HAL
open science

Choreography-based vs Orchestration-based Service Composition in Opportunistic Networks

Fadhlallah Baklouti, Nicolas Le Sommer, Yves Mahéo

To cite this version:

Fadhlallah Baklouti, Nicolas Le Sommer, Yves Mahéo. Choreography-based vs Orchestration-based Service Composition in Opportunistic Networks. WiMob 2017 - The 13th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Oct 2017, Rome, Italy. hal-01699606

HAL Id: hal-01699606

https://hal.science/hal-01699606v1

Submitted on 2 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Choreography-based *vs* Orchestration-based Service Composition in Opportunistic Networks

Fadhlallah Baklouti, Nicolas Le Sommer and Yves Mahéo
IRISA, Université Bretagne Sud, France
{fadhlallah.baklouti, nicolas.le-sommer, yves.maheo}@univ-ubs.fr

***Abstract*—Pervasive networks formed spontaneously by wireless mobile devices capable of ad hoc communication make it possible to get services from these devices, and to compose them dynamically so as to provide people with new and more sophisticated application services. Discovering, invoking and composing services in such networks are challenging tasks, due to the numerous and unpredictable connectivity disruptions caused by mobility and the short radio range of wireless interfaces. By implementing the "store, carry and forward" principle, opportunistic networking and computing can help tolerate connectivity disruptions and support communication in such intermittently-connected networks.**

In this paper, we present a system we have designed to discover, to select, to invoke and to compose REST services dynamically in opportunistic networks. This system implements both orchestration-based and choreography-based composition strategies. In this paper, we compare these two techniques in two different scenarios: one involving nomadic people roaming in an open area, and another one involving people attending a sport event.

I. Introduction

With the proliferation of heterogeneous physical objects capable of wireless communication in our daily environment (e.g., smart-phones, smart-watches, intelligent personal assistant), the Internet of Things is becoming a reality, sparking a renewal of interest for pervasive ad hoc networks formed by such objects spontaneously. So as to be exploited by remote devices, the resources offered by these disparate objects are usually abstracted as services exhibiting a simple and platform-independent interface. By discovering and remotely invoking such services, a device can perform tasks that would otherwise require local installation of specific code, if it had to operate on its own. Service discovery and invocation can rely on opportunistic networking in order to cope with the potentially frequent and unpredictable communication disruptions that occur between network devices: following the "store-carry-and-forward principle", mobile devices store messages in a local cache and carry them when they move in order to be able to forward them opportunistically when encountering other devices.

Opportunistic services can be used in several applications and contexts, such as local social networking, disaster relief, pervasive data collection [1]. As an example, we can imagine that people in an amusement park wish to share information or impressions on the different entertainment attractions. Smartphones held by spectators, but also some other devices administered by the park could host some services that produce pictures, comments, or other data measured on the attractions (length of a waiting queue, height or speed of a ride...).

If the number of services that can be made accessible in this way to a given device can be interesting, the ability to compose those services opens up a much wider space of possibilities. Service composition is performed by chaining service interfaces, syntactically or semantically, in order to provide a new service. In the example of the amusement park, a myriad of derived services could be envisaged, from the simple composition of a picture service and a location service to help visitors to find attractions, to more sophisticated arrangements of services that collect sensor data, services that apply transformations or filter these data, or more general services that deliver pictures and text.

In the kind of opportunistic network we consider in this paper, we will assume that a service is likely to be provided by several devices (for example, several people in an attraction ride could provide, through their smartphone sensors, similar acceleration data that characterize the ride). Yet, composing services still requires a prior discovery of the available services and the selection of the most appropriate for composition. The lack of end-to-end connectivity inherent in opportunistic networks makes this composition a challenging task. Although several routing mechanisms allow the probability of service delivery and the invocation delay to remain acceptable in a given opportunistic network, service composition multiplies the constraints because fulfilling a composite service request involves the invocation of multiple services, any of these service being susceptible to be out of reach at any moment. Simple composition strategies, that mimics those proposed in Internet-like contexts are likely to induce prohibitive delays.

In this paper, we present a new REST service composition system adapted to opportunistic networks, with which we investigate orchestration-based and choreography-based composition strategies. These strategies leverage on a utility-based functions to select, among several services exhibiting the same interfaces, the most relevant services to compose. In the orchestration-based composition strategy, the utility-based function is exclusively executed by the object that initiates the composition process, while in the choreography-based one this function is called by each object that has been selected to participate in the composition process. We have developed our service composition system on top of the C3PO middleware platform [2], [3]. This platform implements several routing strategies based on the "store, carry and forward"

principle, and provides point-to-point and publish/subscribe APIs. In this paper, we compare on two different scenarios the orchestration-based and choreography-based composition strategies running a utility function that computes an average of the *multi-hop inter-service advertisement time* (MHISAT) between a service client and a service provider. MHISAT is the difference between the reception times of two successive service advertisements emitted by a service provider located at a certain number of hops.

