Location-based Service Discovery and Delivery in Opportunistic Networks

Nicolas Le Sommer, Salma Ben Sassi

Location-based Service Discovery and Delivery in Opportunistic Networks

Nicolas Le Sommer and Salma Ben Sassi
Valoria Laboratory
Université Européenne de Bretagne, France
{Nicolas.Le-Sommer, Salma.Ben-Sassi}@univ-ubs.fr

***Abstract*—Opportunistic networks are usually formed spontaneously by mobile devices equipped with short range wireless communication interfaces. Designing and implementing a routing protocol to support both service discovery and delivery in such kinds of networks is a challenging problem on account of frequent disconnections and topology changes. In these networks one of the most important issues relies on the selection of the best intermediate node(s) to forward the messages towards their destination(s).**

In this paper, we propose a new location-based opportunistic routing protocol enabling a bandwidth-efficient service discovery and delivery in a wide area network composed of numerous mobile devices. This protocol implements self-pruning heuristics allowing mobile devices to decide whether they efficiently contribute in the delivery of the messages they receive from their neighbors. This protocol was validated through simulations, which proved its efficiency and cost-effectiveness.

***Index Terms*—Service Provision, Opportunistic Networking, Disconnected Mobile Ad Hoc Networks**

I. Introduction

Mobile devices equipped with short range wireless communication interfaces (e.g., Wi-Fi, Bluetooth), and sometimes with a GPS receiver, are nowadays widespread and used daily by an increasing number of people. Netbooks, mobile internet devices or smartphones are some examples of such devices. Thanks to their short range wireless communication interfaces, these devices can spontaneously form a multi-hop disconnected mobile ad hoc network (DMANET). Designing a routing protocol supporting both service discovery and delivery in such kinds of networks is radically different than devising one for traditional infrastructure-based networks. Indeed due to the mobility of nodes and to the short communication range of wireless interfaces, the topology of DMANETs suffers from frequent and unpredictable changes, entailing an intermittent connectivity between nodes. In DMANETs, devices can thus communicate directly only when they are in range of one another. Otherwise, intermediate nodes can be used to relay a message from a source to a destination following the "store, carry and forward" principle. Since in DMANETs we do not have any knowledge about the network evolution, routes are computed dynamically at each hop while forwarding messages towards their destination(s). Each node receiving a message intended to another recipient, is expected to exploit its local knowledge in order to decide for instance which is the best next hop among its current neighbors to deliver the message. When no forwarding opportunity exists (e.g., no

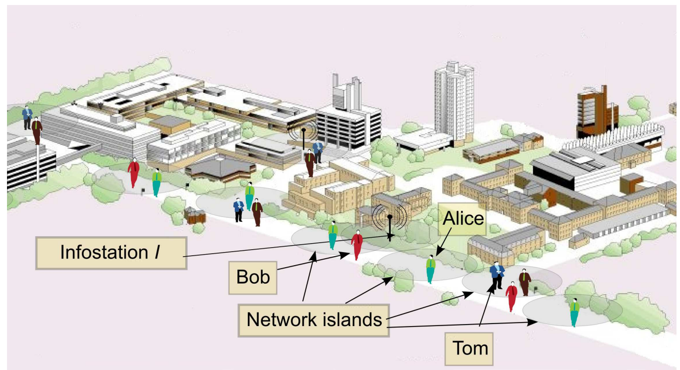

Figure 1. Example of a disconnected MANET.

other nodes are in the transmission range, or the neighbors are evaluated not suitable to that communication) the node stores the message and waits for future contact opportunities with other devices to forward the message. Thanks to this principle, a message can be delivered even if the sender and the recipient are not present simultaneously in the network, or if they are not in the same network partition at the emission time. For instance in the disconnected mobile ad hoc network illustrated in Figure 1, Tom may discover and invoke the services offered by infostation *I*, even if he is not in the communication range of *I*. Indeed, the messages sent by Tom and by infostation *I* may be forwarded opportunistically by intermediate devices, such as those carried by Bob and Alice who are walking along the main street of the city.

