PrivHab: a Privacy Preserving Georouting Protocol based on a Multiagent System for Podcast Distribution on Disconnected Areas

Adrián Sánchez-Carmona*

Department of Information and Communications Engineering (dEIC)
Universitat Autónoma de Barcelona (UAB)
adria.sanchez@deic.uab.cat

Sergi Robles

Department of Information and Communications Engineering (dEIC)
Universitat Autónoma de Barcelona (UAB)
sergi.robles@deic.uab.cat

Carlos Borrego

Department of Information and Communications Engineering (dEIC)
Universitat Autónoma de Barcelona (UAB)
carlos.borrego@deic.uab.cat

Abstract

We present PrivHab, a privacy preserving georouting protocol that improves multiagent decision-making. PrivHab learns the mobility habits of the nodes of the network. Then, it uses this information to dynamically select to route an agent carrying a piece of data to reach its destination. PrivHab makes use of cryptographic techniques from secure multi-party computation to make the decisions while preserving nodes' privacy. PrivHab uses a waypoint-based routing that achieves a high performance and low overhead in rugged terrain areas that are plenty of physical obstacles. The store-carry-and-forward approach used is combined with mobile agents that provide intelligence, and it is designed to operate in areas that lack network infrastructure. We have evaluated PrivHab under the scope of a realistic podcast distribution application in remote rural areas, where these programs have to be recorded into a physical format and distributed to the local radio stations. The usage of PrivHab aims to reduce this spending of resources. The PrivHab protocol is compared with a set of well-known delay-tolerant routing algorithms and shown to outperform them.

Keywords: Routing Protocols, Mobility-Tolerant Communication, Privacy, Location Tracking, Delay and Disruption Tolerant Networks

*Corresponding author.

1. Introduction and Motivation

In 2003, the Food and Agriculture Organization of the United Nations (FAO[1]) implemented a strategic Programme entitled "Bridging the Rural Digital Divide". The programme highlighted innovative approaches to knowledge exchange that were taking advantage of new digital technologies, and that were based on synergies between information management and communication for development.

Thenceforth, many initiatives have been implemented in fields as e-health, e-government, e-education, e-commerce and e-agriculture. The common goal of these initiatives is to universalize the access to knowledge and information in order to improve the life conditions of people living in developing countries. These applications have to overcome barriers like illiteracy, low cultural level of the population, censorship, etc. E-agriculture services, e.g. Agriwatch [**?**], use all the technologies at their reach: web, email, telephone, SMS, videos, printers, mail, etc. but even this way, they are constrained by the need of infrastructure and cannot operate in regions lacking it. It happens that regions where the communication networks are unavailable or spotty, where these services can not be implemented, are usually the ones where these e-agriculture services would be more needed and valuable. Unfortunately, this situation is not likely to change because the low-population density and low-income level make economically infeasible or uninteresting to extend the operators' networks into these regions.

We propose to use PrivHab to reduce the digital divide in developing countries by distributing podcast radio programs among local radio stations or other places of interest using Mobile Agent based Delay Tolerant Networking (MADTN) [**?**]. MADTN, as DTN [**?**], [**?**] uses the store-carry-and-forward strategy to operate in challenged scenarios where there are no simultaneous end-to-end paths, but it substitutes DTN's bundle (just a container of data) by a Mobile Agent, a software entity that carries the data and makes their own intelligent decisions.

Our proposal consists in creating a network of handheld devices carried by persons, and to use mobile agents that will move through this network to transport the data. Thanks to PrivHab, these agents will be able to make their own routing decisions based on the usual whereabouts of the people carrying the devices, while preserving their privacy.

Our main contributions are summarized below:

- We present an e-agriculture application, based on a real need, that improves the podcast distribution in rural areas where we cannot rely on conventional communication networks to distribute them.

[1]More information can be found on http://www.e-agriculture.org/bridging-rural-digital-divide-programme-overview

- We lay the foundations of a multi-agent intelligent system that helps the decision-making of the agents that carry the messages, while providing the enough flexibility to let them make their own decisions.
- We define the habitat, the area where a node is more likely to be found, we explain how to exploit the existence of life-cycles of the network users to define it and we model it in a simple way to allow operating it under the scope of an additive homomorphic cryptosystem.
- We define PrivHab, the first geographical routing protocol that uses the habitat to route the agents based on long-term predictions. To protect this information and to avoid its disclosure, PrivHab cryptographically protects it to ensure the habitat become hidden to the other nodes of the network.

