

Does Mobility Matter?

An Evaluation Methodology for Opportunistic Apps

Jesus Frigal, Marc-Olivier Killijian, Roberto Pasqua, Matthieu Roy, Gilles Trédan
 CNRS, LAAS, 7 Avenue du colonel Roche, F-31400 Toulouse, France
 Univ. de Toulouse, LAAS, F-31400 Toulouse, France
 {surname.name}@laas.fr

Abstract—This paper presents a methodology to guide the evaluation of social distributed applications in mobile environments. Even when applications are already designed, they exhibit a number of tuning parameters upon which network operators can act in order to improve performance. Accordingly, evaluation can be a valuable tool to determine for a particular mobile application which is the most suitable parameters setup from a performance point of view. Our methodology can be of great interest in this tuning process, thus saving both time and money.

The main novelty of this methodology is the use of diversification to recreate mobile environments using both synthetic and real mobility traces. Our work focuses on how micro-mobility may impact social distributed applications. The feasibility of the paper is showed through a realistic microblogging case study.

Keywords—*Mobility models, Evaluation methodology, Opportunistic networks, Network simulation*

I. INTRODUCTION

The recent advances in mobile technology has led to an unprecedented growth in the number of connected devices per person. According to recent studies [1], by the year 2020 each person will own almost 7 devices with connection capabilities, thus leading to a more connected society. The new communication technologies introduced by such devices will enable the deployment of self-managing and self-adaptive networks, not requiring the need of fixed infrastructures (or at least minimising their use). Thus, the traditional client-to-infrastructure paradigm will evolve towards a model where part of the infrastructure will be integrated within mobile clients.

Opportunistic networks [2], also referred to as Delay-Tolerant Networks (DTNs), disruption-tolerant networks or intermittently connected networks, are a relatively recent family of ad hoc network that exploit such a paradigm. In such networks, no end-to-end connectivity between sender and receiver is guaranteed. Instead, data is delivered based on pairwise contact opportunities. The communication is multi-hop in the sense that each intermediate node is used as a router that stores the message until a contact opportunity for further forwarding arises. The intermediate nodes may take custody of data during the blackout and forward it when the connectivity resumes.

To date, the use of these networks has been specially successful in the aerospace domain to establish communications between information processing nodes, satellites or ground stations [3]. Moving from a macro-mobility to a micro-mobility scale, these networks could take benefit from the natural mobility of people to disseminate pieces of data

efficiently among users. Opportunistic networks can foster a variety of pervasive computing applications in mobile environments. There are different authors that have pointed out the potential of opportunistic networks to revolutionise the current market of mobile applications, also referred to as *apps* [4], [5]. Future apps could profit from the physical social interactions between near individuals to offer a new variety of added-value services exploiting human mobility. For example, iBeacon is a recent low-consumption Bluetooth-based technology launched by Apple Inc that starts to exploit the potential of opportunistic communications for indoor positioning [6]. By using indoor navigation assistance, retail stores can, for example, guide consumers to their desired products or track movement patterns of users to fine-tune product placement, thus improving the customer's shopping experience. This type of systems can be used in a lot of contexts (retails, enterprises, hospitals or schools) to improve daily services.

The purpose of future apps will be to allow users to move and interact with information, services, devices and other users in the Internet of Things (IoT) [7]. So, the design of these mobile application will require to study how the users, and their devices, move. Indeed, the success of future apps is increasingly becoming more dependent on their adaptiveness to mobile environments. Hence, most researchers use simulation to discover how their systems respond to the user activity evolution, including mobility. It is thus critical to support such simulations with adequate mobility models. Yet, a lot of the mobility models used in the research community are still ad hoc creations that have not been exhaustively validated. Furthermore, even when the mobility models are apparently proper, tuning their parameters to adapt them to different social contexts is a challenging task [8]. This is a limitation that hinders the recreation of human social scenarios. In spite of the possibility to find some social mobility models in the bibliography [9], they still present problems of representativeness given the complexity of recreating the complexity of human interactions.

The understanding of micro-mobility dynamics (especially in indoor contexts) is therefore essential to develop the next generation of apps. However, the widespread adoption (and acceptance) of such apps will depend on the ability of evaluators to test them under the proper mobility conditions. This means, among other things, developing techniques and methods to tune mobility models to improve their accuracy, thus easing the selection of suitable mobility traces for the system evaluation.

This paper presents an evaluation methodology for opportunistic apps to cope with the prior challenge. Its goal

is to evaluate the performance of such apps when they are subjected to specific mobility constraints. Our methodology evaluates targeting apps using both synthetic traces (obtained from mobility models) and real ones (obtained from real experiences) to validate the evaluation results. This validation technique is known as back-to-back testing in the dependability community [10]. It consists in diversifying the way to obtain the evaluation outputs to compare such results and provide a more robust analysis. Thus, the purpose of this paper is threefold: showing how the results obtained from back-back testing can be used to (i) select adequate mobility traces, (ii) enhance the fine tuning of mobility models and (iii) improve the design of opportunistic apps.