The rest of this paper is organized as follows. Section II discusses related works addressing service composition problems in pervasive and mobile environments. Section III describes the service discovery, the service selection and the service composition processes implemented in our system. Section IV presents the evaluation results we obtained for our system on the two scenarios we considered, namely, an open area in which mobile devices move around following the Levy Walk mobility model, and a sport event that took place in the city of Vannes in France, and where attendees move along the running path. Section V summarizes our contribution and suggests several directions for future works.

II. Related work

Service composition has been widely studied in the past, mainly in infrastructure-based and stable environments [4], [5], defining orchestration-based or choreography-based strategies relying on static composition descriptions, or devising dynamic composition approaches relying on a semantic description of services and on composition ontologies. Service composition is technically performed by chaining interfaces using a syntactical or semantical matching method. The interface chaining is usually represented as a graph or described with a specific language (e.g., BPEL, OWL-S). The data output of one service is piped into the next service input while filtering content and making slight format changes.

Research works dealing with service composition in pervasive environments composed of wireless mobile devices have, for some of them, considered the impact of the users' mobility in the composition process. For instance, Seamless Service Composition (SeSCo) [6] tries to find in the vicinity of the user a new provider of a service used in a composition when this service becomes unreachable following the user's movement. Doing so, it tries to perform only a partial recomposition instead of a full one. SeSCo considers direct connections with remote service providers. In [7], Chakraborty et al. propose distributed service composition protocols for ad-hoc networks. These protocols rely on a coordinating entity (referred to as composition manager) that manages the discovery, integration and execution of a composite service. Each mobile device is likely to be a composition manager (CM). The CM selection process relies on a controlled broadcast-based scheme for soliciting information from nearby nodes, and takes into account services present in the devices, their computation and energy resources and most importantly the topology of services in their vicinity. The service composition requester elects the best possible CM from nearby devices and sends the composition request to it. By implementing a checkpoint-based fault-tolerance strategy, these distributed service composition protocols support, like SeSCo, partial service recompositions to cope with device mobility. Nevertheless, these composition protocols suppose an end-to-end connectivity between devices, and that ad-hoc networks are dynamically routed thanks to protocols such as AODV, DSR or DSDV. It has been proved that such protocols are not suitable when disconnections are frequent and unpredictable, and when networks are fragmented into different communication islands. In practice, ad-hoc networks supporting pervasive environments are usually intermittently-connected networks, and therefore opportunistic or delay-tolerant networking techniques must be used to improve communication in such networks. In [8], Capra et al. present a service composition framework for mobile environments that allows to improve the service composition reliability by considering the mobility pattern of devices and their colocation duration (the probability of service access is related to the colocation duration). Similarly, work in [9] tries to improve service composition dependability by characterizing the service providers' mobility and by tolerating an uncertain mobility of these ones. But yet here, end-to-end communications are assumed between service clients and service providers.

Relying on the "store, carry and forward" principle, opportunistic and disruption-tolerant communication techniques allow to forward messages between service clients and service providers even if there is no end-to-end path between them. This feature must be taken into account in the composition process, since the service may be delivered with a non negligible delay. In such a context, the main objective is to reduce as much as possible this delay in order to provide users with a certain quality of service. In contrast with the above-mentioned works, the service composition solution proposed in [10] is dedicated to opportunistic networks. This solution relies on two metrics: *shortest temporal distance,* which is the minimum time needed to send data from one node to another one, and *service load,* which reflects the workload of a given service. These metrics are used in the composition process in order to select the services to invoke. The presented system implements a choreography-based service composition: when a service terminates, the providers of this service is expected to find a provider to achieve the next service execution defined in the composition graph; this operation is repeated until the end of the composition process. This work does not investigate and evaluate the orchestration-based composition strategy in opportunistic networks.

III. Opportunistic Service Composition

The composition system we have designed implements service discovery, service selection, service invocation and service composition mechanisms. Compositions can be achieved by our system either using an orchestration-based or a choreography-based strategy. Like research works presented in

the previous section, our system supports an incremental and partial composition process, thus allowing to compose services progressively according to their discovery, and to avoid to rebuild the whole composition when a service provider involved in the composition process becomes unreachable.

To improve the composition reliability and to reduce the composition time as much as possible, we limit the communication scope of devices, and we select the most relevant service providers according to a utility-based function. The communication scope of devices is currently specified in number of hops. By limiting this number to a maximum value H_{max}, we prevent service advertisements, service invocation requests and service composition requests from being transmitted to distant nodes and *de facto* we increase the chances to obtain the responses and to finish the compositions in a acceptable time. This number of hops are included in the service messages created by our system, and are taken into account by the C3PO opportunistic computing middleware [2] during the transmission process.

