Services Selection and Composition in Opportunistic Networks

TR-12-10, Dipartimento di Informatica

Marco Conti*, Emanuel Marzini†, Davide Mascitti†, Andrea Passarella*, Laura Ricci†,
* IIT-CNR, Via G. Moruzzi 1 56124, Pisa, Italy
† Department of Computer Science, University of Pisa, Italy
email: {marco.conti, andrea.passarella}@iit.cnr.it, {ricci}@di.unipi.it, {mascitti, marzini}@cli.di.unipi.it

***Abstract*—Opportunistic networking has recently evolved towards the opportunistic computing paradigm which enables service selection and composition in mobile environments. The introduction of services composition introduces new challenges such as defining strategies to detect the best service composition satisfying the user request. Service composition has to take into account nodes heterogeneity, their different mobility patterns and the services published by each node. This requires the exchange of information on the status of the network when pair of nodes comes into direct contact. This information may regard the characteristics of the two nodes or of other nodes they have previously encountered. We propose a system exploiting such information in order to offer the best alternative to a service request. We propose an analytical model which has been validated through a set of simulations.**

I. Introduction

This paper proposes a system for the selection and composition of services in self-organizing mobile networks .

The increasing popularity of mobile devices such as smart phones and tablet computers, characterized by high computing power and different interfaces for communication with other devices opens new applicative scenarios.

For instance, the sharing of resources and services in a mobile environment is a current research challenge. The basic idea is the definition of a system that allows the users to take advantage of the resources and services that others user share without necessarily using only infrastructure such as the cellular one, by using instead direct connections between mobile devices.

The mobility of devices that use of radio interfaces for communications make it difficult to create and maintain a stable network topology that is usable by classical message forwarding algorithms.

A pervasive network of mobile devices exploiting users mobility to provide services also has the advantage of being able to use the social relationships among users to predict the evolution of the network topology. This is due to the nature of human mobility which follows the social relationship established between groups of people.

In order to realize such system the problem of the connection instability has to be considered. Moreover, given the great heterogeneity of the devices, it is important to define a support able to manage the selection and composition of resources and services. This layer is provides to users as a software layer on their devices.

The scenario we consider for this type of applications is that of opportunistic networks. These are a particular type of self-organizing mobile networks where each user that is part of the network exploits its mobility and that of others for transferring messages, disseminating data and accessing/sharing resources [1].

Given the great diversity of resources contained in modern mobile devices, the opportunistic paradigm has recently been extended to create systems for the mutual exploitation of the resources of the devices owned by users of a network, by defining the opportunistic computing paradigm [2].

Opportunistic computing is not limited to consider the mechanisms for transferring data in the network, but, by exploiting the analysis of social dynamics, identifies and decides which nodes of the network to contact to request resources/services to be used to satisfy the user request. Furthermore, the resources available in the network can be composed to obtain complex services.

Given the heterogeneity of resources, these are published by devices such as high-level services, so that they can be used by any device and their composition performed by automatic tools is made possible.

This paradigm can let to make distributed computing overcoming the problem of the intermittent connectivity of mobile devices.

The system that we propose in this paper follows the paradigm of opportunistic computing to build a support for mobile devices that allows users to share and use resources available on the nodes. This support layer is responsible for sharing resources on the devices in the form of services and takes dynamically shared resource information from other participants in the network. (see Section3)

With this knowledge, our system can process service requests generated by the user or by the applications that reside on the device. Then it proceeds to evaluate the possible alternatives which can be exploited to resolve these requests and choose the most convenient, taking into account both the mobility of the nodes and the heterogeneity of the participating devices.

To find the best alternative we propose a system that can be mathematically modelled. We model both the devices mobility and their hardware capabilities in order to derive statistical measures useful for comparing the alternatives. This comparison is used by policies that determine how requests are resolved. These policies are also influenced by the evolution of the network. The mobility is monitored by detecting events of connection and disconnection between nodes.

The knowledge gathered from each node is gained through exchanges of information during the connections to the nodes which are in contact range of the node. We do not implement a system of data dissemination, but each node exploits only the knowledge obtained from nodes with whom the node has entered directly in contact.

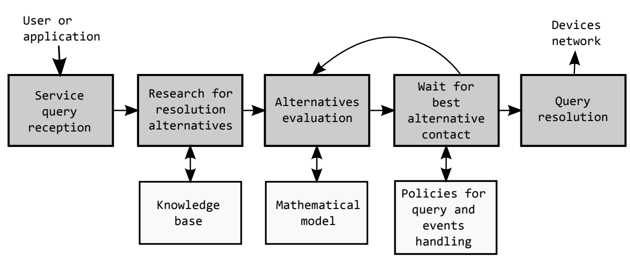

Figure 1. Procedure for resolution of service requests

Services may be composed to derive additional alternatives for the resolution of a request. Service composition is based on an abstract representation of services based on the type of the parameters needed for their execution and on the type of the returned results. The representation is a graph showing the structure of the composition and which can be weighted by a set of statistal values useful for performing the comparison of the available alternatives.

A Knowledge base of the network is constructed through communications and autonomous detection of events. It consists of all the services provided by the users, mobility statistics such as frequency of disconnections and connections with the individual devices, their computing capacity and the load of requests that are submitted.

Through the knowledge built, the model provides estimates of the completion time of a request so that the best alternative can be chosen.

The Figure 1 shows the schema of of the proposed system.

The paper presents a set of experimental results obtained through a set of simulations that characterize the behaviour of the system. The simulator we use, TheOne [3] has been proposed for the simulation of opportunistic environments, and allows the simulation of different mobility models.We have made substantial modifications to the simulator, making it more effective for the management of service composition and more efficient in terms of performance. The simulations are made varying a set of parameters, and highlight the efficacy of the approach we propose also when the system has to manage a large amount of resources.

The originality of this work is to create a system where algorithms and mathematical models together provide a solution to the complex problem of selection and service composition in opportunistic networks. The implementation of this system poses a basis for further research towards the refinement of mathematical analysis and knowledge management.

II. Related work

The main goal of opportunistic computing is to exploit the services offered by different kind of devices, i.e. mobile phones, laptop, tablets, which are currently widespread among the population. This section describes the opportunistic solutions for sharing services in mobile networks recently proposed, starting from a brief discussion of opportunistic networks, and then presenting more recent research on opportunistic computing.

The concept of opportunistic networks is an evolution of the classic paradigm of ad-hoc mobile networks (MANETs) [4]. Opportunistic networks make no assumption about the existence of complete paths between a pair of communicating nodes, however, any pair of nodes may exchange messages at a cost of an higher communication delay, according to the *store, carry and forward* communication model. According to this model the intermediate nodes between sender and receiver store messages to be sent when there is an opportunity for a waiting message to progress toward the receiver.

The definition of algorithms for opportunistic networks requires the investigation of mobility models for the users which take into account the concept of community. The definition of community is based on the social habits of each user regarding his movements and interactions with other ones. Several formal mobility models have been proposed, so that they can be exploited to simulate real-world user behaviour scenarios. Among these ones, we recall the random way point model [5] [6] [7], the temporal model [8] [9], spatial dependencies [10] [11], spatial restrictions [12] [13] and social models [14] [15] [16].

The goal of opportunistic computing is the definition of a system allowing the users to opportunistically compose resources present in the network. The sharing of resources in a heterogeneous mobile network can be used to compose functionalities not available in a single node of the network thus providing a much more rich functionality set. Such a vision requires new solutions for orchestration and management of resources on different devices [17]. Opportunistic computing assumes dynamic network topology, as opportunistic networking. In this sense, many results from the research area of opportunistic networking may be exploited. The peculiar challenge of opportunistic computing is to gain knowledge about the availability of services present in the network, both as regards their allocation on the devices, the delay to access them and their efficiency.

Some recent research proposal take into accounts these aspects. [18] describes a system allowing service discovery and composition in networks with stable connectivity. The proposed system includes a mechanism for modelling services representing their semantics through the use of ontologies . They define a taxonomy of services through a list of concepts describing them [19].
The service composition exploits a mechanism based on graph theory that allows to collect the knowledge of the services offered and to represent this knowledge to evaluate the best alternative to use.

[20] analyses the issue of single service provision. The main aim is to improve the efficiency of service provision by replicating requests to a set of different suppliers. The proper number of requests is computed by considering information from the mobility context and from the node load. The optimal number of requests is computed through an analytical model that minimizes the expected value of the request resolution time, defined as the time required to receive the first response from suppliers.

In this paper we propose a system extending previous techniques and based on an analytical model whose aim is to minimize the time for the provisioning of the service. The proposed system is based on an analytical model. Each node collects information about the status of the network and exploits this information to instantiate a set of parameters of the model. A set of metrics computed by applying the model is then exploited by a distributed algorithm to choose the best service composition satisfying the user request.

III. System blueprint

The system we describe in this report has the goal of enabling a user or an application belonging to an opportunistic network to access local and remote services, compose them and automatically choose among the available alternatives the one with the highest probability to provide the best solution.

