to have the worst average path efficiency. Bounded-based route planning is expected to have an average path efficiency between threshold-based and shortest-path policies. Whether it lands closer to threshold-based or shortest-path depends on the signal strength layout, and is difficult to predict without knowledge of the signal areas.

Figure 5 shows the path efficiency (calculated as the path length divided by the direct path between source and destination) for the three policies. The results are in line with what could be expected. Figure 6 shows the average minimum RSSI (lowest RSSI for each path in the set of paths divided by amount of paths) for the three policies. Somewhat unexpectedly, there is no difference between the shortest path policy and the bounded policy. This can be explained by the fact that the bounded policy actually does not seek to maximise the minimum RSSI, but simply to bound the period of bad signal strength.

Calculating the average signal strength was performed by iterating through the points, calculating the distance from the previous point and averaging the signal values between them. The signal strength between these two points is then multiplied by the ratio of the distance to total route distance. Assuming a high accuracy, this should prove more accurate than the approximation of assuming an equal distance between each node (that is, simply sum the signal values and divide by the amount of data points).

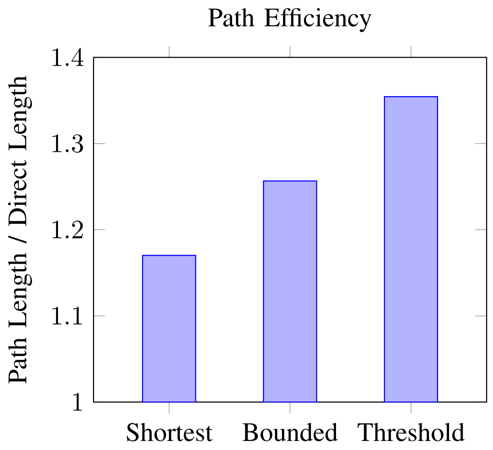


Fig. 5: Calculated average path efficiency for a set of paths using OpenSignal data.

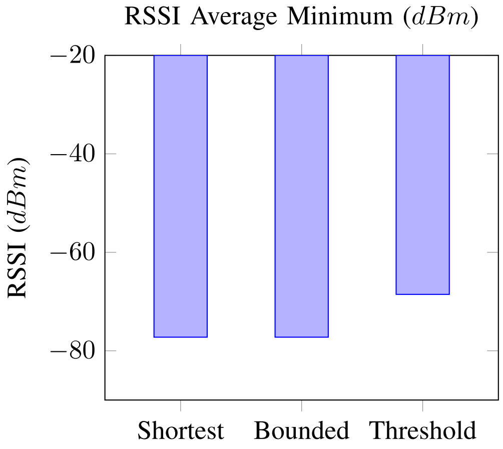


Fig. 6: Calculated average minimum RSSI for a set of paths using OpenSignal data. Path sources and destinations are points of interest in Linköping.

C. Validation

Finally we performed a physical experiment to validate that the routes and rssi values predicted by the tool also had some bearing to what could be measured in reality.

The route shown in Figure 7 was chosen due to having interesting properties; the shortest distance goes through a forested area with a reportedly poor signal. Different route planning strategies may therefore attempt to avoid this area, leading to a longer route but with a higher average signal strength.

The shortest path route, as proposed by the application, was compared with the route proposed by Google Maps and turned out to be equal. Using a threshold of -69 dBm, threshold-based route planning policy attempted to avoid the area entirely, leading to a long bend around the forested area.

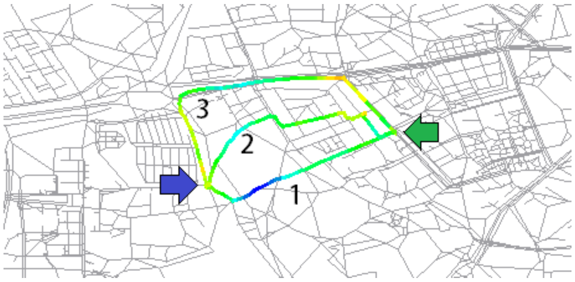


Fig. 7: Measurements plotted in graph. Blue arrow points to source, green to destination. Numbers 1 through 3 marks Shortest-path, Bounded, and Threshold route planning policies.

Using an allowed distance of poor signal of 300m, bounded route planning took a short path through the area of poor signal to reduce the distance, and thus did not avoid the area entirely, but in doing so managed to provide a shorter route than the route provided by the threshold-based strategy.

