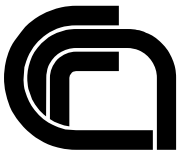

Consiglio Nazionale delle Ricerche

Exploiting opportunistic contacts for service provisioning in bandwidth limited opportunistic networks

A. Passarella, M. Conti, E. Borgia, M. Kumar

IIT TR-18/2010

Technical report

Maggio 2010

Istituto di Informatica e Telematica

Exploiting opportunistic contacts for service provisioning in bandwidth limited opportunistic networks

Andrea Passarella, Marco Conti, Elonora Borgia
IIT-CNR
Via G. Moruzzi, 1
56124 Pisa, Italy
{a.passarella,m.conti,e.borgia}@iit.cnr.it

Mohan Kumar
University of Texas at Arlington
Arlington, TX 76019, USA
mkumar@uta.edu

ABSTRACT

Opportunistic computing has emerged as a new paradigm in computing, leveraging the advances in pervasive computing and opportunistic networking. Nodes in an opportunistic network avail of each others' connectivity and mobility to overcome network partitions. In opportunistic computing, this concept is generalised, as nodes avail of *any* resource available in the environment. Here we focus on computational resources, assuming mobile nodes opportunistically invoke services on each other. Specifically, resources are abstracted as services contributed by providers and invoked by seekers. In this paper, we present an analytical model that depicts the service invocation process between seekers and providers. Specifically, we derive the optimal number of replicas to be spawned on encountered nodes, in order to minimise the execution time and optimise the computational and bandwidth resources used. Performance results show that a policy operating in the optimal configuration largely outperforms policies that do not consider resource constraints.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Performance, Design, Algorithms

Keywords

Opportunistic networks, service provisioning, performance evaluation, analytical modelling

1. INTRODUCTION

Opportunistic computing [10] is a recently proposed mobile computing paradigm, blending the areas of pervasive computing and opportunistic networking. In opportunistic networks, no continuous end-to-end path between nodes can be granted. To cope with intermittent connectivity, nodes opportunistically exploit encountered peers to forward messages and disseminate content to the intended recipients. Note that the conventional networking paradigm to support multi-hop pervasive networks, i.e. MANET, has been shown to be ineffective in general [8] as it does not cope with challenged network conditions that opportunistic networks natively address. In opportunistic networks, nodes essentially contribute and opportunistically avail of each other resources in terms of connectivity and temporary buffer storage. Opportunistic computing generalises this concept, considering the possibility for nodes to opportunistically avail of each others' *general resources*, including, but not limiting to, connectivity and storage. This concept is particularly suitable for pervasive computing environments, which are seen as featuring a huge and extremely variable set of heterogeneous resources available on fixed and mobile devices with wireless networking capabilities. In this context, resources may include heterogeneous hardware components, software processes, multimedia content, sensors and sensory data. While not all resources can be available on any single device, they can be collectively available to anyone through the deployment of effective middleware schemes for opportunistic computing. The opportunistic computing paradigm enables several interesting applications that are already being investigated in the areas of participatory sensing, pervasive healthcare, intelligent transportation systems, and crisis management, as discussed in [10].

In this view, it is natural to abstract resources as *services* that are contributed by *providers* and invoked by *seekers*. While service-oriented computing has long been investigated in the conventional Internet, in single hop wireless networks (e.g., WLANs or cellular networks) and in conventional MANET, it is definitely a novel research area in opportunistic networks. Intermittent connectivity and severely unstable topologies typical of opportunistic networks make conventional service-oriented approaches too cumbersome to be used.

This paper is one of the first attempts, to the best of our knowledge, to tackle key research challenges implied by the opportunistic computing concept. Specifically, we develop a detailed analytical model of the service invocation process between seekers and providers. As better described in Sec-

tion 3, we assume the service invocation and provisioning is carried out as follows: Whenever a seeker has a request for a service execution, it waits to encounter possible providers. When a provider is met, a service execution may be spawned, according to the *replication policy* used by the seeker. Service results are gathered by the seeker from the first provider encountered which has completed the service execution. The goal of the model is to identify the optimal operating point of the replication policy, in which the expected service time (i.e. the time interval between when a request is generated at the seeker and when the seeker receives the output results) is minimised, subject to the available providers' resources in terms of computation and bandwidth. The model is general enough to consider arbitrary bandwidth and computational constraints, heterogeneous mobility of different sets of users, varying density of seekers and providers. Note that modelling resource limitations (particularly in terms of bandwidth) is very challenging in opportunistic networks, and indeed it is common practice to assume unlimited bandwidth scenarios, to simplify the analysis (see, e.g., [22, 14, 6]). While this assumption is acceptable when congestion is not an issue, neglecting bandwidth limitations may lead to very inaccurate conclusions when the effect of congestion plays a role [15].

The model presented in this paper significantly extends an initial model we have described in [20]. Unlike the model in [20] here we consider i) limited bandwidth, ii) heterogeneous mobility processes of different nodes, iii) heterogeneous providers computation capabilities, and iv) varying seekers generation rates.

The model, described in Section 4, is validated against simulation results to assess its accuracy, and then used to characterise the performance of the optimal, resource aware, replication policy (Section 5). Specifically, comparison of the optimal policy with resource-blind policies clearly shows that the latter leads to saturation or under utilisation of available resources. The performance analysis also characterises the behaviour of the optimal policy as a function of the overall load in terms of service requests. The load is varied through several parameters: the percentage of seekers and providers in the network, the rate of requests issued by seekers, and the average computation time required by service executions at providers. With respect to all these parameters, we find that the optimal policy is able to tolerate increasing loads by limiting the corresponding increase of service time, up to the point where the resources become inevitably saturated. Specifically, as the load increases, the optimal policy reduces the replication level so as to avoid resource saturation, thus constantly minimising the expected service time experienced by seekers. Under the optimal policy, saturation occurs only for loads such that the overall system capacity cannot support even a single execution for each generated request. No alternative policy can perform better, as in these cases the only solution would be to drop some service requests without even attempting to serve them. Investigating such "back-pressure" policies is out of the scope of this paper, and subject of future work.

2. RELATED WORK

Service provisioning in opportunistic networks is a challenging problem that, to the best of our knowledge, has just recently started to be addressed. Research on opportunistic networks has mainly focused on routing issues (e.g. [3, 22, 14, 24, 2]), mobility analysis (e.g. [6, 5, 26]) and, more recently, on data-centric architectures for content delivery (e.g., [11, 4]).

The work presented in this paper can be seen as one of the building blocks of the recently proposed concepts of opportunistic computing [10] and people-centric sensing [17]. The latter area already produced significant results, which however are mainly focused on inferring people activities through sensors available on single pervasive user devices (such as last-generation smartphones), and how to exploit such information for augmented mobile social networking services. The process necessary to perform such inferences is mainly executed on single devices, by exploiting sensory data available on the device itself. The work presented in this paper starts investigating opportunistic service provisioning between pervasive devices, and is thus complementary.

Finally, service-oriented architectures have been investigated in the area of ubiquitous and pervasive computing, either assuming single hop wireless environments (WLANs, cellular networks), or well connected MANET [16, 19, 21, 25, 7, 13]. In such environments, which are less challenged from a networking standpoint, considering service composition issues is easier than in opportunistic networks, and indeed much effort has been devoted by the research community to this topic. Examples exist both of static service composition [25, 7, 13], in which the components are identified before requests are issued, and possible providers are sought in the network upon requests, and of dynamic composition [16], in which the composition is built dynamically when requests arise, based on what is available in the network. As already highlighted, in this paper we consider a different, more challenged networking environment, which poses completely novel challenges to service provisioning solutions. Being the networking environment so challenging, we limit the study in this paper to a simple service provisioning scheme, in which seekers can just avail of services directly provided by encountered providers.

3. SERVICE INVOCATION IN OPPORTUNISTIC COMPUTING ENVIRONMENTS

The model for service provisioning in opportunistic computing environments we consider in this paper is as follows. Whenever a seeker needs a service execution it looks for candidate providers in the environment. Whenever a suitable provider is encountered, the seeker may spawn a service execution on it. The execution process entails three stages, i.e., uploading the input parameters, executing the service, and downloading the output results. This process may take several contact times. While this process is ongoing, the seeker may encounter a different provider, and therefore an opportunity arises to spawn an additional service execution in parallel. In general, the same service request may generate multiple parallel executions. A service is complete when the seeker downloads the output results from any of the providers running in parallel. Whether a new parallel execution is spawned or not when an opportunity arises, depends on the *replication policy* implemented by the seeker.

Although quite simple, developing this service provisioning model in an efficient manner requires to address several challenging research questions. In opportunistic computing environments it is particularly important to optimise the

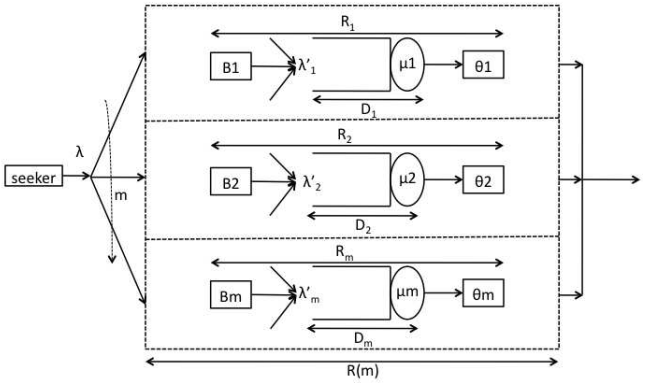

Figure 1: Scheme of the replication process

use of the resources, which are typically contributed by mobile users' devices. The replication policy plays a key role from this standpoint. In a resource unlimited environment (both in terms of computation and bandwidth) clearly the best option would be to replicate service executions onto all encountered providers. In general, running more replicas in parallel tends to reduce the expected service (execution) time (the time required by the seeker to receive results), as results can be collected from any of the providers on which a replica has been spawned. However, when resource constraints are considered, uncontrolled replication may lead to resource saturation and thus to exponential increase of the expected service execution time, as we demonstrate in the following.