The system exploits an evaluation model which does not use the knowledge of the global state of the network, but provides its evaluation only through the knowledge attainable locally and by exchanges between neighbour nodes

We propose a system defining a set of policies to evaluate different alternatives for the resolution of the users' request of services. The policies require a distributed knowledge base of the system which are the basis for the evaluation of the service composition alternatives.

The system must also provide to users a layer capable to call for the execution of services chosen by the aforementioned policies.

In Figure 2 we show the architecture of a node, which includes the following components:

- An algorithm for handling and resolving queries generated by the users.
- An evaluation model which defines metrics for the analysis of the choices available to resolve queries.
- An algorithm for handling exchange of information within the opportunistic network and handling of asyn-

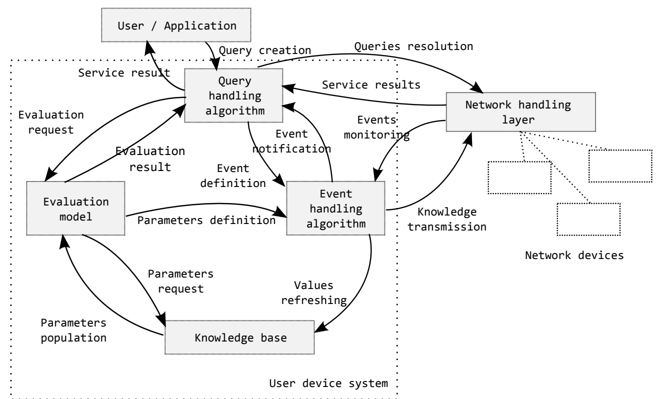

Figure 2. System Components

chronous events that may affect the handling of service queries.

A. Query

This component interacts with the user of the device or with applications that take part to the opportunistic network. It receives service queries and choose the best alternative available according to the evaluation model defined in the following sections.

The role of this component can be summarized as the application of a set of policies to service queries by evaluating the available alternatives to optimize the query resolution by some metrics, taking into account the dynamicity of the network.

Given the interest of the user, there may be different policies to picked, each one giving a specific alternatives classification.

An important characteristics of our approach is that the evaluation of the alternatives includes nodes of the network which are not connected to the requester when the query is submitted. This is possible as long as the node has enough knowledge of the network to estimate the time necessary to establish a new connection.

This requires to consider that the topology of the network may completely change between the query request and the new connection with the node and this may alter the classification and require a new evaluation of the alternatives.

B. Evaluation model

The model has the role of identifying the suitable alternatives for the resolution of a query and evaluate them, using the node knowledge of the network.

This process can be seen as the application of a set of parameters to a function. These parameters are taken from the knowledge accumulated by the node.

The result of this evaluation should be a set of values that enable ordering the alternatives by the requested metric.

The evaluation model gives different results as the network topology evolves as reflected on the local knowledge. The model doesn't require an interaction with the user or with the opportunistic network, but only the knowledge of the network collected by the event handling algorithm.

C. Event handling algorithm

The events handling algorithm must keep monitoring the network and the local system to catch events that may affect either the query handling algorithm or the local knowledge.

The kind of events to handle depends on the parameters required by the evaluation and on the policies in the query handling algorithm.

This algorithm should trigger, when needed, both data communications on the network and the refreshing of the local knowledge base.

These three components are dependent from each other, not only thanks to the interactions but also for the necessity of building a common representation of the opportunistic network elements to be included in the local knowledge base.

IV. System Specification

The system must forecast the time needed to complete the query resolution both for atomic services or for composite ones.

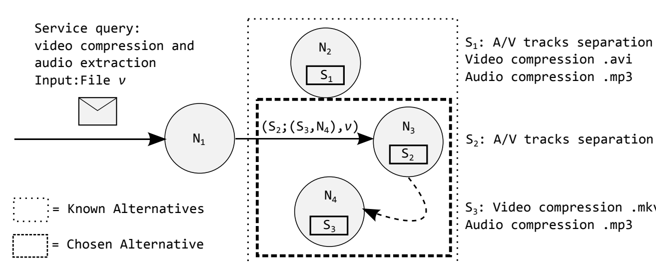

Figure 3. Service composition example

Figure 3 shows a possible use case scenario of our system. A user (in the following, the *seeker*, marked by the node N_1) asks for a service which consists in the transformation of a raw video file into a compressed video file and a separate audio file. The seeker knows three nodes of the network N_2, N_3, N_4 which have published, respectively, the services S_1, S_2, S_3. N_1 recognizes as viable alternatives the service S_1 which offers the required transformations and the sequential composition of the services S_2, S_3. Note that the first step is the check that the sequential composition of S_2 and S_3 is suitable. This can be achieved comparing the output type of S_2 with the type of the input of S_3.

To give an estimation of the time required for the service execution, the system must estimate each step in the resolution process such as the time required to contact the provider, the input parameters transfer time, the time for the request to be processed by the provider and, finally, the time to transfer the results to the seeker or to the next node of the composition.

A. The Service Model

The estimate of the service provisioning is given by a set of metrics used to compare the alternatives.

The network state can be built from statistical measures, which are exploited by the probabilistic model to estimate the resolution time of the query.
We define a random variable modelling the resolution time of the query, and we exploit the following metrics to evaluate the service provisioning alternatives.

- the expected value of the variable defines an estimate of the completion time of a service.
- the expected value and the variance of the variable may be exploited to approximate the probability distribution of the services execution time, enabling to detect the alternative with the highest probability to complete the service before a given time limit.

B. Execution of a Service: Analysis

The execution of a service or a composition of services in an opportunistic network requires the following phases:

- Waiting time to contact the service provider. This value is determined by the inter contact time between the nodes which depends on the relative mobility of the couple *seeker-provider*.
- Data transfer time. Input data for the service must be transferred by the seeker to the provider and back or between nodes providing basic services in the case of service composition. Each data transfer between two nodes of the network may be affected by the connection disruptions due to the mobility of the nodes. This implies that both the duration of contacts and the inter contact periods affect the number and duration of the connections required to complete the transfer.
- Queue waiting time on the provider. This parameter depends both on the frequency of arrival of the queries to the provider and on the time necessary to process them.
- Service execution time on the provider. It depends on the computational capabilities of the provider and on the service type as well.

We characterize each phase by a random variable and characterize that variable by its expected value and variance.

Table I
SYSTEM ELEMENTS

N	Nodes Set
S	Services Set
$N_i \in N$	i-th node of the network
$S_i \in S$	i-th service
I_i	Input type of the i-th service
O_i	Output type of the i-th service
Q_i	i-th service query

Table II
RANDOM VARIABLES

$R_{h,i,j}$	Completion time of service i on provider j, for a service required by node h
$W_{h,j}$	Time h spent by h to establish a contact with j
$B_{h,j}(k)$	Time required to upload data of size k from node h to node j
$\theta_{h,j}(k)$	Time necessary to download data of size k from node h to node j
$D_{s_{i,j}}$	Execution time of service i in provider j
D_{q_j}	Queue waiting time for a service request in provider j
G	Number of queries in a batch received by a provider
$T_{C_{h,j}}$	Contact duration of node h and node j
$T_{IC_{h,j}}$	inter contact period of node h and node j

C. Assumptions

This section defines some basic assumptions we introduce to reduce the complexity of the mathematical analysis. We will exploit some notations used also for the definition of the model which are summarized in tables I,II,III.

Table III
NETWORK STATISTICS

$\delta_{h,j}$	Contact rate between node h and node j
$\delta'_{h,j}$	Intercontact rate between node h and node j
ρ_j	Average load on provider j
$\mu_{i,j}$	Average service time for service i on provider j
λ_j	Query arrival rate on provider j
$V_{h,j}$	Average throughput between nodes h and j

Property IV.1. *For each service request q generated by the seeker $n_h \in N$, it is possible to satisfy the request by a composition of services allocated on the provider nodes $n_j, ..., n_m \in N$ which have been previously contacted by n_h*

This assumption guarantees that the nodes satisfying the user query may be detected at the node where the query is generated, with the knowledge it has previously gained from the network.

Property IV.2. *Each seeker n_h and each provider n_j have the same knowledge about the service s requested by n_h e and provided by n_j.*

This property is needed to model symmetric service knowledge between seekers and providers so that the the types of input/output parameters known by seekers and providers match

Property IV.3. *The i.i.d.random variables T_C and T_{IC} modelling, respectively, the contact and inter contact times between two nodes n_h and n_j follows an exponential probability distributions with rates δ and δ'*

This assumption is based on [21], which analysus a set of human mobility traces that show an exponential distribution of contact and intercontact periods.

Property IV.4. *In each provider n_j and for each service s_i the query batch arrivals follow a Poisson distribution with average rate $\lambda_{i,j}$*

This assumption enables to model the queue waiting time as a $M^{[X]}/G/1$ queue system.

D. Modelling Mobility

We recall that exponential random variables are characterized by the memorylessness property, which implies that the computation of contact and intercontact times does not require to store the history of previous contacts.