In this paper, we propose a new message opportunistic forwarding protocol, called OLFServ, suited to the provision of location-aware software services in disconnected, partially connected or intermittently connected MANETs. This protocol is designed to perform an efficient and geographically-controlled broadcast of service advertisements and service discovery requests. This forwarding protocol is also suited to the invocation of remote service providers. Indeed, it enables service providers to specify in their advertisements their location and the geographical area where they can be discovered and invoked. It allows clients to define their location, the area where their messages must be disseminated, the location of the provider they want to invoke, as well as their moving direction and speed if they know them. Based on these pieces of information, intermediate nodes running OLFServ can decide themselves if they are "good" relays to deliver the messages they receive from their neighbors (i.e., if they contribute to bring a message closer to its destination). For this purpose, OLFServ implements self-pruning techniques in order to select a small subset of intermediate and relevant nodes that will have to broadcast the messages they receive in given geographical areas. In OLFServ, intermediate nodes can update location properties specified by clients and providers in service messages while forwarding these messages in order to refine the area where they must be disseminated progressively, until reaching the destination. In the current scenarios we focus on, service providers are fixed infostations deployed in a city, while clients are devices carried by humans (see Figure 1). Consequently, a client may not stay long enough at the location where he was when he sent its service invocation to receive at this same place the response returned by the provider it invoked. In order to address this issue, OLFServ enables service providers to compute, based on the properties exhibit by a client (location, speed, direction/destination), an ”expected area” where the client is supposed to be when he will receive its response. Finally our protocol implements a random delayed message forwarding in order to avoid message collisions.

The remainder of the paper is organized as follows. Section II motivates our work by discussing some recent research works dealing with routing protocols in disconnected, partially connected or intermittently connected mobile ad hoc networks. Section III presents our new opportunistic and location-aware forwarding protocol. Section IV presents the results we obtained by running our protocol on a network simulator, and we compare these results with other solutions. Section V concludes this paper and gives some perspectives we plan to work along in the future.

II. Related work

Service provision relies on two main processes: service discovery and service invocation. In decentralized and highly changing environments such as those we consider, the service discovery process cannot rely on a centralized approach since no device is stable enough, or accessible permanently, to act as a service register. Therefore, each client is responsible for maintaining its own perception of the services offered in the network, and for discovering these services. The discovery process can be performed reactively, by processing the unsolicited service advertisements broadcast by service providers, or proactively, by broadcasting service discovery requests in the network and by processing the advertisements returned by providers in response. The service invocation, during which a given client actually interacts with the provider of a previously discovered service, is generally performed using a unicast and destination-based communication model. A selection process may precede the invocation, when the opportunity is given to the service client to choose among several providers.

Forwarding protocols designed for opportunistic networking and/or delay tolerant networking can be classified according to both the network knowledge required by mobile hosts in order to run them and the methods used to select the hosts that must forward the messages. Three kinds of methods can be identified: blind-based methods, capability-assessment-based methods, and neighbor-knowledge-based methods. The last category can be divided in two sub-classes: the designating methods and the self-pruning methods. In the first sub-class, the message forwarder (or the initial message sender) is responsible for selecting the next suitable intermediate node(s), whereas, in the second sub-class, intermediate nodes receiving a message from a neighbor node are expected to decide whether they must forward this message or not based on their local knowledge.

Blind-based protocols, such as the Epidemic Routing protocol [1], do not require any knowledge and do not make any attempt to reduce the number of forwarding nodes. The messages are blindly stored and forwarded to all neighboring nodes, generating a flood of messages. The drawback of such an epidemic dissemination lies in the very high number of message copies which are needed to obtain a successful delivery of messages. These protocols are thus not suitable to environments with high density regions, since they would generate too much network traffic and sometimes could even lead to network congestion. This problem is referred to as the broadcast storm problem [2].

