# Characterization of the Impact of Resource Availability on Opportunistic Computing

Alan Ferrari
Networking Laboratory
University of Applied Sciences
of Southern Switzerland
Manno, Switzerland
alan.ferrari@supsi.ch

Daniele Puccinelli
Networking Laboratory
University of Applied Sciences
of Southern Switzerland
Manno, Switzerland
daniele.puccinelli@supsi.ch

Silvia Giordano
Networking Laboratory
University of Applied Sciences
of Southern Switzerland
Manno, Switzerland
silvia.giordano@supsi.ch

## ABSTRACT

With opportunistic computing, devices are no longer restricted to using their own services and resources, but can access services and resources made available by other devices. The performance of opportunistic computing is greatly affected by the resource topology in the network: what resources/services are available, as well as when and where they can be tapped. This paper presents a preliminary investigation of the impact of the resource availability on the performance of opportunistic computing. Specifically, we propose a metric called Expected Resource Availability, ERA, that attempts to capture the impact of the topology of services and resources. The ERA offers a proxy for the applicability of opportunistic computing schemes to a given network: if the ERA is low, any opportunistic scheme can be expected to fail due to a sheer lack of resources and/or connectivity among them. On the other hand, if the ERA is high, success can be expected. To gain perspective on the properties of the ERA, we tackle the problem of service allocation in opportunistic computing, which suffers to combinatorial explosion when looking for the optimal solution. We also present some preliminary simulation results that confirm the validity of the ERA as a metric to gauge whether opportunistic computing can be achieved in a given network.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design, Wireless Communication

## General Terms

Algorithms, Experimentation, Measurement, Performance

## Keywords

Opportunistic Computing, Resource Availability

*MCC'12*, August 17, 2012, Helsinki, Finland.
Copyright 2012 ACM 978-1-4503-1519-7/12/08... $15.00.

## 1. INTRODUCTION

The proliferation of mobile devices that we have witnessed in recent years has changed the face of wireless and mobile computing. Users no longer wish to be constrained by the limitations of their own devices owing to their induced need for additional resources and services that cannot be efficiently provided in a fully localized fashion. A natural direction to fill the gap between user demand and resource-constrained supply is cloud computing. The emerging trend of offloading expensive tasks to the mobile computing cloud [1, 2] goes in this direction, but has a significant energy footprint [3] and suffers from the drawbacks of extreme centralization [4], although cloudlet-type solutions [5] seek a compromise between distributed and centralized operation. Opportunistic computing (OC) [6, 7] provides an appealing alternative to the mobile computing cloud by allowing devices to join forces and leverage heterogeneous resources from other devices. Because this is only possible by leveraging human mobility, opportunistic computing adds even more prominence to the role of the human element, which is already central to mobile phone sensing and now becomes the key enabler of pervasiveness [8].

OC represents a natural evolution of the Opportunistic Networking paradigm. In opportunistic networks, nodes can communicate with each other even in the absence of an end-to-end path at any given time, as long as there are no tight timing constraints. Rather than just leveraging other devices to tap on connectivity, with OC devices can tap on each other's resources and services. OC has recently gained a lot of attention from the scientific community. Currently, the OC paradigm is being investigated in the FP7 Project SCAMPI [9].

A key problem in OC is finding efficient policies to retrieve services or compositions thereof to minimize the delay between the time a request is sent out and the time the desired service output is received. Multiple nodes may request services at a time and the whole cycle from service request to output reception may require multiple inter-device contacts.

In this paper, we illustrate a preliminary study of the impact of the resource availability on the performance of OC. In order to be successful, OC needs enough resources in the network, and enough connectivity to get to them. As a baseline, resources need to be available; to ensure success, they must also have adequate levels of stability. Shared services must in fact be supported in a network environment that is made challenging by topological instability and constrained resources. In our previous work, we have investigated topological instability [10] and limited resources [11] using low-power wireless as a case study. Low-power wireless represents an exceptional test bench because it contains an exacerbated version of the most adversarial features of an OC setting (network instability and constrained resources) while offering easily accessible large-scale experimental resources [12, 13, 14] whose challenges [15] more than make up for the lack of mobility. In this paper, we shift our focus to a basic OC setting to introduce a novel metric, the Expected Resource Availability (ERA), whose goal is to gauge the impact of the topology of services and resources in an OC network. Given the available resources and the ease of connecting to their providers, how successful can we expect *any* OC scheme to be? In other words, the ERA attempts to capture the OC-friendliness of a given networked setting. If the ERA is low, OC is guaranteed to fail no matter how cleverly it is applied owing to a sheer lack of available resources (there are not enough actual resources/services, or there is not enough connectivity to get to them). If the ERA is high, however, the network is literally waiting for OC to take over.