The rest of this paper is structured as follows. Section II introduces the related work. Section III presents our methodology. Section IV shows the case study where we apply our methodology. Finally Section V concludes the paper.

II. RELATED WORK

Mobility models are typically used to simulate and evaluate the performance of mobile systems. The definition of realistic mobility models is one of the most critical aspects of the simulation of applications in mobile environments [8]. In the last years, a variety of mobility models have been proposed and they can be found in the literature [11]. Some of these models are implemented in simulators to recreate network deployments in areas ranging from hundreds of square meters to square kilometres. Some of the classical examples are random-way-point (RWP) [12], Manhattan [11] or random walk [11]. These models have several advantages: they are simple, and it is easy to compare different protocols and systems using them. Although these models have shown their usefulness for the evaluation of network protocols, they typically represent nodes as independent entities with predefined trajectories that do not interact among one another. The work carried out by Cristaldi et al. [13] improves previous models by implementing a mobility model considering obstacles (e.g., walls) that mitigates the signal propagation. This work provides a better spatial model, but does not increase the realism of movements in the space. In the last years more human-like mobility models have appeared [9]. For example, the authors of [14] and [15] proposed social mobility models capturing the interactions between mobile users. Furthermore, Killijian et al. [16] collected the real trace data from a crowded environment that can be used to develop a heuristic to extract mobility characteristics from mobility tracks. Nevertheless, the more complex the model is, the more difficult the parameters setup. Indeed, surveying the recent bibliography [17], the choice of values for model parameters of simulations for mobile (in particular, ad hoc and opportunistic) networks research is relatively variable. In fact, the mobility research community lacks of standard scenarios to validate and to benchmark the different results obtained after simulation. This fact was already reported in [18] in 2005, but the trend has not changed significantly. Obviously, this is a critical point that consequently affects the evaluation of mobile applications.

Another capital challenge is the limited number of available evaluation frameworks beyond the use of simulators. In particular, there is a lack of open source and free tools to guide the analysis and interpretation of results from mobile applications. The proposal of evaluation methodologies is the

first step towards the provision of such a type of tools. Our contribution in this paper goes in this direction.

III. EVALUATION METHODOLOGY

The proposed methodology does not intend to automate the task of practitioners when evaluating a given network application; it rather tries to support and guide the fine tuning of concerned parameters to improve the expected behaviour of evaluated apps. The overall structure of our methodology is presented in Figure 1. Essentially, it is composed of three different stages: the scenario definition, the experiments execution and the data mining. The rest of this section is devoted to explain these phases.

A. Scenario definition

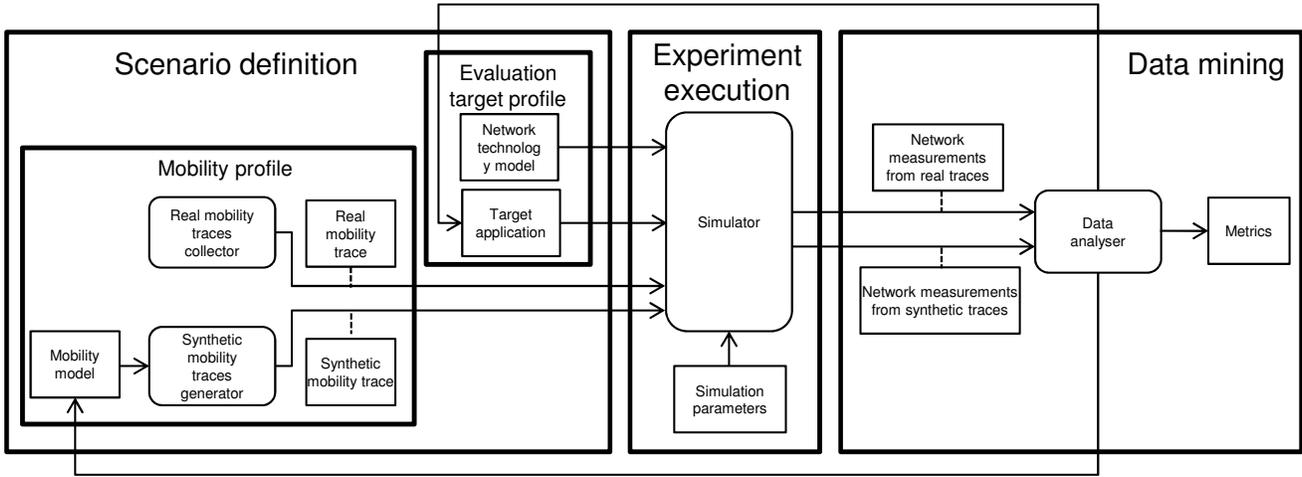
Recreating the environment dynamics and selecting the proper metrics to reflect such characteristics is essential for the quality of evaluation results. Consequently, all the parameters that characterise the experimentation must be precisely determined prior to the experiments execution. Likely, our methodology divides such sensitive parameters into two categories: the evaluation target profile and the mobility profile.