The current utility-based function implemented in our system relies on the average of the *multi-hop inter-service advertisement time* (MHISAT) between a service client and a service provider. More formally for a service client, the result returned by this function for a service provider P is defined by:

$$\mu(P) = \frac{\sum_{k=2}^{n_P} SART_k^P - SART_{k-1}^P}{n_P}$$

where $SART_k^P$ is the reception time of the k-th service advertisement received from P, and n_P is the number of advertisements received from P since the system started. The selection function associated with this utility-based function selects, among a set of providers offering the required service, the provider that has the minimum average.

A. Discovery

Every period of time T_{adv}, each node broadcasts an advertisement message M_{adv}. M_{adv} is defined by a quadruplet $\{D,H,P,T\}$, where D is the set of descriptions of services provided by the service provider P, T is the emission time of the advertisement, and H is the number of hops M_{adv} can make before being delete. H is initialized to H_{max} when M_{adv} is emitted by the service provider, and it is decreased (by -1) when M_{adv} is received by a node. H_{max} is the maximum number of hops to disseminate M_{adv} . The retransmission of M_{adv} is stopped when H is equal to 0.

Each node running our composition system maintains its own service registry. This registry contains the services they provide themselves, and the services they have discovered recently. It is formally defined by the set $\{L,R\}$ where L is the set of local services and R is the set of remote services. R is formed by 6-uplets defined by $\{P,S,D,H',U,\mu(P)\}$, where P is the address of the provider, S is the service name, D is the description of the service S , H' is the number of hops to the provider ($H' = H_{max} - H$), U is the last time the 6-uplet

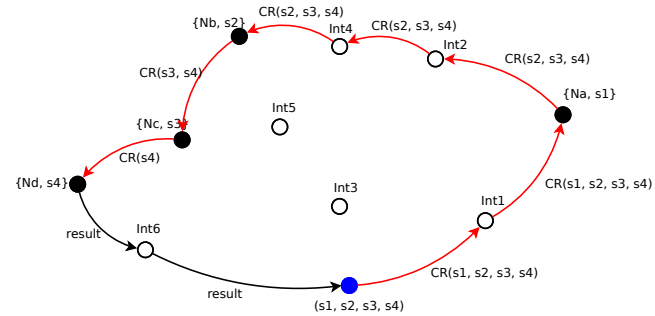


Figure 1: choreography example

has been updated, which corresponds to the reception time of M_{adv}, and $\mu(P)$ is the result of the utility-based function for P. The subset of 6-uplets from the provider P (noted $R[P]$) is updated when a new advertisement is received from P. If $R[P]$ has not been updated since a certain amount of time $T_{inactive}$ or if $\mu(P)$ is greater than a certain threshold T_{far}, the service instances provided by P are considered to be unreachable and therefore they will not be involved in any composition process. The minimum value of $\mu(P)$ is T_{adv}. The *shortest temporal distance* for a service provider P, which is the minimum time needed to send data from one node to another one, is thus defined by $\delta(P) = \mu(P) - T_{adv}$.

B. Choreography-based Service Composition

The choreography-based service composition strategy consists of transmitting to the first selected service provider the composition request, and in delegating to it both the execution of the first service and the selection of the next provider as specified in the composition request. It will pass to this next provider the results of its execution and the composition request (CR). This process will be repeated until the composition is completed. Figure 1 represents a scenario where a CR of 4 services ($s1$, $s2$, $s3$, $s4$) is executed using the choreography-based composition strategy. The requester starts by choosing the node to execute the first service (i.e., $s1$). In this scenario, the nearest node providing $s1$ is Na. Na is at two hops from the requester. Thus, the CR will be sent first to the intermediate node $Int1$, this one will relay CR to the host Na. After finishing the execution, Na updates the CR and chooses the next node which is Nb. The CR, this time, will be transmitted via $Int2$ and $Int4$ to rich Nb since the latter is at 3 hops from Na. Nb will then pass it directly to Nc. Finally, the CR will be sent to Nd which finishes the service composition and forwards to the requester the final result.

C. Orchestration-based Service Composition

In contrast with the choreography-based service composition, in the orchestration-based service composition, the composition request (CR) is never transmitted to the service providers enrolled in the composition process, and the service invocation is never delegated to them. In fact, the providers are selected and invoked by the node that has generated CR itself.

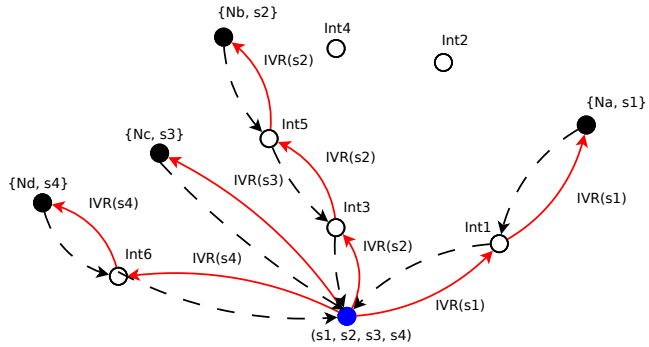


Figure 2: orchestration example

This one selects the providers on the basis of the average of the multi-hop inter-service advertisement time. Figure 2 shows how this composition algorithm works. The requester starts the same CR discussed in the choreography example. It invokes service $s1$ provided by Na. After obtaining the results of this invocation, it invokes service $s2$ provided by Nb. This service request will be forwarded by intermediate nodes $Int3$ and $Int5$ respectively; the response is transmitted by the same path. This process is repeated to invoke the services $s3$ and $s4$ provided respectively by nodes Nc and Nd.