## 2. RELATED WORK

Early analytical work in OC has focused on finding the best service request policy [16] as well as modeling or simulating service composition [17]. A distributed scheme for calculating the level of resources available in the vicinity of a node in a delay-tolerant network is presented in [18], with a specific focus on buffer space, energy, and bandwidth. Our work in this paper leverages our previous work on the decoupling of the impact of the network topology from the one of protocol performance [10], where the impact of network topology is distilled down to one single metric, the Expected Network Delivery (END). The END metric is able to determine when the topological conditions of the network represent a bottleneck that throttles the achievable performance of a data collection protocol. The END was developed for low-power wireless, but its applicability goes way beyond the world of low-end radios, and indeed it forms the basis for the work in this paper, where connectivity is generalized to resource availability: in SCAMPI's OC networked settings, it is no longer enough to be connected, but services and resources are to be shared across nodes. This paper attempts to adapt the logic reasoning that lead to the END in [10] to SCAMPI's heterogeneous networks to capture the impact of a generalized topology of services and resources on the performance of OC schemes.

The next sections of the paper are organized as follows. In Section 3 we introduce the system model, mostly based on [16] and [10], and we derive the ERA metric. To enable the trace-driven evaluation presented in Section 5, in Section 4 we propose a greedy algorithm to tackle the optimal service execution problem.

## 3. SYSTEM MODEL

### 3.1 Basic OC Setup

The main goal of the following section is to formalize the parameters/variables that play a key role in OC, partly following the formalisms employed by Passarella et. al. [16]. Let us define:

- $N$: set of nodes (users, either service seekers or service providers or both).
- $S$: set of services.
- $\mathbf{P}$: a $|N|\text{x}|S|$ boolean matrix containing the relationship between nodes and services.
- $Q_n$ is the set of all queries performed by the service seeker $n$.

Each service request from service seeker $n$ is formalized as a query to the network $q_{i,n}$, where $i$ is the query index. We define $q_{i,n} \triangleq \{s_0 \rightarrow ... \rightarrow s_{|q_{i,n}|-1}\}$ as a set of services where the output of service $s_i$ is the input of service $s_{i+1}$. In general, there will be a chain of service providers. When a user submits a service request (query), the input will have to be uploaded to the first service provider in the chain and travel through the chain; the output will be generated by the last service provider in the chain and will have to be downloaded to the service seeker. In the remainder of the paper, we will use the simplified notation $q$ to refer to a generic service request that will represent the focus of our study.

In order to characterize resource availability, we need to measure the time needed to obtain the object of service request $q$, $T_e(q)$, from query submission to output download. To this end, we employ the following notation:

- $t_s(q)$: time when the user submits the query.
- $T_\Delta(q)$: delay until the first service provider is reached.
- $T_u(q)$: time required to upload the input data to the first service provider.
- $T_c(q)$: computing time at the service providers.
- $T_\Omega(q)$ delay until the last service provider can reach the service seeker.
- $T_d(q)$: time required to download the output data to the service seeker.

The time needed to obtain the output of service request $q$ is therefore given by

$$T_e(q) = T_\Delta(q) + T_u(q) + T_c(q) + T_\Omega(q) + T_d(q) \quad (1)$$

In the next section we characterize services from a user's standpoint.

### 3.2 Service Execution Timing

A user may have a wide range of service execution timing requirements. In order to account for such needs, we refer to the following taxonomy. A service $s \in S$ belongs to a priority class $c$ ($c \in \mathbb{Z}$ and $c \in [0, |S| - 1]$); the higher the value of $c$, the more delay tolerant the service. We assume that services in class 0 can only tolerate delays in the order of seconds. We also define the maximum time a user is willing to wait for a service as $M_c$ for Class $c$ services. Classes are assumed to be disjoint and their union matches $S$.

