Improving QoE via Context Prediction: A Case Study of Using WiFi Radiomaps to Predict Network Disconnection

Nearchos Paspallis
University of Central Lancashire, Cyprus
12-14 University Avenue
Pyla CY-7080, Cyprus
npaspallis@uclan.ac.uk

Salah Eddin Alshaal
University of Central Lancashire, Cyprus
12-14 University Avenue
Pyla CY-7080, Cyprus
sealshaal@uclan.ac.uk

ABSTRACT

This paper proposes a novel way to improve the user Quality of Experience (QoE) by monitoring and predicting their context. The method builds on a fingerprint-based indoor positioning system, which monitors the user's position and uses that to predict the quality of the network connection. Successfully predicting the quality of the network connection allows applications which are sensitive to network fluctuation–such as video and audio streaming apps–to optimize their buffering strategy, thus improving the overall QoE perceived by the end users. Our approach is demonstrated in the context of a case study-based evaluation, using a blend of real and simulated data.

CCS Concepts

•Software and its engineering → Middleware; •Human-centered computing → *Ubiquitous and mobile computing design and evaluation methods;*

Keywords

QoE, context-awareness, context prediction

1. INTRODUCTION

Ever since Mark Weiser proposed his vision of *Ubiquitous Computing* [16, 15], the world has seen major developments towards its realization. Most notably, the smart-phone has become a stable in everyday use. Not surprisingly, it has been called the *pocketable PC* and was cited as the top technology of the decade [10, 11]. The main theme of Weiser's vision was an almost supernatural interaction of humans with an omnipresent technology that is able to do things we want, even before we ask for them. As Weiser put it: "*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*" [16]. In this kind of scenarios, the main resource we aim to economize is human attention. As Garlan et al. have argued: "*The most precious resource in a computer system is no longer its processor, memory, disk, or network, but rather human attention*" [4].

ICPE'17 Companion, April 22-26, 2017, L'Aquila, Italy

ISBN 978-1-4503-4899-7/17/04. . . $15.00

DOI: http://dx.doi.org/10.1145/3053600.3053603

Core to the development of smart-phones are advances in context awareness. In this work, we refer to context with its general definition from Dey: "[Context is] *any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves*" [2, 3].

Most context awareness platforms aim to facilitate the collection of basic context data (as acquired by hardware sensors–e.g. location reported by GPS sensors) and to infer higher level context information by processing one or more lower level context types (e.g. identify user activity by processing movement patterns as sensed by an accelerometer and speed as measured by a GPS). Many of these platforms are surveyed and compared by Bettini et al. [1].

However, a lot of the value of context awareness comes in the form of *predicting* context. This is, for instance, a major enabler of Weiser's vision where technology recedes to the background, taking the initiative to improve the user Quality of Experience (QoE)–in its more general definition–by means of predicting their activities and acting upon that knowledge. For example, one such technology could intelligently predict when the user wants to watch TV (e.g. by sensing he sat on the couch in front of it) and tune to the preferred channel (based on the time of day or the ratings given by the user in the past).

In this paper, we argue that a relatively simple mechanism which collects user context in the form of WiFi radiomaps is able to predict user intention and optimize the QoE accordingly (specifically by anticipating network quality degradation, and acting accordingly). To demonstrate the applicability of the proposed method we illustrate it in a fictional scenario using data sourced from real-world usage. The scenario concerns a mobile user who frequently roams a large building while consuming streamed media (such as online videos) and who experiences varying levels of network quality based on his position inside the building. By predicting the network quality, the user is able to optimize their QoE.

The rest of this paper is organized as follows: Section 2 first discusses related work and sets the premise for the proposed approach. Our approach is then described in more detail in section 3, where the data collection platform is described, along with the proposed context prediction algorithm. The main contribution of the paper is the case-study based evaluation, which is described in section 4. The paper closes with conclusions in section 5.

2. RELATED WORK

The topic of context prediction has been actively studied for many years. Naturally, predicting the context provides many opportunities for optimizing the QoE. Prediction is facilitated by the fact that humans are creatures of habit.

But context prediction has many challenges. Mayrhofer argues that many issues need to be handled by any context prediction approach, including handling issues related to accuracy, fault tolerance, unobtrusive operation, user acceptance, problem complexity and privacy [7]. An approach for context prediction is proposed by the same author in [6], where he also aims to identify "*patterns and interrelations in the user behavior which are not apparent at the lower levels of raw sensor data*". The author collects a set of real life data reflecting common context values (including WiFi access points and their signal strength) and then uses standard methods (e.g. the K-Mean algorithm) to predict higher level context information.

