Recurring Contacts between Groups of Devices: Analysis and Application

Nuno Cruz and Hugo Miranda

Abstract—The capability to anticipate a contact with another device can contribute to improving the performance and user satisfaction of mobile social network applications and of any other relying on some form of data harvesting or hoarding. This paper presents a nine year data set of wireless access logs produced by more than 70,000 devices and 40,000 users. Research on the recurring contact patterns observed between groups of devices permitted to model the probabilities of occurrence of a contact at a predefined date between pairs of devices. As an example, the paper presents and evaluates an algorithm that provides daily contact predictions, based on the history of past pairwise contacts and its application on a reputation service.

Index Terms—Mobility, contact recurrence, modelling

1 Introduction

The knowledge on users mobility patterns can be applied on pervasive computing environments to model applications [22] and routing protocols [16], to harvest computing resources [6] or provide network connectivity [24] to a group of mobile devices and to facilitate the creation of distributed data stores [20].

The capability to anticipate contacts between groups of users can further improve those applications as well as those that use real-time flows by estimating future bandwidth requirements and therefore improve their reservation mechanisms (e.g., [12], [28]).

A contact is usually defined as a moment where two devices are within transmission range or can communicate mediated by a single Access Point (AP). Contact patterns are usually estimated by applying a statistical fitting on multiple metrics of a data set produced by a population of individuals. The goal is to find a statistical distribution that provides a good approximation to the metrics of interest and that can be evaluated in run-time. One of the most frequently cited metrics is the inter-contact time (ICT), defined as the time interval between two consecutive contacts of the same two peers. ICT is used in multiple applications and frequently modelled using a power law distribution [2], [17], [18].

An excessive focus on ICT can limit the range of optimisation strategies made available for application developers. The capability to predict recurring contacts between groups with more than 2 elements or the number of devices in proximity would allow social networks, user assistance applications, geo-replicated databases and even Machine-to-Machine (M2M) networks to minimise data redundancy in cooperative caching, improve availability and reduce communication cost.

Unfortunately, the design of such metrics is severely constrained by the large amounts of mobility data required to give statistical relevance to any modelling effort. This paper gives a step in this direction by presenting and analysing a data set prepared from raw data extracted from the eduroam wireless network site on the Polytechnic Institute of Lisbon. The data set, originally presented in [11], registers all the accesses of the 76,479 distinct devices to each of the network's 239 access points (APs) between 2005 and 2013.

The paper studies the dimension of groups of devices of any size observed in the data set and the repetition patterns of their meetings. In addition, it shows that in most cases, a *Pareto* statistical distribution (with parameters differing with the group size and recurrence period) can be used to model the group contact recurrence probabilities. These findings confirm the relevance of the *Pareto* distribution in the modelling of contacts, complementing, supporting and validating with a large sample what has been previously observed for pairs in data sets with a considerably smaller dimension. In addition, the paper extends existing results for contacts between large groups of devices and using broader time spans.

The modelling of recurring contacts inspired a ranking algorithm that, for some device, creates an ordered list of the devices most likely to be in contact in some future day. The algorithm is fully distributed, with each device constructing its own ranking, supported by its previous observations. The algorithm was validated against a mobility scenario extracted from the same data but with a different methodology and year, and also against a trace of GPS positions of Taxis in Rome. Results show that although distinct, both environments share the same statistical properties, allowing the ranking algorithm to improve the probabilities of success in predicting the devices that will be in range in a future day.

Finally, the paper proposes an application of the ranking algorithm on a reputation management service for mobile

- *N. Cruz is with ADEETC, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Lisbon 1549-020, Portugal and LaSIGE, Faculdade de Cincias, Universidade de Lisboa, Lisboa 1649-004, Portugal. E-mail: ncruz@deetc.isel.ipl.pt.*
- *H. Miranda is with LaSIGE, Faculdade de Cincias, Universidade de Lisboa, Lisboa 1649-004, Portugal. E-mail: hmiranda@di.fc.ul.pt.*

Manuscript received 12 Feb. 2016; revised 20 June 2017; accepted 23 June 2017. Date of publication 20 Nov. 2017; date of current version 1 June 2018. (Corresponding author: Nuno Cruz.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2017.2722408

applications. The goal is to improve the service performance without increasing the number of contacts to a centralised reputation server. To address this goal the ranking algorithm is used to anticipate future contacts, improving the hit rate of the cache of reputation information. Results show that the ranking algorithm can contribute to improve rogue devices detection.

In brief, the contributions of the paper are three-fold. First, it presents and studies the patterns observed in a 9 year data set with 70,000+ WiFi devices. Second, it describes a statistical modelling of the recurrence of contacts on groups of any size. Finally, it presents an algorithm that leverages on the statistical model to predict future contacts between devices that have previously been in contact. The presentation of these results is organised as follows. Section 2 makes a brief survey of the related work and discusses possible applications of this effort. The characterisation of the data set and the methodology used for extracting and analysing the data is presented in Sections 3 and 4. Section 5 addresses our efforts in modelling contacts recurrence into statistical distributions. Experiments on predicting contacts using data from the previous sections is detailed in Section 6. Section 7 presents an application of the contact prediction algorithm, in particular, a reputation system for resource sharing on ad hoc environments. The conclusions and the directions of the future work are the focus of Section 8.

2 Related Work

The Haggle project [27] is a good example of a project that combined research on human mobility with its applications on mobile computing, data dissemination and opportunistic routing. Haggle characterized human mobility on two dimensions: inter-contact time between pairs of devices and contact duration. The project showed that ICTs tend to follow a Power Law with a Exponential Decay, something also supported by other studies (e.g., [17]). Haggle leveraged several applications, for example Bubble Rap [16], a socially influenced routing protocol that used Haggle's mobility traces to infer communities using K-Clique [23] and weighted network analysis algorithms [21]. Research on spatio-temporal communities [22] emerged from using the same data sets of Bubble Rap to combine the study on communities detection with the duration of the contacts. Results were applied in the problem of information dissemination in opportunistic networks.

In [4] the authors study ICTs in two data sets created using distinct approaches. One is based on the view of an external observer. In particular, on data collected from the access logs of WiFi networks. The second, named direct contact, contains records captured directly by the devices. These have been either produced by devices designed specifically to be carried by users or by exploiting the Bluetooth connectivity of mobile devices. Authors observed that the distribution of ICTs can be modelled by a power law when in the range between $600s$ and $86400s$ and attributed it to the human work day pattern. As a follow up, the paper shows how this result impacts current forwarding algorithms and makes suggestions for improvements. In contrast, the work presented in [17] used six data sets to conclude that the modelling of ICTs using a power law is adequate for intervals not exceeding half of the value found in [4]. When the duration is above $43200s$, an exponential distribution shows to be more adequate.

The authors in [25] focused on temporal communities and their relations instead of ICTs. Authors extracted temporal communities from four distinct data sets, the largest of which considering the observation of 97 nodes over 9 months. In spite of the small scale and duration of the study, authors presented two interesting findings: $i)$ a direct implication between the establishment of social communities and temporal communities; and $ii)$ the identification of one class of devices, those with a high contact rate that are rarely seen in temporal communities, but contribute significantly for the efficient content dissemination in opportunistic social networks. Social communities are equally the focus of SocialCast [7]. The project exploits the human tendency to share interests and locations to develop an efficient routing protocol for publish-subscribe on Delay-Tolerant Networks. The authors use Kalman filters for forecasting future contacts, based on previous collocation observations. SocialCast was among the first protocols supporting one-to-many communication for Haggle.

PreKR [15] is a framework that improves forwarding on opportunistic networks by using a kernel regression based estimation for link pattern prediction. Using historical observations of network maps on three data sets, one of which being Bubble Rap, PreKR determines the probability of the recurrence of a link between two devices. Authors show that PreKR outperforms all other prediction methods, including Prophet [19], a protocol for Mobile Ad Hoc Networks that routes messages according to the probability of the next hop to eventually become in contact with the destination. The distinguishing factor was the use of kernel regression, that allowed PreKR to achieve an accuracy of more than 90 percent.

Most of the above works are built on models obtained from data sets that are limited on the duration and on the number and type of devices. coMobile [29] is an interesting exception that leverages from very large scale data sets collected from public transportation access records and mobile cell towers to define a mobility pattern model. Unfortunately, the granularity of the data, reflecting either instantaneous human presence or coarse grain locations can hardly be used to infer contacts between users.

A survey of most mobility models as well as of the modelling process using real traces can be found in [14]. To the best of our knowledge, research on very large data sets, spanning several years and covering thousands of users is limited to the work described in this paper and whose initial results were published in [10]. This paper extends these results by demonstrating their applicability in a practical scenario using a reputation system as the supporting application. The eduroam data collected in the scope of this project was further applied in the development of Mobiplity,[1] a publicly available scenario generator for human mobility using exclusively real data [9].

3 Methodology

The data set used in this study aggregates the log records produced between January 1, 2005 and December 31, 2013 by all the Access Points (APs) composing the eduroam WiFi network at the Lisbon Polytechnic Institute (IPL).

IPL is the 7th largest high education institution in Portugal with approximately 1300 teachers and 15,000 students,

1. http://edata.e.ipl.pt

Fig. 1. Location of IPL sites.

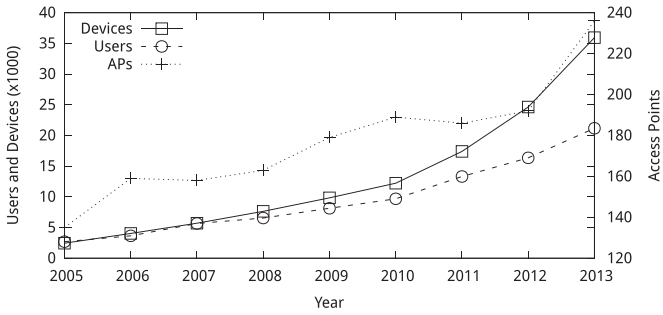


Fig. 2. Devices, users, and access points.

