

This is the submitted version of the following article: Sánchez, A., et al. *Two birds, one stone: using mobility behavioral profiles both as destinations and as a routing tool* in Journal of network and computer applications (Ed. Elsevier), vol. 99 (Dec. 2017), p. 58-72. Which has been published in final form at: DOI 10.1016/j.jnca.2017.10.008 - © 2017. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Two birds, one stone: using mobility behavioral profiles both as destinations and as a routing tool

Adrián Sánchez-Carmona^{a,*}, Carlos Borrego^a, Sergi Robles^a, Gerard Garcia-Vandellós^a

^a*Department of Information and Communications Engineering (dEIC),
Universitat Autònoma de Barcelona (UAB),
08193 Bellaterra, Spain*

Abstract

We present *HabCast*, a profile-cast communication paradigm that learns about the mobility habits of the location-aware nodes of the network and uses this information both to route the messages, and to deliver them only to the nodes that match the target behavioral profile. *HabCast* substitutes destination's identifier by a mobility profile model called habitat, meaning that allows users to send messages "to any nodes who usually roams around this area" instead of sending messages intended to a node. *HabCast* is designed to operate without network infrastructure, using Opportunistic Networking strategies and operates in three phases: approximation, floating and delivery phase. *HabCast* enables new services and applications on Opportunistic Networking by automatically inferring the nodes' behavioral profiles and using them to define the messages' destinations. The overhead introduced by *HabCast* is evaluated using a proof-of-concept implementation, and its performance and feasibility is studied, through simulation, under the scope of a real *carsharing* application.

1. Introduction and motivation

Opportunistic Networking (OppNet) is designed to operate in challenged scenarios where the communication networks are unavailable or spotty and the resources are scarce [4], [26]. Due to its design, based on the store-carry-and-forward strategy, OppNet is able to deal with the absence of simultaneous end-to-end paths through the usage of mobile devices that opportunistically establish contact and exchange messages between them. For this reason, OppNet is usually used as a communication solution in developing countries [37]...

... but it should not be limited to these kind of scenarios. During the recent Hong Kong's protests, *Firechat* has proven the utility of OppNet in a well-connected, highly-populated, urban scenario [3]. OppNet could take advantage of the high density of mobile devices in urban regions of industrialized countries. Besides, its capability of operating without the infrastructure of Internet Service Providers (ISP) can help fighting problems as the lack of Net Neutrality¹, the censorship² or the need of privacy of

the users³.

We envision a future where many applications will be using Opportunistic Networking technology, as *Firechat* does. A future where mobile users carry ultra portable devices delivering highly personalized, context-aware services without endangering their privacy through the dependance on the ISP. A future that is not impossible. However, there are applications whose specific needs make difficult to port them from an Internet environment to OppNet. For this reason, this future will only be feasible if OppNet researchers are able to find ways to overcome these limitations and to provide new features.

Conventional communication paradigms face important limitations in the context of highly dynamic mobile opportunistic networks. Unicast requires explicit identification of the destination node, but it may be hard to know the identities of all the other nodes. On the other hand, multicast requires the maintenance of group membership, and needs to know if they maintain their interests even after being disconnected for a while. The profile-cast paradigm aims to solve these issues by inferring membership in interest groups based on the past behavioral of nodes, removing the need of being explicitly expressed. But even when a message can be sent to the nodes matching a target profile, instead of to a node's identifier, there is still needed to route the message towards them. In this paper, we address a very complex task: to route a message towards an un-

*Corresponding author

Email addresses: adria.sanchez@deic.uab.cat (Adrián Sánchez-Carmona), carlos.borrego@deic.uab.cat (Carlos Borrego), sergi.robles@deic.uab.cat (Sergi Robles), ggarcia@deic.uab.cat (Gerard Garcia-Vandellós)

¹Although Net Neutrality in the U.S.A. is granted thanks to a rule voted in February of 2015, it is still in discussion in other regions, as the European Union: <https://savetheinternet.eu/>

²The reader can find a ranking of the most censored countries on: <https://cpj.org/2015/04/10-most-censored-countries.php>

³This topic has been drawing increasing attention among the population since the NSA scandal broke in early June 2013.

known number of destinations that will not be recognized until reaching them.

Our proposal consists in a profile-cast paradigm of communication for OppNet called *HabCast*. *HabCast* leverages the existence of life-cycles of the users to learn about their whereabouts and uses this information not only to define the profile of the messages' destinations, but also to route the messages towards the area where there will be more likely to find nodes matching the target profile. This way, both the routing and the delivery of every message are made in the basis of the nodes's usual behaviours. We propose the very first system that takes history-based decisions both to route the messages towards their receivers and to decide which to which nodes deliver them.

Our main contributions are summarized below:

- We present *HabCast*, a profile-cast paradigm of communication. *HabCast* benefits from our previous work in [38] to learn about users' whereabouts, and uses this information to route the messages towards the area where the destinations are more likely to be found, then, it delivers the messages to the nodes matching the target behavioral profile.
- We discuss the limitations of traditional unicast, multicast and manycast communications in OppNet, and suggest a set of real applications that could benefit from our *HabCast* proposal.

The outline of the rest of the paper is as follows. We state the problem we aim to solve and list some of the potential applications that could benefit from the proposed solution in Section 2. We discuss the state of the art in Section 3. We summarize the key concepts of our previous work in Section 4. Then, we present our habitat-based profile-cast paradigm of communication, called *HabCast*, in Section 5. This is followed by the description of a proof-of-concept implementation in Section 6. Next, we study the feasibility of our proposal through simulation in Section 7. Finally, Section 8 conclude this paper.

2. Definition of the problem

In this section, we explain why sending messages to a bunch of unidentified nodes based on their mobility profile would allow OppNet to support new applications and services. Firstly, we explain the issues and limitations derived from the current communication's paradigms, and we provide examples of actual Internet applications that do not need it although they use it. Then, we select a set of emergent applications that could be ported to OppNet thanks to our proposal.

2.1. The problem of matching destinations in OppNet

Most previous works in the OppNet field [2, 7, 25] consider one-to-one communications based on nodes identities (IDs). Usually, their main objective is to deliver messages

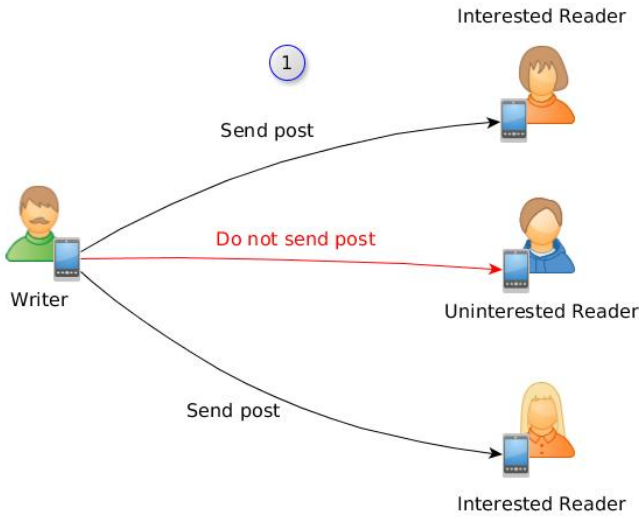
efficiently and promptly, given a destination node ID chosen by the sender, so, they assume that every sender knows the destination's ID of every message. The multicast approach only moves this problem from the sender to the receivers: they are expected to know that *someone* on the network, there are *someone* sending messages they want to receive using a certain ID. The manycast paradigm [8], designed to enable communication with an arbitrary number of group members, provides some flexibility but forces the sender to know the ID of the destination group. The publish-subscribe approach allows nodes to communicate without knowing other's ID, but it is usually based on filtering the received messages [1], or on the willingness of nodes to make their interests public [43].

These communication schemes limit the potential usages of the network. To better illustrate this, the reader may think on the services he or she uses on the Internet. The e-mail, the chat and the RSS are examples of communications directed to a destination. But there are other popular services, as blog posts (see Figure 1), the news web pages or the forums that are different in essence. A message post in a blog is not directly sent to its readers, instead, it "keeps waiting" until someone interested reads it. It is not the writer of the message who decides who is its destination, because he or she cannot know it. The readers are who, while accessing their favourite blogsite, become the destinations of the message. So, the participants perceive the illusion of a profile-cast functionality. Figure 1 illustrates how this process is perceived by the users in contrast with how this process is actually conducted.

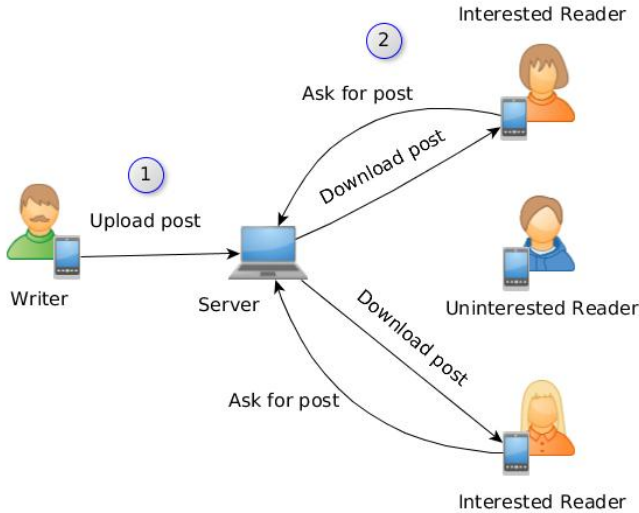
Blogs would never have become so popular if the writers would be forced to specify the ID of all the readers of their posts. Besides, the usage of third parties to build this illusion is not always feasible in OppNet, because there is no guarantee of the existence of a simultaneous end-to-end path. Therefore, we propose a profile-cast flow of communication using users' behavioral profiles as the messages' destination. Instead of explicitly expressing membership in interest groups or receiving and filtering lots of messages, our proposal allows senders to intend messages to any nodes matching a certain behavioral profile. Inasmuch, as the characteristic that differentiates most opportunistic networks from other networks is node mobility, we build nodes' behavioral profiles using their mobility habits. This decision is motivated by the tight coupling between users and their mobile devices, and the possibility of leveraging existing life-cycle patterns. Besides, this way we can also benefit from this mobility habits to route the messages towards the area where the destinations are more likely to be found.