To our knowledge, PrivHab is the first privacy preserving routing protocol that uses a geographical routing based in long-term predictions. For this reason, this is also the first work that considers the privacy of a routing information other than the historic of contacts with the other nodes of the network and that provides the tools that make this possible.

The rest of this article is organized as follows. In Section 2, we present an e-agriculture application of podcast distribution that can be enhanced through the usage of PrivHab. Section 3, summarizes the related work. In Section 4, we present the architecture of the multiagent system. In Section 5, we present the habitat, the core concept of PrivHab. In Section 6, we present PrivHab, a protocol that use the habitats of the nodes to route the messages towards its destination while preserving the privacy of the nodes of the network. In Section 7, we expose the results of the experiments made to measure PrivHab's performance. Finally, Section 8 concludes this paper.

2. Scenario of application

In this section, we present a practical example of an e-agriculture application podcast distribution on disconnected areas. This application could be greatly enhanced by using Mobile Agent based Delay Tolerant Networking, the concept of habitat and PrivHab.

2.1. Podcast distribution

In some places, due to the region's dialect preference and the illiteracy ratios, radio broadcasting is the most important information source for farmers. It plays a key role in the economy development of the region by disseminating important agricultural information. This is the main way these farmers can obtain information as valuable as what are the most appropriate crops for each season, or the most efficient processing techniques of raw materials, among others.

In the Cajamarca region, in Perú, the Non-Governmental Organization (NGO) *Practical Action*[2] records podcast radio programmes targeted to farmers in Compact Discs and physically distributes them to the local radio stations. The podcasts contain small how-to explanations, newsletters, information about prices, etc. This slow distribution method requires the NGO to spend monetary or personnel resources to bring a copy to every small local station. We aim to replace this physical distribution by a digital and automated one.

We propose to create a Delay Tolerant Network using a set of small devices that can be carried by the members of the NGO's staff or by some local villagers that collaborate with them. If it is needed, some devices can also be deployed on strategic locations. We propose to implement an automatic distribution of podcasts using this network. The deployment's cost of the nodes should be low[3], and can be considered as an investment, since the NGO will not need to spend more resources on the podcast distribution.

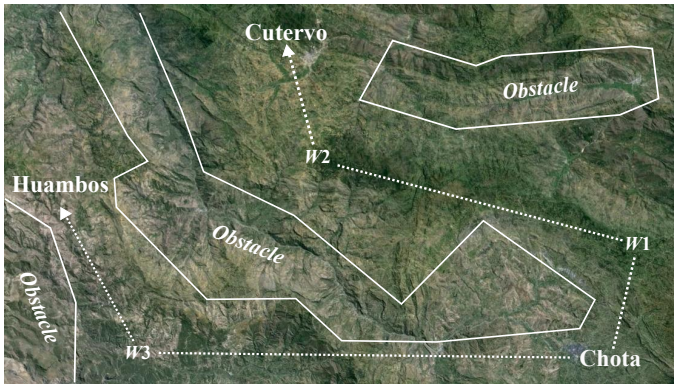


Figure 1: Map of a scenario of application located in a rural area of Cajamarca (Perú). White lines are natural obstacles approximate limits. Podcasts sent from the village of Chota to Cutervo have to be routed through waypoints $W1$ and $W2$ while messages sent from Chota to Huambos have to be routed through waypoint $W3$.

Between the NGO and the local radio stations there could be barriers that nodes carrying the data can not cross, like a cliff, one river without a bridge or a mine field. Moreover, there could be areas that nodes can only cross slowly, like a mountain or forest region, or a river with the nearest bridge located a few kilometres away. Besides, there are some interest locations like markets, churches, or the NGO's offices, that are very likely to have a higher density of nodes. Moreover, there are some zones where nodes can move quickly, due to the quality of tracks or roads, the existence of bridges or the usage of alternative means of transport. Therefore, data should try to avoid the problematic areas and follow paths that take advantage of these interesting zones or locations.