The goal of this paper is therefore to characterise through an analytical model the expected service time as a function of the number of spawned replicas, considering two key resource limitations: the computational capabilities of the providers, and the bandwidth limitations between provider-seeker pairs. The model allows us to derive the optimal operating point for the replication policy, i.e., the number of replicas that should be spawned to minimise the expected service execution time. This also corresponds to the optimal use of the computational and bandwidth resources.

4. ANALYTICAL MODEL

In the model we consider a specific service S_j, and analyse the service execution time as a function of the number of replicas to be spawned by seekers, referred to as m in the remainder of the paper. Specifically, we focus on a tagged seeker, and derive the expected execution time when m executions are spawned on different providers, hereafter referred to as $E[R_j(m)]$. The optimal operating point is the value of m that minimises $E[R_j(m)]$. Note that we assume M_j nodes in the whole network providing the service S_j, thus $1 \leq m \leq M_j$ must hold.

The rationale of the model derivation is captured in Figure 1 (to simplify the notation, we omit index j in the figure). We assume that each service S_j's seeker issues requests according to a Poisson process with rate λ_j, and that there are k_j such seekers in the network. Therefore, the total request rate for service S_j is $\lambda_j k_j$. Each of the m replicas is represented with an horizontal pipe in Figure 1. For each particular request, the i-th provider starting the *execution* in temporal order is represented by pipe i.

Each pipe consists of three stages. The first stage represents the time required to complete the i-th upload of the input parameters, and is referred to as B_{ji}. The second stage represents the time required to execute the request after it is spawned, and is referred to as D_{ji}. The third stage represents the time required by the seeker to complete the download of the output results from the i-th provider, and is referred to as θ_{ji}. The delay on the i-th pipe is thus $R_{ji} = B_{ji} + D_{ji} + \theta_{ji}$, and the service execution time experienced by a node issuing a request is $R_j(m) = \min_{i=1,\ldots,m}\{R_{ji}\}$.

The offered load on pipe i, denoted as λ'_{ji}, can be evaluated as follows. The overall offered load for service S_j is $\lambda_j k_j m$. This rate is split among M_j providers. By denoting with f_{ji} the percentage of executions of service S_j spawned on provider i, the offered load on pipe i becomes $\lambda_j k_j m f_{ji}$. The values of f_{ji} depend on the properties of the seekers and providers mobility process.

Note that, with respect to the scheme in Figure 1, bandwidth limitations affect the delays of the first and third stage, while computational limitations affect the delay of the second stage. Specifically, a contact event might not be long enough to upload all service S_j's input parameters and/or download all output results. In such cases, uploads/downloads are resumed at the next contact opportunity with the same provider.

Hereafter we separately analyse the delays of the three stages, and then derive the service time $R_j(m)$. In general, the distributions of the random variables (r.v.) R_{ji} are fairly complex, and therefore it is possible to provide a closed form expression for their minimum $R_j(m)$ only under particular assumptions. Specifically, we will fully describe the case in which the r.v. R_{ji} are assumed to be exponential. Although in the following we are able to provide a more precise characterisation of some R_{ji} components, note that, under the assumption that the r.v. R_{ji} are exponential, it is sufficient to derive the *average* values of the delays of the stages in order to fully characterise R_{ji}.

Before presenting the analysis, it is worth focusing on a general concept that will be used extensively hereafter. Whenever a new request for service S_j is issued by the seeker, the seeker has to spawn m replicas. The execution of a replica on a provider actually starts when the corresponding input parameters are *completely* uploaded on the provider. Therefore, the m replicas are spawned *on the first m providers (out of M_j) on which the upload of the input parameters completes*. As uploads may require several contacts to complete, the providers on which the replicas are executed are not necessarily the first m providers encountered by the seeker after the request is generated. This has a non-trivial side effect on mapping providers to pipes of Figure 1. Specifically, pipe i corresponds to the provider on which the i-th upload completes, starting from the point in time when the request is issued at the seeker. In other words, if the set of r.v. $\{\beta_l\}_{l=1,\ldots,M}$ denotes the time intervals needed by the seeker to upload the input parameters on providers, starting from the point in time when the request is generated, then pipe i corresponds to provider $\hat{l}$ iff $\beta_{\hat{l}}$ is the i-th shorter time in the set $\{\beta_l\}$.

For the sake of readability, without loss of generality, in the following we omit, in the notation, to indicate the dependence on the particular service S_j we consider. As a

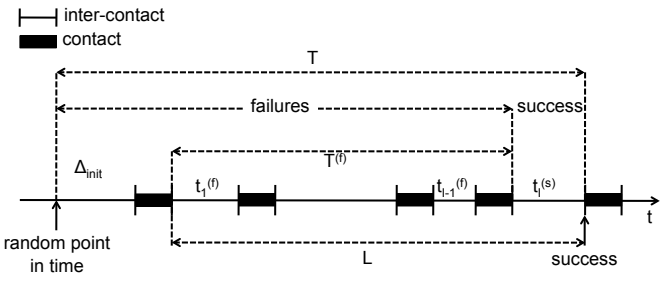

Figure 2: Scheme of the general contact process among nodes.

final remark, we provide hereafter a concise description of the analytical details, preferring to focus more on the rationale behind, and the meaning of the analytical derivations. The detailed proofs of the lemmas and theorems are omitted from the main flow of the paper, and can be found in the Appendices.

4.1 Model of the contact process between nodes

Before analysing the delay of the three stages of the pipes, and from them deriving the expected service time $E[R(m)]$, in this section we provide a general result, which is instrumental to the following parts of the analysis. Specifically, we focus on a tagged node, and analyse the time required by that node to encounter any node in a given subset of the network nodes, starting from a random point in time. We denote as success the event by which the tagged node finds any node belonging to the sought subset, and as T the time for success, starting from a random point in time. This general result will be used afterwards to model the time required by the seeker to meet possible providers on which to spawn executions, by a tagged provider to meet one of the seekers that might spawn requests on it, and by a tagged provider to meet a particular seeker after a service execution spawned by that seeker has completed.

We assume that inter-contact times (contact times) between the tagged node and any other nodes in the subset can be divided in two classes: The first class includes inter-contact (contact) times after which a success occurs, the second class inter-contact (contact) times after which a failure occurs. We assume that inter-contact (contact) times within each class are independent and identically distributed (iid), that contact and inter-contact times are mutually independent, and that the next contact of the tagged node is independent on the previous contacts. Inter-contact (contact) times of different classes may have different distributions, and these distributions can be arbitrary. Note that this is a more general setting with respect to typical assumptions in the models for opportunistic networks, where inter-contact (contact) times are assumed completely iid and following an exponential distribution, at least in the tail (e.g. [22, 5, 12]).

In general, we can represent the process as in Figure 2. To derive $E[T]$ we condition on the specific point in the mobility process of the tagged node from where we start measuring T (remember that T starts at a random point in time), and apply the law of total probability. The starting point of T may fall i) during a contact time with one of the sought nodes; ii) during a contact time with a node not in the sought subset; iii) during an inter-contact time after which a success occurs; iv) during an inter-contact time after which a failure occurs. These events will be denoted by $C^{(s)}$, $C^{(f)}$, $IC^{(s)}$ and $IC^{(f)}$, and their probabilities as $p_c^{(s)}$, $p_c^{(f)}$, $p_{ic}^{(s)}$ and $p_{ic}^{(f)}$. Those probabilities can be derived with routine analysis as shown in Appendix A, and are functions of the average inter-contact and contact times of the two classes (success, failure), throughout referred to as $E[c^{(s)}]$, $E[c^{(f)}]$, $E[t^{(s)}]$, and $E[t^{(f)}]$, respectively, and of the probability that an inter-contact time results in (a contact time is) a success, p_s. As in case i) T is clearly 0, $E[T]$ can be written as

$$E[T] = p_c^{(f)} E\left[T|C^{(f)}\right] + p_{ic}^{(s)} E\left[T|IC^{(s)}\right] + p_{ic}^{(f)} E\left[T|IC^{(f)}\right]. \quad (1)$$

The expression of $E\left[T|IC^{(s)}\right]$ can be derived by noting that in case iii) above, T is the residual inter-contact time after which a success occurs (denoted as $t_+^{(s)}$):

$$E\left[T|IC^{(s)}\right] = E\left[t_+^{(s)}\right] = \frac{E\left[t^{(s)}\right]}{2} + \frac{Var\left[t^{(s)}\right]}{2E\left[t^{(s)}\right]}.$$

In the remaining cases ii) and iv), we have to account for an initial component $\Delta_{init,k}$, representing the time between start of T and the end of the following contact time. After $\Delta_{init,k}$ there is a random number $I-1$ (possibly equal to 0) of inter-contact and contact times after which a failure occurs, and a final inter-contact time after which success occurs. As proved in Appendix A, we obtain:

$$\begin{aligned} E[T|IC^{(f)}] &= E[t_+^{(f)}] + \frac{E[t^{(f)}] + E[c^{(f)}]}{p_s} - E[t^{(f)}] + E[t^{(s)}] \\ E[T|C^{(f)}] &= E[c_+^{(f)}] + \left(\frac{1}{p_s} - 1\right)\left(E[t^{(f)}] + E[c^{(f)}]\right) + E[t^{(s)}]. \end{aligned}$$

In the following analysis, in addition to T, we will also need the time to success starting from the end of a contact, referred to as L. It is easy to see that $E[L]$ can be written as follows:

$$E[L] = \left(\frac{1}{p_s} - 1\right)\left(E[t^{(f)}] + E[c^{(f)}]\right) + E[t^{(s)}]. \quad (2)$$

Note that $E[T]$ and $E[L]$ depend only on the average inter-contact and contact times, and on the probability of an inter-contact time resulting in (a contact time being) a success (p_s), i.e. on the properties of the mobility process and on the probability of encountering one of the sought nodes.

4.2 Analysis of the first stage

In order to model the delay of the first stages $\{B_i\}_{i=1,\ldots,m}$, we first analyse the time required by the tagged seeker to upload the input parameters to any individual provider from the point in time when the request is generated. Recall that these time intervals are denoted as $\{\beta_l\}_{l=1,\ldots,M}$. As discussed at the beginning of the section, pipe i corresponds to the i-th provider on which the input parameters are uploaded, and therefore B_i is the i-th shortest element in the set $\{\beta_l\}$.