This drawback is addressed by protocols implementing methods allowing to assess the capability of neighbor nodes to contribute in the message delivery. These methods traditionally use a probabilistic metric, often called delivery predictability, that reflects how a neighbor node will be able to deliver a message to its final recipient. Before forwarding (or sending) a message, a mobile host asks its neighbors to infer their own delivery probability for the considered message, and then compares these probabilities and selects the best next hop(s) among them. This computation can require a 1-hop, and sometimes a 2-hop, network knowledge. In Context-Aware Routing protocol (CAR) [3] and GeOpps [4], the delivery probabilities are computed using both utility functions and Kalman filter prediction techniques. CAR allows each host to associate a utility function, representing the delivery probability, with every other host. These functions process contextual properties, and especially the change rate of connectivity and the probability of being located in the same cloud as the destination. Instead of using the current contextual information, CAR uses predicted future values of the context in order to have more realistic values. CAR assumes an underlying MANET routing protocol that connects nodes together in the same MANET cloud. To reach nodes outside the cloud, a sender looks for the node in its current cloud with the highest message delivery probability. In CAR, nodes compute delivery probabilities proactively, and disseminate them in their ad hoc cloud (they are exchanged with standard routing tables). Therefore, the contextual properties are exploited to evaluate probabilities just for those destinations that each node is aware of. GeOpps, which is a geographical delay tolerant routing algorithm, exploits information from the vehicles' navigation system to route messages to a specific location. To select the next packet carrier, neighbor vehicles that follow suggested routes to their driver's destination calculate the nearest point that they will get to the destination of the packet. Afterward, they use the nearest point and their map in a utility function that expresses the minimum estimated time that this packet would need in order to reach its destination. The vehicle that can deliver the packet quicker/closer to its destination becomes the next packet carrier. Like CAR, HiBOp [5] also exploits context information in order to compute delivery probabilities. However, HiBOp can be perceived as being more general than CAR since it does not require an underlying routing protocol, and because it is also able to exploit context for the destinations that nodes do not know. HiBOp uses history information in order to improve the delivery probability accuracy, and does not make predictions as CAR. HiBOp maintains an history table in order to estimate the probability of encountering a contextual property in the near future. Propicman [6] also exploits context information and uses the probability of nodes to meet the destination, and infers from that the delivery probability, but in a different way. When a node wants to send a message to another node, it sends to its neighbor nodes the information it knows about the destination. Based on this information, the neighbor nodes compute their delivery probability and return it. The node that wants to send the message will send it only on the two-hop route(s) with the highest delivery probability, if this one is higher than its own. Propicman considers that mobile nodes carried by people are not likely to move around randomly, but in a predictable fashion based on repeating behavioral patterns at different timescales (day, week, month). If a node has visited a place several times before, it is likely to visit this location again in the future.

Prophet (Probabilistic Routing Protocol using History of Encounters and Transitivity) [7] has an approach similar to that of Propicman. However, according to our classification Prophet is in the third category (i.e., the protocols implementing neighbor-knowledge-based methods). Indeed, the selection of the best neighbor node is based on how frequently a node encounter another. When two nodes meet, they exchange their summary vectors, which contain their delivery predictability information. If two nodes do not meet for a while, the delivery predictability decreases. When a node wants to send a message to another node, it will look for the neighbor node that has the highest amount of time encountering the destination, meaning that has the highest delivery predictability to the destination. This property is further transitive.

All the above-mentioned protocols implement various strategies aiming to select the next best hop(s) to deliver a given message. These protocols could probably be used to perform service invocation, which traditionally relies on a unicast communication model, but not to achieve service discovery, which requires an efficient broadcast of service discovery requests and service advertisements. Indeed in order to avoid the broadcast storm problem and a network congestion, these messages must not be broadcast in a blindly epidemic manner, but instead in given geographical areas by protocols implementing both a delayed message forwarding and self-pruning heuristics, thus making it possible for mobile nodes to decide whether they should forward a message or not.

Over the last years, several broadcast protocols implementing such as features have been proposed for connected MANETs. Willians and Camp [8], and later Stojmenovic and Wu [9], have proposed classifications of broadcast protocols for MANETs. They have identified four categories: simple flooding, probability-based methods, area-based methods and neighbor-knowledge-based methods. Like the epidemic forwarding protocol, the simple flooding [2] does not make any attempt to reduce the number of broadcasting nodes and does not require any knowledge. In probability-based methods, each node estimates a potential contribution to the broadcasting process. If this contribution is lower than a given threshold the message is not forwarded [10]. In Area-based and location-based [2] methods, a message can be re-emitted only if the surface or the distance of this area is upper than a given threshold. Finally, neighbor-knowledge-based methods require the knowledge of the 1-hop or 2-hop neighborhood. DFCN (Delayed Flooding with Cumulative Neighborhood) [11] is an example of such protocols. It proposes a bandwidth-efficient algorithm that introduces a benefit parameter allowing hosts to decide if a message must be broadcast or not according to their neighborhood status and to a random delay that aims at avoiding message collisions. These above-presented broadcast protocols are designed for connected MANETs and are not thus suitable to opportunistic networks that are intrinsically disconnected. Moreover, these protocols do not make it possible to restrict the propagation of messages geographically or to define the area where they must disseminated.