The evaluation target profile gathers the target application and the network technology model. The target application refers to the mobile application under evaluation, including its algorithmic specification and the expected context of use where the application will be deployed. The network technology model involves determining the communication paradigm used by the target application to interact with other mobile devices. Typically, it means selecting the network technology (e.g., Bluetooth, Zig-Bee, Wi-Fi, etc.) and the way it is used by the target application (i.e., the parameterisation applied to such a technology. For example, if the signal power is attenuated to reduce the energy consumption).

The mobility profile is in charge of accurately defining the topology dynamics, its characteristics and evolution. It is a fundamental factor that influences the application performance. To cope with this task we consider the use of mobility traces to animate the execution of the target application under evaluation. Mobility traces can be real or synthetic. Real-world traces are collected and recorded from real experimentation. Consequently, they provide a much more realistic representation of some real mobile scenarios. Unfortunately, the generation of representative real-world traces is not trivial. It requires the collaboration of tens or hundreds of volunteers and facilities to track their position with accuracy [16]. This fact makes that real-world traces are typically difficult to obtain. Some public initiatives like *Crawdad*¹ aim at creating free-access repositories where researchers can share their real-world traces. However, not always the available traces are the most adequate ones for the network deployment we need to recreate. Synthetic traces are generated by mobility model simulators. Realistic mobility models are typically more complex. Their accuracy depends on (i) the ability of modellers to identify the rules and parameters that define the realistic trajectories observed in the real world; and (ii) the capacity of evaluators to fine tuning such models. Synthetic traces are reproducible and they have the advantage of isolating specific behaviours not clearly

¹<http://crawdad.cs.dartmouth.edu/>

Fig. 1: Overview of our methodology



expressed in real-world traces. Furthermore, their results are much more scalable (up to thousands of nodes). Our mobility profile considers both types of traces to improve the analysis of results. Although both traces present a different nature, the idea is to choose mobility traces that can be potentially comparable (i.e., they present a similar number of users, context of use, node density, etc.) after an adequate tuning process. The more similar the trend between real-world and synthetic traces is, the easier to refine the parameterisation of the mobility model in charge of generating such synthetic traces.

B. Experiment execution

The experimental approach of our evaluation methodology is based on simulation. There are different network simulators in the bibliography that can be used to simulate opportunistic network deployments, such as NS2[19] or ONE [20]. Accordingly, the simulator receives the evaluation target profile and the mobility profile as inputs. Consequently, it may be necessary to adapt the mobility traces to the format used by the selected simulator and implement the target application as a new module of the simulator.

Concerning experiments can be grouped in experimental campaigns. Several parameters define the configuration of experimental campaigns: a warm-up time devoted to lead the network deployment to a stable state, the duration of the experiments, and the number of repetitions determining the proper amount of experiments that should be performed. Finally, the simulator must be instrumented to collect the adequate measurements that will be used in the data mining phase. As we want to evaluate the target application under the conditions defined by both synthetic and real-world traces, we propose the execution of different experimental campaigns (one per mobility trace).

C. Data mining

This stage addresses the analysis of resulting logs. The data stored in these log files needs to be off-line processed, filtered and correlated, to extract those measures that characterise the

best the application behaviour. So, it is on the evaluators to select proper measures. Throughput, delay or jitter are typical QoS measures used in generic evaluations. Nevertheless, the more specific the application is, the higher the need to define ad hoc measures. In any case, the main goal of this phase is to use such measures as feedback to refine the design the target application. Additionally, by comparing the result of synthetic and real-world traces, it may be also possible to use this feedback to enhance the accuracy of mobility models. The arrows drawn in Figure 1 illustrate this idea. This is an iterative process that can be useful in order to diversify the analysis process and gain confidence in conclusions.

IV. CASE STUDY

This section shows the feasibility of our methodology through a realistic microblogging case study.

A. A microblogging application

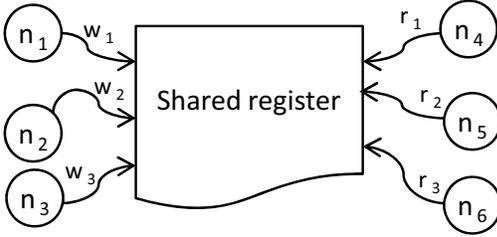
We assume a microblogging application for opportunistic networks that could be useful, for example, in conference environments to promote the communication between organisers and attendants. By using this application, session chairs may announce the beginning of the session after coffee breaks or programme chairs could advertise any modifications in the programme, thus endorsing the interaction with the attendants, and consequently improving their conference experience.