The figure β_l can be modelled as an M/G/1 queue. Requests are generated at the seeker according to a Poisson process with rate λ. The service time of the queue ($\hat{\beta}_l$) includes the time to meet provider l enough times to upload the input parameters of the service. Therefore, it accounts both for the inter-contact process between the seeker and the provider, and the constraint on the available bandwidth. Let us denote by T_l and L_l the time required by the seeker to meet provider l, respectively, starting from an arbitrary

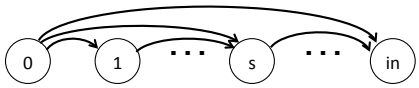

Figure 3: Embedded process for input parameters upload.

point in time or from the end of a contact (as shown in Section 4.1), and by q_l the number of contact times needed to complete the input parameters upload. Then, $E[\beta_l]$ is provided in Lemma 1.

LEMMA 1. *The average delay to complete the upload of the input parameters on provider l is*

$$E\left[\beta_l\right] = \frac{E\left[\hat{\beta}_l\right]}{1 - \lambda E\left[\hat{\beta}_l\right]}. \quad (3)$$

where $E[\hat{\beta}_l]$ is

$$E\left[\hat{\beta}_l\right] = \frac{E\left[T_l\right] - E\left[L_l\right] + E\left[q_l\right]\left(E\left[L_l\right] + E\left[c^{(s)}\right]\right)}{\frac{M}{m} - \lambda\left(E\left[L_l\right] - E\left[T_l\right]\right)}. \quad (4)$$

The figure q_l accounts for the bandwidth limitations, and is derived as follows. Let us denote with *in* the r.v. representing the size of the input parameters to be uploaded. Provider l has to be encountered, in general, more than once by the seeker, as a single contact might not last long enough to upload *in* bytes. We thus consider an embedded Markov process, whose state is the number of bytes already uploaded to provider l, and whose transitions occur upon each contact between the seeker and the provider (see Figure 3).

Assuming that uploads always start at the beginning of a contact, the chain always starts in state 0 at the beginning of the first contact. The chain moves from the state s to the state $s+h$ if there is enough bandwidth and the contact is long enough to transmit h bytes, i.e., with a probability equal to $p\left(h \leq b \cdot c^{(s)} < h+1\right)$ where b is the bandwidth available during a contact. The number of contacts required to complete the upload process is thus the first passage time of the process from state *in*, starting from state 0.

The line of reasoning for completing the analysis of B_i is as follows. Conditioning on the fact that the i-th ordered shortest times in the set $\{\beta_l\}$ is known, say $\{\beta_{k1}, \ldots \beta_{ki}\}$, then B_{i+1} is the minimum over the rest of the variables in $\{\beta_l\}$. Assuming that the random variables (r.v.) β_l are exponential and independent, we can provide closed form expressions both for the probability of the set $\{\beta_{k1}, \ldots \beta_{ki}\}$ being the i-th shortest r.v. in $\{\beta_l\}$, and for the distribution (and average value) of B_i. This is derived in Lemma 2 and Theorem 1. As a matter of notation, in the following $\phi_l = 1/E[\beta_l]$.

LEMMA 2. *The probability of the set $\{\beta_{k1}, \ldots \beta_{ki}\}$ being the i-th ordered shortest variables in the set $\{\beta_l\}_{l=1,\ldots,M}$ is*

$$P(\beta_{k1}, \ldots, \beta_{ki}) = \frac{\prod_{p=1}^{i} \phi_{kp}}{\sum_{p=1}^{M} \phi_p \sum_{\substack{p=1 \\ p\neq k1}}^{M} \phi_p \cdots \sum_{\substack{p=1 \\ p\neq k1,\ldots,k(i-1)}}^{M} \phi_p} \quad (5)$$

THEOREM 1. *The time required by a seeker to complete the upload of the input parameters on provider $i+1$ ($i = 0, \ldots, m-1$) is distributed as follows:*

$$B_{i+1} \sim \sum_{k1=1}^{M} \cdots \sum_{\substack{ki=1 \\ ki\neq k1,\ldots,k(i-1)}}^{M} P(\beta_{k1}, \ldots, \beta_{ki}) \exp\left(\sum_{\substack{p=1 \\ i\neq k1,\ldots,ki}}^{M} \phi_p\right)$$

and its average value is:

$$E\left[B_{i+1}\right] = \sum_{k1=1}^{M} \cdots \sum_{\substack{ki=1 \\ ki\neq ki,\ldots,k(i-1)}}^{M} \frac{P(\beta_{k1}, \ldots, \beta_{ki})}{\sum_{\substack{p=1 \\ p\neq k1,\ldots,ki}}^{M} \phi_p}. \quad (6)$$

Note that the above expression greatly simplifies when the r.v. β_l are also identically distributed, with rate ϕ. In this case, B_{i+1} is exponential with rate $(M-i)\phi$ (and its average value is thus $1/[(M-i)\phi]$). This is a very intuitive result, as B_{i+1} becomes the minimum over $M-i$ iid exponential random variables with rate ϕ.

4.3 Analysis of the second stage

As shown in Figure 1 we model the computation process at each provider with a queue. The service time of the queue represents the computation time at the provider's CPU. We assume that the computation time at providers is exponentially distributed with rate $\mu_l, l = 1, \ldots, M$. Considering a random computation time allows us to account for variations in providers capabilities, and variations in computation times of different requests for the service. To derive the average delay of the stage of the generic pipe i, we assume to know that the generic provider l corresponds to pipe i, and derive the conditioned average delay. This means assuming that the seeker completed the i-th upload of the input parameters on provider l. By computing the probability of this event, we can apply the law of total probability, and derive the average value of the second stages.

Focusing on a particular provider l, it is possible to model the second stage as a batch arrival system (see, e.g., [23]): A batch of requests arrive at the provider when it meets a seeker, and the size of the batch is the number of (sets of) input parameters for requests queued at the seeker whose upload completes during the contact. As a matter of notation, we describe the second stage as an $M^{[X]}/M/1$ system, where X is the r.v. denoting the size of the batch. Lemma 3 provides the average delay of the second stage of pipe i, assuming provider l corresponds to it, denoted by $E[D_i|l]$. In the formula we use the component $E[L_l^{(P)}]$, denoting the average time for the provider l to meet any seeker starting from the end of a contact. $E[L_l^{(P)}]$ can be computed as shown in Section 4.1.

LEMMA 3. *The average delay of the second stage of pipe i, assuming provider l corresponds to it is*

$$E[D_i|l] = \frac{E[X_l^2] + 2E[X_l]}{2\mu_l E[X_l](1-\rho_l)}, \quad (7)$$

where X_l is the size of the batches arriving at provider l. Furthermore, the utilisation of the providers is

$$\rho_l = \frac{\lambda_{X,l} E[X_l]}{\mu_l} = \frac{E[X_l]}{\mu_l(E[L_l^{(P)}] + E[c^{(s)}])}, \quad (8)$$

where $\lambda_{X,l}$ is the rate of batch arrivals at provider l.

The result in Lemma 3 confirm our initial intuition about the possibility of saturation of the service provisioning system. The utilisation of providers increases with the batch size and with the number of seekers "using" the provider (which clearly means an increase of $\lambda_{X,l}$). When the utilisation approaches 1, the average delay on the providers tends to infinity.

Recalling the analysis of the first stage, the probability that provider l corresponds to pipe i is the probability that the time to complete the upload of the input parameters on provider l is the i-th shortest time in the set $\{\beta_l\}_l$, i.e. $P(B_i = \beta_l)$. The analysis of this figure is similar to that related to Lemma 2. Specifically, by exploiting the result of Lemma 2 and recalling that ϕ_l is equal to $1/E[\beta_l]$, we obtain the result in the following Lemma.

LEMMA 4. *The probability of provider l corresponding to pipe $i+1$ $(i = 0, \ldots, m-1)$ is*

$$P(B_{i+1} = \beta_l) = \sum_{\substack{k1=1 \\ k1 \neq l}}^{M} \cdots \sum_{\substack{ki=1 \\ ki \neq k1, \ldots, k(i-1), l}}^{M} \frac{P(\beta_{k1}, \ldots, \beta_{ki})\phi_l}{\sum_{\substack{s=1 \\ s \neq k1, \ldots, ki}}^{M} \phi_s}. \quad (9)$$

Based on Lemmas 3 and 4 it is straightforward to derive the expression of $E[D_i]$.

THEOREM 2. *The average delay to complete the service computation of the i-th replica is*

$$E[D_i] = \sum_{l=1}^{M} E\left[D_i|l\right] P(B_i = \beta_l), \quad (10)$$

where $E\left[D_i|l\right]$ and $P(B_i = \beta_l)$ are as in Equations 7 and 9.

4.4 Analysis of the third stage

The delay of the third stage is the time interval from the end of the execution of a request at the provider, until the point in time when the output results are completely downloaded to the seeker. Note that, in general, when a request execution for a particular seeker completes at the provider, output results from previously completed requests for the same seeker might still be waiting to be downloaded to the seeker. Therefore, we must model the third stage with a queue. Actually, the analysis of the third stage is conceptually similar to that of the first stage, albeit for the fact that the roles of the provider and seeker have to be switched (now it is the provider that has to transfer data to the seeker). Therefore, we model the third stage with an M/G/1 queue, and the service time of the queue is the time for the provider to download the output parameters to the seeker.

Also in this case, to derive the average delay of the stage of the generic pipe i, we assume to know that the generic provider l corresponds to pipe i, derive the conditioned average delay, and finally apply the law of total probability. As a matter of notation, we denote with θ_i the delay of the third stage of pipe i, with $\hat{\theta}_i$ the service time of the queue representing the stage, and with $\theta_i|l$ and $\hat{\theta}_i|l$ the same figures conditioned to the fact that the pipe corresponds to provider l. Recalling that we denote by λ'_i the total offered load on pipe i, and by k the number of seekers, Lemma 5 provides the closed form expression for the average delay conditioned to provider l. In the Lemma we denote by T_l and L_l the time required by provider l to meet the seeker (as can be derived by exploiting the analysis in Section 4.1), and by q_l the number of contact events required to download the output results.