2.2. Applications and benefits

The usage of *HabCast* (will be explained in detail in Section 5), that allows sending messages to a bunch of unidentified nodes based on their mobility profile, would allow OppNet to support a lot of new applications and services. Coming up next, we provide a set of examples



(a) The users perceive a profile-cast communication



(b) The communication is conducted through a third party

Figure 1: The blog post example. The logic flow of the application is shown in a): a direct communication (1) between the writer of the post and any number of interested readers. However, the actual communication flow is shown in b): two separate communications using a third party, the first (1) between the writer and the server, and the other (2) between the server and every interested reader.

of applications that could benefit from it⁴. This list is inspired in actual applications of the recently emerging trend of Collaborative Consumption⁵ [33], but there are no reasons to think that other types of applications (e.g. [12]) can not also benefit from using a profile-cast approach.

⁴The reader might note that this list is not intended to be exhaustive, and that the implementation details of these applications are out of the scope of this paper.

⁵For the newcomers to this field, we highly recommend a TED Talk by Rachel Botsman entitled “The case for collaborative consumption”.

- *Bla bla car*⁶ is a carsharing application designed to connect drivers and passengers that are willing to travel together and share the costs of the journey. Offers of an empty seat, or queries about it, are intended only to those people that share some mobility behaviours with the senders (e.g. because they travel daily from and to the same cities).
- *Vayable*⁷ and *Trip4Real*⁸ are tourism applications designed to get tourists and locals in touch. It tries to benefit from local’s knowledge about their environment to share it with the visitors by showing them the most genuine spots and manners. When a tourist is looking for a guide in a certain location, its messages are basically intended to people that knows and frequents the visited area.
- *Tinder*⁹ is a location-based dating discovery application. Its purpose is to connect people who live nearby based on the attractiveness they find on the photos of each one. Given that this discovery is intended to end in a physical encounter, new photos and updates are intended to people living nearby the sender.
- *Wallapop*¹⁰ is an emerging spanish application for trading second-hand goods. The application puts in contact sellers with potential buyers located nearby, aiming to avoid this way the shipping cost. Most of the products sold are small-value products that are not worth of traveling great distances. Besides, the meeting between the buyer and the seller substitutes the trust and guarantees that the major online stores provide, but the occasional sellers cannot. For these reasons, the announces of products are intended to the users that usually visit the same places that the sender.

All these applications have in common a peer-to-peer conception, where users deal directly between them without the need of a centralized entity, and the focus on the users’ whereabouts (although they all use a very simple approach: a circle of a certain radius centered on the user’s location), given that products, services, and even people, become more interesting as they live nearby or travel or visit the same places. However, as they are implemented on the Internet, they all use a central node that receives all messages and filters them based on the location of every user. But this is not the only way to make it work, we propose moving these applications (or other similar) to OppNet. For example, Figure 2 provides the schema of an OppNet version of *Tinder*, where the users perceive exactly the same operation of the Internet-based application

⁶<https://www.blablacar.com>

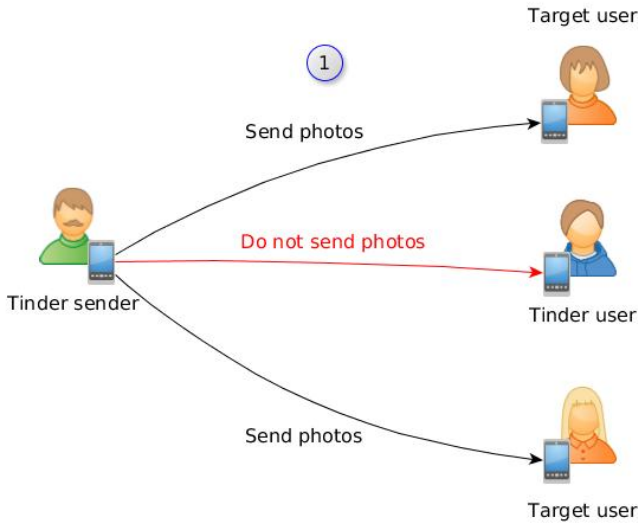
⁷<https://www.vayable.com/>

⁸<http://www.trip4real.com/>

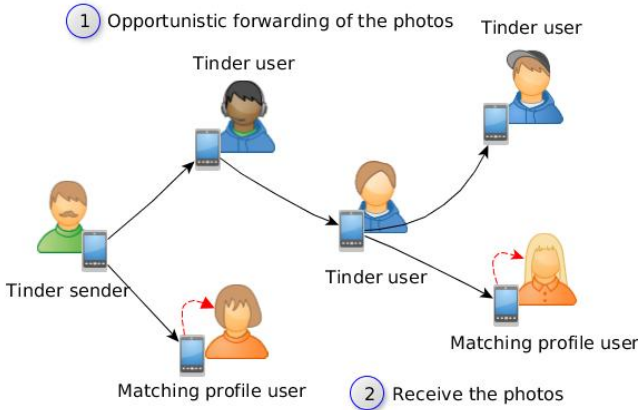
⁹<https://www.gotinder.com/>

¹⁰<http://www.wallapop.com/>

(except for the variable delays) but the real flow of communications is conducted through OppNet, without using any infrastructure.



(a) The users perceive a profile-cast communication



(b) The communication is conducted through OppNet

Figure 2: The OppNet *Tinder* example. The logic flow, shown in a), is the same of the Internet version: a direct communication (1) between the sender of the photos and the target users. However, the OppNet communication flow is shown in b): an opportunistic forwarding (1) between the nodes of the network, that act as relays, but the delivery of the photos (2) is only done to the users matching the target profile.

3. Related work

In this section, we provide the reader with a review of the related work and some conclusions about it.

The major contribution of this paper is a system with three characteristics: 1) it is profile-cast, so it allows to target messages to an unknown group of nodes that with common characteristics; 2) it provides a mechanism to geographically route the messages towards the area where their destinations are expected to be met; and 3) it learns

the behaviours of the nodes to build a model that uses to infer their near future and makes decisions based on it.

In Opportunistic Networking, it is usual to use the information collected in the past to make decisions by inferring the near future of the network. Routing protocols, as Prophet [24], use the historic of past contacts between nodes to infer how likely is for them to meet again. Also, in [6], the authors aim to improve the performance of a set of routing protocols based on the idea that two nodes that have frequently contacted at a particular time in previous cycles are likely to meet around the same time in the next cycles.

Some of the previous work, as the one presented in [30], assume existing infrastructure. *PeopleNet* delivers queries to randomly chosen nodes using the infrastructure, and uses geographic zones for queries to meet. *MobySpace* [22] uses this same idea by building behavioral profiles to find the nodes that are more likely to visit the same locations that the destination. In *MobySpace*, the locations that define the mobility space need to be defined beforehand and require the usage of some infrastructure (access points) to allow their detection. Besides, although the routing decisions are made taking into account the target profile, every node is expected to reveal its behavioral profile to all others. This is the same drawback of *SocialCast* [9], that tries to solve the same problem using a publish-suscribe approach, and relies on complex metrics computed either on the whole network or by the nodes, based on information observed (and stored) during past interactions. *CSI* [16] models the spatio-temporal behaviours of the nodes using behavioral profiles, and forwards one-to-many messages through the nodes that are more similar to the destinations. Besides, the authors realize the importance of the privacy of the nodes and present a privacy-preserving mode of operation. This way the protocol can operate in scenarios where nodes are not willing to send its behavioural profiles to other nodes when needed. But *CSI* lacks flexibility, because its spatio-temporal behaviours are modelled using a fixed vector of locations, and each location has to be decided beforehand and known by all nodes of the network.

Instead, interest-cast proposals, as [32] and [29], present a profile-cast very similar in concept to the presented work, but they are based on discovering the interests of the users of the mobile devices, instead on modelling their mobility behaviours or routing the messages towards them. The authors of this paper presented *Explore & Wait* in [5], a dynamic delivery scheme that explores the network to deliver the messages to those nodes with relatives values, such as minimum, maximum or average, on certain attributes. This is an interesting approach, however, the interests of the users are very useful to deliver the messages to interested users, but not to route the messages towards them.

Geographical multicast routing protocols route one-to-many messages using geographic information about the neighbours and the destinations. At the moment of the

sending, the nodes need to know the identity of every destination to make some calculations. For this reason, they are all based on a previous subscription phase. Moreover, as in *HGMR* [21], they usually require very dense deployments of nodes. There are proposals, as *GMR* [34] that optimize computation time, but suffer serious scalability issues, or as *HRPM* [11] that reduce the encoding of the messages but it is not energy-efficient. Geographical multicast protocols operate in a fully distributed way, but, as *GMP* [40], they tend to need some information about the identity and the position of the destinations that may be very hard to obtain in an Opportunistic Network.

There are proposals that are able to calculate on-the-fly if a message must be delivered to a certain destination. Profile-cast is the solution presented in [17]. It also uses mobility-profiles to represent the likelihood of users to visit geographic locations, it is more flexible than *CSI* because the profiles are built during a training phase and then dynamically updated from the nodes' mobility pattern, but the training phase may still be an important limitation in some scenarios. On the same line, *SANE* [27] is based on the idea that nodes with similar interests meet frequently, and uses this to route messages, but the interests have to be manually defined by each user. In [10], Daly *et al.* push the messages toward nodes with high centrality to improve the chance of delivery. Meanwhile, each node learns the structure of the network locally, and this information is used to make message forwarding decisions. This proposal deals efficiently with the routing, but lacks of a mechanism to deliver the messages to the potentially interested users.

Furthermore, the centralized analysis of user traces is not feasible at all in Opportunistic Networking, because a global view of the information about all users is never available. Nevertheless, the work presented in [15] and [20], that classify users based on their mobility preferences or periodicity, provides an insight about the usefulness of profile-cast paradigms based on mobility behaviours, and points to other potential applications such as behavior-aware advertisements or better network management.