[2]More information about this programme at http://practicalaction.org/podcasting-3
[3]Small devices like Raspberry Pi can be acquired by less than 30$/unit.

This can even imply temporarily moving away the data from its destination. Figure 1 provides an example: when data from Chota is first routed to the waypoint $W1$ instead of directly towards Cutervo, the destination. This is a constraint that make georouting protocols that assume a plain world without obstacles, like LAROD [**?**], unusable. For these reasons, PrivHab allows the sender to define a list of locations (called waypoints) where the data has to pass by in order to reach its destination.

Finally, there are two requirements that can be of great value to the NGO: 1) It has to respect the privacy of its users; and 2) It has to be able to achieve a good performance occupying a small buffer and using fewer resources. A cooperator that has received a device from the NGO in order to distribute the podcasts in an area may not be very concerned about the privacy of its habitat or the amount of buffer occupied by the podcasts. However, if the NGO wants to extend the network cheaply by adding other types of nodes, e.g. volunteers that want to help the NGO by becoming part of the network, it is desirable to reduce as much as possible the impact on the users' devices and lifes. Note that lack of privacy has been identified as one of the main reasons for the unwillingness of users to participate in DTN [**?**].

3. Related work

In this Section, we first explain the reason why we take a reactive approach instead of a planning one. Then we provide the reader with a review of the related work. We present the state of the art of Geographical Routing Protocols. Later, we analyse the different proposals of Privacy Preserving Routing Protocols in Delay Tolerant Networks. Then, we review some Social-based Routing Protocols that are related, somehow, to our proposal.

Although we realize that the presented problem, distributing podcasts through an opportunistic network, is similar to those solved by multi-agent planning, given the characteristics of the scenario (a DTN), a reactive approach may fit better than a planning approach. As said in [**?**], an efficient and fast algorithm for selecting candidates on-the-fly is required when the mobility of the nodes produces a changing topology. This is exactly what DTN routing algorithms do. Besides, due to the absence of simultaneous end-to-end paths and network infrastructure, it may take long to obtain the habitats of the nodes in order to plan an itinerary "a priori" because there are nodes that will never establish a direct communication with the sender. Moreover, given the evolving nature of the habitats, they may change before the agent's arrival, making useless most of the planning effort. Therefore, situations where an agent meet a node with an unexpected habitat that it is useful to bring the message towards its destination, can only be exploited if the decisions are made locally, when this information is still in force.

3.1. Geographical Routing Protocols

Geographical Routing Protocols have been studied both in Ad-hoc Networks and Delay Tolerant Networks. Most protocols only take into account the posi-

tion of the nodes at the moment of the transmission, but not their movement pattern. LAROD [**?**] forwards packets to neighbours inside a certain area located between the forwarder and the destination, without taking into account the mobility patterns of these nodes. In [**?**], a Location Service called LoDIS is presented to improve LAROD by using gossip-based techniques to update the location of the destination at each hop. LoDIS improves the performance of the routing at the cost of the privacy of all nodes, because it periodically broadcasts their locations and speed vectors. GeoDTN+Nav [**?**] is designed for routing in a network of streets, and it has three forwarding modes. In the DTN mode, it requires the nodes to know where they are heading. This requirement can be easily met by certain types of vehicles, like buses or taxis, but it is an important restriction in scenarios where nodes are carried by people. LSGO [**?**] is a georouting protocol designed to work in Vehicular Networks where nodes forward messages to a neighbour based on its location and the link's quality. LSGO's main objective is to avoid retransmissions, but its geographic component, that takes into account only the actual location of the involved nodes, is poor. GSPI [**?**] is a geographic routing protocol for vehicular networks that uses greedy mode on straight roads and to use predictive mode at the intersections, but its predictive mode is short-termed, as it uses the current position and the speed vector of the nodes. GPRP [**?**] improves this approach by dividing roads into two-dimensional road grids and considering every possible node movement while predicting. This restricts the position prediction in the road grid sequence and improves the performance of the network, but makes this propposal hardly applicable to other kinds of scenarios and difficults its deployment.

As it can be seen, almost all proposals use contemporaneous information and short-term predictions, so they fail to take into account long-term trends of nodes' mobility. However, in scenarios where the distances to travel are big, and the density of nodes is low, it is more valuable to know where a node will go in the next hours than where it is currently headed.