LEMMA 5. *The average delay of the third stage of pipe i conditioned to the fact that provider l corresponds to the pipe is*

$$E\left[\theta_i|l\right] = \frac{E\left[\hat{\theta}_i|l\right]}{1 - \frac{\lambda'_i}{k} E\left[\hat{\theta}_i|l\right]}, \quad (11)$$

where $E\left[\hat{\theta}_i|l\right]$ is:

$$E\left[\hat{\theta}_i|l\right] = \frac{E\left[T_l\right] - E\left[L_l\right] + E\left[q_l\right]\left(E\left[L_l\right] + E\left[c^{(s)}\right]\right)}{m - \frac{\lambda'_i}{k}\left(E\left[L_l\right] - E\left[T_l\right]\right)}.$$

Based on Lemmas 5 and 4 it is straightforward to derive the expression of $E[\theta_i]$.

THEOREM 3. *The average delay for the seeker to download the output results of the i-th replica $(i = 1, \ldots, m)$ is*

$$E\left[\theta_i\right] = \sum_{l=1}^{M} E\left[\theta_i|l\right] P(B_i = \beta_l), \quad (12)$$

where $E\left[\theta_i|l\right]$ and $P(B_i = \beta_l)$ are as in Equations 11 and 9, respectively.

4.5 Optimal replication

The expected service execution time experienced by the seeker ($R(m)$) is the minimum delay over the m pipes. As analytical expressions for the minimum of a generic set of r.v. are not available, we approximate $R(m)$ assuming that the r.v. R_i are exponentially distributed with rate $\gamma_i = (E[B_i] + E[D_i] + E[\theta_i])^{-1}$. $R(m)$ is then also exponentially distributed, with rate $\Gamma(m) = \sum_{i=1}^{m} \gamma_i$, and average value $E[R(m)] = \Gamma(m)^{-1}$.

In the general case, $E[R(m)]$ can be computed numerically, while closed formulas can be found under additional assumptions. In the following of the section we derive a closed formula of $E[R(m)]$ under the assumption that the r.v. $\hat{\beta}_l$, $D_i|_l$ and $\theta_i|_l$ do not depend on the particular provider, i.e., they are identically distributed across providers. Under these assumptions, the expressions of the average delay of the three stages in Equations 6, 10, 12 become simpler. For what concerns the second stage, with respect to Equation 10 all the dependencies on the particular provider disappear. As we discussed in Section 4.2, the average delay of the first stage on pipe i becomes the minimum over $M - i + 1$ exponentially distributed r.v. with rate $\phi = 1/E[\beta]$. Recalling the expression of $E[\beta_l]$ in Equation 3 we obtain

$$E\left[B_i\right] = \frac{E\left[\beta\right]}{M - i + 1} = \frac{1}{M - i + 1} \cdot \frac{mK_1}{M - m\lambda(K_1 + K_2)},$$

where K_1 is $E[T] - E[L] + E[q](E[L] + E[c^{(s)}])$, and K_2 is $E[L] - E[T]$. Note that K_1 and K_2 do not depend either on the particular pipe i nor on the number of replicas m. Finally, after similar routine manipulations, noting that the offered load λ' becomes equal to $\frac{\lambda k m}{M}$, the average delay of the third stage becomes

$$E\left[\theta\right] = \frac{K_1}{m - \frac{\lambda m}{M}(K_1 + K_2)}.$$

Note that the only term that depends on the particular pipe i is the delay of the first stage, due to the fact that the pipe corresponds to the i-th provider on which the seeker completes the upload of the input parameters.

Finally, we are in the position of deriving a closed form expression for the expected service time $E[R(m)] = 1/\Gamma(m)$, as in Theorem 4.

THEOREM 4. *The average time for the seeker to avail of service S_j is*

$$E[R(m)] = \frac{(E[D] + E[\theta])^2}{m(E[D] + E[\theta]) - E[\beta] \ln \frac{Q-1}{Q-m-1}}, \qquad (13)$$

where Q is equal to $\frac{E[\beta]}{ED+E\theta} + M + 1$.

The optimal replication level m_{opt} can be found by minimising Equation 13. This requires specifying the specific dependence of ED on m. Once this is done, m_{opt} can be either found analytically or numerically. The result provided by Theorem 4 is therefore general enough to be customised to the features of any specific scenario.

5. PERFORMANCE RESULTS

In this section we analyse the expected service (execution) time of a replication policy which exploits the analytical model of Section 4 (*optimal* policy). Specifically, various facets of such a policy are analysed. First, it is compared with two resource-unaware policies, which replicate requests on the first encountered node only (*single* policy), and on all encountered nodes (*greedy* policy), respectively. Note that the performance in terms of expected service time of these two policies can also be found by using the analysis presented before, as they correspond to $E[R(1)]$ and $E[R(M)]$, respectively. Comparing the three policies highlights the advantage of considering resource limitations in the service invocation process. Then, a sensitiveness analysis of the optimal policy is presented. We consider the impact of i) the probability of nodes being providers ($p^{(p)}$) or seekers ($p^{(s)}$) for the service; ii) the number of nodes in the network (N); iii) the request generation rate at seekers (λ); and iv) the average service computation time at providers ($1/\mu$).

In order to validate the analytical model, we also developed a simulation model based on the OMNeT++ simulator (`http://www.omnetpp.org/`). Simulation results are presented with confidence intervals computed with 95% confidence level. In the simulated environment the nodes move in a square, according to the Random WayPoint (RWP) mobility model, with the modifications described in [18] to guarantee that the model's distributions are stationary (the nodes' average speed is 1.5 m/s, representative of walking speeds). In the simulation model, the value of the allowed replications (m) is an input parameter of the simulation runs, and we repeated the simulation runs with increasing values of m. At the end of each run for a given value of m, we computed the corresponding average service time (with 95% confidence intervals)[1]. For what concerns the optimal policy, we identify the optimal case as the value of m achieving the minimum average service time.

Results presented hereafter show a very good agreement between simulation and analysis. The analytical model is thus able to well predict the trends of behaviour of the various policies under investigation. It is worth noting that the analytical model provides a much more flexible tool than the simulation model. Specifically, the inherent complexity of the simulation model (mainly, the number of events that are generated) makes it practically impossible to explore the policies behaviour over a large range of key parameters, while this becomes possible analytically. This is the case, for example, of the sensitiveness analysis with respect to the request generation rate λ and service computation time $1/\mu$, for which the simulation model becomes too complex (in terms of execution time and memory space), while the analytical model allows us to completely investigate the optimal policy's behavior.

[1]Note that confidence intervals are very narrow, and can hardly be noticed in the plots shown hereafter.

Table 1: Default analysis parameters

$1/\mu$	30s
λ	0.005 req/s
N	100
tx_range	20m
bandwidth	5.5Mbps
input param	100KB
output param	1MB
Area	1000m x 1000m
avg speed	1.5m/s
$p^{(s)}$	0.1,0.2,0.5,0.8
$p^{(p)}$	0.1,0.2,0.5,0.8

Details of the key parameters' values are provided for each investigated scenario. Unless otherwise stated, they are set as in Table 1. These settings represent typical opportunistic networking environments, in which the network is sparse (indeed we obtain average inter-contact times of about 10 minutes and contact times of about 15s). Note that the bandwidth value is a conservative estimate, as it is the typical throughput measured at the application level with 802.11b technologies operating at a nominal rate of 11Mbps [1].

5.1 Resource-aware vs naïve policies

In this section we compare the performance of the optimal (resource aware) policy against that of the single and greedy policies, which do not take resource constraints into account. Table 2 compares the average service time for an increasing percentage of seekers, and a representative percentage of providers ($p^{(p)} = 0.5$, similar results are obtained for the other values of $p^{(p)}$, as well). Results are shown, for each policy, both for the analytical and the simulation model.

For a small number of seekers (i.e., $p^{(s)} \leq 0.2$), the optimal and greedy policies basically coincide, as for such a small percentage of seekers the "overall computational capacity" of the system is large enough to afford greedy replication. However, when the number of seekers increases beyond this point, the greedy policy saturates the system. In these cases, the expected service time is infinite. Simulation results are the average over completed executions, which gives an indication of the exponential increase of the simulated delay. On the contrary, beyond $p^{(s)} = 0.2$ the optimal policy progressively reduces the number of spawned replicas, and significantly outperforms both the single and the greedy policy. Also note that the optimal policy results in just a slight increase of the expected service time as the number of seekers increases.

Table 2: Policies comparison, $p^{(p)}$=0.5

$p^{(s)}$	optimal(an)	optimal(sim)	single(an)	single(sim)	greedy(an)	greedy(sim)
0.1	9886.32	8899.77±135.025	45914.6	44933±1313.36	9888.7	8904.68±137.857
0.2	10030.6	9567.35±108.429	45920.6	46236.7±1007.69	10072.9	9931.05±102.3
0.5	11883.7	12431.5±127.545	45940.8	44559.5±621.437	∞	69154.4±801.452
0.8	15381.7	16945.1±133.773	45965.8	45031.6±509.473	∞	167508±1501.26

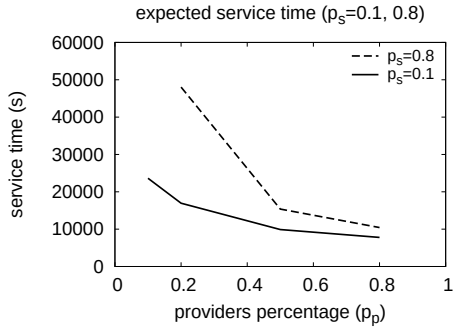

Figure 4: Sensitiveness on the number of providers

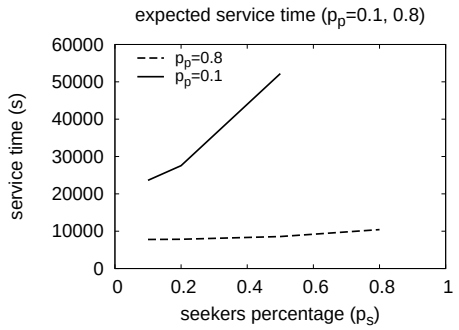

Figure 5: Sensitiveness on the number of seekers

These results clearly indicate the significant performance gain that a resource-aware policy achieves with respect to resource-unaware policies. From now on, we therefore focus on assessing the performance of the optimal policy with respect to a number of parameters. Furthermore, as the analytical model shows to predict very well the trend of simulation results, in the rest of the analysis we use the analytical model only.