We define the function $C(q)$, which returns the service class that a given query belongs to. In the case of service composition, class membership is determined by the class of the services with the tightest delay constraints, *i.e.*, the class with the smallest $q$.

We are now ready to define the Expected Resource Availability.

### 3.3 ERA - Expected Resource Availability

Our goal is to characterize the impact of the network topology on service execution. To this end, we define a novel metric called Expected Resource Availability (ERA).

The first stage needed to compute the ERA consists in evaluating the resource usage at each single node in the network. To this end, we define the Expected User Resource Availability (EURA), which captures the resource availability from the point of view of a given user. In this framework, the EURA serves the same role as the Expected Pathwise Delivery (EPD) in [10]; the EPD, in fact, quantifies the network topology as viewed from a particular node. Let us define service query set $Q_{n,c}$ for user $n$ and class $c$ services as

$$Q_{n,c} \triangleq \{\forall q \in Q_n | C(q) = c\}, \tag{2}$$

where $Q_n \triangleq \cup_{c \in S} Q_{n,c}$. For each subset pertaining to service class $c$, we define a probability density function $f_c$ to characterize the execution time $T_e(q)$ of the requests that belong to class $c$.

For each class $c$ and user $n$, we define $\mathrm{EURA}_{n,c}$ as follows:

$$\mathrm{EURA}_{n,c} = \int_0^{M_{n,c}} f_c(x)dx, \tag{3}$$

where $M_{n,c}$ is the maximum delay acceptable to users of services in class $c$.

Now we can compute the global EURA for a user $n \in N$ as follows:

$$\mathrm{EURA}_n = \frac{1}{|Q_n|} \sum_{c \in S} |Q_{n,c}| \mathrm{EURA}_{n,c}, \tag{4}$$

and the ERA is given by

$$\mathrm{ERA} = \frac{1}{|N|} \sum_{n \in N} \mathrm{EURA}_n \tag{5}$$

## 4. OPTIMAL SERVICE EXECUTION IN OC

Before we proceed to a trace-based investigation of the ERA, we set out to further understand service execution in OC. Specifically, we intend to investigate the Optimal Service Execution (OSE) problem: given a set of contact times, a set of services offered by the users, and a set of service queries with their own timing constraints and requirements (*e.g.,* belonging to different priority classes), what is the optimal assignment of the service queries over the available contact times that minimizes the time from query submission to output retrieval? Solving the OSE will complement our ERA framework by providing a benchmark to compare practical OC solutions against.

The OSE may be viewed as a combinatorial optimization problem. Because problems of this nature are notoriously complex, in this paper we focus on finding a viable solution (though not necessarily optimal) in an acceptable amount of time.

### 4.1 Problem statement

The goal of the OSE Problem is to minimize the time from query submission to output retrieval for all the queries performed by all the users. In principle, finding the best query assignment requires exploring all the combinations (without repetition) of the service queries in the available inter-device contact times. Rather than engaging in the untenable complexity of an exhaustive search, we relax the key constraints and propose a greedy algorithm to handle a simplified version of the OSE and obtain a viable solution.

### 4.2 Greedy Algorithm for OSE

For the sake of tractability, we assume that each query can only call a single service at a time. We also make a set of assumptions to superimpose an execution hierarchy over the service queries to drastically reduce the possible combinations that form the solution space.

We propose a greedy solution based on the assumption that queries are executed in the following order:

1. Query timing constraints (in ascending order of class ID $c$; refer to the classes in Section 3.2);
2. Time when the request was issued;
3. Node id (in ascending order).

The execution hierarchy is not only instrumental to a significant reduction of the size of the solution space, but it also eliminates a large number of non-optimal solutions. For instance, there may be situations where a high-priority service is requested right after a low-priority one, in which case servicing the latter first would be inefficient.

The assumption that each query can only call one service at a time is used to prevent the size of the solution space from growing too large; it is, however, a limiting assumption that we plan to relax in our future work. Removing this assumption will be the first step to the extension of the ERA framework to service composition.

To solve the relaxed OSE problem, we propose a greedy algorithm whose pseudo-code is reported as Algorithm 1. We assume that the access time to $R$ is constant. In the implementation this is handled by means of a relational database. The next section will show our preliminary evaluation of the ERA framework by using Algorithm 1.