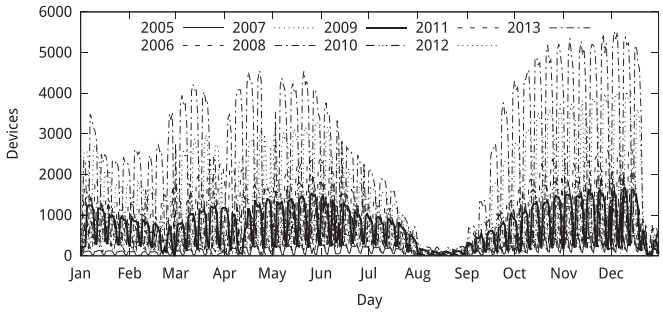


Fig. 3. WiFi devices connected per day.

distributed by 10 distinct campuses in the metropolitan area of Lisbon (see Fig. 1). The eduroam WiFi network results from an international effort aiming to transparently provide wireless Internet connectivity to its members in the campus of all adhering institutions. The IPL's site of the eduroam network is supported by approximately 200 Cisco Systems APs, covering a total of 26 buildings and inter-building areas. Records are originated from all the users that accessed the network at least once, thus including occasional visitors.

Fig. 2 shows a continuous growth of the number of users and devices although at distinct rates, specially since 2010. This is coincidental with an increase in the sales of smart-phones observed at the national level. The increase of the ratio between devices and users from 1.11 in 2005 to 1.70 in 2013 suggests a growing trend on the number of users accessing the network with more than one device. Fig. 3 depicts the number of devices that connect daily. As expected, the plot exhibits an irregular pattern consistent with the different activity levels that can be found on workdays, weekends and summer and winter breaks in a campus. The figure

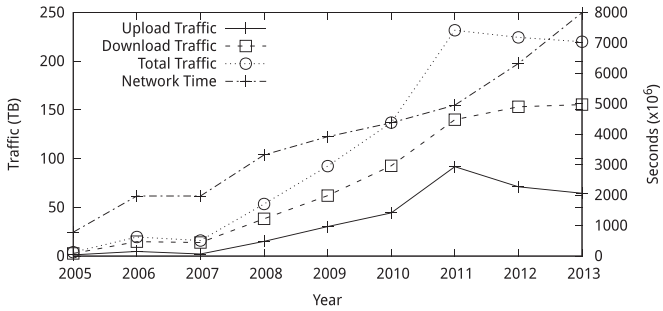


Fig. 4. Traffic and network time.

TABLE 1
Years Per User

#Years	1	2	3	4	5	6	7	8	9
Users	24,106	9,814	4,703	2,527	1,453	863	478	230	178

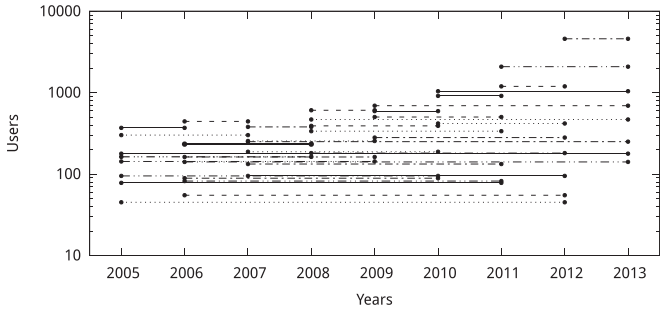


Fig. 5. Number of users observed in consecutive years.

equally indicates a consistent increase in the wireless network usage, with the average number of devices connecting daily to the network increasing 10 times in a 8 year time span (from 155 in 2005 to 1554 in 2012). It should be noted that neither of these gains can be attributed to an increase in the number of students of IPL, whose change cannot be compared to these ratios. The evolution is further observed in the increase on the amount of traffic supported by the network, depicted in Fig. 4. The plot shows an interesting decay both in the upload and total series in the last years of the study which is attributed to the loss of popularity of P2P file sharing networks.

Table 1 aggregates the users by the number of years where they were observed. The table shows that the data set contains more than 20,000 users observed in two or more years, making the data set suitable for long term analysis on user mobility patterns. Of particular interest are the 178 that have persistently appeared in each of the 9 years of the study. Fig. 5 details these results by counting the number of users appearing exactly on each time span of two or more consecutive years. For clarity, the figure omits users found in a single year and those not found on consecutive years. As a consequence of the increasing number of users and devices, the latest years of the study are those that share more users. In particular, 2012 and 2013 share 9452 users, with 48 percent of them having been observed also in 2011.

Fig. 6 depicts the Complementary Cumulative Distribution Function (CCDF) of the users accumulated networking time per year. The figure shows a surprising stability across the years, with approximately 10 percent of the users staying connected for more than $10^6 s$ (277 hours). As depicted

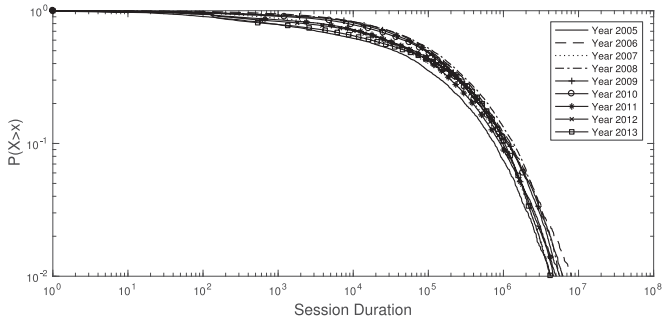


Fig. 6. Network time per user.

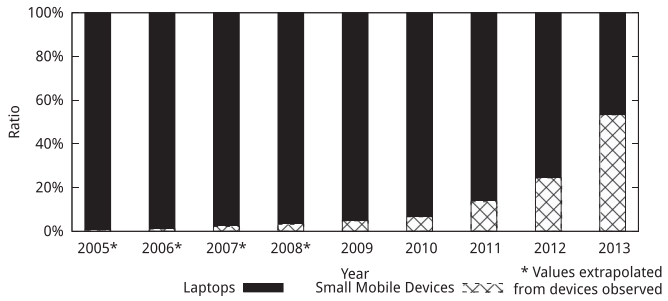


Fig. 7. Proportion of small mobile devices to laptops.

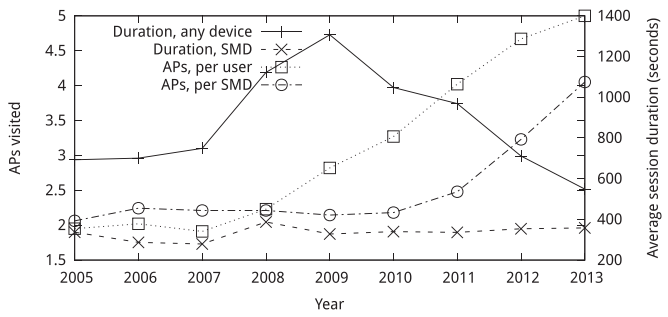


Fig. 8. Sessions duration and number of APs visited.

in Fig. 4, overall, the data sets hours used per year increases from 218×10^3 in 2005 to 2220×10^3 in 2013.

Fig. 7 compares the proportion of Small Mobile Devices (SMD) and Laptops accessing the network in each year, determined from the *vendor*, *parameter request list* and *hostname* fields of the DHCP messages exchanged between the devices and the supporting infrastructure. The analysis of this result is justified by the observation that small mobile devices tend to reproduce more accurately the users' movement patterns. The increasing penetration of SMDs on the eduroam network and its impact on the mobility is further supported by Fig. 8. In the case of SMDs, the figure shows a consistent increase in the number of APs visited per session although the duration of the sessions remains consistently around $400s$. This implies a reduction on the ratio of session time per AP, suggesting that users are increasingly walking in the campus with their smart-phones permanently connected to the WiFi network and therefore, increasing the accuracy of the mobility models extrapolated from this data. Examples of SMDs are smartphones, PDAs and tablets, using Windows CE, iOS and Android. The second class, Laptops, group the larger devices usually running over a classical operating system (Linux, Windows or OS X). The class of each device was determined by its operating system. Table 2 depicts the total dispersion of

TABLE 2
Devices per Operating System
(Sorted Ascending)

Operating System	Devices
Windows CE	432
OS X	1,514
iOS	7,431
Android	10,449
Linux	13,386
Windows	31,948

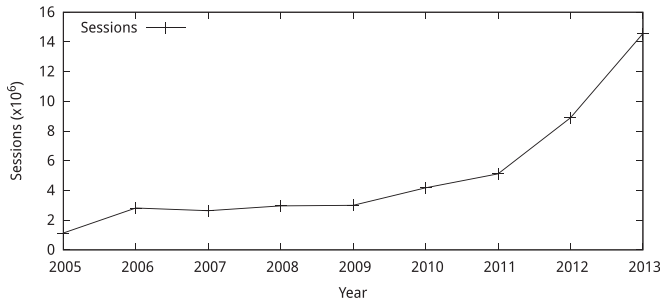


Fig. 9. Absolute number of sessions.

the 65,160 detected devices by the most popular operating systems. These results are only estimates given that: i) users are free to change the data sent by their DHCP client and, ii) recent Apple devices dont send vendor information. Devices with multiple operating system capabilities are represented once per operating system detected.

Contacts data evaluated in this paper are extracted from the RADIUS protocol [26] session logs, which considers the association of each user to a single AP. Fig. 9 reports on the number of sessions recorded each year. Log records contain the device MAC address, AP id, user name, session start and stop dates thus permitting to unambiguously identify the user, device and moment it was created. Prior to the analysis, logs have been purged from the following inconsistencies, that can be attributed to problems with the wireless network card drivers:

- Consecutive sessions between the same device and AP with an interval of less than 5 seconds have been merged in a single session;
- Time overlapping sessions S1 and S2 of the same device to distinct APs have been serialized by setting the stop time of S1 to occur at the moment immediately before the start time of S2. Given that network cards cannot be concurrently associated to more than one AP, this impossibility can only be explained if the device did not disassociate correctly from one AP before associating to the next with the former artificially establishing the session stop time by timeout;
- Sessions with the same start and stop time were removed. Sessions with these characteristics are created when a user has some issue while connecting to the network, although the network considers the user authenticated (thus creating the RADIUS record).