Let T_{IC} be the random variable modelling the intercontact time and T_C that modelling the contact time, then the respective cumulative probability functions F and expected values are the following:

- $F_{T_C}(t) = P\{T_C < t\} = 1 - e^{-\delta_{h,j}t}$
- $F_{T_{IC}}(t') = P\{T_{IC} < t'\} = 1 - e^{-\delta'_{h,j}t'}$
- $E[T_C] = \frac{1}{\delta_{h,j}}$
- $E[T_{IC}] = \frac{1}{\delta'_{h,j}}$

Since a node cannot know beforehand the values of $\delta_{h,j}$ and $\delta'_{h,j}$ they are estimated by computing the average of the previously observed contact and inter contact times.

The first phase to be considered in the computation of the completion time of a service request is the period required by the node h to establish a contact with node j providing the required service/the first service of the composition.

This is modelled by the random variable $W_{h,j}$, which is equal to 0 when the two nodes are in contact when the service request is generated, otherwise is equal to the residual inter contact time, that is equal to $E[T_{IC_{h,j}}]$, due to the memorylessness property .

The expected value of $W_{h,j}$ is recomputed at each connection/disconnection of the two nodes, changing from 0 to $E[T_{IC_{h,j}}]$ when a disconnection occurs or from $E[T_{IC_{h,j}}]$ to 0 in case of a connection establishment.

The evaluation of the value of this variable is symmetrical on the two nodes, because we suppose that the nodes can monitor the connection/disconnections events at the same time so that the local knowledge of the nodes is kept consistent.

E. Queue waiting time

A provider offering a set of services receives a stream of requests form the network nodes and put them in a queue.

We consider a $M^{[X]}/G/1$ queueing model, according to the approach described in IV.4 where network nodes generate queries according to a Poisson distribution and send them grouped in batches. The requests arrive to the provider j follows a Poisson distribution with rate λ_j.

The $M^{[X]}/G/1$ queue system [22], we can extract a definition of the expected value and variance of the random variable

D_{q_j}. This is possible assuming that the first three moments of the general distribution of the service execution time D_{s_j} (with expected value d and $d^{(i)}$ its i-th moment) do exist. The same must be true for the random variable G describing the sizes of the batches. These values can be estimated by monitoring the batches arriving to the provider and the executed services.

So given the random variable G defining the number of queries in a batch, with expected value g and i-th factorial moment $g^{(i)}$, we calculate its expected value and variance by the values G_i of G.

If n is a constant for the number of values of G_i available, we obtain:

- Expected value: $g = E[G] = \frac{\sum_{i=1}^{n} G_i}{n}$
- Variance: $Var(G) = \frac{\sum_{i=1}^{n}(G_i - g)^2}{n-1}$

To complete the characterization of D_q we extract the average rate λ of the query batches arrivals to the provider and compute the average load ρ of the provider as $\lambda * g * d$. With these values we can use the formulas in [22] for the expected value and variance of D_q:

$$E[D_q] = \frac{\lambda g d^{(2)}}{2(1-\rho)} + \frac{g^{(2)} d}{2g(1-\rho)}$$

$$E[D_q^2] = \frac{\lambda g d^{(3)}}{3(1-\rho)} + \frac{g^{(2)}[\lambda d^{(2)}]^2}{2(1-\rho)^2} +$$

$$+ \frac{g^{(3)} d^2}{3g(1-\rho)} + \frac{\lambda [g^{(2)}]^2 d^3 + (1+\rho) g^{(2)} d^{(2)}}{2g(1-\rho)^2}$$

$$Var(D_q) = E[D_q^2] - E[D_q]^2$$

F. Service execution time

The time for the execution of a service i on a provider j is a measure that is influenced both by the device hardware and by the implementation of the requested service.

To characterize the random variable $D_{s_{i,j}}$ expressing the time needed to execute the service i by provider j, we only suppose that $D_{s_{i,j}}$ has an existing value and variance, so that these can be extracted from statistics on the executions of service i.

If $D_s(h)$ is the h-th of n available statistics (n constant) of the service i execution time on provider j, we have:

- Expected Value: $E[D_{s_{i,j}}] = \frac{\sum_{h=1}^{n} D_s(h)}{n}$
- Variance: $Var(D_{s_{i,j}}) = \frac{\sum_{h=1}^{n}(D_s(h) - d)}{n}$

G. Data transfer time

The estimation of the time to transfer data between two nodes requires to take into account the network dinamicity, since disconnections may occur during the transfer process, when the size of the data to be transferred is large.

If a transfer between two nodes starts at the beginning of a contact period, it may be interrupted at the end of each contact, to be resumed when a new contact is established. This means that the total time for the transfer has to be computed by considering a sequence of delays.

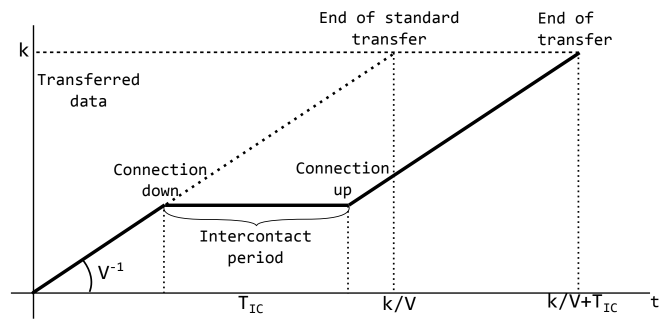

Figure 4. Data transfer interrupted by intercontact periods

We model a scenario where contact and inter contact times between two nodes are characterized by independent exponential distribution with a rate of δ and δ' respectively.

We suppose that the data throughput V is a constant > 0 computed by averaging the values of the throughput monitored during the contact periods, including that required for exchanging local knowledge.

The time to transfer data is modelled through a random variable with parameters k.the size of data, and V. Without interruptions, data transfer can be expressed as k/V (Figure 4), but inter contact periods $T_{IC,i}$ between the nodes required for the transmission $T_{C,i}$, minus one has to be considered. The number of contacts is itself modelled as a discrete random variable N that depends only by the duration of the contact periods.

The random variable for a data transfer starting at a contact period $B(k)$, using exactly n contact periods, can be expressed as follows:

$$B(k)_{|N=n} = \frac{k}{V} + \sum_{i=1}^{n} T_{IC,i}$$

where N can assume a value between 0 and ∞ with probability:

$$P\{N = n\} = P\{\sum_{i=1}^{n} T_{C,i} < \frac{k}{V} \leq \sum_{i=1}^{n+1} T_{C,i}\}$$

The value of the expected value and variance of B (demonstration shown in the Appendix) can be expressed as:

$$E[B] = (-1)*L'_B(0) = \frac{k}{V}*\left(\frac{\delta\delta' + (\delta')^2}{(\delta')^2}\right) = \frac{k}{V}*\left(1 + \frac{\delta}{\delta'}\right)$$

$$Var(B) = E[B^2] - E[B]^2 = \frac{2k\delta}{V(\delta')^2}$$

After the first step of a composition, the system cannot predict the state of the communication between two nodes, so the transfer may start immediately if there is a contact or there must be an additional waiting phase.

Let $\theta(k)$ be the random variable modelling the data transfers, where

- $\theta(k) = B(k)$ if the two nodes are in contact
- $\theta(k) = B(k) + T_{IC}$ otherwise.

Defining as, respectively, p_C and p_{IC} the probabilities that the two previous events happen, the expected value of $\theta(k)$ is (demonstration shown in the Appendix):

$$E[\theta(k)] = E[B(k)] * p_C + E[B(k) + T_{IC}] * p_{IC} =$$
$$= \frac{k}{V} * \left(1 + \frac{\delta}{\delta'}\right) + \frac{\delta}{\delta'} * \frac{1}{\delta' + \delta}$$

While the variance can be expressed as:

$$Var(\theta) = E[\theta^2] - E[\theta]^2 =$$
$$= \frac{2k\delta}{V(\delta')^2} + \frac{1}{(\delta')^2} - \frac{1}{(\delta' + \delta)^2}$$

V. Service composition

Service composition can be realized in different ways, depending on the local knowledge available at the seeker.

The simplest strategy exploits only the local knowledge of the network collected by the seeker to find out both the services to compose and the providers offering them. Once a composition is defined by the seeker, data is transferred to the first seeker, afterwards data transfer between the providers are established depending on the relative mobility of the couples involved in the composition.Two strategies to transfer data among the providers are possible. The first one does not require that the seeker knows the mobility statistics of other nodes, so that it exploits only the local knowledge and force each provider to transfer the results back to the seeker. The second solution requires to ask to the providers their mobility statistics so that the seeker can evaluate data transfers between nodes and enable them during the service resolution.

A. Characterizing compositions

An abstract representation of a service can be defined by considering only the types of the input/output parameters of the service.

The process of identification and evaluation of the services compositions is structured at different representation levels of the nodes and of the services:

- **Abstract level** At this level we consider a directed service graph showing the execution dependencies between services inside the composition. Each path connecting two vertices shows a valid sequence of service executions. A service s_j is defined as a couple (I_j, O_j) where I_j is the type of the input parameter and O_j is the type of the result, these types are codified by natural numbers and are atomic entities.