III. PROTOCOL OLFSERV

In this section we present OLFServ (Opportunistic and Location-based Forwarding protocol for Service provision), a protocol that supports both the discovery and the invocation of location-aware services in opportunistic networks, and especially in those formed by fixed infostations providing software services and by mobile devices carried by humans and behaving as service clients (see Figure 1). OLFServ addresses three main issues: 1) the efficient and geographically-constrained broadcast of service discovery messages, 2) the selection of relevant carriers to perform service invocation, and 3) the support of clients' mobility during the service invocation process. The remainder of this section presents the assumptions on which OLFServ relies, the heuristics it implements and how service discovery and invocation are performed using scenarios.

A. Requirements

Protocol OLFServ relies on 4 main assumptions:

1) Mobile hosts and fixed infostations are aware of their geographical location and able to compare their location with that of another host (for example with that of the recipient of a given message).
2) Mobile hosts are able to perceive their one-hop neighborhood. Such a piece of information is obtained using specific messages (beacon messages).
3) Service providers and clients are expected to include in messages the network address of the destination, but also
   - the geographical location of the destination,
   - their own location,
   - a date of emission,

- a lifetime,
- a number of hops (equivalent to the TTL field of the IP protocol),
- the geographical area where messages can be disseminated,
- and their speed and direction (or destination) if they are mobile.

Intermediate nodes (or at least the last forwarder) are expected to add their own location in messages before forwarding them. Messages must also include the identifiers of the 1-hop neighbor nodes of the mobile host that has originally sent the message in the network or that has forwarded it recently.

4) Each mobile host is able to store the messages it receives temporarily, and able to associate to each of them some information, and especially the identifiers of nodes that are known to have received them.

B. Overview of OLFServ

In OLFServ, communications are currently performed on an IP multicast group. In order to avoid message collisions and the broadcast storm problem when sending messages in the network, OLFServ implements a delayed message forwarding and location-based and neighbor-knowledge-based self-pruning heuristics.

The forwarding delay computed by the mobile hosts has two components: a component that is inversely proportional to the distance from the last forwarder and a random component. For this purpose, we have divided the wireless communication range of devices in several rings, and we have assigned to each ring a range of values so that the values of ring *i* are greater than those of ring *i+1* (see Figure 2). When receiving a message, devices are expected to infer in which ring of the message sender they are located, and then to compute a random value included in the range of values of this ring. The farthest nodes from the sender are therefore expected to forward the message before the closest nodes, thus improving the geographical propagation of messages. Moreover, a node will cancel its message forwarding process if it perceives that all of its neighbors have already received the message it plans to forward, thus limiting the number of messages that roam in the network.

In OLFServ, the forwarding process is driven by four main events, namely the message reception, the expiration of the message delay forwarding, the detection of a new neighbor and the location changes. It is also constrained by location aspects. When receiving a message, hosts running OLFServ compute a new forwarding delay. When this delay has expired, the mobile hosts, which are in the area where this message can be disseminated, will forward this message only if some of their neighbor nodes have not received the message yet. Otherwise, it stores the message locally until it expires.

If the message must be broadcast in a whole given area, as is the case for both service advertisements and service discovery requests, all of the nodes composing the 1-hop neighborhood boundary of the message sender are thus likely to forward the message with the respect to above-presented constraints.

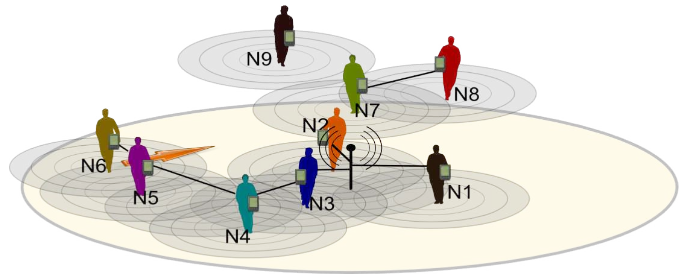

Figure 2. Service discovery with OLFServ in disconnected MANETs.