In another work, Sigg et al. proposed an alignment technique which was based on algorithms that were originally applied by computational biologists to find matching patterns between RNA or DNA sequences [12]. The authors demonstrate that their method is general enough to be applied on both numeric and non-numeric context types.

Petzold et al. [9] assessed branch prediction techniques, originally developed in the context of processor architectures, for context prediction. Similar to this paper, their aim was to predict the context based on the persons' movement patterns inside buildings. They evaluated both local and global context prediction scenarios, identifying the strengths and weaknesses of the two approaches.

Voigtmann and David proposed a *Collaborative Context Prediction* (CCP) approach, which "*takes advantage of existing direct and indirect relations which may exist among the context histories of various users.*" [14]. Their approach was evaluated on a real-world dataset containing smartphone accelerometer data annotated with the movements performed by the users, and it was found that it performs favorably compared to non-collaborative approaches.

3. CONTEXT PREDICTION

Our approach is based on the collection of context data under real-world conditions, using an existing platform: the *Context-Aware Indoor Positioning System* (CAIPS) [8]. This is a platform for general context collection, which was designed for testing and evaluating indoor positioning algorithms. It comes in the form of an Android application which is able to collect raw context information, as it is reported by the built-in sensors of the hosting device. For instance, it is able to collect the following context data:

- the radio-map in the form of triplets of access point BSSID, signal strength and frequency of the carrier channel;
- the user activity, as inferred by the device itself using the built-in API available in Google Play services (this includes basic user activities such as *Still*, *Walking*, *Running*, *Biking*, *Driving*, etc.);
- the selected (i.e. connected) access point, including its BSSID and its signal strength;
- various other context data types such as battery level, magnetometer and accelerometer readings, environmental properties such as air temperature, pressure, humidity, etc.

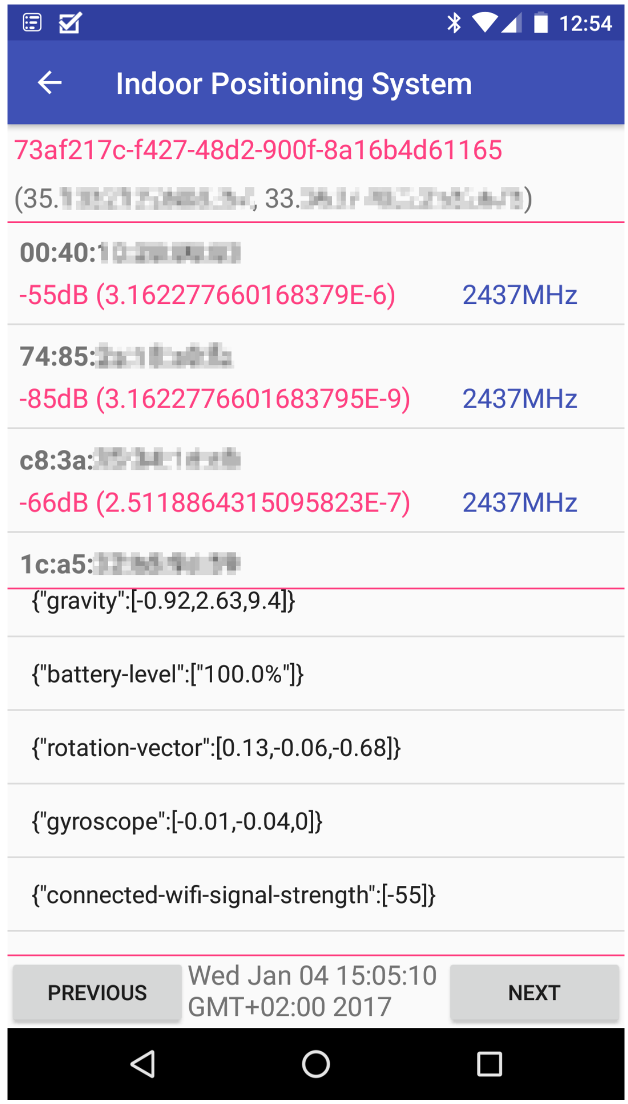

Figure 1: Screenshot of the *Context-Aware Indoor Positioning Platform* (CAIPS) illustrating the view of a typical training instance

3.1 Collecting Context

The app allows the user to collect *instances* of the context, and store them in a local database. These instances can then be used as the training input to a machine learning algorithm, with the aim to tune or evaluate prediction algorithms. Figure 1 displays a screenshot of the user interface of the aforementioned platform when displaying a *training instance*. The top half shows the radio-map and the lower half shows the other context values.