1) *Application target:* The algorithm proposed behind the microblogging application uses the notion of *shared register*. By shared register we understand a data structure consisting in a distributed asynchronous space of memory that is shared by a set of nodes (all the nodes in our case) in the opportunistic network [21].

The access to this shared register is permitted through two operations: read (R) and write (W). As different users may perform R and W operations, we can say that our shared register is characterised by multi-read (MR) and multi-write (MW) operations. Figure 2 illustrates this idea. Given the

lack of centralised entities, the nodes must interact among one another to determining the moment to perform R/W . Quorum systems are an adequate alternative to enforce consistent R/W in a distributed system [22]. Accordingly, before performing a R/W operation, a node needs to receive a minimum number of positive responses (or quorum) from the rest of nodes to proceed. In our case, we will use majority quorums. Our quorum-based algorithm assumes satisfying *atomicity* [21]. This property guarantees that R/W in our register appears to happen at a single point in time, thus avoiding the problem of overlapping R and W . In particular, if two reads r_i and r_j overlap a write w_i then the later read cannot return the old value if the earlier read returns the new one.

Fig. 2: Shared register scheme.



Both R and W are executed in two rounds. The first round, R1, consists in a timestamp synchronisation. Every time a node n_i wants to perform a R/W operation in the shared register, it broadcasts a R/W request $R_REQ_{n_i}$ or $W_REQ_{n_i}$. In order to achieve the maximum number of nodes in the network, the message dissemination follows an epidemic approach [23]. This means that those nodes receiving this request for the first time will respond with an acknowledgement ACK to n_i and additionally they will propagate the request for their neighbour nodes. In the meanwhile, n_i waits for the reception of an ACK from the quorum Q_a , where Q_a is the majority of nodes $\lceil \frac{N+1}{2} \rceil$, where N is the total amount of nodes. The last node in the quorum to notify its ACK will determine the minimum time to reach a consensus ($TTRC$). The $TTRC$ involves that the majority of nodes are aware of the R/W operation of n_i and wait for the node to launch it.

The second round, R2, executes the R/W operation. In the case of the microblogging application, this step involves reading the content of the messages published or writing a new entry in the blog. The procedure is similar to phase R1 in the sense that nodes wait for a read or write quorum Q_b to confirm the operation. Finally, when the R/W operation is completed, it returns the time to read (TTR) and the time to write (TTW). The algorithm pseudo-code is given in Table I.

The network technology we have considered to implement the messages exchange of the microblogging application is based on Ultra-Wide-Band. This choice is justified given its low consumption and its capacity to create ad hoc communications in Wireless Personal Area Networks (WPANs) [24].

2) *Mobility profile*: Our synthetic traces are obtained from animating two mobility models: Small World In Motion (SWIM) [14] and Random-Way-Point (RWP). SWIM is a relatively recent mobility model that recreates human mobility.

TABLE I: R/W operations.

01:	$write(v)$
02:	Round 1:
03:	$send(W_REQ)$
04:	do $receive()$ while $ W_ACK < Q_a $
05:	$TTRC_{Q_a} \leftarrow max_timestamp_{i=1}^{ Q_a }(W_ACK_{n_i})$
06:	Round 2:
07:	$send(write, v, timestamp)$
08:	do $receive()$ while $ Wv_ACK < Q_b $
09:	$TTRC_{Q_b} \leftarrow max_timestamp_{i=1}^{ Q_b }(Wv_ACK_{n_i})$
10:	$TTW = TTRC_{Q_a} + TTRC_{Q_b}$
11:	return ($write_{ack}, TTW$)
12:	$read()$
13:	Round 1:
14:	$send(R_REQ)$
15:	do $receive()$ while $ R_ACK < Q_a $
16:	$TTRC_{Q_a} \leftarrow max_timestamp_{i=1}^{ Q_a }(R_ACK_{n_i})$
17:	Round 2:
18:	$send(read, timestamp)$
19:	do $receive()$ while $ Rv_ACK < Q_b $
20:	$TTRC_{Q_b} \leftarrow max_timestamp_{i=1}^{ Q_b }(Rv_ACK_{n_i})$
21:	$TTR = TTRC_{Q_a} + TTRC_{Q_b}$
22:	return ($read_{ack}, TTR$)

Conversely, RWP is a classical mobility model very well-known in the mobility community. We have considered RWP in our evaluation to show to what extent the inadequacy of a mobility model can influence the evaluation results of target applications.