D. Estimation of the composition time

For a composition C of n services identified by $S1,...,Sn$ emitted by a requester r and provided by k providers (with $k \in [1,n]$), the composition time of C is defined for the orchestration-based strategy by:

$$\tau(C) = \sum_{i=1}^{n} 2 * min_{j=1}^{k}(\delta_r(P_i^j))$$

where P_i^j is one the k providers that offers the service S_i. If a provider do not provide S_i, $\delta_r(P_i^j)$ is equal to $+\infty$ in our estimation of the composition time. If the service is provided by the composition requester, $\delta_r(P_i^j)$ is equal to 0 in our estimation. The minimum of $\delta_r(P_i^j)$, reflects the minimum time needed to invoke a provider of the service S_i. In the orchestration-based strategy, the response must be returned to the node that has initiated the composition request, thus the minimum time is multiplied by 2 to consider this round trip.

Concerning the choreography-based strategy, the estimation of the composition time is defined by:

$$\tau(C) = min_{j=1}^{k}(\delta_r(P_1^j)) + min_{j=1}^{k}(\delta_r(P_n^j)) + \sum_{i=1}^{n-1} min_{j=1}^{k}(\delta_j(P_{i+1}^j))$$

where the first part of the formula is the time to send the composition request to the provider of the first service to compose, the second part of the formula is the time to receive the result of the composition from the last provider involved in the composition process, and the last part of the formula is the sum of the intermediate composition times.

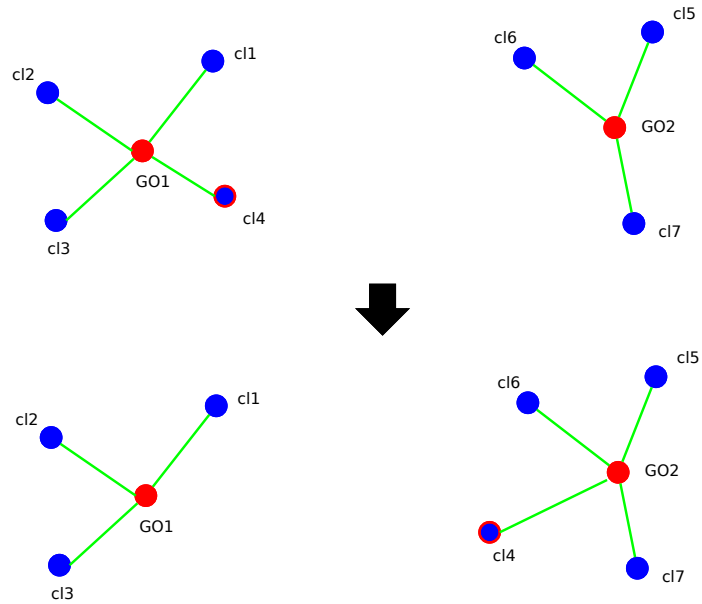


Figure 3: WiFi Direct scenario

IV. Evaluation

A. Evaluation setup

The discovery, invocation and composition system we have proposed has been developed using the C3PO framework [3], and evaluated using an emulator developed within our research team. Two distinct scenarios have been considered in this evaluation: one, involving people roaming in a open area of 500x500 meters, and another one, involving attendees that move along the running path of a sport event in the city of Vannes in France. Both the orchestration-based and choreography-based composition strategies have been evaluated in these two scenarios. Four different configurations are thus considered hereafter (open area/orchestration, open area/choreography, Vannes city/orchestration, Vannes city/choreography). In these scenarios and configurations, people move following the Levy Walk [11] mobility model at a speed between 0.5 m/s and 2 m/s, and use their smartphones while they are moving. The number of people varies between 50 and 250. We assume that these smart-phones can communicate in Wi-Fi Direct [12], using the Android implementation, and that their radio range is a maximum of 80 m. To communicate with Wi-Fi Direct, each two devices must be in the same group (i.e., in the same network cell). A group is managed by a group owner (GO) that acts as a soft access point. The group owner is dynamically chosen among one of the two devices that want to communicate first. A new device can join a group that already exists. In the current Android implementation of Wi-Fi Direct, a GO can have only 7 clients. A client can not be connected to more than one GO, and a GO can not be connected to another GO. As Figure 3 suggests, if the client cl_4 wants to connect to the group owner GO_2, it must disconnect from GO_1 first. This will cause the network to be highly disconnected and formed

Parameter	Value
Open area size	500 m x 500 m
Composition request generation	between 2 and 5 min.
Evaluation Duration	1 hour
Advertisement period	10 seconds
Service registry entry inactivity threshold	20 seconds
Number of service per composition	between 3 and 6
Number of local services per node	5
Number of hops in advertisement messages	between 1 and 3
Number of node per experience	50, 100, 150, 200, 250
Speed range	between 0.5 and 2m/s

Table I: Evaluation parameters

by a small isolated isles of connected devices. The emulations last 1 hour each. We also suppose during these evaluations that devices deliver 5 services, and submit to our system a composition request (including between 3 and 6 services) every x minutes; x is between 2 and 5. All the evaluation parameters are summarized in table I.