With the route planning strategies applied to the start and destination, the next step of evaluation was to physically travel the proposed routes and log the signal strength. The device used for logging was an Android-based smartphone (Google Nexus) with Network Monitor installed. The particular version of Network Monitor is a fork of the original source, and was developed at Linköping University [3]. The data logged by this version of Network Monitor can be exported in several formats, among which is a raw database file format. The example application can read these raw database files, allowing comparisons with the signal map data as provided by OpenSignal. It also allows visual confirmation that the correct route was logged, as the route may be plotted in the graph.

The device used was kept unobstructed while logging data, as the accuracy of the GPS location services drops if the phone is covered (e.g. kept in a pocket). The accuracy is reported as an error in meters using 5m intervals. The lowest non-zero value of this parameter (and thus highest accuracy) is 5m, the next value is 10m, and so on. For all the tests performed in this manner (a few hundred data points) the highest reported error was 15m, and only a single data point reported this value. Most data points reported an error of 5m. A run consisting of 29 data points was performed where the device was kept in a jacket pocket. Three data points reported an accuracy error of 20m, and had an average accuracy error of 14.14m. The same run performed with the phone unobstructed had an average accuracy error of 7.4m.

Figure 8 shows the route length of thre three different policies as predicted by the CARP tool and measured with the device. As expected, these values are almost identical. Finally, Figure 9 shows the differences in average RSSI between measurements and calculations. The measurements show a significantly larger variation between the different policies compared to what was predicted. While variation in signal strength between different devices can be expected [6], these results show a difference of more than 15dBm. This indicates that the predicted values cannot be entirely trusted. On the other hand, the connectivity-aware routing policies seems to be effective in finding routes with better signal strength.

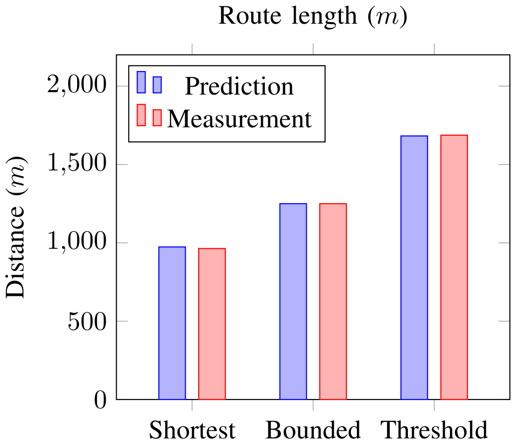


Fig. 8: Differences in distance between measurements and prediction. The difference ranges from 0.01% to 1.05%.

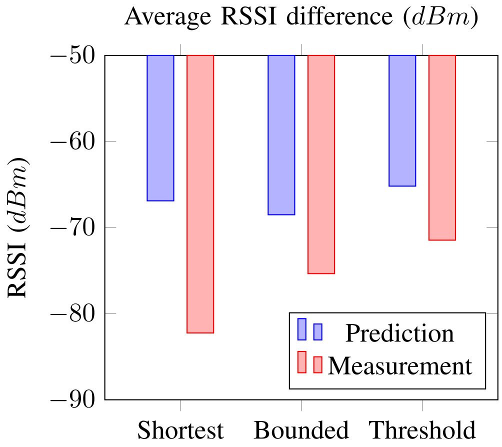


Fig. 9: Predicted and measured average RSSI

V. Related Work

With the increasing use of network-dependent applications in vehicles, route planners and navigation systems need to not only consider physical distances as routing policy but also the network availability.

However, since such a network connectivity is provided through wireless (WIFI and cellular radios) the access could be 6 times more expensive (in terms of energy consumption and the use of processing resources), than a regular access when the network signal quality is quite strong, and may lead to data loss [4]. Accordingly, the network attributes have to be merged with the map physical properties when calculating optimal routes.

Recently, a strong effort has been spent on the calculation and estimation of networks performance [5], [12] like signal strength and bandwidth. The output of such studies can be used then as a profile to improve the network connectivity, reduce the energy consumption or further optimize the wireless interfaces. In this paper, we combine the network signal data provided online by OpenSignal with the map data of Linköping based on OpenStreetMap and design a signal strength aware route planner.