The RADIUS records purged from inconsistencies were then uploaded into one SQL Database. The DBMS played a key role in the continuation of the project. In particular, information extracted using standard SQL commands

TABLE 3
Temporal Communities Observed in the Data Set

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013
Max. TC Size	33	43	44	66	69	74	159	121	108
TCs	370,245	1,113,630	1,309,098	1,682,684	1,700,471	1,935,039	2,775,835	5,633,825	11,366,159
Average TC Size	6.45	8.16	8.32	9.37	9.96	11.08	10.77	10.08	11.5

contributed to diversify data' applications, which include discussions on the evolution of the utilisation of wireless networks and mobile devices, the generation of mobility scenarios and the study of the recurrence of contacts. The interested reader is referred to [11] for a more in depth analysis of the data set and to [9] for its applications on the generation of mobility scenarios.

3.1 Temporal Communities

In this paper, a temporal community (TC) is defined as the set of devices connected simultaneously to the same AP. A TC exists as long as its membership does not change. The addition and/or removal of any member results in the creation of a new temporal community. The approach is oblivious to associations of devices to distinct APs with overlapping coverage and to the repetition of TCs in the same day, which are ignored. Table 3 summarizes the TCs counted using this approach.

The membership of a TC is likely to reflect the presence of multiple groups and/or individual users that coincidentally associated to the same AP at the same time. This paper investigates the repetitive occurrence of groups of any number of devices, independently of in some occasions the members being or not part of a larger TC. This approach is better represented by the *Temporal Sub-Communities* (TSCs) of a TC, defined as each of the $\sum_{i=1}^{n} \binom{n}{i}$ distinct combinations of members that can be arranged from the TC with size n that created it. The paper will focus on the study of TSCs. However, it should be noted that this approach does not exclude the TC itself, as it is included in the set of TSCs it has originated.

The multi-year analysis depicted in Table 3 shows a non-negligible variation of the number and size of the communities. Part of this variation can be attributed to the addition of Access Points (APs) to the network (cf. Fig 2), mostly motivated by the need to resolve localised network bottlenecks. Such addition contributes to a decrease in the dimension of temporal communities as devices have more alternative APs for association on the most frequently accessed locations. In line with previous researchers (e.g., [4]), devices that are connected to the same AP are considered to be within transmission range.

Fig. 10, which depicts the size of the biggest TC observed each day, clearly shows the impact of the academic environment on the network. In the figure it is possible to observe the reduced activity during the Winter (end of December), Summer (August) and Easter (March) breaks (cf. Fig. 3). The irregularity of the plots can also be attributed to weekends and to local organization of events.

3.2 Temporal Patterns

The probability of recurrence of two and three consecutive hits of the same TSCs was evaluated on 4 distinct temporal patterns. The *Consecutive Days (CD)* and the *Consecutive Week Day (CWD)* patterns use intervals of respectively 1 and 7 days. These patterns serve to investigate repetitions inspired by common student activities, like the daily attendance to school and the weekly attendance to classes. The *Consecutive Month Day (CMD)* and *Consecutive Week (CW)* use more irregular patterns. CMD seeks for repetitions in the same day of the month of consecutive months. CW seeks repetitions in any weekday of consecutive weeks.

For clarity, and as an example, consider the observation of a TSC on July 18th, 2012 (Wed). A hit will be considered if the same TSC is observed on July 19th for CD, July 25th for CWD, August 18th for CMD and on any day between the 22nd and the 28th of July (Sun-Sat) for CW.

These temporal patterns are negatively affected by calendar irregularities. No attempt to attenuate the effects of public holidays, weekends or school breaks has been made. This option was chosen to approximate the results from those found by some application using past experiences to estimate the probability of contact repetition.

4 Generic Data Analysis

The accumulated number of Temporal Sub-Communities (TSCs) found in every day of 2012 is depicted in Fig. 11 as a Complementary Cumulative Distribution Function. For clarity, the figure presents TSC sizes in steps of 6. It was observed that the lines of omitted TSC sizes evolve similarly to those that are represented.

A first surprising effect observed in Fig. 11 is the peak of the number of TSCs at size 38, of which more than 10^{20} can be found in 1 percent of the days of 2012. This can be attributed to the methodology used for determining TSCs. Recall from Section 3, that the paper considers all possible combinations of any TC elements as TSCs. In this case, each of the observed TCs of size 74 produces, by itself, $\binom{74}{38} \approx 1.7 \times 10^{21}$ TSCs with size 38. Still, the figure is illustrative of the potential number and size of the groups of devices within transmission range that can be found in academia. Notice for example that in 18 (5 percent) of the days of 2012 it was possible to find at least 10^{10} communities of size 62 and that, in spite of the approximately 150 class days per year at IPL, 292 (80 percent) of the days had more than 100 TSCs with 14 devices.

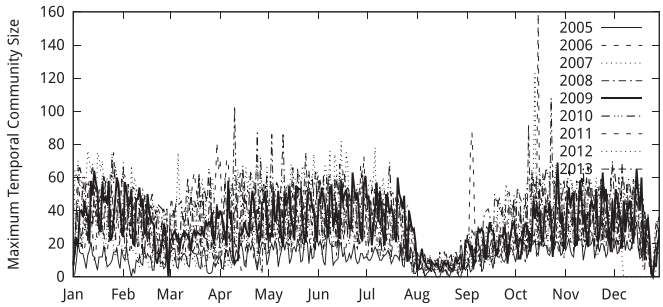


Fig. 10. Maximum temporal community size per day.

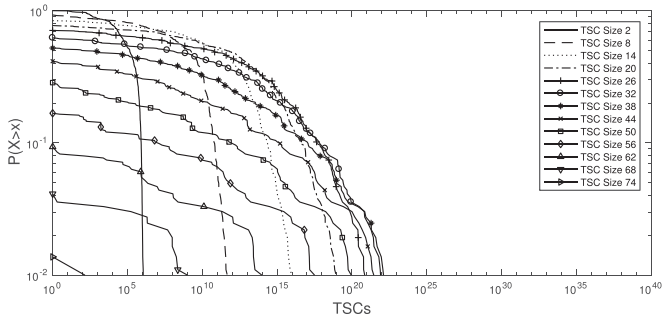


Fig. 11. TSCs per day.

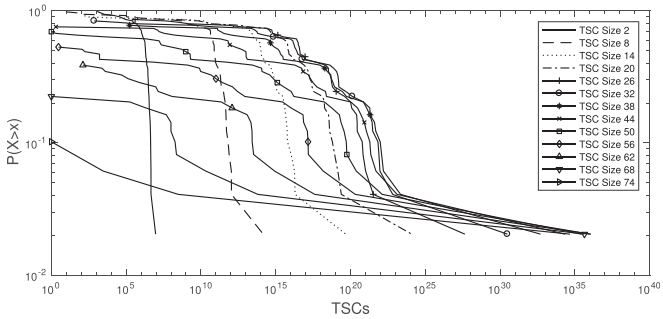


Fig. 12. TSCs per week.

Fig. 12 revisits these results after grouping the TSCs in weeks, to attenuate the spurious effects of occasional TCs of very large size, visible in the significantly distinct shape of the curves. Independently of the 30 class weeks, plots suggest that TSCs of sizes up to 23 tend to occur in a large proportion of more than 90 percent of the weeks and that 20 percent of the weeks have more than 10^5 communities of size 68 or less.

The large number of communities and the high frequency with which they were observed suggest that at least in the academic environment it is not hard to find large concentrations of users, motivating the research on different applications, for example on cooperative distributed databases [20] and on capacity estimation of wireless networks.

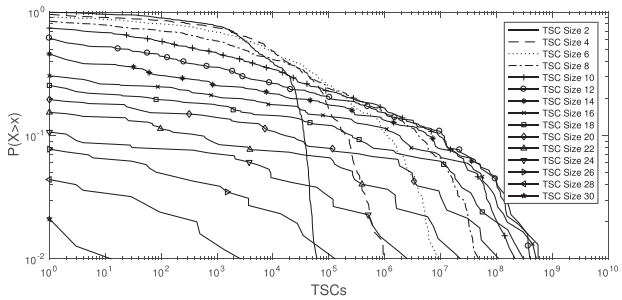
5 Recurrence of Contacts

For each Temporal Sub-Community (TSC) τ observed in some instant t, the study on τ's recurrence will proceed in two steps. First, it will measure the frequency with which τ is observed a second time, respecting one of the temporal patterns defined in Section 3. In other words, we will look for occurrences of τ in instant t', knowing that the relationship between t and t' must necessarily respect one of the temporal patterns. The second step will evaluate the persistence of these occurrences. It estimates the probability of observing τ in a third instant t'', with the time interval between t' and t'' respecting the same temporal pattern that was found between t and t'. The analysis will focus in 2012, considered to present a good trade-off between the manageability of the data set and its recency. Leveraging from these results, the section proceeds with the proposal of a probabilistic model for the occurrence of the third contact.

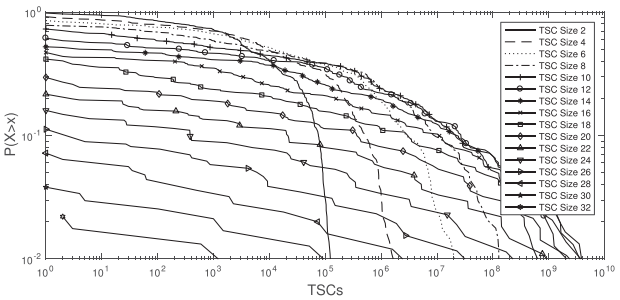
5.1 Observations of Second Contacts

Fig. 13 depicts the CCDF of the TSCs that were observed a second time satisfying each of the Temporal Patterns defined above. The figure clearly demonstrates that the selection of the temporal pattern has a strong impact on the results. Consecutive Month Days is the temporal pattern that performs poorly. This is expected as it is hard to find any human routines depending on the day of the month in the Portuguese academic environment. In contrast, the CD and CWD temporal patterns, which reflect better the typical student schedule, perform reasonably well, specially for TSCs of 6 or less users. In these cases, more than 90 percent of the days presented 100 or more TSCs which were equally observed in the previous day or week.