3.2. Privacy Preserving Routing Protocols

Privacy Preserving Routing Protocols are based on the assumption that nodes are not willing to voluntarily share any information for the good of the network, and that nodes' privacy should be preserved in order to stimulate them to become members of the network. ALAR [**?**] allows a source to send a message through a DTN without revealing its physical location and proposes an anti-localization routing protocol. However, the only information that ALAR protects is the location where the source was when the message was sent. An anonymous communication solution for DTN has been presented in [**?**], but this one is designed to hide the identity of the nodes, not to protect the private information that these nodes use to make routing decisions. SPRING [**?**] is a routing protocol designed to vehicular DTN that bases its operation on the deployment of Roadside Units (RSUs) and on the usage of a group signature technique called CPPA [**?**]. Although its approach is similar to the one of our proposal, SPRING routes messages using a variation of Epidemic [**?**], and

what it hides is the identity of the source node and its location at the moment when the message was sent. In [**?**], the authors present a generic routing protocol that preserves the privacy of the *routing metric* through the usage of the cryptographic tools derived from the "Yao's millionaire problem" [**?**]. This proposal requires both parts of the transaction to be able to calculate their *routing metric* on their own, so it can not be used when the parts need to colaborate to calculate it. A prediction-based privacy preserving routing algorithm is presented in [**?**]. Hasan *et al.* provide a way to calculate the maximum probability of delivery within a community without disclosing node's private information, then, messages that have been disseminated through the community in an epidemic way are routed to other communities if their maximum probabilities of delivery are better. This protocol is designed to work in scenarios where the connectivity, at least inside the communities, is relatively high. SimBet-BF [**?**] protect the nodes' contacts information by blurring them using Bloom Filters at the beginning of every contact. Then, it uses two metrics, the *ego betweenness centrality* and the *similarity* to make the routing decisions. In [**?**] the privacy is also preserved by obfuscating the social network graph announced to the neighbours to make routing decisions. Finally, PRISM [**?**] routes messages towards a location while preserving the privacy of the nodes, but does not allow the source to decide the identity of the message's destination.

Unfortunately, most Privacy Preserving Routing Protocols aim to protect the nodes' contacts information, and their routing usually uses the past contacts of a node to try to predict probability of a new contact in the future. Other informations, as the identity or the locations of the nodes may be protected as well, but to our knowledge, there are no other proposals that preserve more complex informations used to make georouting decisions.

3.3. Social-based Routing Protocols

There are some Social-based Routing Protocols that are related, somehow, to the present work. Social-based routing protocols are based on the idea of using the recent past to model the behaviour of a node to predict how it will behave in the near future. BUBBLE RAP [**?**] classifies nodes using their popularity inside their community. Then, messages are forwarded to more popular nodes until they reach the community of the destination. Its design does not consider hop-distant destinations nor geographic restrictions. So, during the first hops messages can be moved into the opposite direction of their destination while they are forwarded to more popular nodes. MobySpace [**?**] leverages the life-cycles of the nodes to track the most visited by every node points of interest. These life-cycles are modelled this using a multi-dimensional probability vector, and messages are forwarded to nodes with a vector that it is closer to the one of the destination. This is a very interesting approach to our concept of habitat, but lacks adaptability. In MobySpace, the points of interest have to be defined *a priori*, and some infrastructure is needed to allow nodes to detect if they are close to these points. Besides, MobySpace may lead to situations where a node that spends most of the time at point A, very close to B, is considered a bad choice because the destination is expected to be on B, without taking into account that

A is geographically close to B. SANE [**?**] uses the same principles but defines the points of interest in a very broad sense, allowing the usage of more abstract concepts, and compares nodes using a metric called "cosine similarity". HiBOp [**?**] extends this approach using any contextual information about nodes to make routing decisions. One of its drawbacks is the big amount of memory needed to store information about every other node. Besides, the authors do not explain how this contextual information can be updated as the behaviours of the nodes evolve and change, but they recognize that privacy is an important issue to consider and that more work is needed to solve it. CSI [**?**] is a social-based routing protocol that 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.