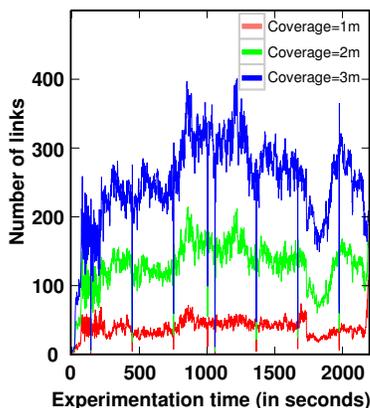
The real traces have been collected by a framework called Social Observation of hUman Kinetics (SOUK) that captures mobility data of crowds. The experience was performed in our research laboratory at LAAS-CNRS [16]. It was conducted during a reception where users were equipped with Ultra-Wide-Band tags to register their precise location in a $12m \times 12m$ room for 90 minutes.

B. Scenario simulation

The microblogging scenario has been implemented in the NS2 simulator given its capacity to implement opportunistic networks [19]. In particular, we have recreated a $12m \times 12m$ square room where 45 conference attendants and organisers may share a coffee break. We assume that all the participants installed the microblogging application in a mobile device equipped with a Ultra-Wide-Band network interface. Theoretically, the effective coverage range of this technology varies from $1m$ to $3m$. Preliminary experimentation (see Figure 3) shows the difference of peer-to-peer links created depending on the coverage range. Obviously, the wider the coverage range, the more links created. Unfortunately, the coverage range is rarely steady as it depends on the environment conditions (e.g., interferences, humidity, signal mitigation effects, etc.). Consequently, in our case study we limited the coverage range of user devices to $1m$ to represent always the worst case possible. This decision is based on the fact that almost all the studied proposed in the domain of network simulation are typically much more optimistic than reality [18], thus affecting the credibility of results.

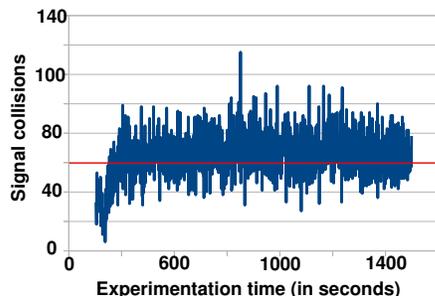
In order not to overload the communication channel, our application transmitted messages only every second. Preliminary experiments show a constant (not growing) trend in the

Fig. 3: Links evolution depending on the coverage range.



number of signal collisions despite considering an epidemic messages forwarding. This can be explained because the node message generation was optimised to avoid sending messages replicas. Figure 4 states this datum.

Fig. 4: Preliminary study of signal collisions in the network.



We considered a coffee break duration of 25 minutes (1500s), of which, the first 5 minutes belong to the warm-up period and were not considered in the final computation. To increase the representativeness of the experimentation, the role of the microblog publisher rotated from one node to another in 45 different experiment campaigns. In addition, for each microblog publisher the initial publication was executed 60 times, each one beginning in a different instant of time, resulting in 2700 experiments for each mobility profile.

The microblogging scenario was executed recreating three different mobility profiles: RWP, SWIM and SOUK. Being RWP and SWIM two mobility models, Table II lists the parameters we considered by default in our preliminary experience.

C. Analysis of results

This section presents three types of analysis coping with the main goals of our methodology: showing how the results obtained from back-back testing can be used to (i) select adequate mobility traces, (ii) enhance the fine tuning of mobility models and (iii) improve the design of opportunistic apps.

TABLE II: Mobility model default input parameters.

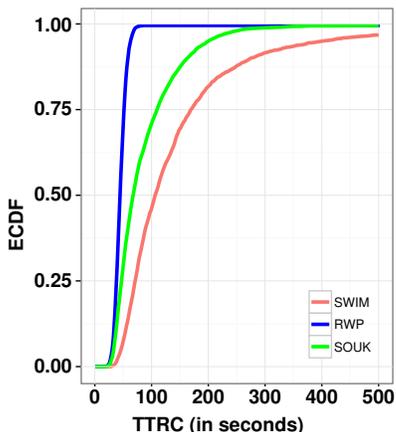
SWIM	N	Node number.	45
	d	Density ratio between the node communication range and the area.	0.007 (1/144)
	α	Proneness to meet other neighbours. It ranges from 0 to 1. The bigger α , the more frequently nodes meet their neighbours. Conversely, the smaller α , the more a node will tend to go to popular places.	0.5
	w	Distribution of the waiting time at a popular place.	1.45
RWP	N	Node number.	45
	s	Node minimum speed.	0 s
	S	Node maximum speed.	3 s
	p	Pause between movements.	10 s