In contrast, for service invocation requests and responses, for which the location of the recipient is known, only the nodes that are closer to the recipient than the previous hop are expected to forward the message towards the destination. Thus, we progressively refine the area where the message is broadcast until reaching the destination, and we drastically reduce the number of messages that are replicated in the network while having a good message delivery ratio. Mobile hosts that receive a response in reply to a service invocation request they forwarded in the past do not forward this request again in the future.

Finally, a mobile host will compute a new forwarding delay in order to send a message when its context changes. For instance, when it reaches the area where the message can be disseminated or when a new node appears in its vicinity. These properties are used in the invocation process in order to broadcast the response in the area where the client is supposed to be.

C. Service discovery

In order to understand how service discovery is achieved with OLFServ, let us consider the scenario of Figure 2. In this simple scenario, we consider a disconnected MANET composed of a fixed infostation *I* and a set of mobile devices carried by humans. Infostation *I* provides a service that is relevant only in a given geographical area, which is represented by the yellow circle in Figure 2. We assume that one of them is interested by the service proposed by *I*, namely node N_7. We also assume that this network, which is composed of three distinct islands in Figure 2, evolves in an unpredictable way according to the nodes' mobility. We make no assumptions about the mobility of people. For instance, some of them can follow regular mobility patterns, while others can have a random mobility behavior or can be stationary.

In order to use the service offered by infostation *I*, N_7 must discover this service first either proactively or reactively. For the sake of illustration, let us consider this last discovery scheme and let us suppose that infostation *I* has injected in the network an advertisement *A* including its location, the geographical area where *A* must be disseminated, a date of emission, a lifetime, a number of hops and the set of nodes that are expected to receive this advertisement (i.e., *I*, N_1, N_2 and N_3). Advertisement *A* will thus be received first by nodes N_1, N_2 and N_3. These nodes will store *A* locally, and then they will compute a forwarding delay in order not broadcast message *A* simultaneously. Since the forwarding delay is a

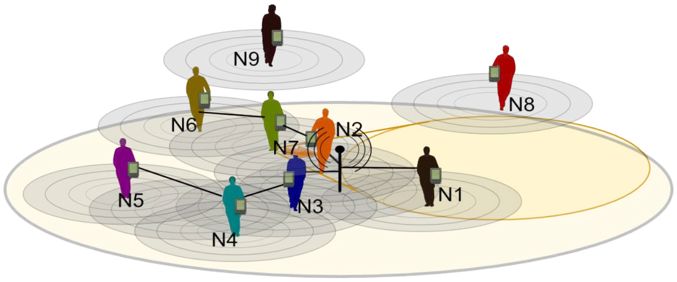

Figure 3. Service invocation with OLFServ in disconnected MANETs.

random value that is inversely proportional to the distance from the last forwarder (in the present case infostation I), nodes N_1 and N_3 will compute a delay in a same range of values. This value will be greater than that computed by node N_2.

Since infostation I has specified in advertisement A that hosts I, N_1, N_2 and N_3 are supposed to have received this message, nodes N_1 and N_2 will not broadcast message A. Message A will thus be forwarded by nodes N_3, N_4 and N_5 successively and will be received by N_6. Thanks to the "store, carry and forward" principle implemented in OLFServ, and which exploits nodes' mobility and contact opportunities, this advertisement will be propagated in the whole area specified by the infostation, and only in this area. Indeed, the self-pruning heuristics implemented in our protocol enable mobile devices to forward messages only if they are in the areas where these messages are relevant. For instance, node N_6, which left the island of the infostation in order to join that of the client (node N_7), will broadcast advertisement A in this new island. This message will be then received by node N_7, which will also forward this message. Then, this message will be received by node N_8. This node will not disseminate message A because it is outside of the area defined by infostation I. Message A will not be received by node N_9 because it is outside of the communication range of node N_7.

D. Service Invocation

Once it has discovered the service offered by infostation I, node N_7 may invoke this service by sending an invocation request that will include especially the location of the provider, its own location, and its speed and direction. This invocation request will be received by nodes N_2 and N_6, and will broadcast only by node N_2, because it is closer to infostation I than node N_7. This request will be then received by infostation I (see Figure 3).