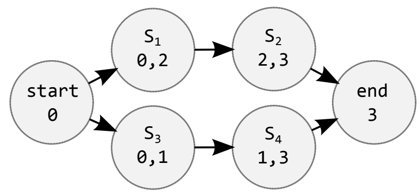

Figure 5. Abstract composition graph

Figure 5 shows a set of services $\{S_1, S_2, S_3, S_4\}$ linked by their type dependencies together with two special nodes $start$ and end representing the start and the end point of the composition to which are assigned the input, respectively the output type of the composition. Each other node is paired with a couple of sets that represents the input, respectively the output type for the corresponding service.

- **Nodes level** The abstract level represents all valid compositions yielding the required service. The evaluation of alternative composition requires to associate to the services of each possible composition the providers offering that service. To build the association (service, node),

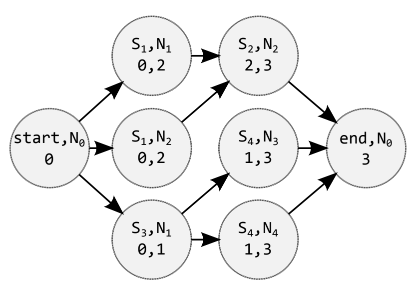

Figure 6. Nodes graph corresponding to the abstract graph in Figure 5

each node of the abstract graph paired with service S_i is replaced with a set of nodes defined by the couples (S_i, N_j) such that N_j provides service S_i, and the edges of the modified graph link the same services pairs of the abstract graph. To nodes *start* and *end* are paired with the seeker generating the query.

The edges of this graph are afterward weighted by a value for representing the cost of the invoking the services in the composition. This step will be described later.

B. Sequential compositions

The nodes level graph can be used to determine which are all the sequential compositions available on the network, just by identifying the set of paths from the *start* node and the *end* node.

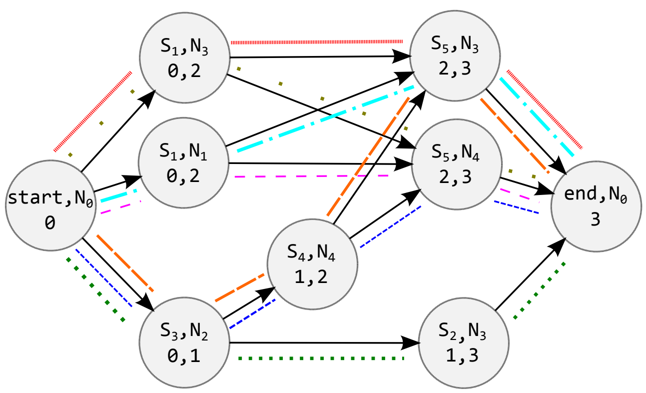

Figure 7. Sequential paths in a node graph

C. Data Exchange between providers

Consider the model where intermediate results of the composition are directly transferred between the providers. In this case the seeker needs to know the average contact and inter contact rates of any pair of nodes involved in the composition. These values may be exchanged according to an epidemic pattern between nodes coming into contact.

Let n_i and n_j be a pair of nodes establishing a contact and exchanging their knowledge of the network. n_i needs to get each average contact rates δ_{n_j,n_h} and inter contact rates δ'_{n_j,n_h} for each node n_h that is known by n_i (so that its services are also known). These values are not averaged inside n_i.

This solution may require to exchange up to n^2 elements, where n is the number of nodes in the network. The trade-off is between the data exchanged and the possibility to exploit direct data transfer between nodes, with may result in more efficient solutions, because data has not to be transferred back to the seeker after each invocation.

D. Query resolution time

In this section we present the evaluation of the alternatives in the nodes composition graph G_N .

Each node (n_i, s_i) of G_N represents the execution of a service s_i on the provider n_i, while each edge $((n_i, s_i), (n_j, s_j))$ is the necessity for provider n_j to wait the data from the result of service s_i to be sent by n_i so that the service s_j can be executed.

R_{n_i,s_i} is the random variable modelling the time to execute the sequence of service up to the end of the execution of service s_i inside the provider n_i, where (n_i, s_i) is a node of G_N.

As we did in the previous sections, W_{n_h,n_i} is the time to wait for the first contact between n_h (seeker) and n_j (first provider), and $B_{n_h,n_i}(k)$ is the time to transfer data of size k from node n_h to node n_i when the contact is established, $\theta_{n_h,n_i}(k)$ is the time to transfer data of size k without knowing whether a contact between the two nodes has been established, finally $D_{q_{n_i}}$ e $D_{s_{n_i,s_i}}$ are respectively the queue waiting time for a query sent to n_i and the time needed for the execution of service s_i by node n_i .

We consider two different scenarios to transfer data between a pair of different nodes, in the first case the seeker acts as an intermediary for each data transfer, in the second case data is directly transferred. The random variables $\overline{\theta}_{n_j,s_j,n_i}(k)$ is defined as follows:

- $\overline{\theta}_{n_h,s_h,n_i}(k) = W_{n_h,n_i} + B_{n_h,n_i}(k)$ if $s_h = start$
- $\overline{\theta}_{n_h,s_h,n_i}(k) = \theta_{n_h,n_i}(k)$ if $s_h \neq start$ and the seeker knows contact and inter contact time for the nodes (n_h, n_i) and if $\theta_{n_h,n_0}(k) + \theta_{n_0,n_i}(k) > \theta_{n_h,n_i}(k)$
- $\overline{\theta}_{n_h,s_h,n_i}(k) = \theta_{n_h,n_0}(k) + \theta_{n_0,n_i}(k)$ if $s_h \neq start$ and n_0 has only local knowledge, or if it uses global knowledge $\theta_{n_h,n_0}(k) + \theta_{n_0,n_i}(k) < \theta_{n_h,n_i}(k)$.

Each edge of the graph pairs with a service invocation consisting in transferring the input data, waiting in the queue of the provider, and executing of the request. This can be expressed by adding to each edge the corresponding values of $\overline{\theta}(k) + D_q + D_s$. The seeker (with service *start* in the graph) considers the value $W + B$ instead of $\overline{\theta}$.

The execution time of the entire composition can be expressed as a random variable R_{end,n_0} that can be recursively defined as the time needed to reach the end of the composition with the addition of the transfer of the results to n_0. Recursively, the time needed to reach the end of the composition can be defined as the sum of all previous steps, plus the transfer the data needed for the last step. This definition can be expanded up to the resolution of R_{start,n_0}, marking the beginning of the composition.

R_{end,n_0}can be defined as follows:

- $R_{n_j,s_j} = R_{n_i,s_i} + \overline{\theta}_{n_i,s_i,n_j}(k_i) + D_{q_{n_j}} + D_{s_{n_j,s_j}}$ if $s_j \neq start, s_j \neq end$ and (n_i, s_i) is the node preceding (n_j, s_j) in the path
- $R_{n_j,end} = R_{n_i,s_i} + \overline{\theta}_{n_i,s_i,n_j}(k_i)$
- $R_{n_j,start} = 0$

The expected value of $R_{n_0,end}$, may be computed by extracting from the recursive expression the explicit formula in terms of known distributions.

A composition path with m service invocations $s_1, ..., s_m$, requested by node n_0, through nodes $n_1, ..., n_m$ of the network, can be expressed as:

$$E[R_{n_0,end}] = E[W_{n_0,n_1}] + E[B_{n_0,n_1}(k)] +$$
$$+ \sum_{i=1}^{m-1} (E[D_{q_{n_i}}] + E[D_{s_{n_i,s_i}}] + E[\overline{\theta}_{n_i,s_i,n_{i+1}}(k_{i+1})]) +$$
$$+ E[D_{q_{n_m}}] + E[D_{s_{n_m,s_m}}] + E[\overline{\theta}_{n_m,s_m,n_0}(k_m)]$$

Figure 8 shows the correspondence between the graph

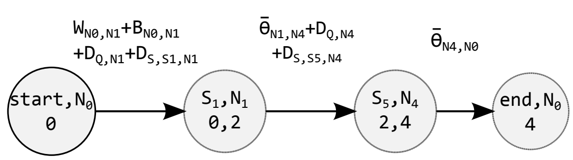

Figure 8. Values of the purple path in Figure 7

for the sequential compositions and the resolution phases of the query. The evaluation of the alternatives corresponds to the shortest path problem in a directed graph from node $(n_0, start)$ to (n_0, end) by considering the weights paired to the edges of the node graph G_N,

The expected value of the execution time may be computed by assigning a weight to each edge, and then using the shortest path algorithm to find the best alternative.