1) *Assessing the impact of mobility traces:* Let us focus on $TTRC$ for this coarse-grained analysis. $TTRC$ is an essential measure to estimate the application performance. The shorter this time, the better for the microblogging application. If we analyse $TTRC$ without contextualising the mobility traces used in each case, the results are relatively surprising. Figure 5 shows the empirical cumulative distributed function (ECDF) resulting from measuring $TTRC$ in our experimental campaigns. The results of $ECDF(0.5)$, which are equivalent to the median, indicate a $TTRC$ of 45s when using RWP, 67s when applying SOUK (+49% with regard to RWP), and 112s when employing SWIM (+148% with regard to RWP). A different reading of the same results show that consensus is always achieved after 91s when using the RWP traces. However, the percentages slows down to 62.5% and 36% if the traces we use are the SOUK and SWIM ones respectively. Additionally, we can observe that 5% of cases the $TTRC$ requires more than 70s for RWP, 200s for SOUK and 400s for SWIM.

In summary, without contextualising the prior analysis, we find the best results for our application when applying the mobility traces generated via RWP. Since mobile nodes move all around the scenario without restrictions, the number of social contacts along the experimentation between nodes is much higher. This leads to a quicker exchange of messages, and consequently the $TTRC$ is obtained earlier. In other words, any message is propagated throughout the network with a higher speed. However, when contextualising these mobility traces within the scenario imagined in Section IV-B, or in other words, when mobile nodes become real users, results become unreal. Real users in a coffee break do not move randomly. Instead, they for example tend to find familiar faces to interact with, or they stay motionless. In this sense, the analysis around the RWP model results useless in our scenario. According to additional experience, not even a different tuning of the model parameters (s , S and p in Table II) could make adequate the mobility traces obtained with RWP to represent our case study.

In consequence, our methodology has shown its usefulness to discard non-adequate mobility traces. Conversely, the trend in the $TTRC$ obtained through the SOUK and SWIM traces is similar. Indeed, the human-interaction factor has been captured by SOUK (given the nature of the social experience) and SWIM (given availability of social-based parameters to create realistic mobility traces). This fact let us think that these two mobility traces are much more proper for the case study.

Fig. 5: Comparison of preliminary $TTRC$.



2) *Scaling the network size*: The goal we pose in this fine-grained analysis is the validation of our microblogging application. So, it is necessary to scale our approach, which means to expand the network size beyond 45 users. Unfortunately, although generating a new SWIM mobility trace is as simple as tuning the mobility model (i.e., parameter N in Table II), the SOUK real mobility trace cannot be scaled. Consequently, the results from applying the new SWIM mobility trace and SOUK would not be comparable. Yet, we can try to face this problem in two steps. First it requires fine tuning the models so that their mobility traces are comparable to real situations. Then, once the model behaviour is validated with regard to real traces, we can scale the mobility traces and infer that the behaviour of an equivalent real scenario would be similar.

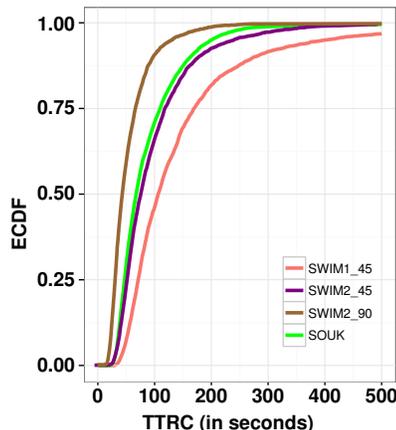
With regard to this first step, we have tested different SWIM configurations by tuning α and w (the value of d is fixed since we cannot change neither the coverage range nor the area). After applying a dichotomic-search-driven tuning [25] along multiple experimental campaigns, our experience indicates that α is the most sensitive parameter to model the behaviour of users in our case study. A configuration of $\alpha = 0.75$ and $w = 1.3$ seems to create mobility traces that are similar to those of SOUK in terms of $TTRC$. To strengthen this claim, we subjected our results to a 2-sample t-test, an statistical test to compare two population means from independent samples. The analysis revealed that at 5% level of significance (leading to a confidence interval=95%), the data does not provide sufficient evidence that the $TTRC$ mean for the new configuration of SWIM and SOUK are different. Figure 6 shows the two configurations of SWIM (the default one, named SWIM1_45 and the optimal one, called SWIM2_45) in comparison to SOUK. As can be seen at first sight, the SWIM2_45 and SOUK curves are very similar.

Once our synthetic traces have been validated with the real ones, we are able to generate easily a larger mobile scenario to assess the application performance in more crowded coffee breaks. Thus, we have considered a new scenario with 90 users animated by a new synthetic mobility trace (SWIM2_90). Figure 6 shows the results of applying this new configuration in comparison to prior ones. This result confirms our intuition: the higher the node density, the better for the $TTRC$. In particular,

the median $TTRC$ is 42s (-59% in comparison to SOUK).