5.2 Sensitiveness on the seekers and providers population

Figures 4 and 5 show the expected service time as a function of the percentage of providers and seekers, respectively. In each plot, two curves are shown, for the minimum and maximum values of the other parameter ($p^{(s)}$ and $p^{(p)}$, respectively).

The expected service time increases when either less providers are available, or more seekers are present. In some configurations, this may result in the resources' saturation, as is the case of $p^{(s)} = 0.8, p^{(p)} = 0.1$. For a given percentage of seekers (providers) the service time decreases (increases) as the percentage of providers (seekers) increases. Note that, unless for particularly congested settings (very low percentage of providers or very high percentage of seekers), using the optimal policy results in a graceful degradation of the performance, as the expected service time gracefully increases (e.g., see the curve for a varying number of providers and $p^{(s)} = 0.1$, or the curve for a varying number of seekers and $p^{(p)} = 0.8$).

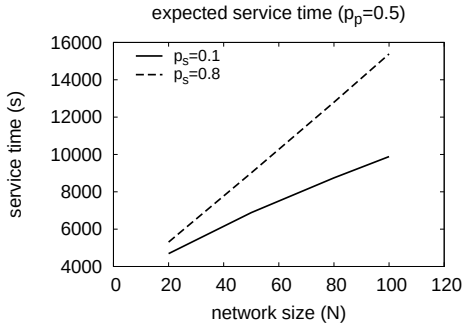

Figure 6: Impact of the network size

5.3 Sensitiveness on the network size

In this section we study the sensitiveness of the optimal policy with the number of nodes in the network (N). To keep a scenario representative of opportunistic networking environments, we scale the size of the simulation area to keep the node density constant (this indeed results in invariant average contact and inter-contact times). Specifically we consider the cases $N = 20, 50, 80, 100$. Figure 6 shows the expected service time as a function of N, for the two extreme values of $p^{(s)}$ and a representative percentage of providers ($p^{(p)} = 0.5$).

The interesting feature shown by the plots is that the expected service time linearly increases with the network size. Overall, the main reason behind this behaviour is due to the delay for downloading the output results to the seeker (i.e., the delay of the third stage). Intuitively, after a provider has completed the execution, it must encounter a single tagged seeker to download the output parameters. The probability of this event can be shown to be proportional to $1/N$, which results in a linear increase of this part of the service delay with N. More in detail, it is possible to write the expected delay on pipe i as the sum of two components, one independent of N, and another one linear with N, i.e., $ER_i = EW_i + NEY_i$. EY_i is dominated by the delay of the third stage, and can thus be approximated by $E\theta$. Assuming, again, that the r.v. R_i are exponential, and recalling that the optimal service time is the minimum over the delays of m_{opt} pipes, $R(m_{opt})$ is also an exponential r.v. with rate $\gamma_R = \sum_{j=1}^{m_{opt}} \frac{1}{EW_i + NE\theta}$. Finally, it can also be shown that, unless when the system is extremely close to the saturation of the second stage, the factor $NE\theta$ dominates over EW_i.

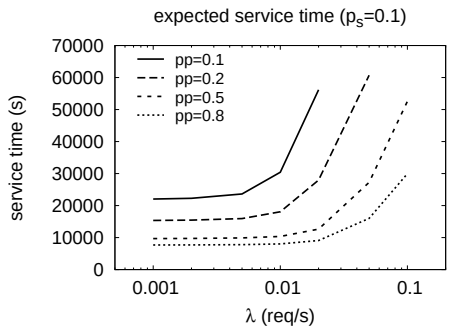

Figure 7: Sensitiveness on λ

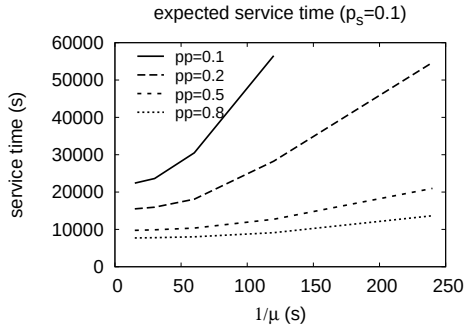

Figure 8: Sensitiveness on μ

After simple manipulations, we find that the expected optimal service time is approximately equal to $E\theta\frac{N}{m_{opt}}$. This confirms the linear dependence of $ER(m_{opt})$ with N found in Figure 6(b). Furthermore, it also justifies the fact that the slope of the line increases with $p^{(s)}$, because, for a given value of $p^{(p)}$, m_{opt} clearly decreases with the percentage of seekers.

5.4 Sensitiveness on the load on providers

The final aspect we analyse is the sensitiveness of the optimal policy with the load on providers. This can be shown by varying either the seekers' request rate λ (Figure 7), or the average providers' computation time $1/\mu$ (Figure 8). Specifically, we scale the value of λ according to a logarithmic scale from 0.001 req/s up to 0.1 req/s, while we scale the value of $1/\mu$ by doubling it from 15s up to 240s. Note that we show results for the lowest percentage of seekers ($p^{(s)} = 0.1$), i.e., we do not further congest the system with a high number of seekers, to better isolate the dependence on λ and μ.

Both plots essentially highlight the same feature of the optimal policy. As expected, the service time increases with the load of the providers, which can manifest either as an increase of the seekers' request rate, or as an increase of the computation time for generating the service results. This increase is moderate when the percentage of providers is high enough (e.g., for $p^{(p)} = 0.8$), meaning that the total "service capacity" of the system is high enough to tolerate increases of the offered load. For lower numbers of providers, the increase of the service time becomes more evident. For a very low percentage of providers ($p^{(p)} = 0.1$) even the optimal policy can work only up to a certain load, while the system inevitably becomes saturated beyond this limit. This is shown in the plots by the fact that the curves stop for $\lambda > 0.02$ req/s and $1/\mu > 120$s, respectively.

6. CONCLUSIONS

In this paper we have derived a complete analytical model for service invocation and provisioning in opportunistic computing environments. The goal of the model is to take into consideration a very general scenario, featuring different mobility patterns, and, very importantly, resource constraints both in terms of computation and bandwidth capabilities. We have also shown how the model simplifies when less general conditions can be assumed.

Employing the proposed model, it is possible to analyse the performance of optimal service invocation, in terms of expected service time (the time required by seekers to receive output results). In this paper, we have highlighted several features. First of all, we have shown that considering resource constraints in service invocation policies is a must to optimise the performance. Naïve, resource-unaware policies either easily saturate the available resources, or significantly under-utilise them, while the optimal, resource-aware policy always optimises their use. Second, we have shown that the optimal policy is able to automatically counteract increased demand on resources, which may arise due to different reasons. Increased resource demand might arise either because of a reduced number of providers, an increased number of seekers, an increased load in terms of request rate or service computation time. In all of these cases, we have shown that the optimal invocation policy results in a graceful degradation of the system performance, until a point where the overall service capacity of the system is too low to cope with the total offered load. Finally, we have highlighted an intrinsic increase of the service time with the network size, which is an inevitable side effect of finding a particular user in larger populations.

To the best of our knowledge, this paper is one of the first considering the challenging problem of service provisioning in opportunistic networks. Despite providing already significant results, there is still ample room for future research directions. Among the most interesting ones, we mention the design of distributed algorithms that implement when possible, or approximate otherwise, the optimal policy found by analysis. Characterising the impact of opportunistic multi-hop forwarding or requests upload and results download is another interesting extension. Finally, it will also be important to understand how the opportunistic computing paradigm can work when services available on different nodes can be composed together in order to provide richer functionality.

Acknowledgments

This work was partially funded by the European Commission under the SCAMPI (258414) FIRE Project and by the US National Science Foundations award CNS-0834493.

7. REFERENCES

[1] G. Anastasi, E. Borgia, M. Conti, E. Gregori, and A. Passarella. Understanding the real behavior of Mote and 802.11 ad hoc networks: an experimental approach. *Pervasive and Mobile Computing*, 1(2):237–256, 2005.

[2] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proc. ACM SIGCOMM*, pages 373–384, August 2007.

[3] C. Boldrini, M. Conti, and A. Passarella. Exploiting users' social relations to forward data in opportunistic networks: The hibop solution. *Pervasive and Mobile Computing*, 4(5):633 – 657, 2008.

[4] C. Boldrini, M. Conti, and A. Passarella. Design and performance evaluation of contentplace, a social-aware data dissemination system for opportunistic networks. *Computer Networks*, 54(4):589 – 604, 2010. Advances in Wireless and Mobile Networks.

[5] H. Cai and D. Y. Eun. Crossing Over the Bounded Domain: From Exponential To Power-law Inter-meeting Time in MANET. In *Proc. of ACM MobiCom*, 2007.

[6] A. Chaintreau, P. Hui, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. Mob. Comp.*, 6(6):606–620, 2007.

[7] D. Chakraborty, F. Perich, A. Joshi, T. Finin, and Y. Yesha. A Reactive Service Composition Architecture for Pervasive Computing Environments. In *Proc. of IFIP PWC*, 2002.

[8] M. Conti and S. Giordano. Multihop ad hoc networking: The reality. *Communications Magazine, IEEE*, 45(4):88–95, 2007.

[9] M. Conti, E. Gregori, and L. Lenzini. *Metropolitan Area Networks*. Springer, 1997.

[10] M. Conti and M. Kumar. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, Jan. 2010.

[11] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE JSAC*, 26(5), 2008.

[12] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: a social network perspective. In *MobiHoc '09: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 299–308, New York, NY, USA, 2009. ACM.

[13] S. Helal, N. Desai, V. Verma, and C. Lee. Konark - a service discovery and deliveryprotocol for ad-hoc network. In *Proc. of IEEE WCNC*, 2003.

[14] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 241–250, New York, NY, USA, 2008. ACM.

[15] A. Jindal and K. Psounis. Contention-aware performance analysis of mobility-assisted routing. *IEEE Transactions on Mobile Computing*, 8(2):145–161, 2009.

[16] S. Kalasapur, M. Kumar, and B. A. Shirazi. Dynamic Service Composition in Pervasive Computing. *IEEE TPDS*, 18(7):907 – 918, 2007.