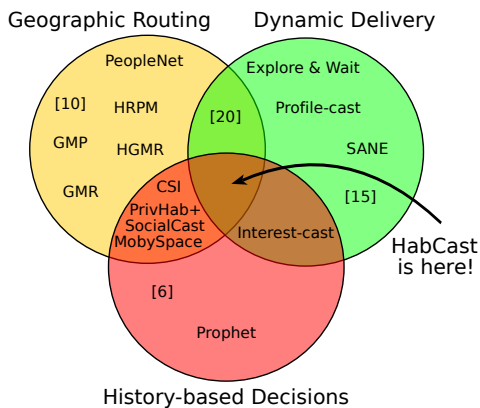


Figure 3: Venn diagram depicting the main aspects covered by every proposal of the related work. To our knowledge, *HabCast* is the very first proposal to cover the Geographic Routing and the Dynamic Delivery at the same time while taking History-based Decisions.

Figure 3 provides a visual summary of the related work. There are profile-cast systems using mobility-based profiles, but they do not offer enough flexibility because they rely on the infrastructure; use a fixed set of locations of interest; or need the users to manually define its profile. Nonetheless, the proposals that provide more flexibility build the profiles based on the users' interests instead of using their mobility, so they do not provide an efficient, geographical or not, routing. These proposals provide filtering methods to deliver the messages to the users matching the target profiles, but fail to provide a mechanism to route the messages towards them. To our knowledge, this is the very first profile-cast proposal that provides a dynamic delivery of messages and a routing mechanism that moves the messages towards their destinations. *HabCast* achieves this by using generating mobility-based profiles that uses to infer the future movements of the nodes and to make decisions both to route the messages and to deliver them to the nodes matching the target profile.

4. Previous work

The work presented in this paper significantly enhances the capabilities of our preliminary Geographic Routing Protocol for Opportunistic Networks called *PrivHab+* [38], where the focus is on routing messages towards a node by comparing the mobility profiles of every intermediate contact. In this section, we summarize the key concepts of *PrivHab+*, the protocol that lays the foundations of the present work.

4.1. The habitat: a model of nodes' whereabouts

The key concept of *PrivHab+* and *HabCast* is the **habitat**. Defined as "the area where someone is more likely to be found, based on his historical whereabouts", the habitat models the area where a node has spent most of his time in the near past, so, it considers both the locations a node visits and the visits' frequency. Considering the power of routine (most of people's life-cycles are repeated on a daily or weekly basis [6]) *PrivHab+* assumes that the best place to try to deliver a message to a node is its habitat. Henceforth, we will use nodes' habitats as a representation of their mobility profiles. Figure 4 shows an example of habitat represented by a heatmap.

In OppNet, the duration of each contact is unpredictable, usually small, so protocols must ensure the exchanges of messages are short to guarantee that every contact is exploited. Besides, the mobile devices that form the network usually do not dispose of high-end computation, storage, or even battery resources, so the overhead introduced by network protocols should be kept at the minimum possible. Due to this efficiency reasons, *PrivHab+* uses an elliptic model of habitat (Figure 5 shows the elliptic model of the same habitat of Figure 4). The elliptic model is not as precise as the heatmap, but calculating, updating and comparing it consumes far less computational and storage

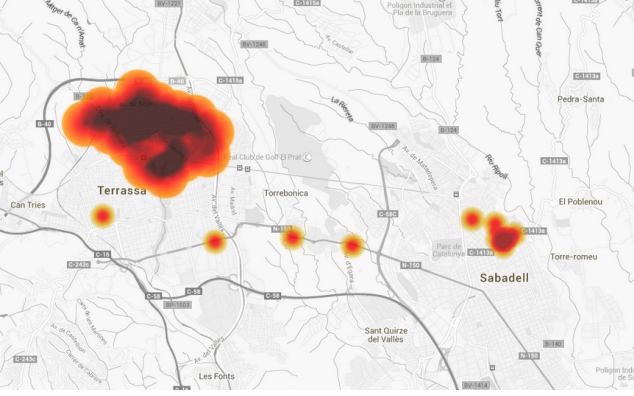


Figure 4: A real habitat of a person living in the north of Terrassa city but working in the neighbour city of Sabadell. The habitat is represented using a heatmap (the darker areas correspond to the most visited locations). The isolated small spots are due to the sampling frequency.

resources. Besides, as all applications from Section 2 use a circular distance model, using elliptical habitats would not harm their accuracy or usefulness, on the contrary, it is actually an improvement because it allows the definition of more complex mobility patterns. Finally, *PrivHab+* is able to compare elliptic habitats by using an additive homomorphic cryptosystem [42] in order to protect the privacy of the nodes.

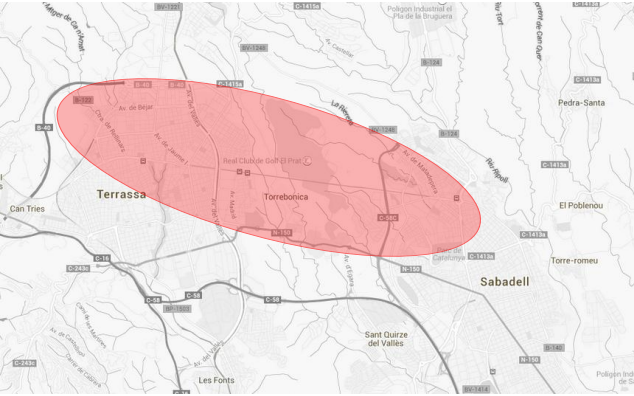


Figure 5: Elliptic model of the habitat of a person living in the north of Terrassa but working in the neighbour city of Sabadell. The considered habitat has been built using the same data as the one of Figure 4.

For the sake of simplicity, we will avoid to explain here all the details about the construction and update of the habitat (the reader shall find them in [38]). In order to understand the present work, it is enough to know that every node calculates its habitat consuming a small amount of energy and computational resources, and that they periodically update it to catch the trend of their mobility pattern.

4.2. *PrivHab+*'s routing

The operation of *PrivHab+* is based on comparing nodes using their habitats. Each time two or more nodes come

close enough to be within communication range, the routing algorithm compares the habitats to decide which node is the most suitable to carry the message towards its destination. Given that *PrivHab+* assumes that nodes are location-aware and that an approximate location of the destination can always be known or guessed by the sender of the message. *PrivHab+*'s operation is explained next:

When the message is far away from the target location, the objective of the routing algorithm is to approximate to it. In order to do this the habitat of the node carrying the message is compared with the habitat of its neighbours, and the message is relayed to the node that is more likely to bring the message closer to the target location. Figure 6 explains exactly how this comparison is resolved. Then, the objective of the routing algorithm is to deliver the message to the destination node. *PrivHab+* uses direct delivery to do this.

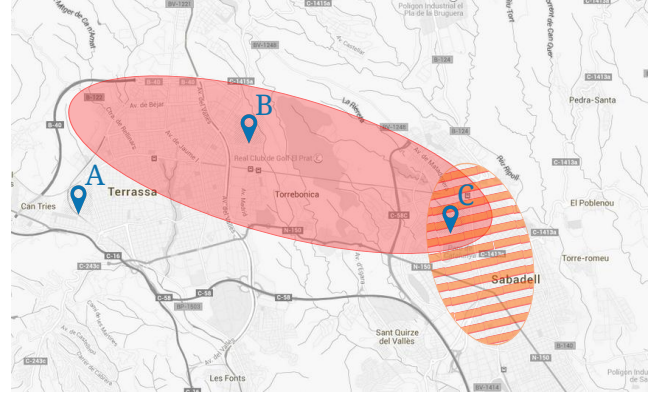


Figure 6: The three possible situations when comparing two habitats. A message intended to location **A**, outside the two habitats, is relayed to the node with the solid habitat because it is closer to **A**. A message intended to location **B** is relayed to the node with the solid habitat because it is the only habitat that encloses **B**. Finally, a message intended to location **C**, enclosed by both habitats, is relayed to the node with the striped habitat, because it is the smallest habitat containing **C**.

4.2.1. Summary

PrivHab+ is a unicast routing protocol, meaning that every message should be sent towards a destination whose ID is known beforehand. Besides, the sender of a message is supposed to use a distributed secure position service [41, 35] or an alternate communication channel to obtain the approximate location of the destination. Summarizing, to send a message using *PrivHab+*, a user needs to know “to who” it is intended, and “to where” has to be routed. As explained before, not only there are scenarios where to obtain this information may be hard or even impossible, but also this way of operation limits the possibilities of certain applications as the ones described in Section 2.

In this paper, we used the habitat's and *PrivHab+* as a foundation to build *HabCast*, a paradigm where the users send messages using a habitat to define “how are” the users that should receive it. In order to accomplish this, *HabCast* unifies all the process by using a target habitat first

for getting close to the destination, then to remain there while waiting for nodes that might be interested, and finally to deliver them the message. Table 1 summarizes the main differences between *PrivHab+* and *HabCast*, and all the details about the last one are presented on the following section.

	<i>PrivHab+</i>	<i>HabCast</i>
Type of system	Geographical Routing Protocol	Profile-cast Communication
Messages' destinations	Unicast	Multicast
Destination's identifier	ID and a guess of its location	Target Habitat
Number of phases	1 (Approx.)	3 (Approx. Floating and Delivery)
Tools used	Habitat	Habitat, Temporal and Geographic restrictions, Similarity

Table 1: Summary of the main differences between our previous work, *PrivHab+*, and *HabCast*.

5. HabCast

In this section we present *HabCast*. Firstly, we explain how to allow the sender to target a habitat instead of a destination ID when sending a message. Secondly, we describe the operation of the three involved routing phases: 1) we describe how to route the messages to bring them close to their potential receivers; 2) then, we explain how to maintain the messages in the area while waiting the appearance of one or more nodes matching the target habitat; 3) finally, we present a method to identify the nodes matching the target habitat and deliver them the messages.

5.1. Selection of potential destinations

The keystone of *HabCast* is to decouple the identifier of the destination from the message's sender. We aim to a paradigm that enables the sending of messages to a "group of nodes with a certain historic of whereabouts". For example, taking in mind some of the applications we listed in Section 2: a *Bla bla car* user may want to send a message to the people living in *City A* but working in *City B* to discover a rideshare opportunity; a *Wallapop* user may be interested in announcing a product to other users that usually hang around the town that he will visit the day after tomorrow; a *Tinder* user without a vehicle of its own may want to share his photos with other users living close and that spend almost all of their time nearby.