In conclusion, we can state the usefulness of our methodology to fine tuning and validating mobility traces with regard to real ones.

Fig. 6: $TTRC$ after using different SWIM configurations.



3) *Addressing the fault-tolerance aspects*: As we have seen in Table I, $TTRC$ is the base to compute TTR and TTW . Although the preliminary version of our algorithm presents $TTRC_{Q_a}$ and $TTRC_{Q_b}$ as equivalent statistic times, we need to analyse the effects of such a fact. Essentially, this means that the nodes selection of Q_b does not depend on Q_a . In other words, without restrictions, for example, all the nodes of Q_a could join Q_b . Obviously, this is an advantage to optimise both TTR and TTW . However, this choice is not so good from a fault-tolerance viewpoint where nodes can fail. Indeed, if we assume the existence of faulty nodes that return permanent false values, it would be desirable that the intersection between Q_a and Q_b was minimum (i.e., $|Q_a \cap Q_b| = 1$) to prevent the same fault from being propagated to both sets.

Fig. 7: Fault-tolerance analysis using the SOUK mobility trace.

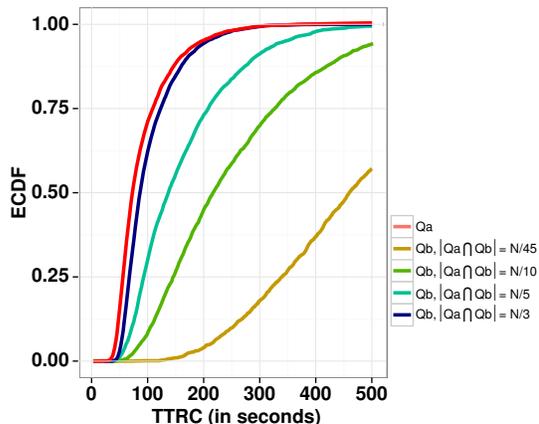


Figure 7 can help us to understand the cost of applying this decision in terms of $TTRC$. Such a figure represents the result of additional experiments using the SOUK mobility

trace to show the $TTRC$ consumed depending on the fault-tolerance strategy considered. For example, without restrictions, $TTRC_{Q_a}$ and $TTRC_{Q_b}$ would be statistically equal, which means that TTR or TTW would be $TTRC_{Q_a} + TTRC_{Q_b}$ in our figure. This would lead to a median TTR or TTW of $134s$ ($67s + 67s$). However, if we consider that just one node must intersect Q_a and Q_b (the minimum intersection to create a majority in both Q_a and Q_b), the median TTR or TTW would increase up to $527s$ ($67s + 460s$), an increment of almost 300%. Such an increment happens due to the fact that the Round 2 of our algorithm will need to be modified to discard the ACK of those nodes that cannot join Q_b because they already belong to Q_a . Thus, Q_b will be formed by $N - Q_a + 1$ but a huge penalty will be paid.

Although this last approach may be the best one to mitigate the faulty nodes problem, the TTR and TTW obtained are absolutely infeasible for the application acceptability. In addition, this strategy fails to address the problem of those nodes which fail silently (without announcing a failure), or that simply leave the room. This fact may lead Q_b to wait for the ACK of such nodes until a timeout expires, hence penalising the application behaviour. Consequently, enabling a moderate intersection between Q_a and Q_b could be an adequate strategy to mitigate this problem while providing a trade-off between performance and fault-tolerance. In particular, in Figure 7 we have analysed the result of enabling the intersection of 10%, 20% and 30% of nodes. As we increase the intersection percentage, we reduce the $TTRC$. The most interesting result is that an intersection of 30% of nodes in Q_b enables obtaining a $TTRC_{Q_b}$ similar to $TTRC_{Q_a}$. Indeed both curves are almost overlapped. As a result, we can improve the fault-tolerance of our application without abusive performance penalties. This fact shows how our methodology can be also used to improve the design of applications considering the expected mobile scenario where they will be finally deployed.

V. CONCLUSION

Opportunistic networks are one of the most interesting evolutions of MANETs. These networks are conceived to be deployed in specific operating contexts. So, the particular conditions of the network deployment must be taken into consideration in the analysis of results. We have designed a simple evaluation methodology to assess the performance of network deployments, meeting system needs, and speeding up the design and tuning among other advantages. Indeed, as the results of our case study confirm, our methodology can assist evaluators in different situations.

Mobility traces are a double-edged sword. If they are correctly selected, mobility has the potential to improve the application behaviour, as large amounts of data can be carried around the network using inter-device transmission. Else, mobility can become a problem for the communication, as device reachability may be highly unsteady. Consequently, interpreting the analysis of evaluation results depending on the context of use of the network deployment is essential to correctly select the mobility traces for our simulation. Accordingly, Section IV-C1 has shown the usefulness of our methodology to rapidly discard those mobility traces that do not fit the application deployment requirements.