[17] H. Lu, N. D. Lane, S. B. Eisenman, and A. T. Campbell. Bubble-sensing: Binding sensing tasks to the physical world. *Pervasive and Mobile Computing*, 6(1):58 – 71, 2010.

[18] W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004.

[19] M. Papazoglou. Service Oriented Computing: Concepts, characteristics and directions. In *Proc. of IEEE WISE*, December 2003.

[20] A. Passarella, M. Kumar, M. Conti, and E. Borgia. Minimum-Delay Service Provisioning in Opportunistic Networks. In *IIT-CNR Tech. Rep., available at* `http://bruno1.iit.cnr.it/~andrea/tr/services-tr.pdf`, 2010.

[21] M. Ravi, P. Stern, N. Desai, and L. Iftode. Accessing Ubiquitous Services using Smart Phones. In *Proc. of IEEE PerCom*, March 2005.

[22] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-copy Case. *IEEE Trans. on Net.*, 2008.

[23] H. Takagi. *Queuing Analysis Volume I: Vacation and Priority Systems, Part I*. North-Holland, 1991.

[24] Y. Wang, H. Wu, F. Lin, and N.-F. Tzeng. Cross-layer Protocol Design and Optimization for Delay/Fault-Tolerant Mobile Sensor Networks. *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Delay and Disruption Tolerant Wireless Communication*, 26(5):809–819, 2008. A preliminary version was presented in IEEE ICDCS'07.

[25] X.Gu and K. Nahrstedt. Dynamic QoS-aware multimedia service configuration in ubiquitous computing environments. In *Proc. of IEEE ICDCS*, July 2002.

[26] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang. Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing. In *Proc. ACM Intl. Conf. on Mobile Computing and Networking (Mobicom)*, pages 195–206, September 2007.

APPENDIX

A. CONTACT PROCESS BETWEEN NODES

In this Appendix we provided a detailed discussion of some derivations related to the model of contact process between nodes, presented in 4.1.

Recall that the goal of this part of the model if finding the time (T) required for a tagged node to meet any node in a given subset starting from a random point in time. To model T, we condition on the specific point of the tagged node's mobility process from where we start measuring T. This point may fall i) during a contact time with one of the sought nodes; ii) during a contact time with a node not in the sought subset; iii) during an inter-contact time after which a success occurs; iv) during an inter-contact time after which a failure occurs. These events are denoted by $C^{(s)}$, $C^{(f)}$, $IC^{(s)}$ and $IC^{(f)}$.

First of all, we derive the probabilities of these events, referred to as $p_c^{(s)}$, $p_c^{(f)}$, $p_{ic}^{(s)}$ and $p_{ic}^{(f)}$. Let us define a stochastic process $\{Y(t), t \geq 0\}$ that can be in four possible states, each corresponding to one of the conditions i)-iv) above (hereafter, these states will be referred to as I to IV, respectively). Essentially, $Y(t)$ describes the state of the contact and inter-contact process between the tagged node and the nodes in the sought subset. Considering the sequence of points in time when a success occurs $\{S_n, n = 0, 1, \ldots\}$, and as we assumed that contact and inter-contact events do not depend on previous contacts and inter-contacts, it follows that $Y(t)$ is a regenerative process that regenerates after each success. Let us finally denote by $\{F_n, n = 0, 1, \ldots\}$ the length of the n-th (regeneration) cycle, i.e. $F_n = S_n - S_{n-1}$. The probabilities $p_c^{(s)}$, $p_c^{(f)}$, $p_{ic}^{(s)}$ and $p_{ic}^{(f)}$ can be computed as the probabilities of the states of $Y(t)$, which can be found by analysing a generic cycle of the process, shown in Figure 9. Denoting by J the number of inter-contacts (and contacts) in a cycle, the cycle is characterised by an initial contact following the success, $J-1$ inter-contacts (and the following contacts) after which a failure occurs, and a final inter-contact after which a success occurs. Let us focus on state I, i.e., the process is in the contact time after success. By applying results on regenerative process (see, e.g., [9]), the probability of state I is the ratio between the average time spent by $Y(t)$ in state I during a cycle, and the average cycle length $E[F_n]$. By noting that the probability of a success p_s is $1/E[J]$, we obtain the following expression:

$$\begin{aligned} p_c^{(s)} &= \frac{E[c^{(s)}]}{(E[J]-1)(E[t^{(f)}]+E[c^{(f)}])+E[c^{(s)}]+E[t^{(s)}]} \\ &= \frac{p_s E[c^{(s)})]}{(1-p_s)(E[t^{(f)}]+E[c^{(f)}])+p_s(E[c^{(s)}]+E[t^{(s)}])}. \end{aligned}$$

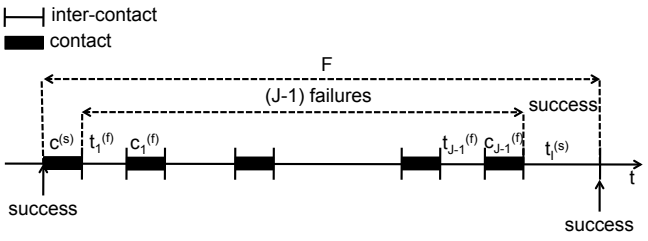

Figure 9: Form of the generic regeneration interval

The other probabilities can be found according to the same line of reasoning, thus obtaining

$$\begin{aligned} p_c^{(f)} &= \frac{(1-p_s)E[c^{(f)})]}{(1-p_s)(E[t^{(f)}]+E[c^{(f)}])+p_s(E[c^{(s)}]+E[t^{(s)}])} \\ p_{ic}^{(s)} &= \frac{p_s E[t^{(s)})]}{(1-p_s)(E[t^{(f)}]+E[c^{(f)}])+p_s(E[c^{(s)}]+E[t^{(s)}])} \\ p_{ic}^{(f)} &= \frac{(1-p_s)E[t^{(f)})]}{(1-p_s)(E[t^{(f)}]+E[c^{(f)}])+p_s(E[c^{(s)}]+E[t^{(s)}])}. \end{aligned}$$

The second result we derive in this Appendix are the expressions for $E[T|IC^{(f)}]$ and $E[T|C^{(f)}]$, which are the average values of T, conditioned to the fact that T starts during an inter-contact time and a contact time after which a failure occurs, respectively. Let us focus on $E[T|IC^{(f)}]$ first. Recall that in this case T starts with an interval $\Delta_{init,k}$, until the end of the next contact time. After $\Delta_{init,k}$ there is a random number $I-1$ (possibly equal to 0) of inter-contact and contact times after which a failure occurs, and a final inter-contact time after which success occurs. As in this case T starts in a random point in time during an inter-contact time after which a failure occurs, the following expression holds true:

$$T|IC^{(f)} = t_+^{(f)} + c^{(f)} + \sum_{i=1}^{I-1}(t_i^{(f)} + c_i^{(f)}) + t^{(s)}.$$

As we assumed that the next contact of the tagged node does not depend on the previous contacts, the number of intervals I follows a geometric distribution with parameter p_s. Furthermore, the lengths of contact and inter-contact times are independent on I, and therefore we obtain

$$E[T|IC^{(f)}] = E[t_+^{(f)}] + \frac{E[t^{(f)}]+E[c^{(f)}]}{p_s} - E[t^{(f)}] + E[t^{(s)}].$$

Using the same line of reasoning, we also obtain the expression of $E[T|C^{(f)}]$:

$$E[T|C^{(f)}] = E[c_+^{(f)}] + \left(\frac{1}{p_s}-1\right)\left(E[t^{(f)}]+E[c^{(f)}]\right) + E[t^{(s)}].$$

The final result we obtain in this Appendix is the expression of $E[L]$, i.e. the average time for the tagged seeker to encounter any node in a given subset, starting from the end of a contact. We can follow the same approach used for the previous results, and notice that L starts with a random number $I-1$ (possibly equal to 0) of inter-contact and contact times after which a failure occurs, and a final inter-contact time after which success occurs. Therefore L can be written as

$$L = \sum_{i=1}^{I-1}(t_i^{(f)} + c_i^{(f)}) + t^{(s)},$$

and it's average value is as follows:

$$E[L] = \left(\frac{1}{p_s}-1\right)(E[t^{(f)}]+E[c^{(f)}]) + E[t^{(s)}].$$

B. FIRST STAGE DERIVATIONS

In this Appendix we provide the detailed derivations of the first stage delays, i.e., Lemmas 1, 2 and Theorem 1.

LEMMA 1. *The average delay to complete the upload of*

the input parameters on provider l is

$$E\left[\beta_l\right] = \frac{E\left[\hat{\beta}_l\right]}{1 - \lambda E\left[\hat{\beta}_l\right]}.$$

where $E[\hat{\beta}_l]$ is

$$E\left[\hat{\beta}_l\right] = \frac{E\left[T_l\right] - E\left[L_l\right] + E\left[q_l\right]\left(E\left[L_l\right] + E\left[c^{(s)}\right]\right)}{\frac{M}{m} - \lambda\left(E\left[L_l\right] - E\left[T_l\right]\right)}.$$

PROOF. The key part of the analysis is modeling the service time of the queue, $\hat{\beta}_l$. Three cases must be separately considered. First of all, in general, some queued requests (and even a request being served) might be dropped. This happens if provider l is not among the first m providers on which the input parameters upload completes. The probability of this event is throughout referred to as $p_{m,l}$. We set $\hat{\beta}_l$ equal to 0 for these requests. Note that this results in understimating $\hat{\beta}_l$, as the time spent on partially served requests is neglected.

The second and third cases to be considered are the service time for requests arriving when the queue is empty or not empty, respectively. We denote these values with $\hat{\beta}_l|_{NE}$ and $\hat{\beta}_l|_E$, respectively. Let us focus again on Figure 2. In case the request arrives when the queue is empty, the initial part of $\hat{\beta}_l$ is the time required to encounter provider i starting from a random point in time (denoted as T_l). This can be derived using to the general result described in Appendix A. When, on the other hand, a request arrives when the queue is not empty, it starts being served after the input parameters of the previous request has been uploaded. By assuming that uploads always complete at the end of a contact[2], in this case the service time starts with an inter-contact time, and, again, the time until the first contact with provider i (L_l) can be computed exploiting the result of Appendix A.