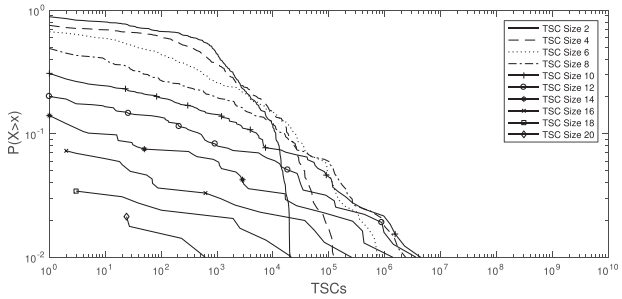
The best results are presented by the CW temporal pattern, where it was not hard to find 10,000 communities of



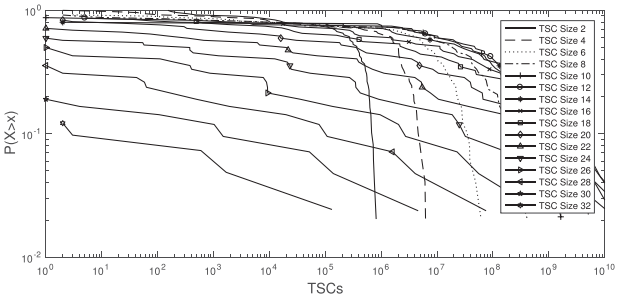
(a) TSCs on two consecutive days (CDs)



(b) TSCs on two consecutive week days (CWDs)



(c) TSCs on two consecutive month days (CMDs)



(d) TSCs on any day of two consecutive weeks (CWs)

Fig. 13. Temporal patterns for two consecutive periods.

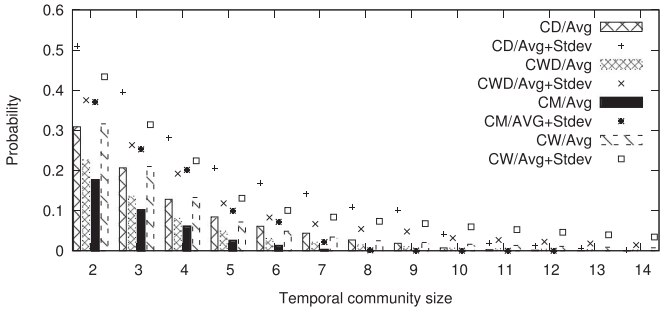


Fig. 14. Probabilities for three consecutive periods.

sizes 6 or less equally observed in the previous week in 90 percent of the days. This is not a surprising result considering the great flexibility of the constraints imposed by CW in comparison with CD and CWD.

5.2 Observations of Third Contacts

Fig. 14 shows the average and standard deviation of the proportion of TSCs that repeated in a third consecutive instant from those that were observed twice. Results contribute to decrease the relevance of the observations of large TSCs in two consecutive periods. Although it is frequent to find large TSCs, their membership tends to vary with time. Therefore, the occurrence of large TSCs can only be used by applications depending on ad hoc concentrations of users.

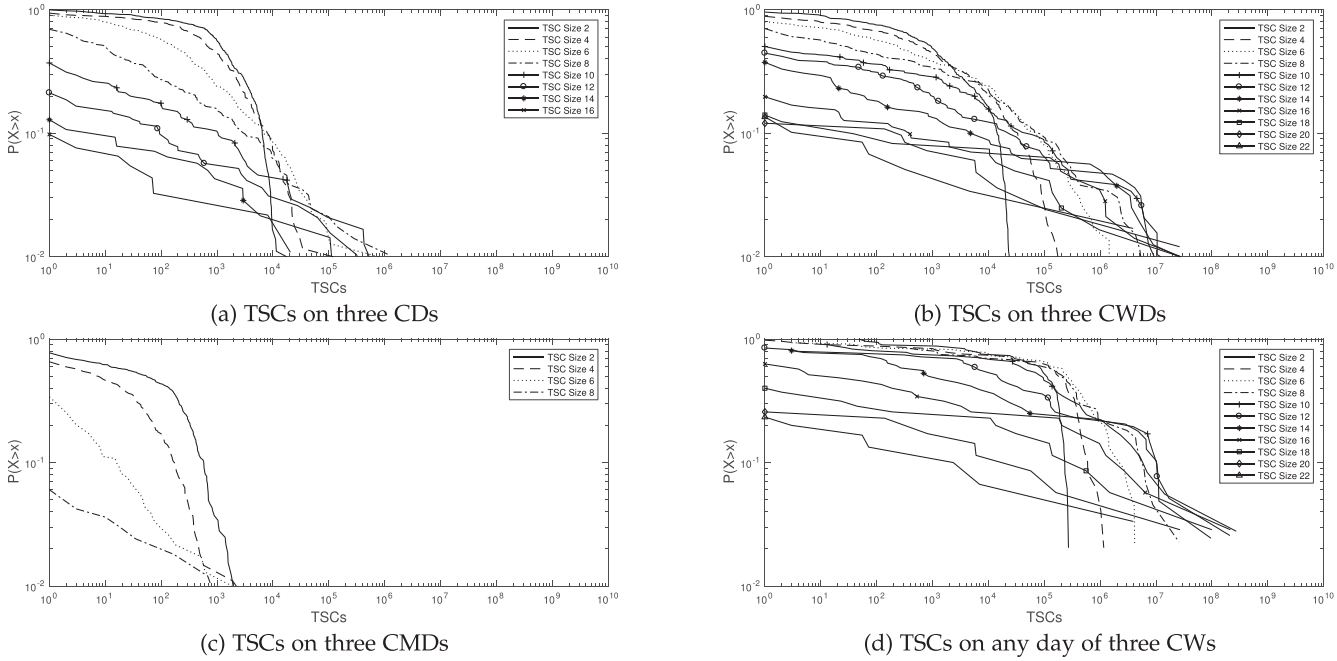
Concerning TSCs with small number of members, the results permit a separation of the CD and CWD temporal patterns, with the daily one showing to be more predictable. CD and the more relaxed CW temporal patterns are the only able to obtain probabilities of repetition above 10 percent for TSCs of up to 4 members and to show a surprisingly 30 percent for TSCs of size 2. This can be considered as a non-negligible probability of finding the same device on, respectively, the next two days and weeks. CW outperforms CD in the stability of the predictions, as it presents a smaller standard variation of the sample.

Fig. 15 depicts these results using CCDFs of the absolute number of occurrences observed. As suggested by Fig. 14, TSCs with significant results are those with small membership sizes. It is also interesting to notice the distinct pattern exhibited by each temporal pattern. Still, it should be noticed that it is not hard to find a considerable number of TSCs satisfying the CD and CW criteria for 3 consecutive intervals. In the case of CW, in 90 percent of the days of 2012 it is possible to find 100 TSCs of 6 members that were observed on 3 consecutive weeks. As for CD, more than 100 TSCs of size 6 can be found in 10 percent of the days, In line with what was observed before, the CWD and CMD temporal patterns show worst results, in both the number of TSCs found and on the probability of their recurrence. Although relatively rare, the study was able to find TSCs of size up to 22 occurring in the same day of the week for 3 consecutive weeks, possibly reflecting the weekly attendance to some lecture. For TSCs with very small sizes (up to 4) more than 100 repetitions were found in 50 percent of the cases.

5.3 Probabilistic Model

To model the temporal patterns observed in TSCs, the results above were fitted to statistical distributions using Matlab's Akaike information criteria. Fig. 16, which aggregates TSC sizes by distributions, shows that the Generalized Pareto distribution is the most adequate to model the behaviours observed in the paper. Our findings are consistent with results found for modelling other aspects of human mobility, for example those detailed in [5], [13], [17], although using considerably smaller scale samples.

Interestingly, the CD, CWD and CMD temporal patterns exhibit an exception where only a single size of the TSCs is better represented by Generalized Extreme Value. As depicted in Table 4, the sizes of the TSCs with an abnormal behaviour are distinct for each temporal pattern and no relation between the values could be found. Therefore, these



(a) TSCs on three CDs

(b) TSCs on three CWDs

(c) TSCs on three CMDs

(d) TSCs on any day of three CWs

Fig. 15. Temporal patterns for three consecutive periods.

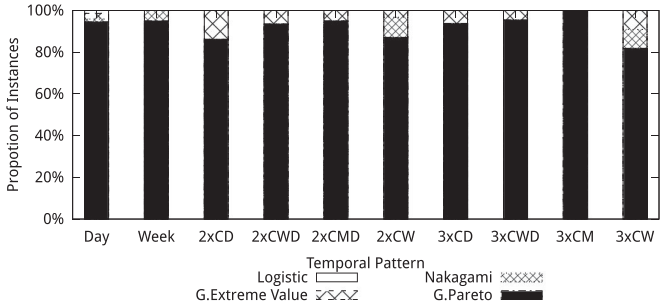


Fig. 16. Distribution fitting.

TABLE 4
TSC Size with Exceptional Distribution

Temporal Pattern	TSC Size
CD	6
CWD	3
CMD	2

cases are considered as an anomaly and the reminder of the text handles them indifferently from the remaining.

A practical application of these results can be obtained by reproducing the approach discussed in Section 5.2 to create a fitted function that returns the probability of occurrence a third time of a group of size n. Results are approximated by function PC presented in Eq. (1), modelled by constants a and b presented in Table 5 and graphically depicted in Fig. 17.

$$PC_{tp}(n) = a_{tp} \times e^{b_{tp}n}. \quad (1)$$

Standard deviation can be approximated by a function $SC_{tp}(n) = a_{tp} \times e^{b_{tp}n}$, which is similar to function PC although with a distinct set of constants a and b, equally depicted in Table 5.