The tool that allows *HabCast* to represent this kind of destinations is the elliptic habitat. A node's habitat

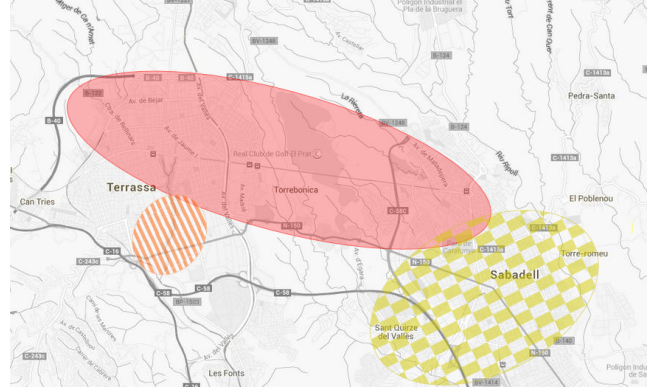


Figure 7: The users' habitats from the examples: 1) A person living in a city but working in the neighboring town has a habitat like the solid red one; 2) A person that hangs around a town and its surroundings has a habitat like the yellow squared one; 3) A person without a vehicle that does not travel usually has a habitat like the orange striped one.

models its usual whereabouts, so it is a representation of its mobility profile. Besides, as the habitat is automatically calculated and updated by the node itself, there is no need of explicitly defining or changing it. Furthermore, by defining a habitat carefully, lots of different behaviours, as the destinations of all the examples from the previous paragraph, can be characterized (see Figure 7). Therefore, *HabCast* substitutes the identifier of the destination by a target habitat, meaning that allows users to send messages "to any nodes whose habitat is similar to a certain one".

Summarizing, *HabCast* substitutes the classical "Destination Identifier" field by a "Target Habitat" one. This way, the sender of a message does not have to explicit the identifier of the destination nodes. Instead, it defines a target habitat and a tolerance index, and the message is sent to any node whose habitat is similar enough to the target habitat. Figure 8 shows the structure of the *HabCast* message's fields (note that the figure also includes other fields that will be explained in the next paragraphs).

Origin Identifier	
Target Habitat	
Expiration time	Tolerance index
Data	

Figure 8: *HabCast* message's fields.

5.2. A three-phases routing

The needs of an *HabCast*'s message vary over time, as it departs from its origin and approaches its destinations to finally reach them. Firstly, once a new message is sent,

the priority is to bring it close to their potential receivers (**approximation**); then, when the message has arrived the zone where the destinations are expected to be, it has to be maintained in the area until their potential destinations appear (**floating**); finally, when a node matching the target habitat is contacted, it is crucial to identify it in order to deliver the message (**delivery**).

Therefore, *HabCast* uses the elliptic model of habitat as destination, and its operation is separated on three different phases, each one covering one of these needs. These three phases will be amply described in the following paragraphs.

5.3. Approximation phase

During the Approximation phase, the main objective is to bring the message close to its potential receivers. On one hand, the only information available on routing time about the receivers is that their habitat will be similar to the target habitat. On the other hand, by definition, the area where every node is more likely to be found is the area modelled by their habitat. Therefore, the best place to look for nodes with a habitat similar to the target habitat is inside the target habitat itself, so, during the Approximation phase, *HabCast* moves the messages towards the target habitat, Figure 9 provides an scheme of this phase.

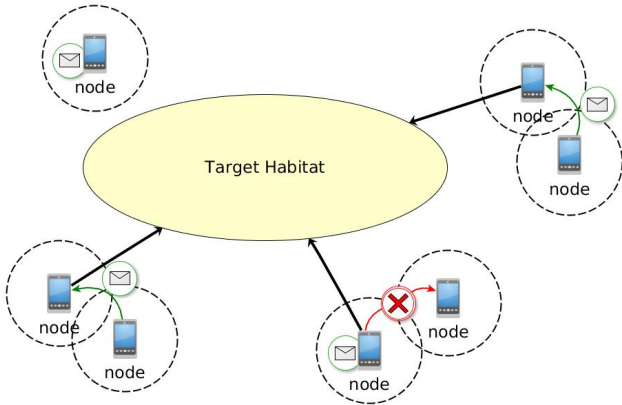


Figure 9: During the Approximation phase, every time two nodes meet outside the target habitat, their habitats are compared and the message is relayed to the other node only if it is more likely to bring the message towards the target habitat than the carrier. The straight arrows indicate the node of each pair that it is more likely to move the message closer to the target habitat.

The routing protocol used to route the messages towards the target habitat is *PrivHab+* (more details on Section 4). *PrivHab+* fits perfectly with this phase because it has been designed exactly to accomplish this purpose¹¹.

¹¹*PrivHab+* needs a target location as a guess of the destination's location in order to make routing decisions, and, in *HabCast*, the destinations are expected to be met inside the elliptic target habitat, therefore, the centre point of the target habitat is used as target location during this phase.

At every encounter, *PrivHab+* moves the message to the node that is more likely to carry it near the target habitat.

It is worth noting that, thanks to the usage of the habitats to make routing decisions, this approach captures communication opportunities not only among the “friends” nodes that frequent the same geographic locations, also with “familiar strangers” that usually take the same bus or work in the office next door, and even with “half-way carriers” that travel to somewhere between the actual location and the target habitat.

5.4. Floating phase

As soon as the message enters the area of the target habitat, the Approximation phase finishes and the Floating phase starts. In this phase, the objective is to maintain the message inside the target habitat and wait until any potential destination is contacted. Using an approach similar to the one used in [31] [23], *HabCast* tries to flood the target habitat with copies of the message by using a lifespan controlled adaptation of the Epidemic routing [39].

Concretely, *HabCast* applies two restrictions to control the lifespan of the messages during the Floating phase:

1. **Geographic restriction:** the flooding of the messages is restricted to their target habitat. When a node carrying a message establishes contact with a neighbour inside the area of its target habitat, a copy of the message is sent using the Epidemic routing. However, at the moment that a node leaves the area while carrying a message, it stops the flooding and switches again to the Approximation phase to bring back the message by using *PrivHab+*.
2. **Temporal restriction:** the lifetime of the messages is limited by the “Expiration time” field. Nodes periodically check if this time has arrived, and all copies of the message are deleted when this happens. With this measure, *HabCast* avoids that old messages remain in the area indefinitely, wasting resources when they are not useful anymore.

Summarizing, with the geographic restriction, the flooding of the messages is focused on the area of interest defined by the elliptic target habitat, and the copies of messages that leave the area try to return to it. On the other hand, with the temporal restriction we ensure that this area does not remain flooded by copies after the expiration of the message. Figure 10 provides an scheme of this phase.

5.5. Delivery phase

When a node receives a message, it has to deliver the message to the upper layers if the node is the message's destination. In IP, for example, this is done if the IP address of the destination message's field matches with the IP address of the node (its identifier). In *HabCast* there are no destination ID field, so, the elliptic habitat of the

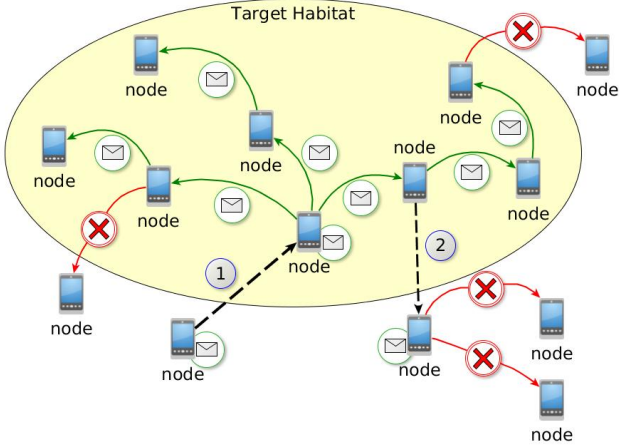


Figure 10: When a node carrying a message enters (1) the target habitat, it starts the Floating phase. During this phase, the node floods the network with copies of the message that are sent to all other nodes inside the area, but not to those that are located outside it. Finally, when a node carrying a message leaves (2) the target habitat, it stops sending copies to every node it meets and returns to the Approximation phase again.

node has to be compared with the elliptic target habitat, and the node will be a message's destination if and only if they match.

Therefore, the objective of the Delivery phase is to ensure that every node with a habitat similar enough to the target habitat recognizes itself as a destination of the message and delivers the message to the upper layers.

The following paragraphs provide the tools needed to compare the habitats and to decide if a node is a message's destination.

5.5.1. Delivery phase: Elliptic habitat similarity

In order to make the decision of delivering the message, the node has to decide if its own habitat is *similar enough* to the target habitat. Forwarding, we will define how *HabCast* defines the elliptic habitats' *similarity*.

First of all, the reader must note that the *geometric similarity* of ellipsoids can not be used by *HabCast*, because this definition only considers the shape¹² of the ellipses and not their size or location. Figure 11 shows three habitats that are *geometrically similar* between them, but that can not be considered *similar* for *HabCast*'s purposes.

In order to identify if a habitat (H_1) matches the target habitat (H_2), we need to define a new *elliptic habitat similarity* metric. This metric should increase as the area that is inside both H_1 and H_2 (from now on: $H_1 \cap H_2$) increases. At the same time, the habitat similarity should decrease as the area that is contained by H_1 but not by H_2 (from now on: $H_1 \not\subset H_2$), and *viceversa* ($H_2 \not\subset H_1$) increases.



Figure 11: The three elliptic habitats (stripped, squared and solid) shown in the map are *geometrically similar*, because they share the same shape (the relation between the minor axis and the major axis). However, for *HabCast*'s purposes, we can not consider them as similar habitats, because they have very different sizes, orientations, and do not cover the same area.

Equation 1 defines habitat similarity and Figure 12 illustrates the different components of the equation.

$$(H_1 \cap H_2) - (H_1 \not\subset H_2) - (H_2 \not\subset H_1) \quad (1)$$

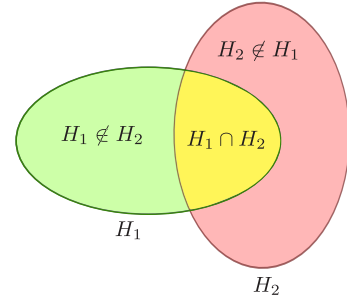


Figure 12: Venn diagram that illustrates the different parts of the habitat similarity equation. Both H_1 and H_2 are complete ellipses. Differences $H_1 \not\subset H_2$ (green) and $H_2 \not\subset H_1$ (red) are the area of H_1 not contained by H_2 and viceversa. Finally, $H_1 \cap H_2$ (yellow) is the common area of the two habitats.