To our knowledge, CSI is the only one proposal that takes into account the privacy of the nodes. Unfortunately, in all other cases, social-based routing protocols expect nodes to broadcast their information about the locations they visit or the details about their interests to the neighbours.

4. A Multiagent System

In this section, we first justify the decision of using Mobile Agents to solve a network problem. Then, we describe the multiagent system needed to execute PrivHab. Finally, we list and define the different agents and entities involved.

4.1. Usage of Mobile Agents' Technology

Due to the challenging characteristics of the scenario, to deploy a DTN it is not enough to achieve a fast and reliable podcast distribution. There are long distances between the senders and the receivers of the messages, so each one has to be carried by several nodes to reach its destination. Besides, most of the nodes near the source are likely to never meet with the nodes near the destination, making very difficult to obtain information about how to reach them. MADTN, using Mobile Agents, brings us a set of characteristics that PrivHab could benefit in order to deal with these challenges.

A Mobile Agent is a software entity that it is autonomous, intelligent, mobile, proactive, and represents a third part. To our consideration, all of these characteristics are beneficial to PrivHab. Agents need autonomy because they have to find their way to its destination in a changing and partially unknown environment; agents also need to be intelligent enough to make decisions that lead them towards their goal; mobility is capital because agents cannot control nodes' movement, so they need to migrate when finding a more useful one; proactivity allows agents to not only react to changes, but also to initiate context-aware actions (e.g. to start the delivery phase when the agent is near the destination);

and representativity is the characteristic that allows applications with different needs to use the same network in a different way, with the agents making decisions on their behalf.

4.2. Entities involved

PrivHab's goal is to improve the routing of the MADTN agents that carry the messages. The agents involved in this multiagent system are listed and explained below.

- **Habitat agent**: This agent calculates and periodically updates the habitat of the node (more details in Section 5). This agent also informs the Carrier agent of the current location, this way the Carrier agent can track if the node had approached enough the current waypoint and has to start considering the next one.
- **Interactor agent**: Every time a node meets a neighbour, this agent performs the PrivHab's exchange of messages to compare the habitats of the two nodes and decide who is the best choice to carry the message (more details in Section 6). When the exchange of messages has finished, this agent informs the Carrier agent of the result obtained, whether the neighbour is considered a worse or a better choice to carry the data.
- **Carrier agent**: This agent carries the message, and his goal is to deliver it to its destination. In order to achieve this, the Carrier agent moves through the network and makes decisions concerning the best way to reach a location. It uses the result of PrivHab's execution, along with other contextual information, to make a routing decision. The three decisions that the Carrier agent can made are: a) staying at the current node and waiting for other neighbours; b) migrating to the neighbour; and c) being cloned, so one agent remains at the node and the other one migrates to the neighbour.

Figure 2 depicts the agents and entities that form the system. Apart from the three agents, there are two more concepts that need to be defined here: 1) the message contains the data (e.g. the podcast) that a node has sent to a receiver, it also contains the identifier of both the sender and the receiver, and a list of locations (waypoints) that the message has to pass by in order to reach its destination; and 2) the node is a location-aware mobile device (e.g. a Raspberry Pi or a smartphone), usually carried by a person or placed in a vehicle or in a certain strategic location.

5. A habitat-based routing

In this section, we present the cornerstone of our novel georouting protocol: the habitat of a node. We define the concept and show how we model it using a circle, how it is automatically calculated and the parameters involved in the calculations. Then, we explain the characteristics of the circular model. Finally, we provide some examples of automatically calculated habitats.