Based on the above discussion, if we use the standard definition of the queue utilisation $\rho_l = \lambda E\left[\hat{\beta}_l\right]$, we can write $\hat{\beta}_l$ as

$$\hat{\beta}_l = 0 \cdot p_{m,l} + (1 - p_{m,l})\rho_l \hat{\beta}_l|_{NE} + (1 - p_{m,l})(1 - \rho_l)\hat{\beta}_l|_E. \quad (14)$$

We now derive the expressions for $\hat{\beta}_l|_{NE}$, $\hat{\beta}_l|_E$ and $p_{m,l}$. Remember we have denoted by q_l the number of contact events needed by the seeker to complete the upload of the input parameters on provider l (modelled through the Markov chain shown in Figure 3). $\hat{\beta}_l|_{NE}$ and $\hat{\beta}_l|_E$ can be computed as the sum of the time to make contact with provider l for the first time, plus a number of additional contact and inter-contact times required for completing the upload:

$$\hat{\beta}_l|_{NE} = \sum_{k=1}^{q_l}\left(L_{l,k} + c_k^{(s)}\right) \quad (15)$$

$$\hat{\beta}_l|_{NE} = T_l + \sum_{k=1}^{q_l - 1}\left(c_k^{(s)} + L_{l,k}\right) + c^{(s)} \quad (16)$$

In both cases the seeker has to meet the provider q_i times. When the queue is not empty, each encounter results in a time interval equal to $L_l + c^{(s)}$, i.e., the time required to meet provider l starting from the end of a contact, plus the contact time with provider l ($c^{(s)}$). When the queue is empty, the first encounter with provider l occurs after T_l, because the service time starts at a random point in time with respect to the underlying mobility process. Then, to complete the upload, q_l contacts are required (each lasting for $c^{(s)}$), and each contact is separated by an interval equal to L_l. To derive the average values of $\hat{\beta}_l|_{NE}$ and $\hat{\beta}_l|_E$ we assume that the r.v. q_l is independent of L_l and $c^{(s)}$, and thus obtain:

$$\begin{aligned} E\left[\hat{\beta}_l|_{NE}\right] &= E\left[q_l\right]\left(E\left[L_l\right] + E\left[c^{(s)}\right]\right) \\ E\left[\hat{\beta}_l|_E\right] &= E\left[T_l\right] - E\left[L_l\right] + E\left[q_l\right]\left(E\left[L_l\right] + E\left[c^{(s)}\right]\right) \end{aligned} \quad (17)$$

To finally derive the expression of $E\left[\hat{\beta}_l\right]$ based on Equation 14 we need to characterise $p_{l,i}$, i.e., the probability that the delay of the first stage related to provider l (β_l) is greater than the shortest m delays. By recalling that B_l represents the delay of the first stage on the i-th pipe, $p_{m,i}$ can be expressed as follows:

$$p_{m,l} = P(\beta_l = B_{m+1}) + P(\beta_{ji} = B_{m+2}) + \ldots + P(\beta_l = B_M). \quad (18)$$

The terms $P(\beta_l = B_k), k = 1, \ldots, M$ will be completely derived in the following of the analysis, as they are required for characterising other parts of the model, as well. As far as $p_{m,l}$ is concerned, we can anticipate that $p_{m,i}$ turns out to be a complex function of the average values of *all* the r.v. β_l, and therefore, finding the average values of the r.v. β_l by exploiting Equation 14 requires to solve a complex non-linear system. In general, this is possible only numerically. To provide a closed formula, we derive $p_{m,l}$ in the case in which the r.v. β_l are independent and identically distributed. In this case, it is straightforward to see that $p_{m,l}$ is equal to $\frac{M-m}{M}$. Based on this assumption, from the expression of $\hat{\beta}_l$ in Equation 14, and by using Equations 17, we can derive the average value of the service time of the queue, $E\left[\hat{\beta}_l\right]$ as:

$$E\left[\hat{\beta}_l\right] = \frac{E\left[\hat{\beta}_l|_E\right]}{\frac{M}{m} - \lambda\left(E\left[\hat{\beta}_l|_{NE}\right] - E\left[\hat{\beta}_l|_E\right]\right)} \quad (19)$$

$$= \frac{E\left[T_l\right] - E\left[L_l\right] + E\left[q_l\right]\left(E\left[L_l\right] + E\left[c^{(s)}\right]\right)}{\frac{M}{m} - \lambda\left(E\left[L_l\right] - E\left[T_l\right]\right)}. \quad (20)$$

According to the general results of M/G/1 queue, to derive the average value of β_l we should compute the second moment of $\hat{\beta}_l$. Unfortunately, this would make the analysis untractable, unless we consider a set of quite unrealistic assumptions, such as the independence of the r.v. L_l and T_l. Therefore, we approximate the analysis by considering an equivalent M/M/1 queue, in which the service time is exponential and its average value is provided by Equation 20. It immediately follows that the average delay for uploading the input parameters to provider l is:

$$E\left[\beta_l\right] = \frac{E\left[\hat{\beta}_l\right]}{1 - \lambda E\left[\hat{\beta}_l\right]}.$$

This concludes the proof. □

LEMMA 2. *The probability of the set $\{\beta_{k1}, \ldots \beta_{ki}\}$ being the i-th ordered shortest variables in the set $\{\beta_l\}_{l=1,\ldots,M}$ is*

$$P(\beta_{k1}, \ldots, \beta_{ki}) = \frac{\prod_{p=1}^{i}\phi_{kp}}{\sum_{p=1}^{M}\phi_p \sum_{\substack{p=1 \\ p\neq k1}}^{M}\phi_p \cdots \sum_{\substack{p=1 \\ p\neq k1,\ldots,k(i-1)}}^{M}\phi_p}$$

[2]Note that the latter assumption can become weak when the contact times are much longer than the time required to upload the input parameters. However, in this case the whole model can be simplified as shown in [20].

PROOF. The proof of Lemma 2 assumes that the r.v. β_l are independent and exponentially distributed with rate $\phi_l = 1/E[\beta_l]$. Let us focus on the case where $i = 2$, i.e. we want to find the probability that β_{k1} and β_{k2} are the shortest and the second shortest r.v. in the set. We can write $P(\beta_{k1}, \beta_{k2})$ as

$$\begin{aligned} P(\beta_{k1}, \beta_{k2}) &= P(\beta_{k1}) \\ &= P(\beta_{k2}\text{is the second-shortest}|\beta_{k1}\text{is the shortest}). \end{aligned}$$

As we assumed that the r.v. β_l are exponential and independent, the probability that β_{k1} is the shortest r.v. in the set $\{\beta\}_l$ is $\frac{\phi_{ki}}{\sum_{p=1}^{M}\phi_p}$. The probability that β_{k2} is the second-shortest given β_{k1} is the shortest, is the probability that β_{k2} is the shortest r.v. in the set $\{\beta_l\} \setminus \{\beta_{k1}\}$, i.e. $\frac{\phi_{k2}}{\sum_{\substack{p=1 \\ p\neq k1}}^{M}\phi_p}$. Generalising this line of reasoning, we obtain the expression of $P(\beta_{k1}, \ldots, \beta_{ki})$ shown in the Lemma. □

THEOREM 5. *The time required by a seeker to complete the upload of the input parameters on provider $i+1$ ($i = 0, \ldots, m-1$) is distributed as follows:*

$$B_{i+1} \sim \sum_{k1=1}^{M} \cdots \sum_{\substack{ki=1 \\ ki\neq k1,\ldots,k(i-1)}}^{M} P(\beta_{k1}, \ldots, \beta_{ki}) \exp\left(\sum_{\substack{p=1 \\ i\neq k1,\ldots,ki}}^{M} \phi_p\right)$$

and its average value is:

$$E\left[B_{i+1}\right] = \sum_{k1=1}^{M} \cdots \sum_{\substack{ki=1 \\ ki\neq ki,\ldots,k(i-1)}}^{M} \frac{P(\beta_{k1}, \ldots, \beta_{ki})}{\sum_{\substack{p=1 \\ p\neq k1,\ldots,ki}}^{M}\phi_p}.$$

PROOF. Let us focus first on the distribution of the first two stages, i.e. B_1 and B_2. B_1 is clearly the minimum of the r.v. $\{\beta_l\}$, and is thus exponential with rate $\sum_{p=1}^{M}\phi_p$.

The distribution of B_2 can be derived according to the following line of reasoning. Assume that B_1 is equal to β_{k1} for some $k1$. Then, B_2 is distributed as the minimum of the r.v. in the set $\{\beta_l\} \setminus \{\beta_{k1}\}$, i.e. it is exponential with rate $\sum_{\substack{p=1 \\ i\neq k1}}^{M}\phi_p$. By applying the law of total probability and recalling the result of Lemma 2, we obtain that B_2 is hyperexponential, and is distributed as follows:

$$B_2 \sim \sum_{k1=1}^{M} \frac{\phi_{k1}}{\sum_{p=1}^{M}\phi_p} \exp\left(\sum_{\substack{p=1 \\ p\neq k1}}^{M} \phi_p\right)$$

where $\exp(x)$ denotes an exponential distribution with rate x. The above line of reasoning can be extended to all the r.v. B_i. Specifically, for a given variable B_{i+1} we can consider the generic set of r.v. shorter than B_{i+1}, denoted as $\beta_{j,k1}, \beta_{j,k2}, \ldots, \beta_{j,ki}$. Note that we assume that this set is ordered. Then, $B_{j,i+1}$ is the minimum in the set $\{\beta_l\} \setminus \{\beta_{k1}, \beta_{k2}, \ldots, \beta_{ki}\}$, it is exponential with parameter $\sum_{\substack{p=1 \\ p\neq k1,\ldots,ki}}^{M}\phi_p$. The results of the Theorem follow immediately by applying the law of total probability, and exploiting the result in Lemma 2. □

C. SECOND STAGE DERIVATIONS

In this Appendix we provide the detailed derivations related to the second stages, i.e. we prove Lemmas 3 and 4.