Given that the area inside boths habitats is added and the area in which the habitats differ is substracted, by definition, a $Similarity(H_1, H_2) > 0$ means that the common area both H_1 and H_2 cover is greater than the area in which H_1 and H_2 differ. Figures 12 and 13 provide two examples of habitat similarity. The habitats shown in Figure 12 have negative similarity, so, they are considered *not similar*, while the two of Figure 13 can be considered *similar* because they have positive similarity.

Calculating habitat similarity using the elliptic model of habitat requires the calculation of the habitat difference. Equation 2 shows how to calculate the area in which two habitats differ:

$$H_A \not\subset H_B = Area(H_A) - (H_A \cap H_B) \quad (2)$$

Therefore, by starting with Equation 1 and applying

¹²Two geometrical objects are called similar if one can be obtained from the other by uniformly scaling, possibly with additional translation, rotation and reflection.

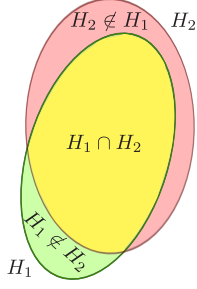


Figure 13: The two habitats are *similar* because their common area ($H_1 \cap H_2$) is bigger than the different area ($H_1 \setminus H_2$ and $H_2 \setminus H_1$) they cover

Equation 2 we obtain Equation 3, the formula of the elliptic habitat similarity.

$$\begin{aligned} & (H_1 \cap H_2) - (H_1 \setminus H_2) - (H_2 \setminus H_1) \\ & (H_1 \cap H_2) - (H_1 - (H_1 \cap H_2)) - (H_2 - (H_1 \cap H_2)) \\ & 3 \cdot (H_1 \cap H_2) - H_1 - H_2 \end{aligned} \quad (3)$$

The calculation of the areas H_1 and H_2 is quick and easy¹³. However, the calculation of the habitats' intersection ($H_1 \cap H_2$) is more complex, it can be calculated by approximating the ellipse curves with polygons [13], based on the Gauss-Green formula to determine segment areas [18], or using a probabilistic method as Monte Carlo¹⁴ to obtain an approximate result consuming minimum computational resources (in Section 6 we provide a comparison of the overhead and error ratio introduced by each method).

5.5.2. Delivery phase: matching the target habitat

During the delivery phase, a node carrying a message has to decide if its habitat matches the message's target habitat. To decide this, the node uses the previously defined metric of habitat similarity between H_O and H_T , where H_O is its own habitat and H_T is the target habitat. This decision is made by comparing how much area H_O and H_T have in common and how much area they differ.

Hence, the nodes decide if their habitat match the target habitat by calculating the *indulgent habitat similarity* using Equation 4. If the indulgent habitat similarity is positive, then the node is a message's destination and has to deliver it to the upper layers.

$$T \cdot (H_O \cap H_T) - (H_O \setminus H_T) - (H_O \setminus H_T) \quad (4)$$

HabCast lets message's sender to tweak the similarity calculation by using the parameter "Tolerance index" ($T \in$

$[-1, 1]$). This T is a measure of how much greater can be the area where H_O and H_T differ than their common area, to still consider that the two habitats are similar. For example, a $T = 0.2$ means that H_O matches H_T even if the area they differ is a 20% greater than their intersection, and a $T = -0.3$ means that H_O only matches H_T if their intersection is at least a 30% greater than the area they differ.

Finally, Figure 14 summarizes the operation of the three *HabCast*'s phases.

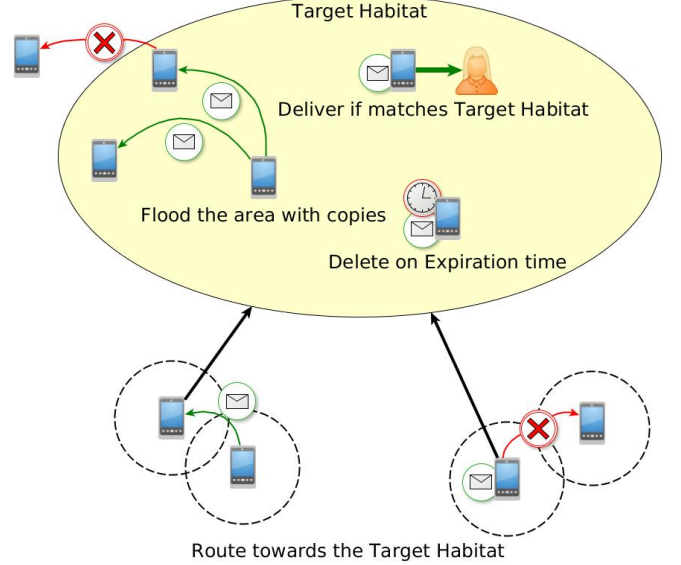


Figure 14: Summary of the operation of *HabCast*. Firstly, messages are routed to the nodes that are more likely to bring the messages closer to the target habitat. Once inside this area, the copies are sent to every other node met in the target habitat. When a node's habitat matches the target habitat, the message is delivered to the upper layers. Finally, the copies are deleted when the expiration time is reached.

6. Proof-of-concept implementation

In this section we present some details about the proof-of-concept we have implemented. Then, we provide measurements of the computational and communication overhead introduced by the presented protocol.

6.1. Implementation details

We have deployed a proof-of-concept implementation written in C of the presented protocol on two different sets of devices: three Raspberry Pi boards¹⁵, and two desktop

¹³The area of an ellipse is defined by $\pi \cdot a \cdot b$ where a is the semi-minor axis and b is the semi-major axis.

¹⁴Monte Carlo [28] methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results [14]. In this case, to calculate ($H_1 \cap H_2$) the method can be applied by selecting a random set of points, all contained by H_1 , and testing if they are also contained by H_2 . Finally, the result is calculated using: $(H_1 \cap H_2) = \text{Area}(H_1) \cdot \frac{\text{amount of positive tests}}{\text{amount of tested points}}$.

¹⁵Raspberry Pi Broadcom BCM2835 SoC full HD, 700MHz Low Power ARM1176JZ-F, 512MB SDRAM, 512MB SD with Raspbian, equipped with a Wi-Fi Wireless Adapter (802.11n up to 150Mbps), a GPS receiver NL-302U (baud rate: 4800 bauds) and a dual output 5000mAh battery.

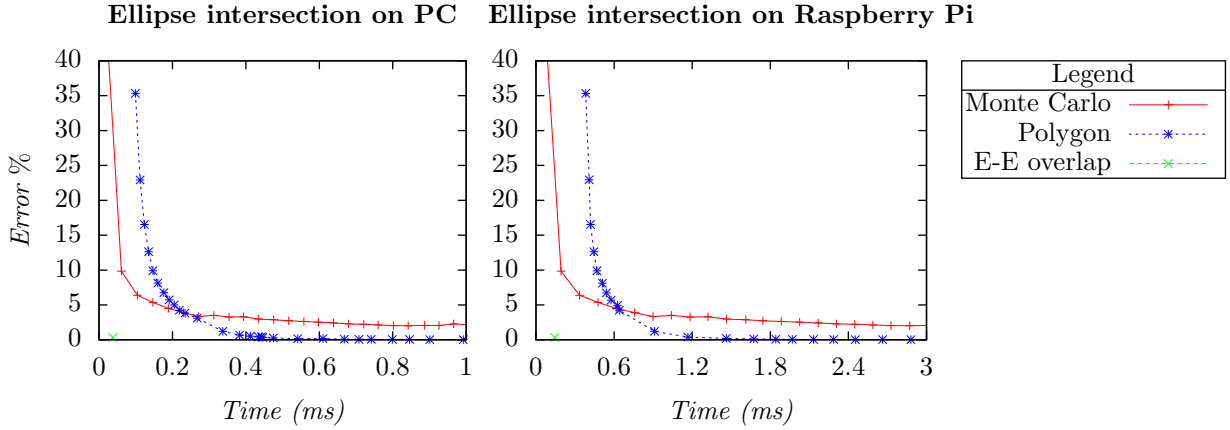


Figure 15: Time required to calculate the intersection of ellipses using three different approaches, plotted against the relative error of the result. The obtained results are very similar on both devices. However, the Raspberry needs approximately the triple time to perform every operation.

PCs¹⁶. The objective of the implementation is test the proposal, and to obtain a measure of the overhead that *HabCast* adds to every transaction.

Although we are aware that numbers vary with the platforms, we have chosen the Raspberry Pi boards because they are very cheap low-end devices, and we plan to use them to deploy a cheap prototype network to run field experiments in a near future. The PCs have been chosen as representatives of future high-end mobile devices.

All interaction with the GPS (Global Positioning System) is performed through the GPSD¹⁷ library. Cryptographic operations, including Paillier’s [42], have been implemented using OpenSSL. Measurements of time have been done using the standard C library.

6.2. Experiments and results

We have established an opportunistic network using the chosen devices and we have used the proof-of-concept implementation to send 500 messages of sizes between 1KB and 16MB. We have repeated the tests five times, using Paillier’s length keys of 512, 1024 and 2048 bits. We have measured the average time needed during each one of *HabCast*’s three phases.

The amount of time consumed to make the routing decision during the approximation phase is shown in Table 2. *HabCast*’s execution time during the approximation phase depends on the key length used. When using keys of 512 bits, a low-end device can make the routing decision in half a second. The execution time increases to 3.4 seconds when using keys of 1024 bits. The usage of keys of 2048 bits or more in low-end devices is discouraged because of

Device	Key length (bits)	Time (ms)	Overhead 1024KB (%)	Overhead 4MB (%)
Raspberry Pi	512	524.4	107.43	27.08
	1024	3,495.19	712.95	180.55
	2048	25,601.53	5,247.86	1,322.50
i5 Laptop	512	32.23	6.74	1.66
	1024	89.51	18.37	4.62
	2048	402.91	85.52	20.81

Table 2: Execution time of *HabCast* to make a routing decision during the approximation phase, in both devices, the Raspberry Pi and the desktop PC, using different key lengths. The overhead is calculated as the extra amount of time needed to send a message of 1024KB or 4MB.

the high overhead times they produce. In a high-end processor, the overhead introduced is less than half a second even when using extra-large keys of 2048 bits.