B. Results

The evaluation results presented hereafter show the impact of the number of hops between the composition requester and the remote services providers on the composition time, the number of nodes involved in a composition process, the composition success ratio and the number of compositions executed by a node.

Composition time: Figure 4a shows the median composition time (MCT) against the maximum number of hops between the request composition emitter and the service providers involved in the composition process. The MCT increases together with the number of hops in the four configurations we consider. The choreography-based composition strategy (CCS) provides a lower composition time than the orchestration-based composition strategy (OCS); thus offering a better response time to the request composition emitters. Indeed, in the open area scenario, the MCT increases from 242 ms to 379 ms for the OCS, and from 11 ms to 101 ms for the CCS. In Vannes city scenario, the MCT increases from 239 ms to 469 ms for the OCS, and from 8 ms to 158 ms for the CCS. Moreover, the distributions of the composition times represented in Figure 5 for 2-hop communications, shows that OCS has a close to 0 % of service compositions with composition time less than 200 ms. Composition times for the OCS are spread between 200 ms and 600 ms with a maximum pick between 15 % and 25 % in the open area and between 10 % and 14 % in Vannes city. Whereas in the case of CCS, most of the composition times are located between 10 ms and 200 ms with a maximum pick between 20 % and 25 % in the open area and between 10 % and 14 % in Vannes city. Both of these picks take place around 10 ms of composition time. The explanation resides in the fact that in with OCS, the services are exclusively invoked by the composition initiator, and that the service responses must be returned to this one in order to continue the composition process. The service responses are likely to pass through intermediate nodes, increasing de facto the composition time. With CCS, only the result of the last service invocation is