In the literature, different studies [1], [8], [10], [2] have been focusing on the delivery of a routing policy that merges both classical physical attributes (distance, toll, etc) together with different profiles like user-aware, traffic-aware and self-aware route planning. In this context, we are only interested in works that consider the network connectivity when calculating routes.

In [1], a mobile user-aware route planner application has been developed. Basically, it collects GPS data of a user's everyday locations and provides different patterns (rather than street names) that can be used by the user as landmarks to identify different places. Both GPS data and the user profile (given by landmarks) are transformed to a route model that be used when the client request for a direction. However, such informal landmarks can be subject of ambiguity when calculating routes.

In [10], the authors propose a new technique to calculate optimal routes by considering the traffic state through the road map. The proposed technique collects data regarding the traffic situation from different nodes of the road map and considers such an acquirement when calculating a route. The attribute describing the traffic state is used with the same weight as the classical road attributes, leading accordingly to routes that are physically longer but less congested. However, a standard quantification of the traffic state is missing in the literature.

In [2], the authors introduce a network-aware path planner for systems distributed across large area networks. Such a proposal constructs network-aware communication paths that enhance application performance by taking into account both application performance preferences and availability of the network resource.

VI. Conclusions

In this paper we have demonstrated how online crowd-sourced data repositories (OpenStreetMap and OpenSignal) can be leveraged to provide connectivity-aware route planning in urban envinroments.

Despite some limitations of the OpenSignal API, with regard to accuracy and availability of samples for certain areas, our results indicate that the data provides the basis to make appropriate routing decisions when network connectivity is of high importance. Moreover, by considering a number of points of interest in the city of Linköping and paths between these points, we have shown that there are often alternative paths to the shortest path that result in slightly longer physical distance, but significantly better signal strength.

Acknowledgements

This work has been supported by CENIIT project 14.04.

References

[1] J. Chung and C. Schmandt. Going my way: A user-aware route planner. In *The 27th international conference on Human factors in Computing Systems*. ACM Press, 2008.

[2] X. Fu and V. Karamcheti. Planning for network-aware paths. In J.-B. Stefani, I. Demeure, and D. Hagimont, editors, *DAIS'03*. Springer Berlin Heidelberg, 2003. doi: 10.1007/978-3-540-40010-3˙17.

[3] R. Holm. Energy-efficient mobile communication with cached signal maps. Master's thesis, Linköping University, 2015.

[4] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan. Bartendr: a practical approach to energy-aware cellular data scheduling. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, MOBICOM 2010, Chicago, Illinois, USA, September 20-24, 2010*. ACM, 2010. doi: 10.1145/1859995.1860006.

[5] S. Sonntag, L. Schulte, and J. Manner. Mobile network measurements - it's not all about signal strength. In *Wireless Communications and Networking Conference (WCNC), 2013*. IEEE, 2013. doi: 10.1109/WCNC.2013.6555324.

[6] B. Viel and M. Asplund. Why is fingerprint-based indoor localization still so hard? In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, 2014. doi: 10.1109/PerComW.2014.6815247.

[7] D. Villeneuve and G. Desaulniers. The shortest path problem with forbidden paths. *European Journal of Operational Research*, 165(1), 2005. doi: 10.1016/j.ejor.2004.01.032.

[8] D. Wilkie, J. P. van den Berg, M. C. Lin, and D. Manocha. Self-aware traffic route planning. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.

[9] T.-C. Wu, C. Nguyen, C. Ankrom, J. Yang, D. Persse, F. Vahidy, J. C. Grotta, and S. I. Savitz. Prehospital utility of rapid stroke evaluation using in-ambulance telemedicine a pilot feasibility study. *Stroke*, 45(8):2342–2347, 2014.

[10] J. Xu, L. Guo, Z. Ding, X. Sun, and C. Liu. Traffic aware route planning in dynamic road networks. In S.-g. Lee, Z. Peng, X. Zhou, Y.-S. Moon, R. Unland, and J. Yoo, editors, *The 17th International Conference on Database Systems for Advanced Applications*. Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-29038-1˙41.

[11] I. Yildiz. On the estimation of user mobility pattern for location tracking in wireless networks. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*. IEEE, 2002. doi: 10.1109/GLOCOM.2002.1188151.

[12] M. Zhong, P. Hu, and J. Indulska. Revisited: Bandwidth estimation methods for mobile networks. In *WoWMoM'14*, 2014. doi: 10.1109/WoWMoM.2014.6918916.