6 Contact Prediction Algorithm

The capability to anticipate user contacts is valuable to a multitude of applications, which can be arranged according to the observer, in 3 distinct categories. In the *omniscient observer* category, a centralised server has access to the list of all contacts that have occurred in the past. An example is a reputation server that combines the contacts of all the service members to anticipate those that will occur in the future. The omniscient observer perspective is the one supporting the theoretical analysis of the previous section. In the *localised* category, some external observer, for example an access point, creates a local perspective of the contacts that occur between all the service members within his observation area. Finally, in the *peer view* category, each device anticipates future contacts exclusively from those where he has participated in the past. By not using any centralised entity, *peer view* is the one with a broader applicability, covering more restrictive wireless network models, for example mobile ad hoc networks (MANETs).

This section reports on our efforts to create an algorithm capable to anticipate future contacts with a reasonable accuracy in the peer view perspective. The algorithm leverages from the probabilistic model defined for the *ominiscient observer* perspective and weights the probabilities of contacts with the duration of the previous occurrence of each contact.

TABLE 5
Probability Function Parameters

Temporal Pattern (tp)	Probability		Std. dev.	
	a	b	a	b
CD	0.7167	-0.4204	0.3335	-0.205
CWD	0.6081	-0.4971	0.2375	-0.2216
CMD	0.5947	-0.601	0.4295	-0.357
CW	0.7701	-0.4431	0.1499	-0.1391

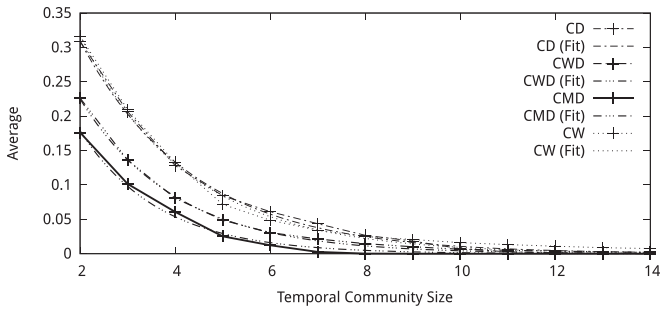


Fig. 17. Probability fitting results.

As a result, the algorithm presents a rank, ordering devices by the likelihood of contacting them again in the future.

6.1 Algorithm Description

The algorithm accepts a target date and a list of the contacts observed in the past, tagged with their date, duration and peer ID and outputs a list of peers, ordered by the likelihood of finding them on the target day. Members of the output list are all the peers from the input contact list that satisfy at least one of the CD, CWD or CMD temporal patterns (see Section 3.2) for the two previous consecutive instances and which, if observed on the target date, will result in a third consecutive occurrence of the pattern.

The algorithm ranks peers according to the *score* function, calculated independently for each device d and given in Eq. (2).

$$score(d) = f_{CD}(d_{CD}) + f_{CWD}(d_{CWD}) + f_{CMD}(d_{CMD}), \quad (2)$$

where d_{CD}, d_{CWD}, d_{CMD} are the duration (in seconds) of the contact between the node running the algorithm and device d in the last event of the corresponding pattern. The *score* for each node is therefore dictated by a sum of 3 components, one for each temporal pattern where d was observed. Each of the components is given by the product:

$$f_{tp}(d) = w_{tp} \times PC_{tp}(2) \times CD_{tp}(d), \forall tp \in \{CD, CWD, CMD\}, \quad (3)$$

which combines, $i)$ the weight attributed to the temporal pattern (w_{tp}); $ii)$ the probability of the occurrence of the third contact, as given by Eq. (1) (PC_{tp}); and $iii)$ the weight attributed to the duration of the last recorded contact between the two peers (CD_{tp}).

Preliminary experiments showed that CD_{tp} should privilege longer contacts, mapping them on proportionally heavier weights. Equations 4 and 5 present the two classes of functions experimented to model this property. Both assume that the contact duration is bounded between $60s$ and $86400s$ (one day), considered to be the interval

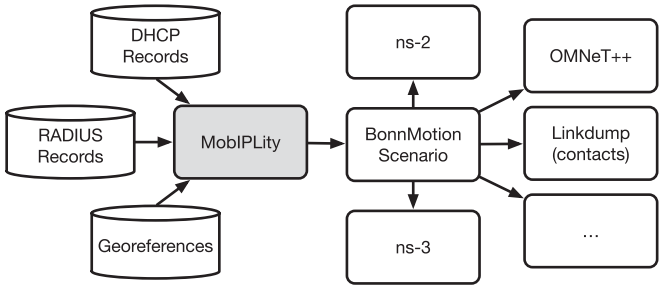


Fig. 18. MobIPLity.

representing social interactions between peers.

$$CD(d) = \frac{kd}{86400 + (k-1)d}, \quad k \geq 1, d \geq 60 \tag{4}$$

$$CD(d) = \left(\frac{d}{86400}\right)^2, \quad d \geq 60 \tag{5}$$

Functions differentiate by the direction of their curve. The family of functions of Eq. (4) increases the weight of shorter contacts with k. In contrast, Eq. (5) tends to decrease the relevance of shorter contacts, increasing the weight more rapidly as the duration approaches 86,400.

Overall, the algorithm leverages from the probabilities of occurrence of a third repetition of a contact between a pair of nodes, which were found in the analytic study presented in the previous section, to derive a ranking algorithm. The algorithm uses the duration of the contacts and multiple temporal patterns between the same pair of nodes as tie breakers. Expectations are that these tie breakers correctly identify the contacts that are more likely to occur, providing to applications accurate estimates of upcoming contacts with other nodes.

6.2 Evaluation

Evaluation was performed by running contact data sets against the algorithm and comparing its ordering with the contacts that have been actually observed. The capability of the algorithm to correctly order the expectations of contacts was evaluated using two metrics. Both measure the number of hits (defined as a prediction of contact that has effectively occurred), although using different perspectives:

The *Rank of the First Miss* (RFM) returns the position in the algorithm's prediction list of the first contact that was not observed. RFM is useful for application programmers as it provides an indication of the number of highly reliable predictions of the list.

The second metric compares the proportion of hits across the percentiles 10, 25, 50, 75 and 100 of the list. Percentiles permit to evaluate the quality of the ranking. Expectations are that the 100 percentile mirrors the analytic results discussed in Section 5. Therefore, the quality of the ranking will be evaluated by the increase in the proportion of hits in the lowest percentiles, which will confirm the capability of the algorithm to put hits effectively observed on the highest positions of the list.

6.2.1 Trace Generation on MobIPLity

The ranking algorithm was experimented using a mobility scenario generated by MobIPLity [9] with all devices that connected to the eduroam network on IPL during the year of 2013. MobIPLity uses the records produced by the RADIUS service to create mobility scenarios that closely reflect observed user behaviour. RADIUS identifies the participants and the timestamp of each association/dissociation event. The geographical coordinates of the access point used contribute with the estimate of the location of each participant. Scenarios are produced in bonnmotion [1] format, a popular mobility scenario generator capable of interacting with numerous network simulators, as detailed in Fig. 18.

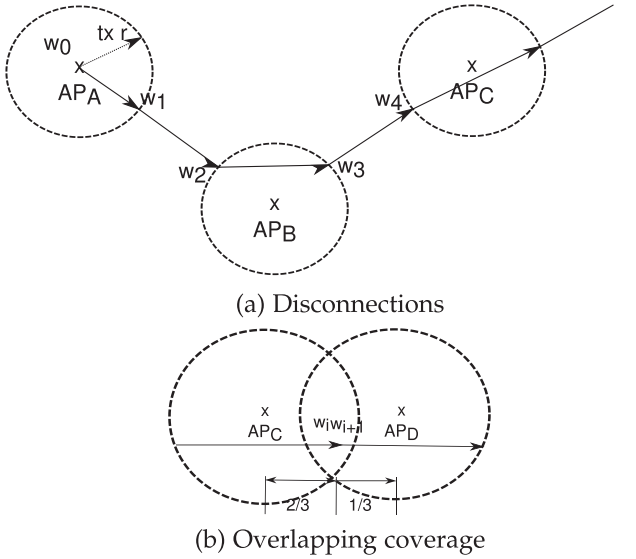


Fig. 19. Trace extraction examples.

The MobIPLity trace-set mobility model is created from a set $E \subseteq D \times A \times \{in, out\} \times T$ where D is the set of wireless devices, A the set of access points of the network annotated with their geographical coordinates and T are time stamps. The set is populated with 2 events $(d, a, in, t_1), (d, a, out, t_2)$, for each RADIUS log record, with t_1 and t_2 time-stamping respectively d's association/dissociation to AP a. Finally, let $E_d \subseteq E$ be the subset of E containing all the events recorded for device d.

E_d is expected to respect two invariants: $i)$ devices are always associated with an access point before being disassociated from it; and $ii)$ in any point in time, a device is associated at most to one access point. Note that invariant $i)$ is trivially assured by the access points software and invariant $ii)$ by the corrections applied to the RADIUS logs that have been outlined in Section 3.

We define $E'_d = e_{d,0}, e_{d,1}, \ldots, e_{d,n}, d \in D, e_{d,i} \in E_d, i > 0$ as the temporally ordered set of events for device d. Note that invariant $i)$ ensures that $e_{d,2j}, j \geq 0$ are events of type *in* and, conversely, $e_{d,2j+1}, j \geq 0$ are all events of type *out*.

A trace $W_d = w_0, w_1, \ldots, w_{2n-1}, w_i = F(e_{d,2j+i}), 0 \leq i \leq 2n-1$, $j, n \geq 0$ for some device d, is defined as a sequence of way-points w_i. The way-points are defined by a geographical coordinate and a time stamp, returned by a function F applied to consecutive events (not necessarily starting on $e_{d,0}$) in E'_d.