The user is allowed to collect individual trainings on demand, or configure the app to periodically collect trainings in predefined intervals. Once a training sessions is completed, the user can export the full context data in JSON format. The CAIPS app is publically available as an open source project (under LGPL) on Github[1].

3.2 Context Prediction Algorithm

Once an adequate set of context data is collected, we proceed to the assessment of algorithms which could potentially help predict the user context. The aim is to predict network disconnections, by monitoring user movements. As the user is moving primarily inside a building, we employ a standard fingerprint-based algorithm to infer user position [13]. In particular, in this work we use the K-Nearest Neighbor algorithm (with K=3) which was found to provide sufficiently accurate results (specifically, mean distance between estimated and actual position was less than 7 meters) [8].

At the same time, we collect context information about significant *Points of Interest* (POIs) in the building, including the signal strength of the connected access point. This data is used in combination with the position prediction to provide a prediction of the WiFi quality, thus allowing network streaming applications to optimize their operation and improve the user-perceived QoE. The process along with sample data are illustrated in a case study in section 4.

4. CASE STUDY-BASED EVALUATION

The case study-based evaluation is partly based on real-world data, covering two floors of a university building, and partly on artificial data created to reflect the author's understanding of user habits with respect to daily movements. Our hypothesis is that by modeling the motion patterns we can predict with reasonable accuracy the movement of a user and thus also predict when the network quality might degrade.

4.1 Motion map

In the first step, we model the motion map using blueprints of the building used in our experiments. As the same modeling is clearly used for the *indoor positioning* algorithm, we identify well-separated points of interest, along a set of 24 POIs. These are grouped in terms of function and relevance to the user (e.g. 'Entrance/exit' is where you would normally enter or exit the building).

4.2 Network strength

In the next step, we used real-world measurement to identify the signal strength of the connected access point for each one of these POIs. The results are listed in table 1 where the relevant points are shown along with the observed WiFi signal strength of the connected access point (measured in decibel). Note that even though there is some variance is signal strength across the building, the stairwells have significantly low strength indicating that effectively WiFi is not accessible in those locations.

4.3 Motion patterns

In the next step, we provide a model of the user motion patterns. This is used to model a typical route of a user for a given day. The basic idea of these motion patterns is that it reflect user activities that occur regularly because of repetitiveness of some tasks (e.g. lecturing in a specific classroom, or leaving the building at the end of the day).

[1]https://github.com/nearchos/CAIPS

POI	WiFi strength (dB)	Notes
A	-81	Edge of building
B	-61	Entrance/Exit
C	$-\infty$	Stairwell (weak signal)
D	-62	Transition point
E	-54	Transition point
F	-67	Edge of stairwell
G	-59	Edge of stairwell
H	$-\infty$	Stairwell (weak signal)
I	-43	Office (frequent use)
J	-51	Transition point
K	-74	Entrance/Exit
L	-60	Edge of building
M	-64	Transition area
N	-53	Admin office (frequent use)
O	-59	Entrance/Exit
P	-56	Transition point
Q	-63	Edge of stairwell
R	$-\infty$	Stairwell (weak signal)
S	-66	Edge of stairwell
T	-47	Lab (frequent use)
U	-70	Transition point
V	-62	Transition point
W	-56	Library (frequent use)
X	-52	Cafeteria (frequent use)

Table 1: *Points of Interest* (POIs) with annotations

Even though the model was not formed based on real-world data, we argue it can serve as a realistic model capable of predicting user movement with relatively high accuracy. Arguably, a similar model could have been *trained* using standard machine learning algorithms, and a sufficiently large set of data sample.

Id	Time	Motion path	Notes
i	08:00	B C A D F J I	Arriving to the building
ii	09:00	I J M Q R S T	Going to a timetabled class
iii	12:00	T S R Q M J I	Returning to office
iv	13:00	I J M Q R S V X	Going for lunch
v	13:30	X V S R Q M J I	Returning to office
vi	14:30	I J M Q U W	Going to the library
vii	14:45	W U Q M J I	Returning to office
viii	17:00	I J F D A C B	Leaving the building

Table 2: Fabricated model illustrating user's most common motion patterns in the building

4.4 Motion prediction

We claim that by having a model for the motion pattern we can predict user movement relatively accurately. Assume for example that our position changes from POI I (office) to J (transition point) at 13:17. In that case the more likely movement is described by pattern *iv*, meaning that the user will go through point R which has poor network quality.

In this case it is sufficient to consider only the last two positions and the current time (e.g. 'I-J' at 13:17) to predict that the user is most likely on motion path *iv*. While the algorithm works accurately whenever there is no ambiguity, it can obviously deliver even more accurate predictions when considering 3 or more of the last visited POIs.