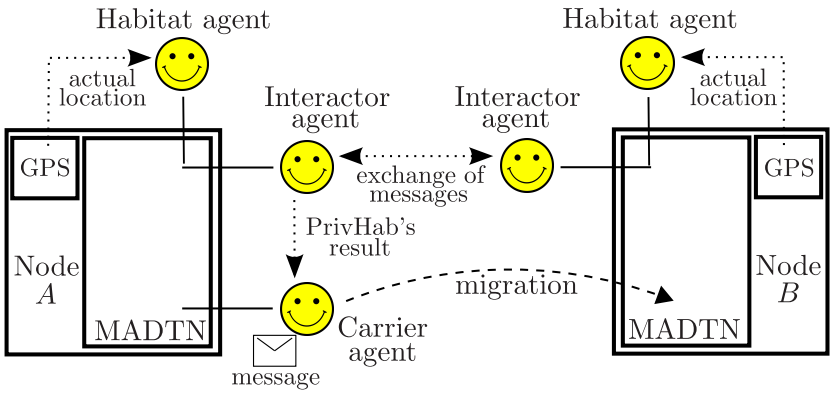


Figure 2: Schema of the multiagent system. Dotted lines depict the main interactions between entities, while slashed lines depict the movement of the agents. The Habitat agent updates the habitat using information from the GPS receiver. The Interactor agent exchanges PrivHab's messages with the other nodes and informs the Carrier agent of the result of the execution. The Carrier agent carries the message and makes the decision of migrating, staying or being cloned.

5.1. An approach towards the heatmap

In the described scenario, each node is a small device that may be carried by a person, placed in any vehicle or located in a static known place. Therefore, the movements of every node will be strongly related to their carrier. A static node will obviously remain immobile. A node carried by a person will probably spend much time in the vicinity of the carrier's home or workplace. A node placed in a vehicle will often pass by the same points if it is a regular-itinerary vehicle like a bus, or it will be inside a particular area if it is a taxi or similar. In any case, to know the places where a node has been in the past is useful to infer if a node will visit these places again in the future[4].

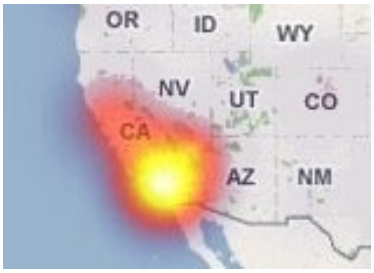


Figure 3: Heatmap of a node that spends much time in the south of the state of California. The dark red area corresponds to the area that is usually visited, and the intense yellow spot corresponds to the region where the node spends most of his time.

[4]The similarity of the movements patterns of a node to its future movements is above 0.8 for two days, and 0.75 for a week, and remains 0.6 for five weeks [**?**].

To have a heatmap of a node and its neighbours to route an agent would be ideal. For example, a Carrier agent would want to migrate to a node with a heatmap like the one shown in Figure 3 if it is carrying a message destined to the south of California, but would not if the message is destined to Utah or Wyoming. The heatmap is an extremely accurate, perhaps the most accurate, **habitat** (the area where someone is more likely to be found) representation. However, creating and maintaining this data is a resource consuming task that does not fit well with the small devices of the presented network.

Therefore, we propose to model each nodes' habitat using the simplest geometric shape: the circle. This way, nodes can automatically calculate and store their habitat consuming the minimum computational resources by using a mobile average, and they can use it to make routing decisions quickly.

5.2. Definition of the habitat

We model each habitat using a circle. Each habitat H is characterized by two elements: a centre point and a radius. From now on, we will refer as $C = (x, y)$ to the centre point of the current habitat, and we will use R to denote their radius. A habitat is defined by the tuple $H = (C, R)$.

5.3. Calculation of the habitat

The Habitat agent updates the node's habitat in order to capture the trend of the node's mobility pattern. The update process of a habitat consists in obtaining the location of a node and adding it to his habitat's model. Nodes use the Exponentially Weighted Moving Average (EWMA) to update their previous version of the habitat, named H_{old}, with a frequency of ω updates/hour. The Global Positioning System (GPS) can be used to obtain their location, from now on, we will refer as $L = (x_s, y_s)$ to the location of a node at the moment of the update. We assume that every geographic coordinate (a pair latitude - longitude) can be mapped[5] to cartesian coordinates and that this mapping is known by all the nodes of the network.

Step zero. Initialization of the habitat

At the initialization step, H_0 is initialized with the centre point at the same coordinates of the location L_0 (node's location when the calculation starts) and $R = 0$.

First step. Update of the centre

The first step to update a habitat is to update the centre. The centre point of the current habitat H is calculated by averaging using EWMA the centre point C_{old} and the current location L. The only parameter involved is α (more details about α can be found in Subsection 5.4). This first step is depicted in Figure 4, where C is calculated averaging C_{old} and L using EWMA.