The output of function F depends of the position of the way-point on the sequence and of the type (in, out) of the event. The general case is depicted in Fig. 19a. w_0 is set with the time stamp of $e_{d,2j}$ and the coordinates of the access point in the first of the sequence of events selected. Subsequent transformations of pairs of events on pairs of Way-points $w_{2i+1}, w_{2i+2}, i \geq 0$ will return coordinates overlapping a vector $\overrightarrow{AP_A AP_B}$, with AP_A, AP_B being the coordinates of the

TABLE 6
Evaluation of Contact Duration Functions

CD_{CD}	CD_{CWD}	rfm	p10
$k = 4$	$k = 5$	3.61	0.40
$k = 4$	$k = 6$	3.60	0.40
$k = 2$	$k = 2$	3.58	0.40
$k = 3$	$k = 6$	3.57	0.40
Eq. (5)	Eq. (5)	3.16	0.39

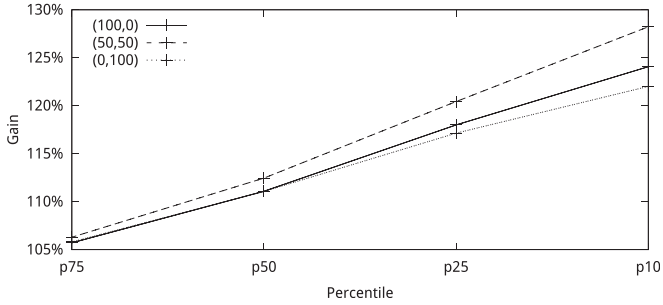


Fig. 20. Improvement observed (baseline p100=100 percent).

access points in the corresponding events $e_{d,2j+2i+1}, e_{d,2j+2i+2}$. The precise locations are dictated by the predefined transmission radius of the access points, as w_{2i+1} (resp. w_{2i+2}) will be placed at the intersection of the vector with the transmission radius of AP_A (resp. AP_B). Time stamps of w_1 and w_2 are copied from the corresponding events. Notice that, according to the definition of E'_d above, events $e_{d,2j+2i+1}, e_{d,2j+2i+2}$ are respectively an *out* and an *in* record, thus signalling the moment at which d abandoned the area covered by AP_A and the moment at which d associated with AP_B. The algorithm is successively repeated for each pair of events and way-points.

In the particular case where the coverage area of two consecutive access points visited by the device is not empty (Fig. 19b), the algorithm reflects the conservative approach of wireless interface drivers. The two way-points receive the time stamp of the *in* record and are set at 2/3 of the distance between the access points, chosen to reflect expected driver behaviour by selecting the strongest signal only after ensuring that the difference between both signals is enough to ensure better connectivity.

Traces are terminated at an *out* event by creating a way-point with the coordinates of the access point. It is assumed that the device abandoned the eduroam network and therefore that the trace must be terminated when the speed for traversing the distance between two consecutive access points falls below some threshold. Alternatively, traces are terminated when two consecutive connections to the same AP exceed a time threshold, suggesting that the users have abandoned and returned to the location.

6.2.2 Evaluation Results in MobIPLity

Contact data was extracted from MobIPLity traces by configuring the *LinkDump* application of bonnmotion to extract the periods in which two peers were within a 50 m range from each other for a minimum of $60s$. To prevent disturbance on the results due to the distinct patterns found on weekends, the original data set was purged from the events occurring on Saturdays and Sundays.

It should be noted that the data set used in the evaluation is considerably distinct from the one used in Section 5 for the analytic evaluation and which supported the rationale of the ranking algorithm. The latter evaluated the 2012 data set and defined a contact as the simultaneous association of two or more devices to the same access point, using RADIUS records. This section uses the 2013 data set and the distinct methodology for defining contacts presented above. In addition to exposing the ranking algorithm to a considerably distinct data set from the one that inspired it, this approach permits to verify if the properties observed during the year 2012 are reproducible on a different year.

Parameters for the algorithm were experimentally tuned in order to obtain the best results for the metrics used in the evaluation. Table 6 depicts the experimental results for multiple variations of the CD functions with equal weights for w_{CD} and w_{CWD}. Results evidence a minimal impact of the k constant when the function of Eq. (4) is used, in contrast with the results exhibited by Eq. (5). In practice, this result evidences a preference of the algorithm for a fast growing of the weight of the contact duration in the ranking. As a result, the reminder of the text presents results using Eq. (4) with $k = 4$ for all the CD_{tp} functions.

Table 7 shows the average and standard deviation of the metrics when different weights are used. These results average the rankings produced for all devices and days, provided that the ranking contained 20 or more devices. The table shows some encouraging results. In particular, that the algorithm can, on average, correctly predict the contact with the first 3.6 devices of the list and that 40 percent of the highest 10 percent ranked devices have been found as predicted. The contribution of the algorithm becomes more evident by noticing that a random sort of the list (p100) would equally distribute the 31 percent of the devices on the list that were effectively observed by all the percentiles.

Fig. 20 further emphasis this result by evidencing the 28 percent performance gain of p10 over p100. A combined analysis of the Fig. 20 and of the Table 7 highlights the distinct contribution of each of the temporal patterns to the algorithm, with the participation of the CMD or the use of individual patterns consistently presenting worse results than a 50,50 percent combination of weights of CD and CWD.

In the elaboration of the results above, a ranking list is always prepared, independently of the connectivity of the

TABLE 7
Results (Average Per Day and Standard Deviation)

w_{CD}	w_{CWD}	w_{CMD}	Rank Totals	rfm (σ)	p10 (σ)	p25 (σ)	p50 (σ)	p75 (σ)	p100 (σ)
50	50	0	56,892,626	3.609 (6.75)	0.402 (0.38)	0.377 (0.35)	0.352 (0.32)	0.333 (0.30)	0.313 (0.29)
50	0	50	38,423,918	3.352 (7.13)	0.361 (0.37)	0.340 (0.34)	0.317 (0.32)	0.301 (0.30)	0.283 (0.28)
33	33	33	65,604,286	3.326 (6.30)	0.363 (0.37)	0.340 (0.34)	0.316 (0.32)	0.297 (0.30)	0.279 (0.28)
0	50	50	40,827,514	2.690 (3.81)	0.360 (0.37)	0.342 (0.35)	0.322 (0.33)	0.306 (0.31)	0.288 (0.29)

TABLE 8
Results Excluding not Connected Days (Average per Day and Standard Deviation)

w_{CD}	w_{CWD}	w_{CMD}	Rank Totals	rfm (σ)	p10 (σ)	p25 (σ)	p50 (σ)	p75 (σ)	p100 (σ)
50	50	0	39,620,508	5.122 (8.11)	0.635 (0.28)	0.596 (0.24)	0.557 (0.22)	0.526 (0.21)	0.495 (0.20)
33	33	33	43,290,708	4.967 (7.82)	0.620 (0.28)	0.579 (0.25)	0.538 (0.23)	0.507 (0.21)	0.476 (0.20)
50	0	50	25,271,814	4.871 (8.82)	0.593 (0.29)	0.559 (0.26)	0.522 (0.24)	0.495 (0.23)	0.466 (0.21)
0	50	50	25,861,694	3.917 (4.64)	0.621 (0.28)	0.591 (0.25)	0.556 (0.23)	0.528 (0.22)	0.498 (0.21)

TABLE 9
Per Day of the Week Metrics for Setup with the Highest Improvement (wCD=50,wCWD=50)

	rfm	p10	p25	p50	p75	p100
Monday	3.15 (4.62)	0.4 (0.38)	0.38 (0.35)	0.35 (0.32)	0.34 (0.31)	0.32 (0.29)
Tuesday	4.13 (7.57)	0.45 (0.38)	0.42 (0.35)	0.39 (0.33)	0.37 (0.31)	0.35 (0.3)
Wednesday	3.97 (8.26)	0.41 (0.38)	0.39 (0.34)	0.36 (0.32)	0.34 (0.3)	0.32 (0.28)
Thursday	3.67 (6.65)	0.41 (0.38)	0.38 (0.35)	0.36 (0.32)	0.34 (0.31)	0.32 (0.29)
Friday	3.04 (5.65)	0.34 (0.36)	0.32 (0.33)	0.29 (0.3)	0.27 (0.28)	0.26 (0.26)

TABLE 10
Taxis in Rome Trace Results

w_{CD}	w_{CWD}	Rank Totals	rfm (σ)	p10 (σ)	p25 (σ)	p50 (σ)	p75 (σ)	p100 (σ)
50	50	3487	1.884 (1.39)	0.426 (0.49)	0.409 (0.43)	0.380 (0.33)	0.340 (0.28)	0.312 (0.25)

device. However, many cases were found where some devices did not connect to any other device in one full day, although the algorithm predicted some connections. Table 8 confirms that this cannot be considered a negligible aspect. The table presents the same metrics after excluding the lists prepared for these devices. Not surprisingly, lists become much more accurate, with p100 approaching an average of 50 percent. In other words, on average, 50 percent of the devices predicted by the algorithm are effectively found. More demanding metrics, in particular RFM and p10 are in line with the improvement of p100: on average, the first 5 devices of each list are effectively found as predicted as well as more than 63 percent of the percentile 10 of each ranking list.

Table 9 exhibits the distribution of the results by day of the week. It is interesting to notice that the performance of the algorithm is not uniform across all the weekdays. The ranking algorithm presents better results for Tuesday, Wednesday and Thursday. The worst results of Mondays can be attributed to the weekend discontinuity impact on the CD temporal pattern. Surprisingly, Fridays present the worst performing results, although no evident explanation could be found.

6.3 Evaluation with Taxi Traces

To understand the applicability of the algorithm in a broad range of scenarios, the algorithm was experimented in a data set containing 1 month GPS traces of 320 taxis in Rome [3]. The data set was sanitised to include only positions in the metropolitan area of Rome, and to mark as off-line the taxis not reporting their position for intervals above $120s$. The transmission range and all other variables required for defining a contact replicate the values used in the previous scenario.