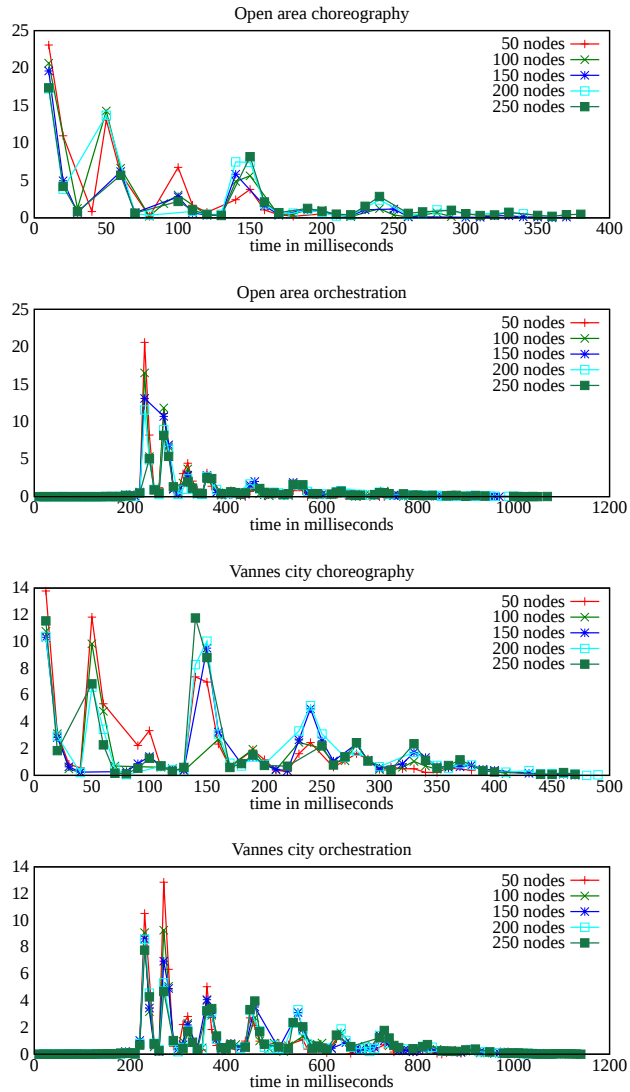
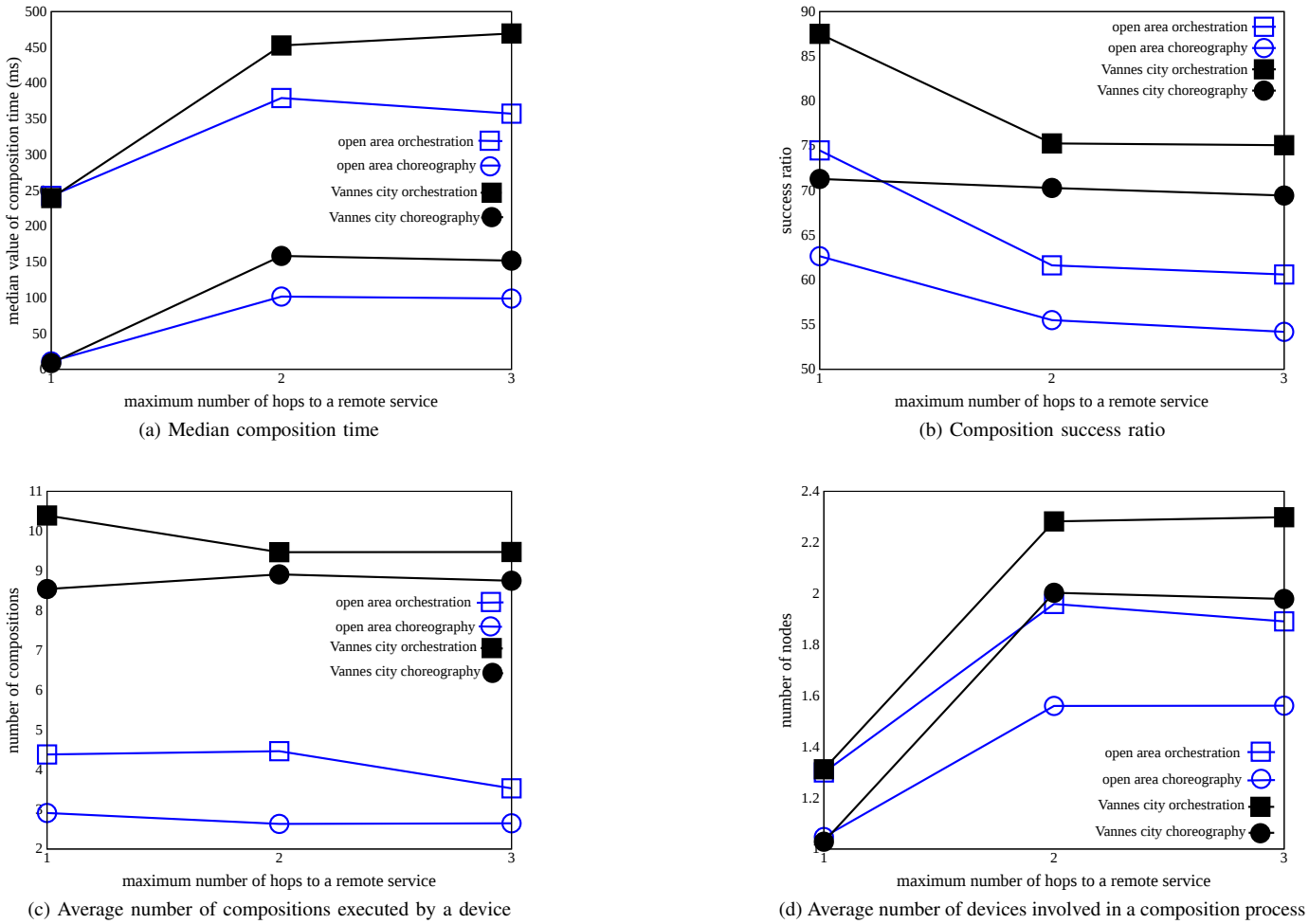


Figure 5: Composition time distribution for 2 hops

returned to the composition initiator, the intermediate results are used by service providers involved in the composition process to invoke the next service provider as specified in the composition request, and are never returned to the composition initiator. In addition with OCS, more messages are exchanged during the composition process than with CCS.

Success ratio: Figure 4b shows the service composition success ratio against the maximum number of hops between the composition initiators and the service providers involved in the composition process. The success ratio in Vannes city scenario is better than in the open area for the both composition strategies. This better success ratio is explained by the fact that, in the sport event scenario, the movements of attendees are constrained by the natural environment and by the fact that they move along the running path, whereas in the open area people can go anywhere and their mobility is not constrained. In addition, the OCS provides a better ratio



(a) Median composition time

(b) Composition success ratio

(c) Average number of compositions executed by a device

(d) Average number of devices involved in a composition process

Figure 4: Impact of the number of hops on the composition process

(60.6 % to 74.4 % in the open area and 75 % to 87.5 % in Vannes city) than the CSS (54 % to 62.6 % in the open area and 69.4 % to 71.28 % in Vannes city) because the composition initiator receive intermediate responses, and thus can continue the composition itself until receiving the final response. With the CCS, the intermediate responses are received and stored by the service providers involved in the composition process. Since, each of these providers can invoke a service at k-hops (with k between 1 and 3 in our evaluations), the final response is likely to be returned by a provider, which is $k * |CR|$ hops away from the composition initiator, where k is the number of communication hops considered in the evaluation, and $|CR|$ the number of services defined in the composition request CR. With the OCS, the intermediate responses and the finale responses is always returned by a provider at most k-hops from the composition initiator. The success ratio obviously decreases when the number of hops increases.