Given two adjacent vertices (n_i, s_i) and (n_j, s_j) of the node graph, if a directed edge $((n_i, s_i), (n_j, s_j))$ exists, its weight is computed as $E[R_{n_j,s_j} - R_{n_i,s_i}]$, having:

- $E[\overline{\theta}_{n_i,s_i,n_j}(k_i)] + E[D_{q_{n_j}}] + E[D_{s_{n_j,s_j}}]$ if $s_j \neq end$
- $E[\overline{\theta}_{n_i,s_i,n_j}(k_i)]$ if $s_j = end$

E. Evaluation algorithms

The minimum cost path in a general graph is generally computed by the Dijkstra algorithm *Dijkstra* [23]. It is worth noticing that working with *Directed acyclic graphs (DAG)*, more efficient solutions exploiting the characteristics of these graphs, so achieving lower complexity [24] can be used.

VI. Experimental Results

In this section we evaluate the model presented in the previous sections through a set of simulations. We first describe the simulator chosen and the modifications required to define a suitable environment for our simulation. Then, we discuss the experimental results.

A. TheOne: Opportunistic Network Environment Simulator

In order to validate our analytical model, we developed a set of simulations by exploiting TheOne (Opportunistic Network Environment) [3]. This simulator is capable of:

- generating node movements by using different mobility models
- routing messages between nodes with several routing algorithms proposed for delay tolerant networks
- visualizing mobility traces and messages in real time through the graphical interface
- importing real-world mobility traces
- collecting reports to produce simulation results

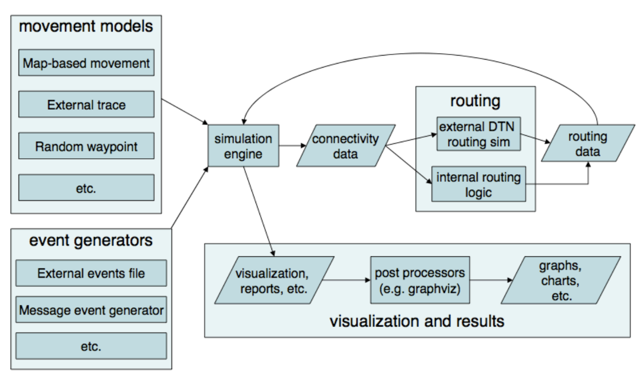

Figure 9. TheOne: general structure

Figure 9 shows an overview of TheOne. Each module of the environments defines different alternatives to define a custom simulation. To validate our model we choose RandomWayPoint mobility model, with the modification described in [25] [26]. For this reason we import external traces of nodes movement.

The messages are directly sent from the sender to the receiver and, if the sender and receiver aren't connected when the message is generated, the sender stores the message until next connection. We have chosen this routing paradigm because is the simpler mode to send a message.

B. Customizing the simulator

In this section first we present the main limitations of TheOne, then we describe the most significant modification we have performed to adapt it to our needs.

1) Limitations: TheOne is a very flexible and suitable simulator to simulate opportunistic networks but it presents some known limitations:

- The simulation may require considerable computing power and memory when a large number of nodes or of events is considered.
- All nodes are always active during the simulation. In the real world a user turns off its radio interface to increase the life of its battery. This aspect which has an impact on the contact and inter contact times, may not be easily modelled in the simulator.
- A limited amount of messages can be managed by each node. When increasing the number of nodes/messages to be sent, the time of completion grows exponentially so becoming computing intractable.
- A message is treated as an atomic entity, no fragmentation of messages in packets is defined. If the connection between the two nodes is interrupted during the message sending, the message is saved to be sent at the next connection or deleted, depending on the policy used.
- Physical and MAC layer are not implemented iin the simulator
- In the real world the transmission speed, as well as the loss of signal, is influenced by obstacles, distance and interferences between the involved nodes. The simulation does not handle these aspects.

2) Simulator extensions: To adapt the simulation to our scenario, we have introduced new features as well as changed the management of some functionalities:

- Service implementation. A service is defined by a pair of integers. A service [x, y] is an entity capable of transforming the input data of type x into the output data of type y. Each node is able to perform a number of services.
- Handling requests. The environment must include the possibility to build a list of possible paths representing service composition, according to a given policy.
- Messages. TheOne is designed to manage small amount of messages of limited size. In our simulation nodes exchange both network knowledge and the input/output of services which may have a huge size. For this reason we have modified the communication mechanism.
  Our simulation requires that the simulator manages connection falls during data transfer. In this case the data which have not been yet sent at the moment of the disconnection are saved, and then, at the beginning of next contact, communication is resumed. This behaviour requires fragmentation of messages with disconnections handling.

C. General properties

In this section we discuss the general properties of the system and the policies we have implemented.

1) Simulation settings: For each of scenario characterized by a setting of simulation parameters, 5 different run of the simulation are performed. Each run is characterized by a different simulation seed and by a different random way point trace of node movements. The number of nodes is fixed at 30 nodes that moves in a square of size $500m \times 500m$. The duration of the simulation time is 70000 s. The first 10000.0s are considered a warm up interval, where nodes move and exchange their information of services. When two nodes meet, they update their statistics (contact statistics and load status). After the warm up period each node starts creating new service requests for a period of 30000.0s. In the last 30000.0s nodes do not create any new request, but continue to process the pending service requests.

The connectivity module is customized as a blue tooth interface with 90m of range transmission and 2 Mbps of transmission speed. The amount of input/output data is selected following a uniform distribution in the interval [20KB-2MB]. We chose this interval by studying the delay introduced by the upload/download phases when compared to the other phases.

As described above the services are associated with a pair of integer, representing input/output types. We consider acyclic services, so that the value of input parameter must be less than output one. In our simulation input type are selected in [0,7] interval and output type in [1,8]. Each combination of input/output types define a service which is assigned to 25% of nodes.

We proposed two different scenarios for service requests. The first one randomly selects in the interval [0,7] for input type and [1,8] for output type of a service. The second, referred as *(case08)*, includes all requests with input 0 and output 8.

2) Policies: Our main performance index are the average completion time and the average load on a provider. We consider four different policies and compare these statistics in each of them.

- MINIMUM EXPECTED VALUE (MEV). The choice of the composition that seeker use to execute the service request is selected following the our model. The statistics collected when two node met are used to calculate shortest path of composition.
- RANDOM (RAN). For each request of service, this policy chooses a random path. For each step of selected path a random provider is chosen. This policy doesn't take into account any information about the nodes contact inter contacts.
- Always FIRST (AFIR). For each step of the composition, the policy select one of the connected provider. If all provider are disconnected, the seeker wait until the first connection with some provider.
- ATOMIC (ATOM). Service Composition is not taken into account. The system waits for a provider that can execute the entire required service

The basic idea is to compare the different policies in order to show the disadvantages of the policies that do not exploit any network knowledge. Instead the MEV policy exploits network knowledge to avoid overloading provider or considering providers with high inter contact time.

3) Request creation: TheOne allows us to configure the frequency of new requests creation and their assignment to seekers. When a new request is created, a seeker is selected randomly and the request is assigned to it.

In our simulations the number of requests may be tuned as shown in Table VI.1.

Table VI.1 Interval time for a new request

1: 20.0; 40.0 (3 - 1.5 request/min)
2: 17.0; 35.0 (3.53 - 1.71 request/min)
3: 15.0; 30.0 (4 - 2 request/min)
4: 10.0; 20.0 (5 - 3 request/min)
5: 8.0; 10.0 (7.5 - 5 request/min)
6: 5.0; 8.0 (12 - 7.5 request/min)
7: 3.0; 5.0 (20 - 12 request/min)

By increasing the number of requests we can increase the load, up to the saturation of the system. As we will see in the following sections, the effectiveness of the policy MEV with respect to the AFIRST, RANDOM and ATOMIC policy is more remarkable for higher load values.

D. Experimental results

As described in the previous sections we have considered two macro scenarios, exploiting, respectively, only the knowledge gained by the direct contacts with other nodes or not local knowledge collected by the other nodes during their contsacts.

1) Local knowledge: The results presented in Figure 10 show how MEV remain fairly stable when varying the number of requests, as opposed to the AFIR and RAN policies, which have the worst performance. These politics present a massive degradation in the case of a high frequency of service requests. ATOM policy follows the trend of MEV with average times barely greater then MEV.

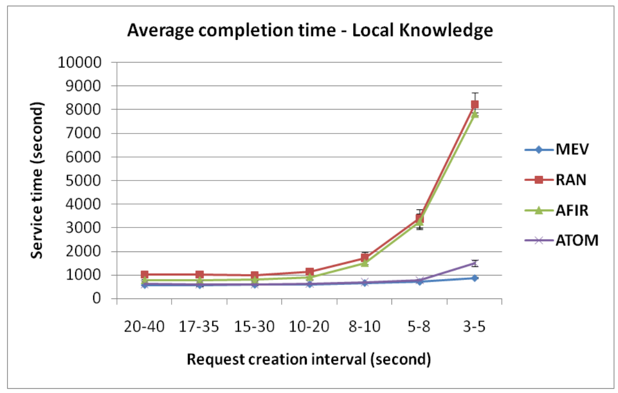

Figure 10. Average completion time

(Figure 11) shows the same results for the *case08*. Note that RAN, FIR and ATOM performance are degraded considerably when increasing the frequency of requests. The performance difference between the *case08* and the precedent scenario for the ATOM policy is due to the fact that providers that can offer the service 0-8 are selected for all the requests.