Most of the time spent on the delivery phase is used to calculate the intersection of ellipses. For this reason, we have compared the time required to calculate it with three different approaches: approximating the ellipses with polygons, using the Monte Carlo method and using the ellipse-ellipse overlap algorithm presented in [18] (we used our implementation of the two first approaches, and the code provided by the authors¹⁸ for the last one).

Figure 15 shows the results of this comparison in terms of execution time and relative error. The execution time of the Monte Carlo method has been obtained using 200 to 5,000 random samples, the execution time of the polygon approximation has been obtained using polygons of 3 to 250 vertices, the ellipse-ellipse overlap algorithm complexity is constant, so it is depicted with a single dot instead of with a line. Regarding the Monte Carlo and the polygon approximation, they intersect in both devices when

¹⁶Intel Core i7-4770: 3,40 GHz, 16GB RAM, WiFi 802.11 b/g/n, with Gentoo Linux x86_64 v4.4.4-gentoo, equipped with a GPS receiver NL-302U (baud rate: 4800 bauds).

¹⁷<http://www.catb.org/gpsd>

¹⁸<https://github.com/chraibi/EEOver>

using around 1000 – 1100 samples and 12 – 13 vertices, under these numbers, the Monte Carlo approach is faster, above them, the polygon approximation is more precise. Nevertheless, the results show that the ellipse-ellipse overlap algorithm is the fastest option, and it is also the most precise one. Therefore, we recommend to use it in any scenario.

6.3. Case study: simulated scenario

In order to provide a realistic measure of *HabCast*'s overhead on the scenario that will be used to carry on the simulations (more details about the scenario on Section 7), we have chosen to use low-end mobile devices, a key lenght of 1024, the ellipse-ellipse overlap algorithm to calculate the ellipse intersection and a message size of 512 KB in average. Table 3 provides the break up of *HabCast*'s overhead during the three phases using these settings.

Time spent (ms)	Approx.	Floating	Delivery
Communication	243.41	243.41	0
Computation	3,495.19	0	0.14

Table 3: Detailed communicational and computational *HabCast*'s overhead during each phase. The settings have been chosen in order to match the scenario studied on the simulations.

The most costly phase is the approximation, that requires 3.5 s of computation to decide if a message has to be forwarded (243 ms more) or not. Then, during the floating phase, the computation time needed is negligible (decide if the node is inside or outside the target habitat) and all the time is spent by forwarding copies of the message to the neighbours. Finally, the delivery phase does not require any communication because the node has already a copy of the message, and it needs 0.14 ms to decide if the node is a destination or just another forwarder.

7. Feasibility study

In this section, we explain the scenario we have chosen to study *HabCast*'s feasibility, and how we have obtained the data needed to model and simulate it. Afterwards, we provide the obtained results, we evaluate *HabCast*'s feasibility and we compare it's perfomance and characteristics with some other profile-cast approaches based on well-known OppNet routing algorithms. As explained in Section 3, there are no other profile-cast proposals that include the automatic calculation of profiles and its usage to geo-route the messages towards matching profile nodes, so, this Section is not intended to provide a comparison with other protocols but to demonstrate that *HabCast* allows the deployment of an OppNet version of one of Section's 2 applications.

7.1. Proposed scenario

In order to study the feasibility of our proposal, we have designed a scenario based on the *Bla bla car* application. On the basis of the ideas we have presented in Section 1, we aim to move this application into an OppNet scenario without using infrastructure. Therefore, users are supposed to carry a small mobile device and all the communication is done by exchanging messages between them at every encounter.

In order to design a realistic scenario, we have located it in a concrete geographic region and we have used actual demographic data. The chosen region was Catalonia. Firstly, we selected all catalan cities with a population above 90,000, according to Catalonia's Official Statistics Institute¹⁹. Then, in order to model the movement patters of the users, we have gathered the data about all *Bla bla car* travels between every pair of these cities during the months of November 2015 and December 2015 from the official website. From the analysis of this data, we have learned four items that we used to build our model:

- *Bla bla car* users tend to do return travels. Given any day and any pair of cities $city_A$ and $city_B$, the amount of travels from $city_A$ to $city_B$ is approximately the same the amount of travels from $city_B$ to $city_A$, and the users that do these travels are almost the same users.
- Travels are split in two different time slots. Approximately half the travels are done during the morning, and the other half is done during the afternoon.
- The amount of daily travels between these cities remained very stable around 500 during the two studied months. Concretely, the mean is 502 travels per day.
- The likeliness of a travel's destination depends on where the travel departs. Table 4 provides the percentage of travels starting in the row city that are destined to the column city.

7.2. Simulation details

In our model of this scenario, we have used a Catalonia's map (Figure 16) and 250 nodes representing 250 *Bla bla car* users that carry a small portable device like the Raspberry Pi from Section 6.

We have used the information from the previous paragraphs to implement a mobility pattern that takes into account their home city (randomly chosen considering the population of every city) and a destination city (randomly chosen using the probabilities from Table 4). Every node travels from one city to the other twice a day at 100 ± 20 km/h and roams, following a random waypoint movement model, around the center of the city at 3 ± 1 km/h during the rest of the day, the amount of simulated time is one week.

¹⁹<http://www.idescat.cat/pub/?id=aec&n=250&lang=es>

	Barcelona	Lleida	Hospitalet	Badalona	Terrassa	Sabadell	Girona	Tarragona	Mataró	Santa Coloma	Reus
Barcelona	N/A	64.96	0	0	1.71	1.71	1.71	20.51	2.56	0	6,84
Lleida	22.61	N/A	22.32	21.74	2.61	2.90	1.45	2.03	0.29	22.32	1,74
Hospitalet	0	67.27	N/A	0	0	0	1.82	22.73	1.82	0	6,36
Badalona	0	71.15	0	N/A	0	0	1.92	20.19	0	0	6,73
Terrassa	0	66.67	13.33	0	N/A	6.67	0	13.33	0	0	0,00
Sabadell	4.55	50.00	4.55	0	9.09	N/A	0	27.27	0	0	4,55
Girona	11.11	18.52	11.11	11.11	0	0	N/A	18.52	0	11.11	18,52
Tarragona	20.39	6.80	22.33	18.45	1.94	4.85	2.91	N/A	0	20.39	1,94
Mataró	37.50	25.00	25.00	0	0	0	0	0	N/A	12.50	0
Santa Coloma	0	68.81	0	0	0	0	1.83	21.10	0.92	N/A	7,34
Reus	18.18	13.64	18.18	15.91	0	2.27	6.82	6.82	0	18.18	N/A

Table 4: Percentage of travels departing from the row city with the column city as a destination. The values have been obtained from the analysis of the data of all *Bla bla car* travels between these cities during two months (Nov-Dec 2015).



Figure 16: The region where the proposed scenario is located. We have placed on the map the 11 catalan cities with a population above 90,000, according to Catalonia’s Official Statistics Institute.

Nodes have a buffer that lets them carry up to 100 messages simultaneously, and a wireless interface featuring a communication range of 30 meters. Every 35.7 minutes²⁰, one node of the network randomly picks up a pair of cities and sends a profile-cast message destined to any node whose profile indicates that it travels between these cities (more details about profiles will be provided below).

In order to provide context to the study of *HabCast*’s feasibility, and given that, as seen in Section 3 there are not other proposals that can be directly compared with it, we

have modified two well-known OppNet routing protocols as Epidemic [36] and Prophet [24] to adapt them to a profile-cast operation. The main condition to select these two protocols was that the resulting profile-cast version could operate in a completely automatic mode, without the need of any interaction with the users to define their profile (e.g. as in *SANE* [27]), nor the deployment of any infrastructure to locate or define the places of interest (e.g. as in *MobySpace* [22]), and that it does not require nodes to share their private profiles with the rest of the network (e.g. as in *CSI* [16]). Therefore, we have studied the operation of the network using the following different types of routing:

1. ***HabCast***. Nodes calculate their habitat and use *HabCast* to make routing decisions. The simulator adds the computational and communication overhead from Table 3 in Section 6 to each transaction. A target habitat is used as messages’ destination, the delivery of the messages is done as described in Section 5.
2. **Epidemic-like profile-cast**. Nodes exchange copies of all messages they do not hold at every encounter. A target route identifier *Starting city - Destination city* is used as messages’ destination, the delivery of the messages is done to any node with a pair *Home city - Destination city* (nodes do know this) that matches the target identifier.
3. **Prophet-like profile-cast**. The Prophet routing algorithm with one modification: the probabilities of delivery are not calculated for every node ID, but for every route identifier *Starting city - Destination city*. This target route identifier is used as message’s destination, the delivery of the messages is done to

²⁰There are 40.33 seat reservations per day, on average, for travels between the selected cities. Data obtained from the analysis of the amount of “complete cars” from the official website during Nov-Dec 2015.

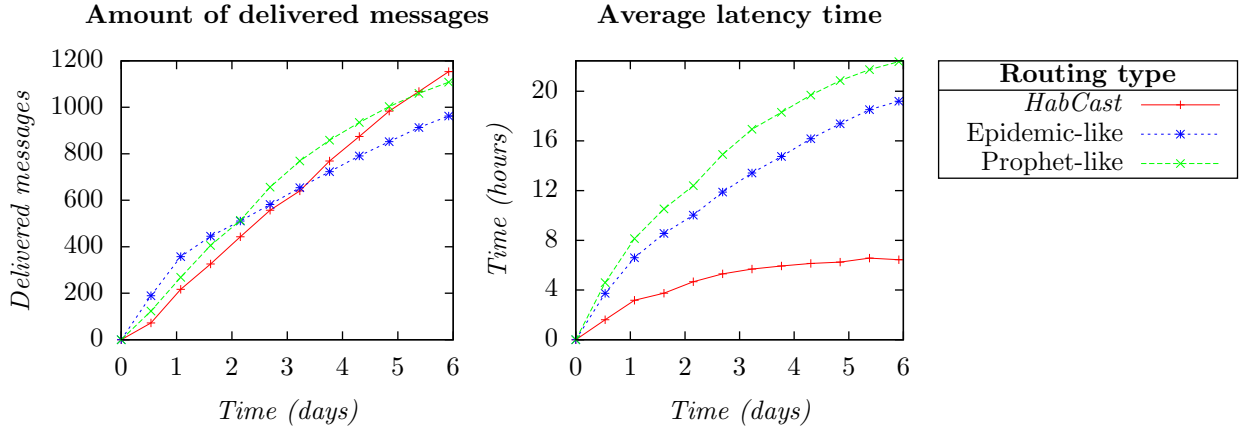


Figure 17: Obtained results in terms of delivered messages and latency. Although the three routing protocols deliver a similar amount of messages, since the first moment of the simulation, *HabCast* delivers them between 4 and 10 hours sooner.

any node with a pair *Home city - Destination city* (nodes do know this) that matches the target route identifier.