Algorithm 1 Simple pattern matching algorithm

```
procedure PredictNextPoint(patterns, path, time)
    result ← MatchWithAllCharacters
    if result == 1 then
        print 'Found a match'
    else
        if result == 0 then
            PredictNextPoint(path - firstCharacter, time)
        else
            nextPoint ← getClosestMatchInTime()
procedure MatchAllCharacters(patterns, path)
    list result ← ∅
    for do item in patterns
        if item in path then
            result ← result + item
    return result
```

To verify our hypothesis, we implement a basic string-matching algorithm based on the one proposed by Karp et al. in [5]. The algorithm, denoted as Algorithm 1, consists of two procedures: The former takes as input an array containing the patterns (e.g. those listed in table 2), the search path (e.g. 'I-J') and the selected time (e.g. '13:10'). The latter–which is used by the first–takes two arguments only, the array of patterns and the search path. The algorithm produces as output the most likely next POI. The algorithm was also prototyped in *Python* and is publicly available on Github[2].

5. CONCLUSIONS AND FUTURE WORK

We have proposed an approach that allows to monitor indoor position of a user in a building, and argued that when there is a fixed pattern in the user movements, then it can be identified and used to predict future movements. We have also presented a case study where relevant points of interest are identified in a real building, and then sued to measured the WiFi signal strength at those points. The results show that some points could suffer from weak or no connectivity. At the same time, predicting the motion path of a user can also provide information on whether the user will pass by such weak connectivity point, thus allowing for services running on a mobile device to prepare accordingly, thus optimizing the perceived QoE.

In the future, we intend to collect more data spanning multiple days with the aim of automatically forming the movement pattern model. Furthermore, we will evaluate whether it is possible to predict network connectivity straight from the WiFi fingerprint measurements (i.e. without the intermediate step of inferring the indoor position first).

6. REFERENCES

[1] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161 – 180, 2010. Context Modelling, Reasoning and Management.

[2] A. K. Dey. *Providing architectural support for building context-aware applications.* PhD thesis, Georgia Institute of Technology, 2000.

[3] A. K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.

[4] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing*, 1(2):22–31, Apr. 2002.

[5] R. M. Karp, R. E. Miller, and A. L. Rosenberg. Rapid identification of repeated patterns in strings, trees and arrays. In *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, STOC '72, pages 125–136, New York, NY, USA, 1972. ACM.

[6] R. Mayrhofer. *An Architecture for Context Prediction.* PhD thesis, Johannes Kepler University of Linz, Austria, October 2004.

[7] R. Mayrhofer. Context prediction based on context histories: Expected benefits, issues and current state-of-the-art. In T. Prante, B. Meyers, G. Fitzpatrick, and L. D. Harvel, editors, *Proc. ECHISE 2005: 1st International Workshop on Exploiting Context Histories in Smart Environments*, pages 31–36, Munich, Germany, May 2005. In conjunction with the Third International Conference on Pervasive Computing (PERVASIVE 2005).

[8] N. Paspallis and M. Raspopoulos. An open platform for studying and testing context-aware indoor positioning algorithms. In *Proceedings of the 25th International Conference on Information Systems Development (ISD2016)*, pages 314–321, Katowice, Poland, Aug. 2016. AIS.

[9] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer. Global and local state context prediction. In *Artificial Intelligence in Mobile Systems (AIMS)*, Seattle, WA, USA, 2003. In conjunction with the Fifth International Conference on Ubiquitous Computing.

[10] J. J. Romero. Smartphones: The pocketable pc, Jan. 2011.

[11] P. E. Ross. Top 11 technologies of the decade. *IEEE Spectrum*, 48(1):27–63, Jan 2011.

[12] S. Sigg, S. Haseloff, and K. David. An alignment approach for context prediction tasks in ubicomp environments. *IEEE Pervasive Computing*, 9(4):90–97, October 2010.

[13] A. Varshavsky and S. Patel. Location in ubiquitous computing. In J. Krumm, editor, *Ubiquitous Computing Fundamentals*, chapter 7, pages 285–320. CRC Press, Boca Raton, FL, USA, 2009.

[14] C. Voigtmann and K. David. Collaborative context prediction. In K. David, K. Geihs, J. M. Leimeister, A. Roßnagel, L. Schmidt, G. Stumme, and A. Wacker, editors, *Socio-technical Design of Ubiquitous Computing Systems*, chapter 8, pages 131–150. Springer International Publishing, Cham, 2014.

[15] M. Weiser. Hot topics: ubiquitous computing. *IEEE Computer*, 26(10):71–72, Oct 1993.

[16] M. Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, Jul 1999.

[2] https://github.com/SalahEddin/IndoorMovementPrediction