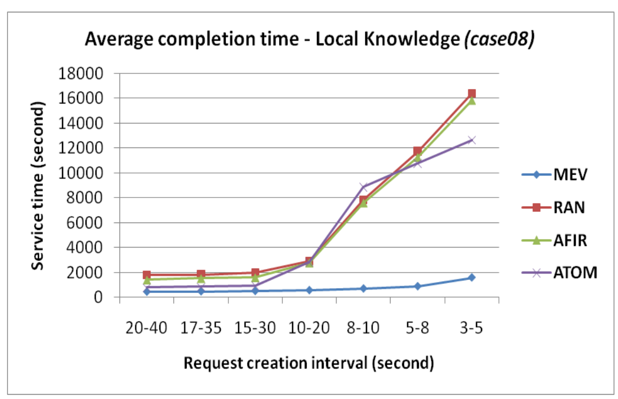

Figure 11. Average completion time (*case08*)

Figure 12 and Figure 13 show the trend of the average workload on the providers. In both cases MEV is the better policy. In the *case08*, RAN and AFIR policy exceed the saturation threshold. The trend of ATOM similar to that of MEV in Figure 13 due to the fact that ATOM generates only one service invocation for each request. Even given this advantage, MEV overperforms ATOM in the average load.

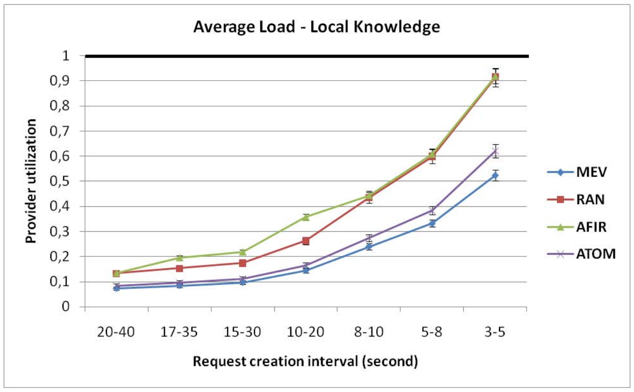

Figure 12. Average provider load

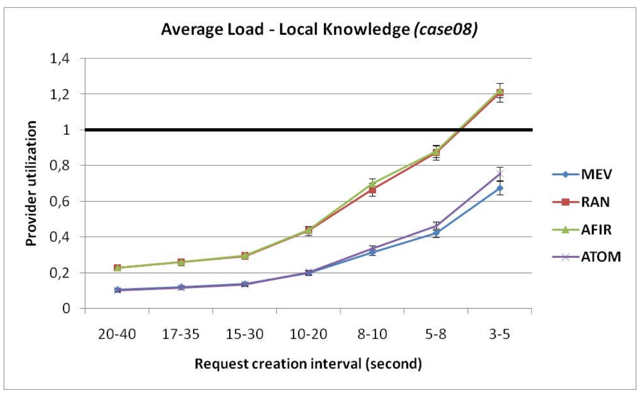

Figure 13. Average provider load (*case08*)

2) Not local knowledge: The use of not local knowledge does not change the trend of the policies. For the average completion time the distances between MEV and other policies increases, in both cases analysed (Figure 14 e Figure 15).

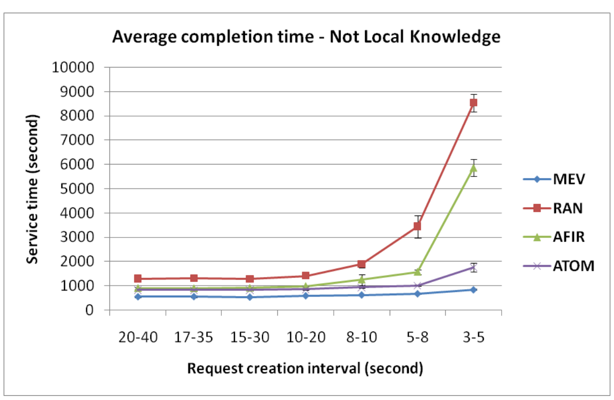

Figure 14. Average completion time

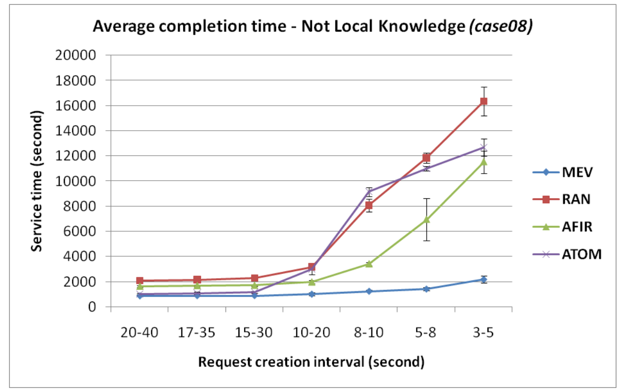

Figure 15. Average completion time (*case08*)

As far as concerns the average load (Figure 16, Figure 17), MEV never exceeds the value of 0.8, as opposed to other policies. RAN and AFIR exceed even in this case the saturation threshold; ATOM has the same behaviour of the local knowledge scenario, but also *case08* overloads the nodes offering the service 0-8 without a composition, leaving the other nodes without requests, given that only the 25% of providers have the (0;8) service.

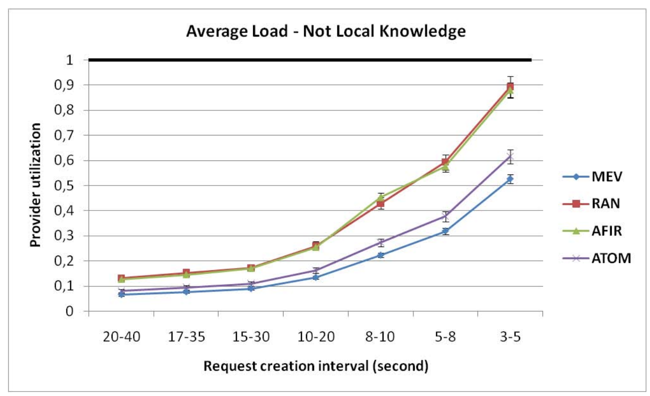

Figure 16. Average provider load

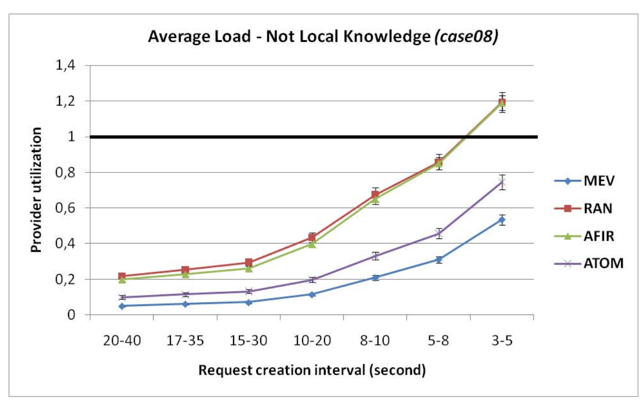

Figure 17. Average provider load (*case08*)

VII. Future direction

There are several directions to further develop the system proposed in this paper, both as regards the addition of further functionalities and for the refinement of the proposed approach. As far as regards the improvement of the mathematical model, we plan to extend it in several directions such as investigating different diffusion techniques for the network knowledge and the management of the general compositions where, for instance, service can be executed concurrently.

As far as concerns the knowledge dissemination, it would be interesting to study how to apply in our system solutions for transferring and disseminating data in the opportunistic network. Through a more extensive knowledge of the network, it is possible, for instance, to estimate if it is more convenient to satisfy a request through nodes in direct contact or by considering multi-hop scenarios.

Another area of possible development is the management of the uncertainty of the collected statistics in the mathematical model. It would be interesting also to consider the past history of the collected statistics to foresee the behaviour of the nodes.

VIII. Conclusions

In this paper we have presented a system able to satisfy user service requests in an opportunistic environment. We have defined a mathematical model to evaluate alternatives known by the device where the request is generated.

The support layer processes the available knowledge of the network collected at the request node, and composes known services to evaluate new alternatives for request resolution. To evaluate alternative service compositions, we have developed a probabilistic model which provides an approximation of the completion time of a request.

We have introduced a formal representation of the compositions through graphs. Then we have considered two types of compositions of services, which are differentiated by the knowledge that each node has about the network. The first case of composition uses only local knowledge that each node collects during contacts with other nodes. The second uses also mobility information that involve the node encountered and other nodes with which it had contacts.

Appendix

Variance and Expected values for data transfer times

In this section we show how to get the explicit formulation for the expected values of the data transfer times $B(k)$ and $\theta(k)$ we described in subsection 4.G.

As assumed before, we model a scenario where contact and intercontact times between a nodes couple are random variables (T_C and T_{IC} respectively) following identical and independent exponential distributions with average rates δ and δ'.