Table 10 shows the metrics presented by the algorithm. Unfortunately, the smaller number of nodes and the shorter length of the data set prevented experiments with the CMD temporal pattern and forced to a reduction of the minimum size of the rankings from 20 to 5. Surprisingly, p10 metric shows values comparable to the ones obtained from MobIPLity, for the same configuration parameters. The differences in the RFM can be attributed to the smaller dimension of the data set, which necessarily reduces the ranking list and, proportionally impacts RFM. The difference between p10 and p100 loses significance, with the gain decreasing to 11 percent.

Table 11 show the outcome of the per day of the week analysis in this data set. It is interesting to observe that Monday is the worst performing day of the ranking algorithm. This result is attributed to the discarding of weekends that was kept from the MobIPLity analysis in an attempt to keep the comparison fair. However, the social constraints that encouraged the introduction of the exception for MobIPLity have no significance in a taxis scenario, where devices are expected to operate on all days of the week. To the extent of our knowledge, this was the unique characteristic of the algorithm which did not adapt to both scenarios.

6.4 Discussion

In contrast with our expectations, differences in results between CD and CWD temporal patterns tend to be orthogonal to the environment. As an example, one could consider that the CD temporal pattern better represents faculty (with a daily schedule), CWD would better represent students that meet in classrooms following a weekly schedule and taxis movements would not reflect neither of these assumptions. However, the results obtained using the data from MobIPLity present similar outcomes to the ones obtained in Section 5.3, using the same data but on different years. Furthermore, and considering that the Taxis in Rome trace also present the same results, we are encouraged to consider that these results support the applicability of the algorithm and probability modelling on multiple environments.

Results suggest that performance could be improved by considering weekdays in the ranking algorithm. However, this claim must be supported by additional experiments in other traces, and therefore, is left as future work.

TABLE 11
Per Day of the Week Metrics for Taxis in Rome, Setup with the Highest Improvement (wCD=50,wCWD=50)

	rfm	p10	p25	p50	p75	p100
Monday	1.27 (0.61)	0.19 (0.39)	0.2 (0.33)	0.18 (0.21)	0.16 (0.17)	0.15 (0.16)
Tuesday	1.87 (1.18)	0.48 (0.5)	0.42 (0.42)	0.41 (0.3)	0.39 (0.28)	0.35 (0.21)
Wednesday	1.95 (1.41)	0.44 (0.5)	0.41 (0.45)	0.4 (0.33)	0.34 (0.28)	0.31 (0.25)
Thursday	2.03 (1.5)	0.45 (0.5)	0.43 (0.45)	0.4 (0.35)	0.35 (0.3)	0.32 (0.25)
Friday	2.16 (1.58)	0.57 (0.5)	0.54 (0.41)	0.49 (0.3)	0.46 (0.27)	0.42 (0.24)

7 Sample Application

The contributions of the contact prediction algorithm to real world mobile applications were evaluated by extending a reputation system designed to support offline transactions [8]. The reputation system does not expect that mobile applications have a permanent Internet connection. Instead, it assumes that the applications periodically communicate with a central trusted entity (CTE). During the interactions with the CTE, mobile devices: $i)$ report to the CTE the experiences of the most recent transactions performed while offline and; $ii)$ retrieve from the CTE up to date reputation information about other users, in order to reduce the probability of having failed transactions in the future. The reputation information r_p ($r_p \in [-1, 1]$) of each participant p is affected by the outcomes of previous transactions as reported to the CTE by the participants. Transactions are assumed to imply the exchange of some virtual currency, deposited by the participants in a bank account at the CTE.

Applications use reputation information to decide if some transaction should be performed with other participants. Reputation information is signed by the CTE and therefore cannot be corrupted. However, considering that it can be obtained from the application's local cache, from other peers in the neighbourhood or directly from the CTE, one must assume that it can be outdated. The scenario of interest for this paper occurs when a misbehaving device presents outdated reputation information, representing the good behaviour it had in the past.

The contributions of the contact prediction algorithm emerge from the reputation system assumption on the limited amount of memory made available by the participants to cache reputation information of other users. The paper compares three possible caching policies. In the *probabilistic dissemination policy*, the CTE keeps a list of all participants ordered by their reputation, from the worst to the best. The reputation data delivered for caching to each participant is randomly constructed, by orderly traversing the list and inserting in the participant's local cache the $i\mathrm{th}$ member with probability $1/i$. The preference given to devices with a bad reputation is justified by noticing that correct devices will have no motivation for not presenting to potential candidates for transactions their most recent reputation certificates. On the second option, the *ranking policy*, the output of the contact prediction algorithm is used in order to create a rank of the devices ordered by the higher probability of being in range on the following period. From this ranking, the local device cache is filled with the devices that are considered selfish. The third policy is a variation of the previous, where the local cache includes all devices from the ranking, with the requirement that at least half of the cache includes devices that behave selfishly (bellow a predefined threshold). We call this last policy, the *biased ranking policy*. The biased ranking policy is motivated by the introduction of the more ambitious requirement of reducing the number of exchanged messages by dismissing the need of consulting neighbouring devices cache.

To evaluate the reputation system, we used the traces from the 2013 MobIPLity data set. After extracting the trace we used bonnmotion to obtain all pairs of devices that were within a $50\ \mathrm{m}$ range for at least $60s$. For each contact we considered that a transaction was possible every $10s$.

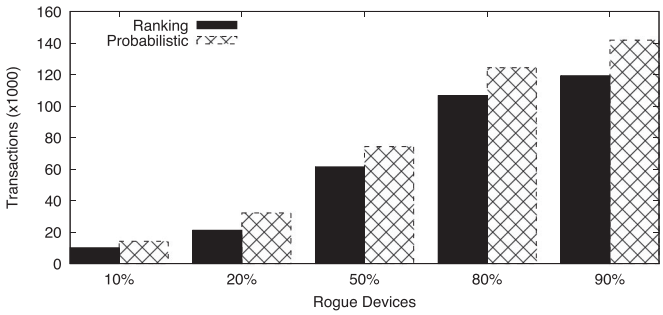
Simulation proceeds in days, with all devices concurrently connecting to the CTE at midnight. Each connection to the CTE is first used by the devices to report the previous contacts with the peers. In reply, the CTE uses the results presented in Section 5 to construct, for each connecting device, a set of the devices more likely to be observed in the following day and whose reputation is bellow a predefined threshold (in the experiment fixed at 0.0).

For the simulation, transactions between devices are considered to have a duration of 10 seconds. Each transaction has a cost of 10 virtual coins, delivered by the requester to the provider when a successful transaction is completed. Each device has an initial budget of 1000 coins, thus imposing a limit of 100 operations to selfish devices.

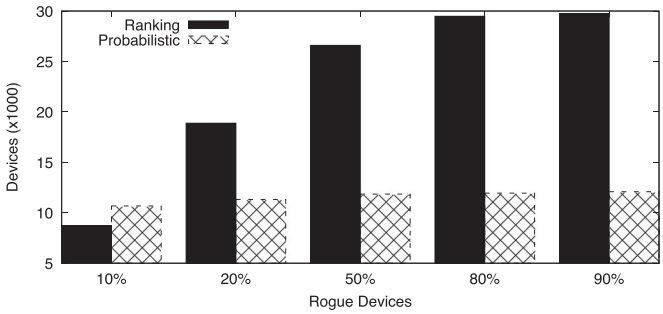
As depicted in Fig. 21, the ranking policy is able to reduce by 22 percent the number of bad transactions and increase in 38 percent the early detection of rogue devices using the local reputation cache. The exception is on the number of detected rogues by local cache when the proportion of rogue devices is 10 percent, which can be attributed to the fact that the probabilistic dissemination policy includes any reputation certificates, while the ranking policy used a predefined low threshold to determine which devices have their reputation certificate sent by the CTE. Despite the differences the number of detected bad transactions still shows improvement.

For the biased ranking policy we defined the selfish threshold as 0, meaning that half of the cache includes the highest ranking devices according with the contact prediction algorithm, independently of their reputation. The other half includes only devices from the ranking with a reputation lower than 0.

Fig. 22 shows the improvement on the number of good transactions, where reputation information came from the local cache, obtained using the biased ranking policy when compared with the probabilistic dissemination policy. This is further detailed in Fig. 23 where the percentage of successful transactions when the biased ranking policy is used is depicted according to the source of the certificate. Possible sources are the local cache or the



(a) Bad transactions



(b) Rogues detected by local cache

Fig. 21. Evaluation results of the ranking algorithm.

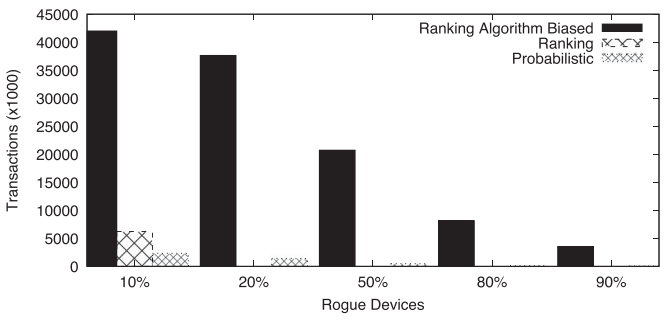


Fig. 22. Successful transactions where reputation came from local cache.