Number of compositions executed by a node: Figure 4c shows the average number of compositions executed by a node against the maximum number of hops between the composition initiators and the service providers involved in the composition process. The four curves present the same shape of nearly a horizontal line which shows that the average number of compositions executed by a node is not impacted by the communication scope (i.e., the number of hops). The number of compositions achieved in Vannes city scenario is greater than that performed in the open area scenario. This difference resides in the fact that the mobility of people in the sport event scenario is more constrained than that in the open area. Consequently, the contact opportunities are more frequent, and more service providers can be enrolled in the composition processes, thus allowing to complete more compositions. Moreover, the OCS allows to execute a small number of additional compositions. In fact, in the open area it is between 2.63 and 2.9 compositions for the CCS against 3.52 to 4.46 compositions for OCS and in Vannes city it is around 8.5 compositions for the CCS against around 9.5 compositions for the OCS. The explanation of these results is similar to that given in the success ratio paragraph.

Number of nodes involved in a composition: Figure 4d shows the average of the number of nodes involved in a composition process. This average is presented against the

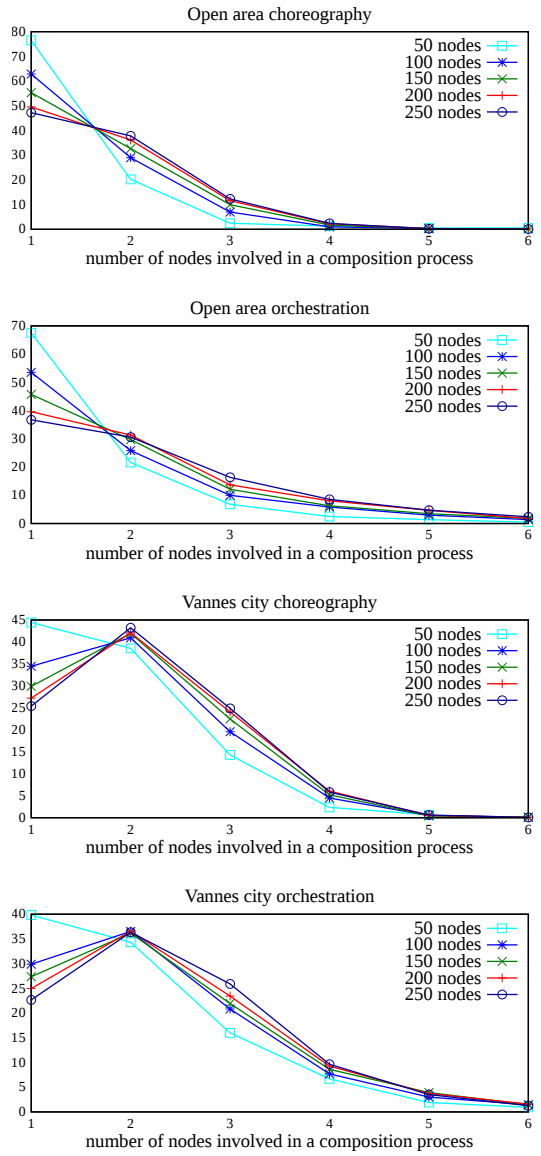


Figure 6: distribution of node number per composition with two hops away remote services