The data throughput V is a constant > 0, computed by averaging the values of the throughput monitored the contact periods, including those required to exchange local knowledge. Also the transfers are depending on a parameter k which is the size of the data to be transferred.

Duration of the data transfer time can be expressed as k/V plus the periods of loss of contact ($T_{IC,i}$ between the seeker and the provider).

The number of these periods are equal to the number of contacts periods $T_{C,i}$ used to do the transfer, minus one. The number of the contact periods can eventually be expressed as a discrete random variable N, which depends only on the duration of those contacts.

The random variable modeling the time to transfer data starting in a contact period $B(k)$, using exactly n contact peroids, is:

$$B(k)_{|N=n} = \frac{k}{V} + \sum_{i=1}^{n} T_{IC,i}$$

Where N may take values from 0 to ∞ with probability:

$$P\{N = n\} = P\{\sum_{i=1}^{n} T_{C,i} < \frac{k}{V} \leq \sum_{i=1}^{n+1} T_{C,i}\}$$

To associate the distribution of N to B, we can note that $\sum_{i=1}^{n} T_{IC,i}$ and $\sum_{i=1}^{n} T_{C,i}$ are Erlang random variables with average rates respectively δ' and δ and we will call them $S_{IC,n}$ and $S_{C,n}$.

Given that we are interested in getting a formulation of the expected value of $B(k)$, we can use the Laplace transform of B to ease the analisys. Also, we can formulate the expected value of N using its Z-transform $\Pi_N(z)$.

The transform of B is $L_B(s) = L_{k/V+S_{IC,N}}(s) = e^{-sk/V} L_{S_{IC,N}}(s)$ and thanks to a property of the Laplace transform of an Erlang variable which has a random variable as the number of components, we can write $e^{-sk/V} L_{S_{IC,N}}(s) = e^{-sk/V} \Pi_N(L_{T_{IC}}(s))$, successfully separating the number of intercontacts from the intercontacts distribution.

This composite function can be changed thanks to the definition of Z-transforms into $e^{-sk/V} \sum_{n=0}^{\infty} P\{N = n\} * (L_{T_{IC}}(s)^N) = e^{-sk/V} \sum_{n=0}^{\infty} P\{N = n\} * \left(\frac{\delta'}{\delta'+s}\right)^n$

The probability of $N = n$ depends on the probability that n

contact periods won't be enough to finish transfering the data, but that with another contact the transfer will end:

$$P\{N=n\} = P\{\sum_{i=1}^{n} T_{C,i} < \frac{k}{V} \leq \sum_{i=1}^{n+1} T_{C,i}\}$$

We considerate the sum of n contact periods as an Erlang with average rate δ that we call $S_{C,n}$:

$$P\{N=n\} = P\{S_{C,n} < \frac{k}{V} \leq S_{C,n} + T_{C,n+1}\} =$$

$$= P\{S_{C,n} + T_{C,n+1} \geq \frac{k}{V} \wedge S_{C,n} < \frac{k}{V}\} =$$

$$= \int_0^{k/V} P\{x + T_{C,n+1} \geq \frac{k}{V} \wedge S_{C,n} = x\} dx =$$

$$= \int_0^{k/V} P\{T_{C,n+1} \geq \frac{k}{V} - x | S_{C,n} = x\} * P\{S_{C,n} = x\} dx =$$

We substitute the probability values with the cumulative probability distribution of the intercontact times $F_{T_{IC}}$ and the density function of $S_{IC,n}$:

$$= \int_0^{k/V} (1 - F_{T_C}(\frac{k}{V} - x)) * f_{S_{C,n}}(x) dx =$$

$$= \int_0^{k/V} (1 - (1 - e^{-\delta(\frac{k}{V} - x)})) * \frac{\delta^n x^{n-1} e^{-\delta x}}{(n-1)!} dx =$$

$$= \int_0^{k/V} e^{-[\delta(\frac{k}{V} - x) + \delta x]} * \frac{\delta^n x^{n-1}}{(n-1)!} dx =$$

$$= \int_0^{k/V} e^{-\delta \frac{k}{V}} * \frac{\delta^n x^{n-1}}{(n-1)!} dx =$$

$$= e^{-\delta \frac{k}{V}} \delta^n * \int_0^{k/V} \frac{x^{n-1}}{(n-1)!} dx = e^{-\delta \frac{k}{V}} * \frac{(\frac{\delta k}{V})^n}{n!}$$

substituting this value to the formulation of the trasform of B, we have:

$$L_B(s) = e^{-sk/V} \sum_{n=0}^{\infty} e^{-\delta \frac{k}{V}} * \frac{(\frac{\delta k}{V})^n}{n!} * \left(\frac{\delta'}{\delta' + s}\right)^n =$$

$$= e^{-(s+\delta)k/V} \sum_{n=0}^{\infty} \left(\frac{\delta k \delta'}{V(\delta' + s)}\right)^n * \frac{1}{n!} =$$

Given that $\sum_{n=0}^{\infty} c^n/n! = e^c$, we have:

$$= e^{\frac{-(s+\delta)k}{V}} * e^{\frac{\delta k \delta'}{V(\delta'+s)}} = e^{\frac{-(s+\delta)k(\delta'+s) + \delta k \delta'}{V(\delta'+s)}} = e^{\frac{-(\delta+\delta'+s)ks}{V(\delta'+s)}}$$

To obtain the expected value of B, we calculate the value of the derivate function of the transform:

$$L'_B(s) = \frac{k}{V} * \left(\frac{(\delta\delta' + (\delta'+s)^2)}{(\delta'+s)^2} * e^{(-\frac{ks(\delta+\delta'+s)}{V(\delta+s)})}\right)$$

And we calculate it for $s = 0$:

$$E[B] = (-1) * L'_B(0) = \frac{k}{V} * \left(\frac{\delta\delta' + (\delta')^2}{(\delta')^2}\right) = \frac{k}{V} * \left(1 + \frac{\delta}{\delta'}\right)$$

Using the transform $L_B(s)$ it is also possible to extract the value of the variance of B, given that we only need to find the second moment $E[B^2]$ of B and substract the square of the expected value. We can obtain the second moment by calculating the second derivative $L''_B(s)$ of the laplace trasform of B and put $s = 0$.

With simple calculations we have that:

$$L''_B(s) = \frac{k(k(\delta\delta' + (\delta'+s)^2)^2 + 2\delta\delta' V(\delta'+s)}{V^2(\delta'+s)^4} * e^{\frac{-(\delta+\delta'+s)ks}{V(\delta'+s)}}$$

$$E[B^2] = L''_B(0) = \frac{k(k(\delta\delta' + (\delta')^2)^2 + 2\delta V(\delta')^2}{V^2(\delta')^4} =$$

$$= \frac{k(k(\delta\delta')^2 + k(\delta')^4 + 2k\delta(\delta')^3 + 2\delta V(\delta')^2)}{V^2(\delta')^4} =$$

$$= \frac{k(k\delta^2 + k(\delta')^2 + 2k\delta\delta' + 2\delta V)}{V^2(\delta')^2} =$$

$$Var(B) = E[B^2] - E[B]^2 = \frac{k(k\delta^2 + k(\delta')^2 + 2k\delta\delta' + 2\delta V)}{V^2(\delta')^2} -$$

$$- \frac{k^2}{V^2}\left(\frac{(\delta')^2 + 2\delta\delta' + \delta^2}{(\delta')^2}\right) = \frac{2k\delta}{V(\delta')^2}$$

To calculate the expected value of the time to transfer data not knowing whether to start from an intercontact or contact period, we use thee results from B:

- $\theta(k) = B(k)$ if there is a contact time.
- $\theta(k) = B(k) + T_{IC}$ otherwise, thanks to the memorylessness propery of exponential distributions.