All simulations have been performed using *The Opportunistic Network Simulator* (The ONE) [19], and have been repeated twenty times using different random seeds, then, the average results of the twenty repetitions have been calculated and will be presented in the following paragraphs.

7.3. Simulation results

In first place, we studied the amount of delivered messages, because it measures how many nodes matching the target habitat have received the message and provides a measure of the system's success. Figure 17 shows the amount of delivered messages and the average latency over time for the three studied routing protocols.

The amount of delivered messages is similar between the three studied approaches. We consider this an *HabCast*'s achievement, because *HabCast* is able to deliver almost the same amount of messages, but it does it faster and consuming less resources, as it will be seen on the next paragraphs.

The study of the average latency time shows that, using *HabCast*, nodes receive the messages around 6 hours before they were created. It is up to every application's user to decide if this latency is acceptable, but, taking into account that the majority of *Bla bla car* travels are announced at least 72 hours before departure, we consider that an average latency of 6 hours does not endanger the operation of the network nor the feasibility of the application.

Figure 18 shows the cumulative probability of delay, and illustrates the comparison between how quick the three routing types deliver the messages. There is a huge gap between the amount of messages *HabCast* or the other two routing types deliver in few hours. As shown, *HabCast*

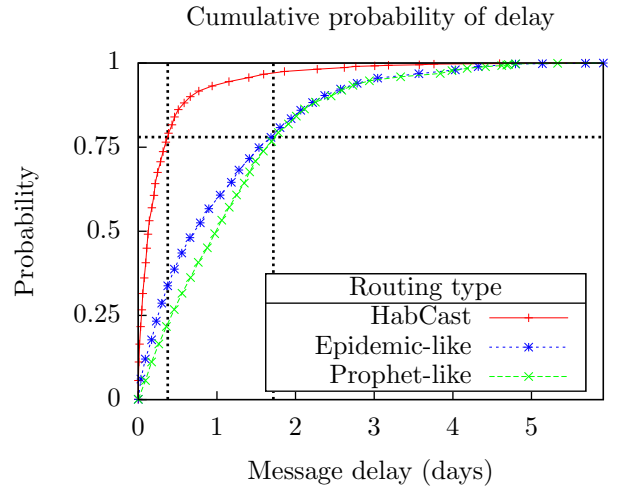


Figure 18: Cumulative probability of delay, in days, related to the type of routing used. From the messages delivered using *HabCast*, the 78% have been delivered in less than 8 hours. From the messages delivered using Epidemic-like and Prophet-like, only the 25% have been delivered in this amount of time, and they require up to 42 hours to deliver the 78%.

greatly outperforms Epidemic-like²¹ and Prophet-like because it delivers the 78% of the messages in less than 8 hours, while Epidemic-like and Prophet-like deliver the 33% and 25%, respectively, in the same amount of time. Moreover, they take 42 hours to deliver the 78% of messages.

The delivery ratio is a very used metric in OppNet to study the performance of routing protocols. Due to the nature of our proposal, we propose a similar metric based on the delivery ratio, but much more accurate for evaluating this kind of delivery schemes. *HabCast*'s messages are

²¹Note that Epidemic only provides an upper bound in theoretical scenarios with unlimited buffer space, which is not the case under study.

not intended to a concrete destination, but to any node matching the target habitat, and there is no way to know how many nodes match it. Nevertheless, this is a information that we can obtain from simulations. Hence, we implemented an oracle entity that, every time a message is sent, analyzes all nodes of the network to find how many nodes match its target profile. We used this information to obtain the data needed to build Figure 19 and to calculate the delivery ratio.

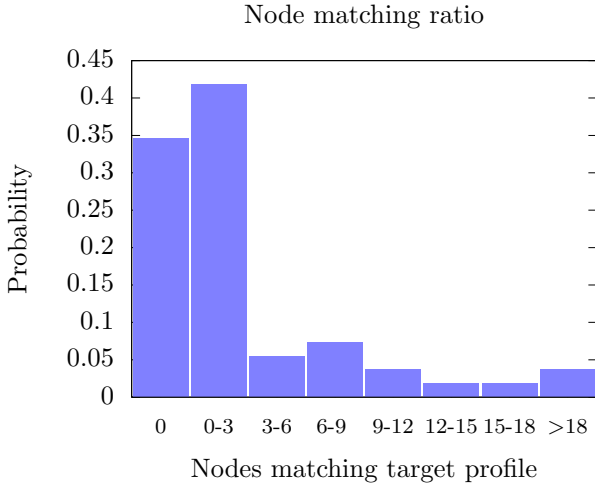


Figure 19: Probability of the existence of a certain amount of nodes matching every target profile during the simulation. One third of the messages target profiles that do not match with any node, more than 40% target profiles that match with three or less nodes. The rest of the messages target profiles that match any number of nodes with a distribution close to uniform.

By knowing the amount of nodes that the target profile of every message sent (this is, the amount of potential destinations of the messages) we calculated the adjusted delivery ratio, defined as the ratio between the amount of copies delivered and the amount of nodes matching the target profile of every message. The results show that *HabCast* has delivered, on average, a message to 1,160.1 of the 1,282.9 potential destinations, obtaining a 90.42% delivery ratio, higher than the 86.31% obtained by the Prophet-like protocol, and than the 76.05% obtained by the Epidemic-like one. Figure 20 shows the delivery ratio obtained by *HabCast* regarding the amount of potential destinations of the messages. Note that the delivery ratio increases with the amount of nodes that match the target habitat. The reason is that *HabCast*'s spreading of messages is performed during the floating phase, which is done at the area where is more likely to find a destination node, so, it is very convenient to encounter a higher amount of nodes in this area in order to reach as destinations as possible.

Finally, Table 5 provides a comparative between the three routing types considering the ratio of relays per delivery done, the amount of aborted relays, the hop-count of the delivered messages and the distance between the lo-

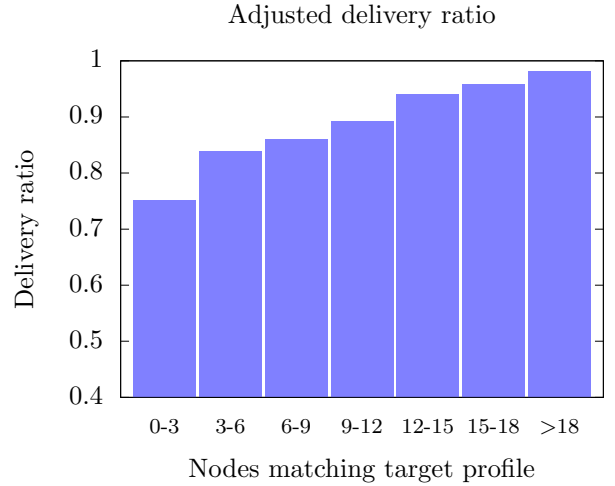


Figure 20: The ratio between the amount of delivered messages and the amount of nodes that match the target profile. With *HabCast*, the amount of potential receivers of a message that do not receive it decreases as the number of nodes matching its target profile grows. Therefore, it is more likely to not deliver a message to a node that matches the target profile if there are few nodes matching it.

cation where messages are created and the locations where they are delivered. The first column shows that *HabCast* performs an efficient filtering during routing, this way, it wastes fewer resources because it restricts the flooding of messages to the area where the destinations are expected to be found (during the floating phase). This is an important advantage of *HabCast* against other proposals that require the nodes to do this filtering of messages on their own. Therefore, as the amount of messages filling the buffers and flooding the network is smaller, nodes can exploit better every opportunistic encounter because they have time to process all messages. The high amount of aborted relays of both Epidemic-like and Prophet-like indicates that there are messages still unprocessed when the contacts end, and this rarely happen when using *HabCast*, whose amount of aborted transactions is 25 times lower. Besides, the amount of hops performed by the delivered messages indicates that both *HabCast* and Prophet-like are able to find similar routes, although Prophet-like needs to flood network to find them. On the last column, the similar delivery distance of the three types of routing simply reinforces what Figure 17 showed: not only the three types of routing deliver almost the same amount of messages, they also deliver them to almost the same nodes.

Summarizing, the results obtained demonstrate the feasibility of the proposal. Even in a big scenario with only 250 nodes, the average latency and the amount of delivered messages achieved by *HabCast* are good enough to implement an actual infrastructure-dependant application in a fully OppNet mode. As *HabCast* has proved to be efficient and to not consume a high amount of resources, it makes sense to enlarge the network by adding nodes that use the application but do not announce their travels (be-

Protocol	Relays per delivery	Aborted relays	Hop-count	Delivery distance
<i>HabCast</i>	24.20	1,034.75	5.15	73.61 km
Epidemic-like	321.83	27,576.4	12.38	73.24 km
Prophet-like	228.29	26,090.8	6.80	71.47 km

Table 5: Obtained results: amount of relays needed to perform every delivery, amount of aborted relays, number of hops performed by the delivered messages and distance travelled by the delivered messages. Although the three routing protocols deliver a similar amount of messages, *HabCast* does it using fewer network's and node's resources.

cause they only book seats), or even nodes that use other *HabCast*-based applications (note that these are less interested users that would not participate if they are required to provide a big buffer or a significant battery consumption, so they would value the efficiency more than *hardcore* users). This way, the overall results will improve and other applications that require of lower response times will also become feasible.