Since N_7 has specified its location, its speed and its moving direction, infostation I can estimate the area where the client is expected to be when it should receive the response. So when returning its response, infostation I includes its location and the area where N_7 is supposed to be. For instance, the orange circle in Figure 3. The response will be then routed towards this area using a forwarding scheme comparable to that used for the invocation. When the response reaches this area, the precise location of N_7 is not known. Therefore, the response will be disseminated in the whole area following a broadcast scheme comparable to that used for the service discovery in order to improve the service delivery. When they receive a response for an invocation they have stored previously, the mobile devices remove the invocation message from their local cache and stop to forward it.

IV. Experiments and Results

In order to evaluate our protocol, we have made a series of simulations using the Madhoc simulator[1], a metropolitan ad hoc network simulator that features the components required for both realistic and large-scale simulations, as well as the tools essential to an effective monitoring of the simulated applications. This simulator, which is written in Java, allow us to run our middleware platform on it.

A. Experiments

The simulation environment we consider is an area about $1km^2$, which is composed of 6 squares of 100m $\times$ 100m linked by predefined paths. An infostation is installed in each square. An infostation offers only one service. A same service is offered by two distinct infostations. The discovery and invocation area of each infostation is a circle with a radius of 300m. This environment is populated with 100 mobile devices equipped with Wi-Fi interfaces, which move along predefined paths in order to reach a given destination located in one of the predefined squares. The destinations of nodes are chosen randomly. When it has reached its destination, a mobile device strolls during 5 minutes in the square where it is located. Afterward, it chooses another destination. Each mobile host moves at a average speed between 0.5 and 2 m/s. The communication range of mobile devices and of infostations varies from 60 to 80m. Among the 100 mobile hosts, 70 act as clients of the services offered by the 6 infostations, whereas the 30 others only act as intermediate nodes. In our scenario, the infostations broadcast service advertisements every 30 seconds. Once they have discovered a relevant service, the clients invoke this service periodically (every 5 minutes) with a different request. All messages have a lifetime of 10 minutes and a number of hops of 8 hops. Messages are forwarded only in areas where they are relevant. Moreover, our middleware platform selects the best service provider when several service providers have been discovered (i.e., the closest provider) [12]. In OLFServ, beacon messages are sent every 20 seconds.

B. Results

In this section, we focus on a particular experiment whose objective was to measure the ability to satisfy the client invocation efficiently. For this purpose, we compared the service provision with OLFServ, with one implementing a 1-hop discovery and invocation model, and with another one implementing a purely epidemic discovery and invocation model. In the 1-hop model, clients must be in the vicinity of a provider offering the service they require in order to discover and to invoke this service. In contrast in the epidemic model, messages are forwarded by mobile devices opportunistically

[1]http://agamemnon.uni.lu/~lhogie/madhoc/

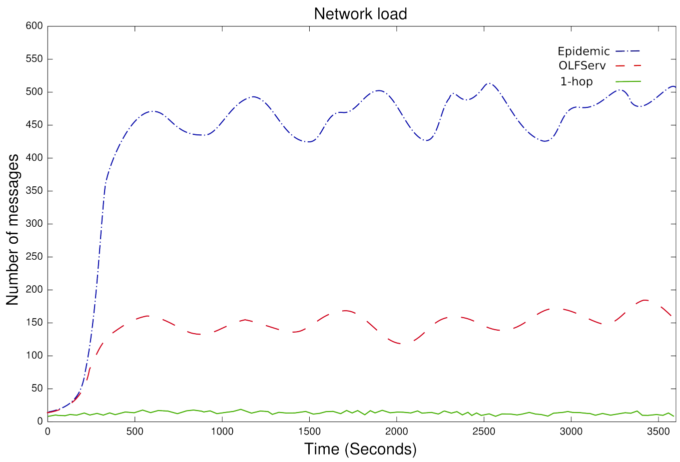

Figure 4. Network load.

	1-hop	epidemic	OLFServ
Number of clients having discovered a provider	52/70	70/70	70/70
Average delay of success invocation(sec.)	3	122	39
Standard Deviation for delay of success invocation (sec.)	2	94	21
Average invocation success ratio	0,96	0,55	0,94
Average of number of invocations	1,8	9,6	4,7

Table I
SIMULATION RESULTS FOR SERVICE DISCOVERY AND INVOCATION.

and without any geographical constraints. In this model, service can be discovered and invoked from anywhere.