If we indicate with p_C and p_{IC} the probabilities that the previous cases happen, we can calculate the expected value of $\theta(k)$ as:

$$E[\theta(k)] = E[B(k)] * p_C + E[B(k) + T_{IC}] * p_{IC} =$$

We substitute the values;

$$= \frac{k}{V}\left(1 + \frac{\delta}{\delta'}\right) * \frac{E[T_C]}{E[T_C] + E[T_{IC}]} +$$

$$+ \left(\frac{k}{V}\left(1 + \frac{\delta}{\delta'}\right) + \frac{1}{\delta'}\right) * \frac{E[T_{IC}]}{E[T_C] + E[T_{IC}]} =$$

$$\frac{k}{V}\left(1 + \frac{\delta}{\delta'}\right) * \frac{\delta'}{\delta' + \delta} + \left(\frac{k}{V}\left(1 + \frac{\delta}{\delta'}\right) + \frac{1}{\delta'}\right) * \frac{\delta}{\delta' + \delta} =$$

$$= \frac{1}{\delta' + \delta} * \left((\delta' + \delta) * \frac{k}{V} * \left(1 + \frac{\delta}{\delta'}\right) + \frac{1}{\delta'} * \delta\right) =$$

$$= \frac{k}{V} * \left(1 + \frac{\delta}{\delta'}\right) + \frac{\delta}{\delta'} * \frac{1}{\delta' + \delta}$$

We use the same approach to extract, from the definition of B, the value of the variance of $\theta(k)$:

$$Var(\theta) = E[\theta^2] - E[\theta]^2$$

- Second moment:

$$E[\theta^2] = E[B^2] * p_C + E[(B + T_{IC})^2] * p_{IC} =$$
$$= E[B^2] * p_C + E[B^2 + 2B * T_{IC} + T_{IC}^2] * p_{IC} =$$
$$= E[B^2] * (p_C + p_{IC}) + 2 * E[B] * E[T_{IC}] * p_{IC} +$$
$$+ E[T_{IC}^2] * p_{IC} =$$
$$= E[B^2] + 2 * E[B] * E[T_{IC}] * p_{IC} + E[T_{IC}^2] * p_{IC}$$

- Square of the expected value:

$$E[\theta]^2 = (E[B] * p_C + (E[B] + E[T_{IC}]) * p_{IC})^2 =$$
$$= E[B]^2 * p_C^2 + (E[B]^2 + E[T_{IC}]^2 + 2 * E[B] * E[T_{IC}]) * p_{IC}^2 +$$
$$+ 2(E[B] * p_C * (E[B] + E[T_{IC}]) * p_{IC}) =$$
$$= E[B]^2 * (p_C^2 + p_{IC}^2 + 2 p_C p_{IC}) + 2 * E[B] * E[T_{IC}] *$$
$$* (p_{IC}^2 + p_C p_{IC}) + E[T_{IC}]^2 * p_{IC}^2 =$$
$$= E[B]^2 + 2 * E[B] * E[T_{IC}] * (p_{IC}^2 + p_C p_{IC}) + E[T_{IC}]^2 * p_{IC}^2$$

With the second moment and the square of the expected value, we can show the formulation of $Var(\theta)$:

$$Var(\theta) = E[\theta^2] - E[\theta]^2 =$$
$$= E[B^2] + 2 * E[B] * E[T_{IC}] * p_{IC} + E[T_{IC}^2] * p_{IC}$$
$$- E[B]^2 - 2 * E[B] * E[T_{IC}] * (p_{IC}^2 + p_C p_{IC}) - E[T_{IC}]^2 * p_{IC}^2 =$$

We note that $E[B^2] - E[B]^2$ is equal to $Var(B)$.

$$= Var(B) + 2E[B]E[T_{IC}](p_{IC} - p_{IC}^2 - p_C * p_{IC}) +$$
$$+ E[T_{IC}^2] * p_{IC} - E[T_{IC}]^2 * p_{IC}^2 =$$
$$= Var(B) + 2E[B]E[T_{IC}](p_{IC}(1 - p_{IC} - p_C)) +$$
$$+ E[T_{IC}^2] * p_{IC} - E[T_{IC}]^2 * p_{IC}^2 =$$

Given that $(1 - p_{IC} - p_C) = 0$, we can delete $2E[B]E[T_{IC}]$.

$$= Var(B) + E[T_{IC}^2] * p_{IC} - E[T_{IC}]^2 * p_{IC}^2 =$$

We substitute the explicit values of the probabilities and the expected values:

$$= \frac{2k\delta}{V(\delta')^2} + \frac{2}{(\delta')^2} * \frac{\delta}{\delta' + \delta} - \frac{1}{(\delta')^2} * \frac{\delta^2}{(\delta' + \delta)^2}$$

$$= \frac{2k\delta}{V(\delta')^2} + \frac{2\delta\delta' + 2\delta^2 - \delta^2}{(\delta')^2(\delta' + \delta)^2} = \frac{2k\delta}{V(\delta')^2} + \frac{2\delta\delta' + \delta^2}{(\delta')^2(\delta' + \delta)^2} =$$
$$= \frac{2k\delta}{V(\delta')^2} + \frac{(\delta' + \delta)^2 - (\delta')^2}{(\delta')^2(\delta' + \delta)^2} =$$
$$= \frac{2k\delta}{V(\delta')^2} + \frac{1}{(\delta')^2} - \frac{1}{(\delta' + \delta)^2}$$

REFERENCES

[1] L. Pelusi, A. Passarella, M. Conti, *Opportunistic networking: data forwarding in disconnected mobile ad hoc networks*. Communications Magazine, IEEE, volume 44, numero 11, November 2006.

[2] M. Conti, M. Kumar, *Opportunities in opportunistic computing*. Computer, IEEE, volume 43, numero 1, January 2010.

[3] A. Keränen, J. Ott, T. Kärkkäinen, *The ONE Simulator for DTN Protocol Evaluation*. SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, 2009.

[4] L. Pelusi, A. Passarella, M. Conti, *Beyond MANETs: dissertation on opportunistic networking*. technical report, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.8900, May 2006.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, *A performance comparison of multi-hop wireless ad hoc network routing protocols* Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98), ACM, October 1998.

[6] T. Camp, J. Boleng, and V. Davies, *A Survey of Mobility Models for Ad Hoc Network Research*, Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483-502, 2002

[7] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser *An Analysis of the Optimum Node Density for Ad hoc Mobile Networks*, Proceedings of the IEEE International Conference on Communications(ICC), Helsinki, Finland, June 2001.

[8] B. Liang, Z. J. Haas, *Predictive Distance-Based Mobility Management for PCS Networks*, Proceedings of IEEE Information Communications Conference (INFOCOM 1999), Apr. 1999.

[9] C. Bettstetter, *Smooth is Better than Sharp: A Random Mobility Model for Simulation of Wireless Networks*, Proc. ACM Intern. Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), Rome, Italy, July 2001.

[10] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, *A group mobility model for ad hoc wireless networks*, Proc. ACM Intern. Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), August 1999.

[11] M. Sanchez and P. Manzoni, *A Java-Based Ad Hoc Networks Simulator*, Proceedings of the SCS Western Multiconference Web-based Simulation Track, Jan.1999.

[12] J. Tian, J. Hahner, C. Becker, I. Stepanov and K. RothermeL, *Graph-based Mobility Model for Mobile Ad Hoc Network Simulation*, Proceedings of 35th Annual Simulation Symposium, in cooperation with the IEEE Computer Society and ACM. San Diego, California. April 2002.

[13] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, *Scenario-based performance analysis of routing protocols for mobile ad-hoc networks*, International Conference on Mobile Computing and Networking (MobiCom'99), 1999, pp. 195–206.

[14] Frans Ekman, Ari Keränen, Jouni Karvo and Jörg Ott, *Working Day Movement Model*, MobilityModels'08, Hong Kong SAR, China. May 26, 2008

[15] M. Musolesi and C. Mascolo, *A Community Mobility Model for Ad Hoc NetWork Research*, Proc. of ACM REALMAN, 2006.

[16] Chiara Boldrini, Marco Conti, and Andrea Passarella, *Users Mobility Models for Opportunistic Networks: the Role of Physical Locations*, IEEE Conference on Wireless Rural and Emergency Communications (WRECOM 2007), 2007

[17] M.Conti, *From opportunistic networks to opportunistic computing*. Communications Magazine, IEEE, volume 48, numero 9, September 2010.

[18] S. Kalasapur, M. Kumar, B. A. Shirazi, *Dynamic Service Composition in Pervasive Computing*. IEEE TPDS, vol.18, num. 7, 2007.

[19] D. Bianchini, V. D. Antonellis, M. Melchiori, *An Ontology-Based Architecture for Service Discovery and Advice System*. Proceedings 16th International Workshop on Database and Expert Systems Applications, August 2005.

[20] A. Passarella, M. Kumar, M. Conti, E. Borgia. *Minimum-Delay Service Provisioning in Opportunistic Networks* Technical report, disponibile online: *http://www.iit.cnr.it/sites/default/files/IIT-TR-03-2010.pdf* February 2010.

[21] W. Gao, Q. Li, B. Zhao, G. Cao, *Multicasting in Delay Tolerant Networks: A Social Network Perspective*. INFOCOM, 2010 Proceedings IEEE, 2010

[22] H. Takagi, *Vacation and Priority Systems, Part 1*. Elsevier Science Publishers B.V., 1991

[23] P. Crescenzi, G. Gambosi, R. Grossi *Strutture di dati e algoritmi: Progettazione, analisi e visualizzazione*. Pearson-Addison Wesley 2006

[24] S. Dasgupta, C. Papadimitriou, U. Vazirani *Algorithms* McGraw-Hill 2007

[25] W. Navidi and T. Camp, *Stationary Distributions for the Random Waypoint Mobility Model*. IEEE Transactions on Mobile Computing, vol. 3, no. 1, pp. 99-108, January-March 2004

[26] W. Navidi, T. Camp, and N. Bauer, *Improving the Accuracy of Random Waypoint Simulations Through Steady-State Initialization*. Proceedings of the 15th International Conference on Modeling and Simulation (MS ’04), pp. 319-326, March 2004.