**Algorithm 1** Greedy OSE

```
input:
  • $N$, set of nodes in the network;
  • $S$, set of services offered by $N$;
  • $\mathbf{P}$, boolean matrix containing the relationships between nodes and services;
  • $R$, contacts between nodes. A contact $r \in R$ that begins at time $s$ and ends at time $f$ between $n_1$ and $n_2$, both in $N$, is represented as a 4-tuple $r = \{n_1, n_2, s, f\}$;
  • $Q$, set of all queries performed by all the users.

definitions:
  • USER($q$), where $q \in Q$, returns the user ($n \in N$) who submitted the query
  • REM($n, t_1, t_2$) removes from R all the entries containing the node $n \in N$ from the time $t_1$ to the time $t_2$

procedure:
for each priority class $c$ ordered by $M_c$ do
  $Q_c = \forall q \in Q | C(q) = c$
  Sort $Q_c$ in increasing order of $t_s(q)$
  for each $q \in Q_c$ do
    $N^{'} = \{n \in N | \mathbf{P}(n, q) == true\}$
    $s \triangleq 0, t \triangleq \infty$, k$\triangleq \infty$
    for each $n \in N^{'}$ do
      Find $s'$ and $t'$ for $\{\text{USER}(q), n, s', t'\}$, i.e., the earliest contact between USER($q$) and $n$, where $s' \geq t_s(q)$ and $(t' - s') \geq T_u(q)$
      Find $s''$ and $t''$ for $\{\text{USER}(q), n, s'', t''\}$, i.e., the earliest contact between USER($q$) and $n$, where $s'' \geq s' + T_u(q)$ and $(t'' - s'') \geq T_d(q)$
      if $(t - s) > (t'' - s')$ then
        $T_e(q) = s'' + T_d(q) - t_s(q)$
        $T_\triangle(q) = s' - t_s(q)$
        $T_\Omega(q) = s'' - t'$
        $s = s'$, $t = t''$, $k = s''$
      end if
    end for
    REM(USER($q$), $s, s + T_u(q)$)
    REM(USER($q$), $k, k + T_d(q)$)
    REM($n, s, s + T_u(q)$)
    REM($n, k, k + T_d(q)$)
  end for
end for

return: $Q$
```

# 5. EVALUATION

This section presents an investigation of the ERA based on the Cambridge iMote traces collected within the context of the Haggle Project [19] as well as our own synthetic traces generated by tweaking the probability of inter-node contact measured in the Cambridge traces. The former are used to show the effectiveness of the ERA when the service distribution of the nodes varies; the latter serve to show the effectiveness of ERA when the distribution of the inter-node contacts varies.

## 5.1 Cambridge iMote Simulation

We begin by applying Algorithm 1 to the Cambridge iMote traces, where contacts among 12 devices were recorded over a 6 day timeframe [19]. Figure 1 shows the Contact Time distribution and the Inter Contact Time distribution.

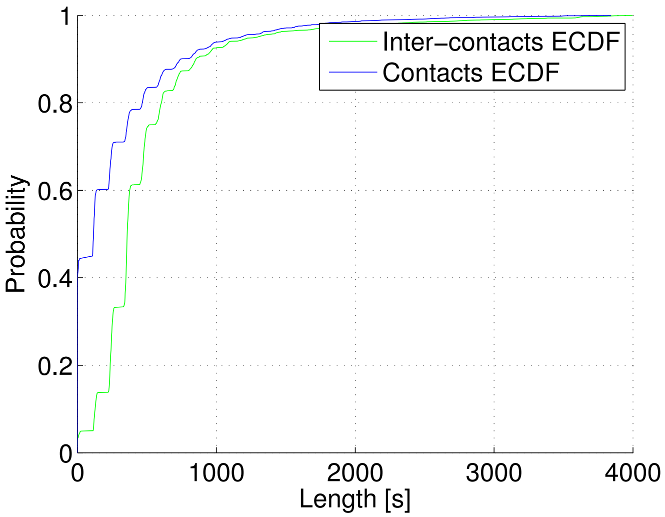

**Figure 1: Empirical Cumulative Distribution (ECDF) of the Contact and Inter contact Times in the iMote Cambridge Dataset**

Services are assigned to nodes uniformly at random. Reasonable values for the timing parameters in Section 3 have been obtained based on the contact and inter-contact time statistics. Service queries are reassigned to nodes, also uniformly at random, at fixed intervals. Specifically, we assume that a new service query is issued on a half hourly basis (more precisely, we employ a fixed interval of 2000 sec, which makes sense given the contact and inter-contact time statistics).