LEMMA 3. *The average delay of the second stage of pipe i, assuming provider l corresponds to it is*

$$E[D_i|l] = \frac{E[X_l^2] + 2E[X_l]}{2\mu_l E[X_l](1-\rho_l)},$$

where X_l is the size of the batches arriving at provider l. Furthermore, the utilisation of the providers is

$$\rho_l = \frac{\lambda_{X,l}E[X_l]}{\mu_l} = \frac{E[X_l]}{\mu_l(E[L_l^{(P)}] + E[c^{(s)}])},$$

where $\lambda_{X,l}$ is the rate of batch arrivals at provider l.

PROOF. A new request is ready at provider l as soon as its input parameters are uploaded by the respective seeker. Therefore, in general, whenever provider l encounters a seeker, a batch of new request becomes available for being executed. Assuming the batch size X is iid, we can model the second stage as an $\mathrm{M}^{[X]}/\mathrm{M}/1$ batch arrival system.

We first need to derive the rate of batch arrivals, $\lambda_{X,l}$. The inter-arrival time between batches is the time interval between the end of a contact between provider l and a seeker, and the end of the next contact with any other seeker. Recalling the result of Section 4.1, we can say that the average time interval from the end of a contact with a seeker and the point in time when the next seeker is encountered is equal to $EL_l^{(P)}$, P denoting the whole set of seekers. The average time between batches is therefore $EL_l^{(P)} + Ec$. As the inter-contact time between any two nodes is assumed to be exponential, the inter-contact time between the tagged provider and any seeker is the minimum over a set of exponential iid random variables, and it thus exponential too. Therefore, the rate of batch arrivals is $\lambda_{X,l} = \frac{1}{EL_l^{(P)}+Ec}$.

The expected delay of the second stage is the average delay of the $\mathrm{M}^{[X]}/\mathrm{M}/1$ system. If $E[W_i|l]$ denotes its expected waiting time, $E[D_i|l]$ is then $E[W_i|l] + 1/\mu_l$, and can be computed according to Equation 4.13(a) (page 47 of [23]). Specifically, we obtain the formula reported in the Lemma:

$$\begin{aligned} E[D_i|l] &= E[W_i|l] + \frac{1}{\mu_l} \\ &= \frac{\lambda_{X,l}E[X_l]\frac{2}{\mu_l^2}}{2(1-\rho_l)} + \frac{E[X_l^2]\frac{1}{\mu_l}}{2E[X_l](1-\rho_l)} + \frac{1}{\mu_l} \\ &= \frac{E[X_l^2] + 2E[X_l]}{2\mu_l E[X_l](1-\rho_l)}. \end{aligned}$$

Finally, according to [23], the utilisation of provider l (ρ_l) is as follows:

$$\rho = \lambda_{X,l}E[X_l]\frac{1}{\mu_l} = \frac{E[X_l]}{\mu_l(E[L_l^{(P)}] + E[c^{(s)}])}.$$

This concludes the proof. □

LEMMA 4. *The probability of provider l corresponding to pipe $i+1$ ($i = 0, \ldots, m-1$) is*

$$P(B_{i+1} = \beta_l) = \sum_{\substack{k1=1 \\ k1\neq l}}^{M} \cdots \sum_{\substack{ki=1 \\ ki\neq k1,\ldots,k(i-1),l}}^{M} \frac{P(\beta_{k1}, \ldots, \beta_{ki})\phi_l}{\sum_{\substack{s=1 \\ s\neq k1,\ldots,ki}}^{M}\phi_s}.$$

PROOF. This probability actually corresponds to the probability that the r.v. β_l takes the $(i+1)$-th shortest value in the set $\{\beta_l\}$. If we condition this probability to the set of r.v. taking the shortest i values, i.e., we assume that $B_{j1} =$

$\beta_{k1}, B_2 = \beta_{k2}, \ldots, B_{i)} = \beta_{ki}$ for a given set $\beta_{k1}, \ldots, \beta_{ki}$ such that β_l is not in this set, then the conditional probability that β_l is equal to B_i is the probability that β_l takes the minimum value in the set $\{\beta_l\} \setminus \{\beta_{k1}, \ldots, \beta_{ki}\}$, which is $\phi_l / \sum_{\substack{p=1 \\ p \neq k1,\ldots,ki}}^{M} \phi_p$. Furthermore, the probability that of the event $\{B_1 = \beta_{k1}, B_2 = \beta_{k2}, \ldots, B_i = \beta_{ki}\}$ is provided by Equation 5. The expression of $P(B_{i+1} = \beta_l)$ shown in the lemma follows immediately by applying the law of total probability. □

D. THIRD STAGE DERIVATIONS

In this Appendix we provide the detailed derivations related to the delay of the third stages, i.e., we prove Lemma 5.

LEMMA 5. *The average delay of the third stage of pipe i conditioned to the fact that provider l corresponds to the pipe is*

$$E\left[\theta_i|l\right] = \frac{E\left[\hat{\theta}_i|l\right]}{1 - \frac{\lambda_i'}{k} E\left[\hat{\theta}_i|l\right]},$$

where $E\left[\hat{\theta}_i|l\right]$ is:

$$E\left[\hat{\theta}_i|l\right] = \frac{E\left[T_l\right] - E\left[L_l\right] + E\left[q_l\right]\left(E\left[L_l\right] + E\left[c^{(s)}\right]\right)}{m - \frac{\lambda_i'}{k}\left(E\left[L_l\right] - E\left[T_l\right]\right)}.$$

PROOF. The analysis of the delay of the third stage is similar to that related to the first stage. The line of reasoning is as follows. Assume we now the provider that corresponds to pipe i. When the service execution ends, the provider has to download the output results to the seeker. In general, other results might still be waiting for download to the same seeker. Therefore, we can model the process of results download with an M/G/1 queue, in which the service time is the time required to download the output parameters to the seeker, from the point in time when the request starts being served (i.e., all previous requests have completed). As in the case of the M/G/1 queue used to model the first stage, we have to distinguish three cases to compute the service time: i) the download does not occur because the seeker has already got the results from another provider; ii) the queue is empty when the results are ready to be downloaded, and iii) the queue is not empty when the results are ready to be downloaded. Recall that, as a matter of notation, we denote with θ_i the M/G/1 queue delay, with $\hat{\theta}_i$ its service time, and with $\theta_i|_l$ and $\hat{\theta}_i|_l$ the same figures conditioned to the fact that the pipe corresponds to provider l. The derivation of $\theta_i|_k$ is quite similar to that of β_i carried out in Appendix B.The service time $\hat{\theta}_i|_l$ can be computed as in Equation 14, by replacing all figures related to β with the corresponding figures related to θ, with slight additional modifications. The expressions of $\hat{\theta}_i|_{l,NE}$ and $\hat{\theta}_i|_{l,E}$ can be derived from Equations 16 by considering that T_l and L_l now describe the time required for the provider l to meet with the tagged seeker, and q_l is the number of contacts required to complete the download. The term ρ_l becomes $\rho_l = \frac{\lambda_i'}{k} E\left[\theta_i|_l\right]$, as the offered load of the M/G/1 queue is the offered load of the third stage, divided by the number of seekers generating that load (k). The term $p_{m,l}$ needs some more elaboration. The main difference is that the result downloads for a given request does not start at the same time (while the parameters upload do start at the same time). Therefore $p_{m,l}$, i.e., the probability that a results download is cancelled, should be evaluated as follows. $p_{m,l}$ can be seen as the probability that the i-th pipe, provided it corresponds to provider l, is not the first one to complete. By recalling the general notation in Section 4, this is actually $P(R_i|_l > \min_{s \neq i} R_s|_{\sim l})$, where R_i is the delay of the pipe i, and the notation $R_s|_{\sim l}$ corresponds to the delay of the s-th pipe when all providers *but* l may correspond to that pipe. As for the case of the first stage, the only case in which a simple closed form for $p_{m,l}$ can be provided is when the r.v. R_i are iid. In that case, it is straightforward to obtain $p_{m,l} = \frac{m-1}{m}$. Finally, the closed form expression for the average value of $\theta_i|_l$ can be immediately obtained from Equation 3 after applying the replacements discussed above, obtaining the results shown in the Lemma. □

E. DERIVATIONS FOR THE OPTIMAL REPLICATION

In this section we provide the detailed derivations of the expected service execution delay $E[R(m)]$ as per Theorem 4.

THEOREM 4. *The average time for the seeker to avail of service S_j is*

$$E[R(m)] = \frac{(E[D] + E[\theta])^2}{m(E[D] + E[\theta]) - E[\beta] \ln \frac{Q-1}{Q-m-1}},$$

where Q is equal to $\frac{E[\beta]}{ED+E\theta} + M + 1$.

PROOF. Having assumed that the r.v. R_i are independent and exponentially distributed with rate γ_i, it follows that $R(m)$ is exponential with rate $\Gamma(m) = \sum_{i=1}^{m} \gamma_i = \sum_{i=1}^{m} 1/E[R_i]$. Recalling the simplifications described in Section 4.5, $E[R_i]$ can be written as

$$E[R_i] = \frac{E[\beta]}{M - i + 1} + E[D] + E[\theta].$$

After simple algebraic manipulations, the rate $\Gamma(m)$ can be written as follows:

$$\begin{aligned} \Gamma(m) &= \sum_{i=1}^{m} \left(\frac{1}{E[D] + E[\theta]} - \frac{E[\beta]}{(E[D] + E[\theta])^2} \frac{1}{Q - i} \right) = \\ &= \frac{m}{E[D] + E[\theta]} - \frac{E[\beta]}{(E[D] + E[\theta])^2} \sum_{i=1}^{m} \frac{1}{Q - i}, \end{aligned}$$

where $Q = \frac{E[\beta]}{E[D]+E[\theta]} + M + 1$. Approximating Q with the closest integer number, the factor $\sum_{i=1}^{m} \frac{1}{Q-i}$ can be approximated using the sum of the harmonic series, as $\ln \frac{Q-1}{Q-m-1}$. Therefore, $\Gamma(m)$ becomes

$$\Gamma(m) = \frac{m}{E[D] + E[\theta]} - \frac{E[\beta]}{(E[D] + E[\theta])^2} \ln \frac{Q-1}{Q-m-1},$$

and $ER(m)$ immediately becomes as shown in the Theorem. □