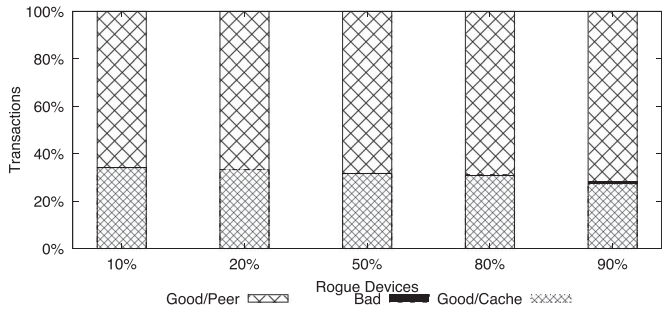
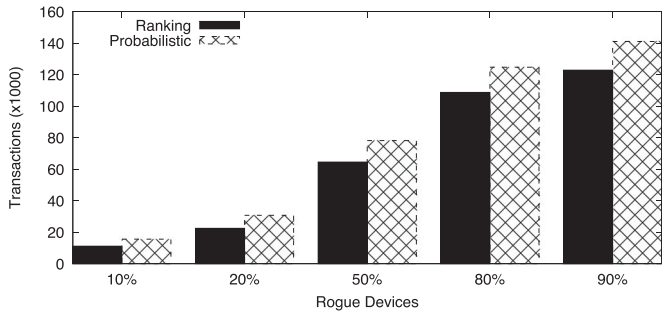
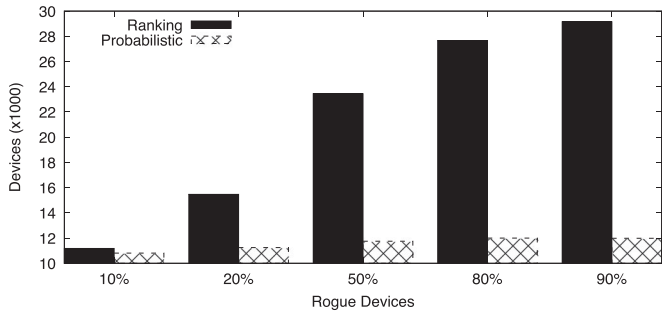


Fig. 23. Total transactions with ranking algorithm biased ranking policy.



(a) Bad transactions



(b) Rogues detected by local cache

Fig. 24. Improvement when using a biased ranking policy.

reputation sent directly by the peer device. Between 27 and 34 percent of all transactions reputation certificates were obtained from the local cache of the mobile devices. The low number of bad transactions is attributed to the benefits of using the contact prediction for the dissemination of reputation certificates.

Fig. 24 shows the impact of the introduction of the biased ranking policy on the bad transactions and number of rogue devices detected by local cache. The figure shows that the gains are retained, when comparing with the probabilistic dissemination policy.

8 Conclusions

Developers of mobile applications can be faced with the need of anticipating the number or affiliation of the groups of devices to be found in the future. This paper leverages on a large data set of accesses to the wireless network of an academic institution to extract the complete set of temporal communities observed between 2005 and 2013. The paper derives a statistical model that characterizes the different temporal community sizes assuming four distinct recurrence patterns that mirror typical user schedules.

The fitting of the observations showed that the recurrence of three temporal patterns can be modelled by generalized Pareto distributions, confirming and supporting the results already observed in studies with smaller scale samples for the Inter-Contact Times.

The paper presents an algorithm to predict contacts with peer devices on a nearby future using the knowledge of previously observed contacts and temporal patterns. The algorithm was evaluated against two data sets: one prepared by extracting data from a different year of the same data set that inspired this research and another prepared from a publicly available trace of taxis in Rome. Evaluation results showed that the algorithm presents good prediction capability and that its performance is comparable in both scenarios, which raises our expectations of its applicability as a generic prediction tool.

The prediction algorithm was applied to a reputation system for resource sharing in ad hoc networks. Experiments using the algorithm to drive the selection of the subset of reputation information to be cached by the participants provided a noticeable improvement in the

detection of rogue devices when compared with a probabilistic dissemination algorithm.

Analysis of the data continues. As future work, authors plan to apply the lessons learnt in the modulation of recurrence of temporal communities on real world applications. In addition, work will continue in the analysis and expansion of the data set and on the verification of the applicability of the results presented in the paper in other data sets.

Acknowledgments

The work described in this paper was partially supported by the project DoIT (PTDC/EEI-ESS/5863/2014) of the Fundação para a Ciência e Tecnologia, Portugal.

References

[1] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A mobility scenario generation and analysis tool," in *Proc. 3rd Int'l Conf. Simul. Tools Techn.*, 2010 pp. 51:1–51:10.

[2] C. Boldrini and A. Passarella, "HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships," *Comput. Commun.*, vol. 33 no. 9, pp. 1056–1074, 2010.

[3] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD data set roma/taxi (v. 2014–07-17)," Jul. 2014. [Online]. Available: http://crawdad.org/roma/taxi/

[4] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Trans. Mobile Comput.*, vol. 6 no. 6, pp. 606–620, Jun. 2007.

[5] A. Clauset, C. Shalizi, and M. Newman, "Power-law distributions in empirical data," *Soc. Ind. Appl. Math. Rev.*, vol. 51 no. 4, pp. 661–703, 2009.

[6] M. Conti, S. Giordano, M. May, and A. Passarella, "From opportunistic networks to opportunistic computing," *IEEE Commun. Mag.*, vol. 48 no. 9, pp. 126–139, Sept. 2010.

[7] P. Costa, C. Mascolo, M. Musolesi, and G. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE. J. Sel. Areas Commun.*, vol. 26 no. 5, pp. 748–760, Jun. 2008.

[8] N. Cruz and H. Miranda, "A hybrid trust and trade service for mobile collaborative computing," in *Proc. 7th Int'l Conf. Next Generation Mobile Apps Serv. Technol.*, 2013, pp. 1–6.

[9] N. Cruz and H. Miranda, "MobiPLity: A trace-based mobility scenario generator for mobile applications," *EAI Endorsed Trans. Ubiquitous Environ.*, vol. 15, no. 5, Jul. 2015.

[10] N. Cruz and H. Miranda, "Recurring contact opportunities within groups of devices," *EAI Endorsed Trans. Ambient Syst.*, vol. 15, no. 6, Aug. 2015.

[11] N. Cruz, H. Miranda, and P. Ribeiro, "The evolution of user mobility on the eduroam network," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2014, pp. 249–253.

[12] B. Dezfouli, M. Radi, and O. Chipara, "Mobility-aware real-time scheduling for low-power wireless networks," in *Proc. IEEE 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[13] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453 no. 7196 pp. 779–782, Jun. 2008.

[14] A. Hess, K. A. Hummel, W. N. Gansterer, and G. Haring, "Data-driven human mobility modeling: A survey and engineering guidance for mobile networking," *ACM Comput. Surv.*, vol. 48, no. 3, pp. 38:1–38:39, Dec. 2015.

[15] D. Huang, S. Zhang, P. Hui, and Z. Chen, "Link pattern prediction in opportunistic networks with kernel regression," in *Proc. The 7th Int. Conf. Commun. Syst. Netw.*, Jan. 2015, pp. 1–8.

[16] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10 no. 11, pp. 1576–1589, Nov. 2011.

[17] T. Karagiannis, J. Y. Le Boudec, and M. Vojnovic, "Power law and exponential decay of intercontact times between mobile devices," *IEEE Trans. Mobile Comput.*, vol. 9 no. 10, pp. 1377–1390, Oct. 2010.

[18] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "SLAW: A new mobility model for human walks," in *Proc. IEEE Conf. Inf. Comput. Commun.*, Apr. 2009, pp. 855–863.

[19] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7 no. 3, pp. 19–20, Jul. 2003.

[20] H. Miranda, S. Leggio, L. Rodrigues, and K. Raatikainen, "An algorithm for dissemination and retrieval of information in wireless ad hoc networks," in *Proc. 13th Int. Euro-Par Conf.*, 2007, pp. 891–900.

[21] M. E. J. Newman, "Analysis of weighted networks," *Phys. Rev. E*, vol. 70, no. 5, p. 056131, Nov. 2004.

[22] M. Orlinski and N. Filer, "The rise and fall of spatio-temporal clusters in mobile ad hoc networks," *Ad Hoc Netw.*, vol. 11 no. 5, pp. 1641–1654, 2013.

[23] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435 no. 7043 pp. 814–818, Jun. 2005.

[24] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks," *IEEE Commun. Mag.*, vol. 44 no. 11, pp. 134–141, Nov. 2006.

[25] A.-K. Pietiläinen and C. Diot, "Dissemination in opportunistic social networks: The role of temporal communities," in *Proc. 13th ACM Int'l Symp. Mobile Ad Hoc Netw. Comput.*, 2012, pp. 165–174.

[26] C. Rigney, "RADIUS accounting," Internet Eng. Task Force—Network Working Group, June 2000, Updated by RFCs 2867, 5080, 5997.

[27] J. Su, et al., "Haggle: Seamless networking for mobile applications," in *UbiComp 2007: Ubiquitous Computing*, Berlin, Germany: Springer, 2007, pp. 391–408.

[28] Z. Wang, H. Shah-Mansouri, and V. Wong, "How to download more data from neighbors? a metric for D2D data offloading opportunity," *IEEE Trans. Mobile Comput.*, vol. 16 no. 6 pp. 1658–1675, Jun. 2017, doi: 10.1109/TMC.2016.2604260.

[29] D. Zhang, J. Zhao, F. Zhang, and T. He, "CoMobile: Real-time human mobility modeling at urban scale using multi-view learning," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inform. Syst.*, 2015, pp. 40:1–40:10.

Nuno Cruz received the PhD degree with a thesis on collaborative computing between mobile devices from the University of Lisbon. He is currently an assistant professor in the Instituto Superior de Engenharia de Lisboa do Instituto Politcnico de Lisboa. He is currently an integrated researcher with the Large-Scale Informatics Systems Laboratory. He collaborated on several projects with the industry mainly on computer networks related topics. His main research interests include mobile computing, mobility prediction, wireless networks, railway communications, smart cities, and intelligent transport systems.

Hugo Miranda received the PhD degree from the Universidade de Lisboa. He is an assistant professor with the Faculdade de Ciências da Universidade de Lisboa, where he teaches courses on computer networks, middleware, and system administration. He is an integrated researcher of Lasige. His research interests include protocol composition frameworks and middleware for mobile computing, with a particular emphasis on mobile ad hoc networks. He was one of the leading developers of the Appia protocol composition framework and of the PAMPA message dissemination algorithm for MANETs. He was a co-editor of the *Middleware for Network Eccentric* and the *Mobile Applications* book.