For the probability density function $f_c$, we have chosen a Rayleigh because it provides the best qualitative fit to our empirical data. For the purposes of this preliminary study, we assume that each service has a given probability of being offered. Figure 2 shows the ERA as a function of this probability averaged over all nodes and services. It is intuitively pleasing that the ERA grows as services are more likely to be offered.

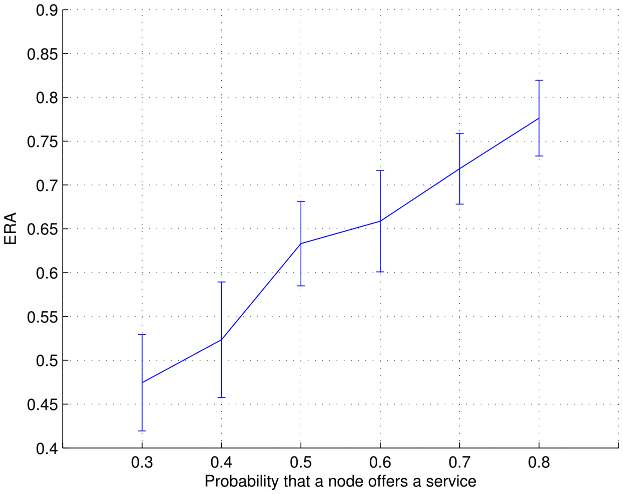

**Figure 2: The value of the ERA metric grows with service availability, as shown by these results based on the iMote Cambridge Dataset.**

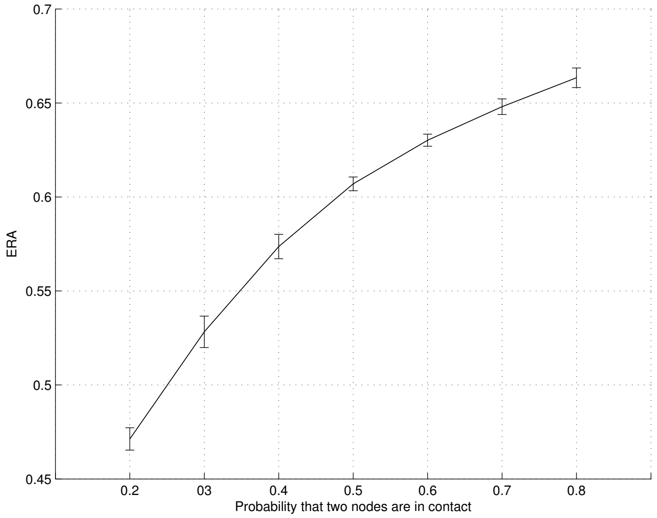

**Figure 3: The value of the ERA metric grows as inter-node contacts become more likely, in which case there are more opportunities for connectivity in the network.**

## 5.2 Impact of contact times

We obtain additional synthetic traces by tweaking the inter-node contact rate in the Cambridge traces. Figure 3 shows that the ERA correlates well with the inter-node contact probability, confirming that the ERA can capture the effort that a user must put into service retrieval (the more nodes meet, the lower the effort).

# 6. CONCLUSION

In this paper we proposed a novel metric called Expected Resource Availability (ERA), which helps characterize the impact of the service topology on an OC scenario. The development of the ERA is based on the END framework in [10]. The END captures the network-wide impact of topological instability due to low-power wireless link dynamics; the ERA metric can be viewed as the generalization of the END to service and resource availability as well as stability. The goal of the ERA metric is to determine the impact of the topology of services and resources in a heterogeneous OC scenario, providing a proxy for the applicability of opportunistic computing schemes to a given network.