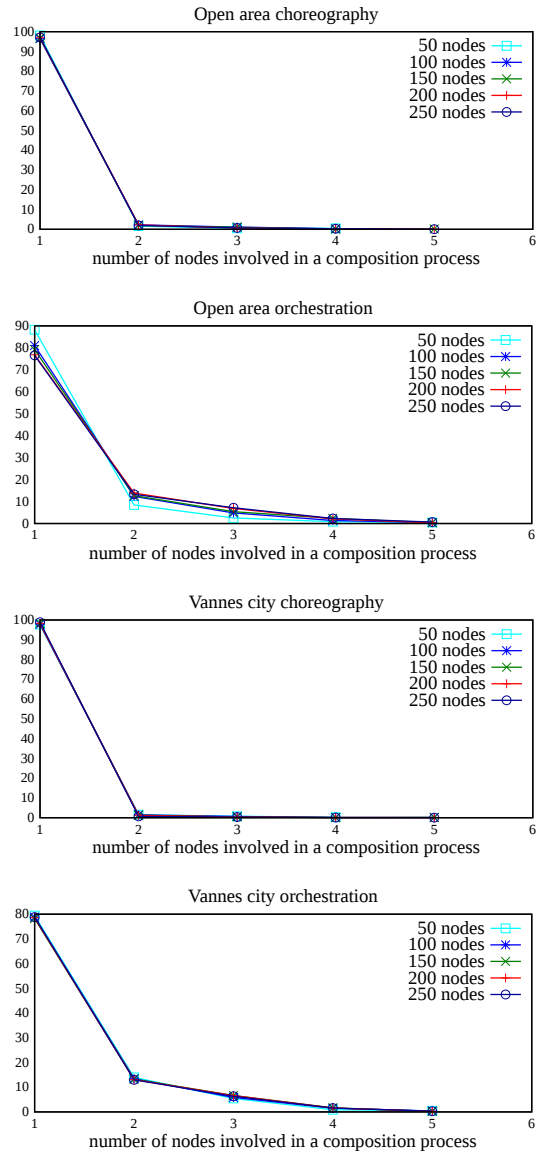


Figure 7: distribution of node number per composition with one hop away remote services

number of hops between the composition initiator and the services providers. This average increases with the number of hops. This is totally logical since more service providers can be reached, and thus can be involved in a composition process. The growth is not as important between 2-hops and 3-hops than between 1-hop and 2-hops. This is the result of the significant redundancy of service providers we have in our scenarios. As shown in Figure 7, a large number of compositions are performed using 2 hops. Consequently, when we have a significant redundancy of service providers, it is not necessary to increase the communication scopes of devices (i.e., a number of hops equal to 2 is enough).

V. Conclusion

Throughout this paper, we proposed a service discovery and composition algorithm for opportunistic environments. The algorithm relies on a utility function based on the average of the *multi-hop inter-service advertisement time* between a service client and a service provider. The algorithm comes with two different strategies: choreography-based and orchestration-based. In the simulation section, we thoroughly compared the two approaches behaviors. The orchestration approach outperformed the choreography approach in almost all the criteria. However, the choreography approach has a shorter composition time.

Future works may include changing the utility function to improve the overall performance of the algorithm. One

of several options would be to rely on average location coordinates and distance in order to determine if a remote node is near enough to be included in the composition process. Furthermore, a hybrid approach, combining orchestration and choreography could be devised in order to exploit the best of the both worlds.

REFERENCES

[1] M. Conti and M. Kumar, "Opportunities in Opportunistic Computing," *IEEE Computer*, vol. 43, pp. 42–50, 2010.

[2] N. Le Sommer, P. Launay, and Y. Mahéo, "A Framework for Opportunistic Networking in Spontaneous and Ephemeral Social Networks," in *10th ACM MobiCom Workshop on Challenged Networks (CHANTS 2015)*, (Paris, France), pp. 1–4, ACM Press, Sept. 2015.

[3] F. Laforest, N. Le Sommer, S. Frénot, F. De Corbière, Y. Mahéo, P. Launay, C. Gravier, J. Subercaze, D. Reimert, É. Brodu, I. Daikh, N. Phelippeau, X. Adam, F. Guidec, and S. Grumbach, "C3PO: a Spontaneous and Ephemeral Social Networking Framework for a Collaborative Creation and Publishing of Multimedia Contents," in *International Conference on Selected Topics in Mobile and Wireless Networking (MoWNet 2014)*, (Rome, Italy), pp. 129–134, Elsevier, Sept. 2014.

[4] M. Garriga, C. Mateos, A. Flores, A. Cechich, and A. Zunino, "RESTful service composition at a glance: A survey," *Journal of Network and Computer Applications*, vol. 60, pp. 32–53, jan 2016.

[5] N. Ibrahim and F. Le Mouël, "A Survey on Service Composition Middleware in Pervasive Environments," *International Journal of Computer Science Issues*, vol. 1, pp. 1–12, Aug. 2009.

[6] S. Kalasapur, M. Kumar, and B. Shirazi, "Seamless Service Composition (SeSCo) in Pervasive Environments," in *1st ACM International Workshop on Multimedia Service Composition (MSC'05)*, (Hilton, Singapore), pp. 11–20, ACM, 2005.

[7] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha, "Service composition for mobile environments," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 435–451, 2005.

[8] L. Del Prete and L. Capra, "Reliable Discovery and Selection of Composite Services in Mobile Environments," in *12th Enterprise Distributed Object Computing Conference (EDOC'08)*, (Munich, Germany), pp. 171–180, IEEE, Sept. 2008.

[9] J. Wang, "Exploiting Mobility Prediction for Dependable Service Composition in Wireless Mobile Ad Hoc Networks," *IEEE Transactions on Services Computing*, vol. 4, pp. 44–55, Jan. 2011.

[10] U. Sadiq, M. Kumar, A. Passarella, and M. Conti, "Service Composition in Opportunistic Networks: A Load and Mobility Aware Solution," *IEEE Transactions on Computers*, vol. 84, pp. 2308–2322, Aug. 2015.

[11] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the Levy-Walk Nature of Human Mobility," *IEEE/ACM Transactions on Networking*, vol. 19, pp. 630–643, June 2011.

[12] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *IEEE Wireless Communications*, vol. 20, pp. 96–104, June 2013.