Figure 4 presents the network load for the simulations we have made, and in Table I we summarize the simulation results we have obtained. In contrast to the invocation with both the epidemic protocol and OLFServ protocol, the 1-hop service provision protocol has the advantage of offering a low network load, but to the detriment of the service provision since only 1,8 invocation requests have been made in average by clients during the simulation (i.e., 1 hour). As shown in Figure 4 and Table I, the OLFServ protocol has approximately the similar invocation success ratio as the epidemic protocol (0,94*4,7= 4,418 for OLFServ and 0,55*0,6=5,28 for the epidemic protocol), but with a lower network load. We also observe that OLFServ provides better results regarding the invocation delay than the epidemic routing protocol. Indeed, with the epidemic protocol the provider can be invoked from anywhere. Some clients can thus obtain some responses even if they are outside the areas specified by the providers, but with a more bigger delay introduces by the store, carry and forward principle. Moreover in the epidemic protocol, mobile devices broadcast their messages periodically every 20 seconds, whereas in OLFServ, messages can be forwarded by mobile devices as soon as they receive new messages. Moreover, our protocol makes it possible to support the mobility of clients, and thus to improve the delivery of responses.

V. CONCLUSION AND FUTURE WORKS

We presented in this paper a new opportunistic and location-based forwarding protocol supporting service discovery and invocation in disconnected mobile ad hoc networks. This protocol addresses three main issues, namely the efficient and geographically-constrained broadcast of service discovery messages, the selection of relevant carriers to perform service invocation, and the support of clients' mobility during the service invocation process.

Despite the good preliminary results we obtained, we plan to investigate new directions with OLFServ. For instance, we plan to implement source routing mechanisms in order to help in the selection of intermediate nodes when forwarding service responses. Finally we would investigate how a n-hop neighborhood perception can contribute to improve the self-pruning heuristics we have implemented in our protocol (typically with n=2). In the future, we also plan to perform more evaluation, and especially by varying the number of mobile devices in our simulation (for instance between 100 to 1000).

REFERENCES

[1] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," tech. rep., Duke University, Apr. 2000.

[2] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 151–162, ACM/IEEE, 1999.

[3] M. Musolesi and C. Mascolo, "CAR: Context-Aware Adaptive Routing for Delay Tolerant Mobile Networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 246–260, 2009.

[4] I. Leontiadis and C. Mascolo, "GeOpps: Geographical Opportunistic Routing for Vehicular Networks," in *IEEE Workshop on Autonomic and Opportunistic Communications (Colocated with WOWMOM07)*, (Helsinki, Finland), pp. 1–6, IEEE Press, jun 2007.

[5] C. Boldrini, M. Conti, I. Iacopini, and A. Passarella, "Hibop: a history based routing protocol for opportunistic networks," in *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks WoWMoM 2007* (M. Conti, ed.), pp. 1–12, 2007.

[6] H. A. Nguyen, S. Giordano, and A. Puiatti, "Probabilistic routing protocol for intermittently connected mobile ad hoc network (propicman)," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM 2007), AOC Workshop*, pp. 1–6, IEEE Explore, jun 2007.

[7] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *Proceedings of The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, August 2004.

[8] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," in *Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, (Lausanne, Switzeland), pp. 194–205, ACM, jun 2002.

[9] I. Stojmenovic and J. Wu, "Broadcasting and Activity-Scheduling in Ad Hoc Networks," in *Mobile Ad Hoc Networking* (S. G. S. Basagni, M. Conti and e. I. Stojmenovic, eds.), pp. 205–229, IEEE Press, 2004.

[10] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," in *INFOCOM 2003*, (San Francisco), IEEE Computer Society, Apr. 2003.

[11] L. Hogie, P. Bouvry, F. Guinand, G. Danoy, and E. Alba, "A bandwidth-efficient broadcasting protocol for mobile multi-hop ad hoc networks," in *Fifth International Conference on Networking and the International Conference on Systems (ICN / ICONS / MCL 2006). April 23-29. Mauritius*, p. 71, IEEE Computer Society, 2006.

[12] S. Ben Sassi and N. Le Sommer, "Towards an Opportunistic and Location-Aware Service Provision in Disconnected Mobile Ad Hoc Networks," in *2nd International ICST Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications (Mobilware 2009)*, vol. 7 of *LNICST*, (Berlin, Germany), pp. 396–406, Springer, April 2009.