8. Conclusions and future work

We have presented *HabCast*, a profile-cast paradigm of communication in which membership in interest groups is not explicitly expressed by users, it is rather inferred based on their past behavioral profiles, and where the users' behavioral profiles are used as the destination of the messages. *HabCast* takes advantage of a concept from our previous proposal *PrivHab+*, the habitat, to model the usual whereabouts of the nodes of the network and build their private mobility profile. *HabCast*'s completely automatic setup provides flexibility, because it is immune to users' oversights and responds well to changes in the usual patterns.

HabCast has been designed to operate in three phases. In every phase, the habitat plays a central role: 1) a tool used to approach the destinations during the Approximation phase; 2) the area where the destinations can be found during the Floating phase; and 3) the definition of the destinations during the Delivery phase. The usage of the habitat during the whole process makes *HabCast* recommendable to use in scenarios where nodes are related, directly or indirectly, to a person, because people usually repeat their life-cycles.

We also have described a set of real applications that could benefit from *HabCast*'s innovations to move from the Internet to an opportunistic network and we have studied *HabCast*'s performance under the scope of one of these applications. First, we have developed a proof-of-concept implementation to measure *HabCast*'s performance in high-end devices, but also on small devices. Both the computation and the communication overhead introduced by *HabCast* is proven to be affordable and to not degrade the performance of the network. Then, simulations based on a realistic *Bla bla car* scenario have shown that *HabCast*'s performance is good enough to make feasible an OppNet

version of these applications, or any other that operates in a peer-to-peer way and focuses on users' whereabouts. Besides, *HabCast* has proven very efficient in terms of consumed network resources.

As future lines of research, we plan to study the best way a message sender can define the target habitat to maximize the amount of interested users that receive it, to improve *HabCast* to make it compatible with more complex models of habitat, and to compare the habitat with other automatically generated types of profiles. We also plan to model the same scenario using data gathered from other real applications to study *HabCast*'s performance and feasibility in other contexts, and to deploy a network prototype using Raspberry Pi devices to run field experiments using *HabCast*. Finally, we will continue searching applications that could benefit from this novel profile-cast paradigm to operate without infrastructure.

9. Acknowledgments

This work has been partially funded by the Ministry of Economy and Competitiveness of Spain, under the reference project TIN2014-55243-P, by the Catalan Government under the reference project 2014-SGR-691 and by the Autonomous University of Barcelona under the reference number 472-03-01/2012.

References

- [1] A dtn routing scheme based on publish/subscribe model. *International Journal of Informatics Society*, 6(2):49 – 56, 2014-11.
- [2] Abey Abraham and Jebapriya S. Article: Routing strategies in delay tolerant networks: A survey. *International Journal of Computer Applications*, 42(19):44–48, March 2012. Full text available.
- [3] Archie Bland. Firechat—the messaging app that's powering the hong kong protests. *The Guardian*, 29, 2014.
- [4] Carlos Borrego, Sergio Castillo, and Sergi Robles. Striving for sensing: Taming your mobile code to share a robot sensor network. *Information Sciences*, (0), 2014.
- [5] Carlos Borrego, Adrián Sánchez-Carmona, Zhiyuan Li, and Sergi Robles. Explore and wait: A composite routing-delivery scheme for relative profile-casting in opportunistic networks. *Computer Networks*, 123:51 – 63, 2017.
- [6] Eyuphan Bulut, Sahin Cem Geyik, and Boleslaw K. Szymanski. Utilizing correlated node mobility for efficient dtn routing. *Pervasive and Mobile Computing*, 13:150 – 163, 2014.
- [7] Yue Cao and Zhili Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys Tutorials, IEEE*, 15(2):654–677, 2013.
- [8] Casey Carter, Seung Yi, Prashant Ratanachandani, and Robin Kravets. Manycast: Exploring the space between anycast and multicast in ad hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 273–285. ACM, 2003.
- [9] Paolo Costa, Cecilia Mascolo, Mirco Musolesi, and Gian Pietro Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 26(5):748–760, 2008.
- [10] Elizabeth M Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM, 2007.

- [11] S. M. Das, H. Pucha, and Y. C. Hu. Distributed hashing for scalable multicast in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(3):347–362, March 2008.
- [12] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39(1):319–333, 2014. cited By 85.
- [13] Peter M Gruber. Asymptotic estimates for best and stepwise approximation of convex bodies ii. In *Forum Mathematicum*, volume 5, pages 521–538, 1993.
- [14] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [15] Wei-jen Hsu, Debojyoti Dutta, and Ahmed Helmy. Mining behavioral groups in large wireless lans. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 338–341. ACM, 2007.
- [16] Wei-jen Hsu, Debojyoti Dutta, and Ahmed Helmy. CSI: A paradigm for behavior-oriented delivery services in mobile human networks. *CoRR*, abs/0807.1153, 2008.
- [17] Wei-jen Hsu, Debojyoti Dutta, and Ahmed Helmy. Profile-cast: Behavior-aware mobile networking. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 3033–3038. IEEE, 2008.
- [18] Gary B. Hughes and Mohcine Chraïbi. Calculating ellipse overlap areas. *Computing and Visualization in Science*, 15(5):291–301, 2014.
- [19] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.
- [20] Minkyong Kim and David Kotz. Periodic properties of user mobility and access-point popularity. *Personal and Ubiquitous Computing*, 11(6):465–479, 2007.
- [21] D. Koutsonikolas, S. Das, Y. C. Hu, and I. Stojmenovic. Hierarchical geographic multicast routing for wireless sensor networks. In *2007 International Conference on Sensor Technologies and Applications (SENSORCOMM 2007)*, pages 347–354, Oct 2007.
- [22] Jérémie Leguay, Timur Friedman, and Vania Conan. Evaluating mobyspace-based routing strategies in delay-tolerant networks. *Wireless Communications and Mobile Computing*, 7(10):1171–1182, 2007.
- [23] I. Leontiadis, P. Costa, and C. Mascolo. Persistent content-based information dissemination in hybrid vehicular networks. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–10, March 2009.
- [24] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.
- [25] Mengjuan Liu, Yan Yang, and Zhiguang Qin. A survey of routing protocols and simulations in delay-tolerant networks. In Yu Cheng, DoYoung Eun, Zhiguang Qin, Min Song, and Kai Xing, editors, *Wireless Algorithms, Systems, and Applications*, volume 6843 of *Lecture Notes in Computer Science*, pages 243–253. Springer Berlin Heidelberg, 2011.
- [26] A. Martín-Campillo, J. Crowcroft, E. Yoneki, and R. Martí. Evaluating opportunistic networks in disaster scenarios. *Journal of Network and Computer Applications*, 36(2):870–880, 2013. cited By 57.
- [27] A Mei, G. Morabito, P. Santi, and J. Stefa. Social-aware stateless forwarding in pocket switched networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 251–255, April 2011.
- [28] Reza Farrahi Moghaddam, Vahid Asghari, Fereydoun Farrahi Moghaddam, Yves Lemieux, and Mohamed Cheriet. A monte-carlo approach to lifespan failure performance analysis of the network fabric in modular data centers. *Journal of Network and Computer Applications*, 87:131 – 146, 2017.
- [29] S. Moghaddam and A. Helmy. Interest-aware implicit multicast (icast): Opportunistic mobile data dissemination without per-group management. pages 55–60, 2014. cited By 0.
- [30] Mehul Motani, Vikram Srinivasan, and Pavan S Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 243–257. ACM, 2005.
- [31] Jörg Ott, Esa Hyytiä, Pasi Lassila, Jussi Kangasharju, and Sougata Santra. Floating content for probabilistic information sharing. *Pervasive and Mobile Computing*, 7(6):671 – 689, 2011. The Ninth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2011).
- [32] E. Pagani, L. Valerio, and G.P. Rossi. Weak social ties improve content delivery in behavior-aware opportunistic networks. *Ad Hoc Networks*, 25(PB):314–329, 2015. cited By 1.
- [33] Ro Rogers Rachel Botsman. *What’s mine is yours: the rise of collaborative consumption*. HarperCollins Publisher, 2010.
- [34] J. A. Sanchez, P. M. Ruiz, J. Liu, and I. Stojmenovic. Bandwidth-efficient geographic multicast routing protocol for wireless sensor networks. *IEEE Sensors Journal*, 7(5):627–636, May 2007.
- [35] Joo-Han Song, Vincent W.S. Wong, and Victor C.M. Leung. Secure position-based routing protocol for mobile ad hoc networks. *Ad Hoc Networks*, 5(1):76 – 85, 2007. Security Issues in Sensor and Ad Hoc Networks.
- [36] Thrasyvoulos Spyropoulos, Rao Naveed Rais, Thierry Turletti, Katia Obraczka, and Athanasios Vasilakos. Routing for disruption tolerant networks: Taxonomy and design. *Wirel. Netw.*, 16(8):2349–2370, November 2010.
- [37] Adrián Sánchez-Carmona, Sergi Robles, and Carlos Borrego. Podcast distribution on gwanda using privhab: A multi-agent secure georouting protocol. In Javier Bajo, Josefa Z. Hernández, Philippe Mathieu, Andrew Campbell, Antonio Fernández-Caballero, María N. Moreno, Vicente Julián, Amparo Alonso-Betanzos, María Dolores Jiménez-López, and Vicente Botti, editors, *Trends in Practical Applications of Agents, Multi-Agent Systems and Sustainability*, volume 372 of *Advances in Intelligent Systems and Computing*, pages 29–37. Springer International Publishing, 2015.
- [38] Adrián Sánchez-Carmona, Sergi Robles, and Carlos Borrego. Privhab+: A secure geographic routing protocol for dtn. *Computer Communications*, pages –, 2015.
- [39] Amin Vahdat, David Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.
- [40] Shibo Wu and K. S. Candan. Gmp: Distributed geographic multicast routing in wireless sensor networks. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS’06)*, pages 49–49, 2006.
- [41] Xiaoxin Wu and B. Bhargava. Ao2p: ad hoc on-demand position-based private routing protocol. *Mobile Computing, IEEE Transactions on*, 4(4):335–348, July 2005.
- [42] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, lester and pierre: Three protocols for location privacy. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4776 of *Lecture Notes in Computer Science*, pages 62–76. Springer Berlin Heidelberg, 2007.
- [43] H. Zhou, J. Chen, J. Fan, Y. Du, and S. K. Das. Consub: Incentive-based content subscribing in selfish opportunistic mobile networks. *IEEE Journal on Selected Areas in Communications*, 31(9):669–679, September 2013.