We presented a preliminary evaluation of the potential of the ERA based on real-world traces, where we show that the ERA metric correctly captures some of the basic interplays between resource availability and performance. Specifically, the ERA responds well to increasing service or resource availability as well as to the increasing frequency of inter-node contacts. The preliminary evaluation presented in this paper shows that the ERA is a promising tool to understand the impact of the service topology on the performance of OC.

In our future work, we will further investigate the properties of the ERA. We will take into consideration not only the network topology but also the resource usage, as we argue that OC needs not only a reasonably expeditious service execution, but also mechanisms to prevent the saturation of the available resources. We also plan to study the interplay between the ERA and energy availability, and we plan to account for the social behavior of users [20], which may result in the presence of identifiable service query patterns.

## Acknowledgements

This work was partially supported by the European Commission under the SCAMPI Project (ICT grant agreement 258414).

# 7. REFERENCES

[1] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: Making Smartphones Last Longer with Code Offload. In *MobiSys'10*, San Francisco, CA, USA, June 2010.

[2] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting Mobile 3G Using WiFi. In *5th Annual International Conference on Mobile Systems, Applications and Services (MobiSys'10)*, San Francisco, CA, USA, June 2010.

[3] A. Miettinen and J. Nurminen. Energy efficiency of mobile clients in cloud computing. In *2nd USENIX Workshop on Topics in Cloud Computing (HotCloud'10)*, Boston, MA, USA, June 2010.

[4] T. Benson, S. Sahu, A. Akella, and A. Shaikh. A First Look at Problems in the Cloud. In *2nd USENIX Workshop on Topics in Cloud Computing (HotCloud'10)*, Boston, MA, USA, June 2010.

[5] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4):14–23, 2004.

[6] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134 –141, november 2006.

[7] M. Conti and M. Kumar. Opportunities in Opportunistic Computing. *IEEE Computer*, pages 42–50, January 2010.

[8] S. Giordano and D. Puccinelli. The Human Element as the Key Enabler of Pervasiveness. In *10th IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-NetÕ11)*, Favignana Island, Italy, June 2011.

[9] Scampi Consortium. Scampi, service platform for social aware mobile and pervasive computing, March 2012.

[10] D. Puccinelli and O. Gnawali and S. Yoon and S. Santini and U. Colesanti and S. Giordano and L. Guibas. The impact of network topology on collection performance. In *EWSN*, pages 17–32, 2011.

[11] D. Puccinelli and S. Giordano. On the Interplay of Low Power Wireless Connectivity and Energy Consumption. In *2nd International Workshop on Networks of Cooperating Objects (CONET 2011)*, Chicago, IL, USA, April 2011.

[12] G. Werner-Allen, P. Swieskowski, and M. Welsh. MoteLab: a Wireless Sensor Network Testbed. In *4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, CA, April 2005.

[13] V. Handziski, A. Koepke, A. Willig, and A. Wolisz. TWIST: A Scalable and Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Networks. In *2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN'06)*, Florence, Italy, 2006.

[14] M. Doddavenkatappa, M.C. Chan, and A.L. Ananda. Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed. In *7th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom 2011)*, Shanghai, China, April 2011.

[15] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An Empirical Study of Low-Power Wireless. *ACM Transactions on Sensor Networks (To appear)*, 2010.

[16] A. Passarella, M. Kumar, M. Conti, and E. Borgia. Minimum-delay service provisioning in opportunistic networks. *IEEE Trans. Parallel Distrib. Syst.*, 22(8):1267–1275, 2011.

[17] U. Sadiq, M. Kumar, A. Passarella, and M. Conti. Modeling and simulation of service composition in opportunistic networks. In *MSWiM*, pages 159–168, 2011.

[18] G. Sandulescu, P. Schaffer, and S. Nadjm-Tehrani. Exploiting resource heterogeneity in delay-tolerant networks. *Wireless Communications and Mobile Computing*, January 2012.

[19] J. Scott and R. Gass and J. Crowcroft and P. Hui and C. Diot and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/infocom2006 (v. 2006-01-31), January 2006.

[20] A. Foerster, K. Garg, H. A. Nguyen, and S. Giordano. On Context Awareness and Social Distance in Human Mobility Traces. In *3rd International Workshop on Mobile Opportunistic Networks (MobiOpp 2012)*, Zurich, Switzerland, February 2012.