Besides the decisions already taken by application designers, programmers and system integrators, there exist a number of parameterisation issues that must be carefully considered to test the application behaviour in realistic contexts of use. In this direction, the analysis done in Section IV-C2 has illustrated the usefulness of our methodology to check the consistency of different mobility model configurations before scaling network deployments.

Finally, the analysis of Section IV-C3 stated that our methodology can be used to improve the overall behaviour of distributed mobile applications. In particular, our methodology has been used to establish a trade-off between performance and fault-tolerance in the design of our application. This is a first step towards guiding (i) the efforts to design and develop mechanisms addressed to prevent, detect and tolerate network threats, and (ii) the experimental procedures to validate those mechanisms and assess the robustness of deployed networks.

With regard to our case study, in the future work we would like to study additional points. From a social viewpoint, we would like to characterise our microblogging application through more measures, such as the amount of connected components in the network or the inter-contact time. From a practical viewpoint, we aim at considering a variable amount of nodes N in the network, which is a more realistic condition, but very challenging in the domain of distributed systems.

We claim that this type of evaluation methodologies can be very useful to improve the knowledge of opportunistic networks. Thus, reducing the gap between the future apps that can be developed for opportunistic networks and the ability to evaluate their performance.

ACKNOWLEDGEMENT

This work is partially supported by the ANR French project AMORES (ANR-11-INSE-010) and the Intel Doctoral Student Honour Programme 2012.

REFERENCES

- [1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the Future*, 2012.
- [2] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, 2006.
- [3] "Disruption Tolerant Networking," [Online]. Available: <http://www.nasa.gov/content/disruption-tolerant-networking/#.U39dwyjwApo>, 2014.
- [4] A. Lindgren and P. Hui, "The quest for a killer app for opportunistic and delay tolerant networks," in *Proceedings of the 4th ACM workshop on Challenged networks*. ACM, 2009, pp. 59–66.
- [5] P. Meroni, E. Pagani, G. P. Rossi, and L. Valerio, "An opportunistic platform for android-based mobile devices," in *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*. ACM, 2010, pp. 191–193.
- [6] "What is iBeacon? A Guide to Beacons," [Online]. Available: <http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/>, 2014.
- [7] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [8] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.
- [9] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella, "Human mobility models for opportunistic networks," *Communications Magazine, IEEE*, vol. 49, no. 12, pp. 157–165, 2011.

- [10] M. Vouk, "Back-to-back testing," *Information and software technology*, vol. 32, no. 1, pp. 34–45, 1990.
- [11] M. Musolesi and C. Mascolo, "Mobility models for systems evaluation," in *Middleware for Network Eccentric and Mobile Applications*. Springer, 2009, pp. 43–62.
- [12] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic properties of the random waypoint mobility model," *Wirel. Netw.*, vol. 10, no. 5, pp. 555–567, Sep. 2004.
- [13] S. Cristaldi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, "Obstacles constrained group mobility models in event-driven wireless networks with movable base stations," *Ad Hoc Networks*, vol. 9, no. 3, pp. 400–417, 2011.
- [14] A. Mei and J. Stefa, "Swim: A simple model to generate small mobile worlds," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2106–2113.
- [15] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 3, pp. 630–643, 2011.
- [16] M.-O. Killijian, M. Roy, G. Trédan, and C. Zanon, "Souk: social observation of human kinetics," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 2013, pp. 193–196.
- [17] C. Song, T. Koren, P. Wang, and A.-L. Barabási, "Modelling the scaling properties of human mobility," *Nature Physics*, vol. 6, no. 10, pp. 818–823, 2010.
- [18] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet simulation studies: the incredibles," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 50–61, 2005.
- [19] T. Issariyakul and E. Hossain, *Introduction to network simulator NS2*. Springer, 2011.
- [20] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 55.
- [21] M. Roy, F. Bonnet, L. Querzoni, S. Bonomi, M.-O. Killijian, and D. Powell, "Geo-registers: An abstraction for spatial-based distributed computing," in *Principles of Distributed Systems*. Springer, 2008, pp. 534–537.
- [22] S. Nagaraj, N. Sirisilla, and H. S. Madhusudhana, "Method and system for managing network resources based on a dynamic quorum," Oct. 9 2012, uS Patent 8,285,825.
- [23] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," in *NETWORKING 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*. Springer, 2006, pp. 827–839.
- [24] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*. IEEE, 2007, pp. 46–51.
- [25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein *et al.*, *Introduction to algorithms*. MIT press Cambridge, 2001, vol. 2.