[5] Any cartographic projection can be used.

$$C = L * \alpha + C_{old} * (1 - \alpha) \tag{1}$$

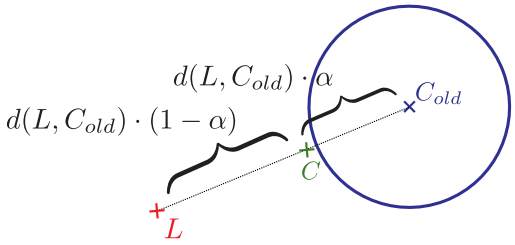


Figure 4: The new centre point C is calculated averaging the old centre C_{old} and the new location L. Note that the centre point C has moved towards L using an α factor.

Second step. Update of the radius

After C has been calculated, the radius R is updated by averaging using EWMA the radius R_{old} of the previous habitat and $d(L, C)$, the distance between L and the centre point C. This second step is depicted in Figure 5.

$$R = d(L, C) * \alpha + R_{old} * (1 - \alpha) \tag{2}$$

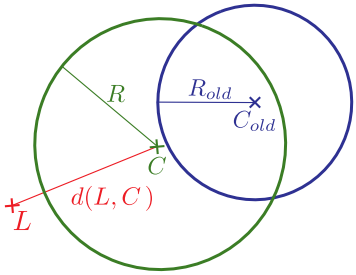


Figure 5: The old radius R_{old} is used together with the distance $d(L, C)$ that separates the new location L and the centre point C to calculate the radius R of the habitat.

As $d(L, C)$ is the radius of a hypothetical circle with centre point C that contains L. Then, it will be greater than R_{old} if L is outside the circle with centre point C and radius R_{old} and it will be smaller than R_{old} if L is contained inside this circle. Therefore, the radius R of the current habitat, which is calculated using R_{old} and D, will increase if L is out of H and will decrease if L is contained by H.

5.4. Characteristics and examples of habitats

A habitat calculated using $\alpha = \frac{2}{T\omega+1}$ models the mobility habits of a node during the last T hours. The amount of hours T a habitat models is called the habitat's time span, it is the span of time modelled by the habitat, and it has

to be known and shared by all nodes of the network. In a mobile average, each time a location is used to update the habitat, previous locations lose weight. Concretely, in EWMA, the last $T\omega$ locations weight the 86% of the total, while previous locations weight the remaining 14%.

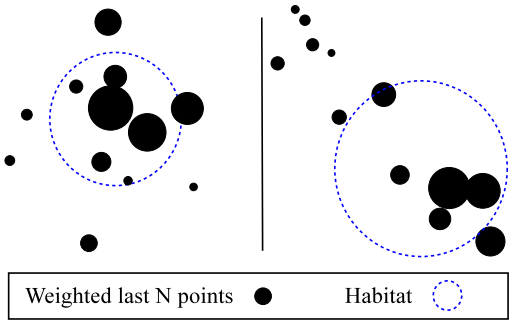


Figure 6: Two examples of habitats (dotted circles) calculated with $T\omega = 12$ ($\alpha = 0.1538$), black bubbles depict the last 12 locations, sized according to their relative EWMA weights.

Figure 6 shows two examples[6] of habitats and the locations used to update them. The bubbles representing locations are sized according to their relative weights, so the bigger they are, the more recent they are. Note that the circular habitat model is not designed to contain all the sampled locations. Its purpose is to achieve a compromise between containing all, giving more importance to the last ones, and considering the trend (the more recently sampled locations are more important than the older ones) of the node's movements.

6. The PrivHab Protocol

In this section, we first describe the PrivHab routing algorithm and its previous assumptions. Then, we introduce some important background concepts that are crucial for PrivHab to protect nodes' privacy. We explain how to use homomorphic encryption to solve two geometric problems: point inclusion and distance between a circle and a point. Following, the details about every message that has to be exchanged by the Interactor agents during the execution of PrivHab are presented. Finally, we provide some discussion about the *secure* nature of the protocol and the privacy of the participants.

6.1. Previous assumptions

PrivHab is designed to operate in scenarios where the approximate locations the message has to pass to reach the destination can be known or guessed by the sender. They may be known beforehand, may be inferred from the

[6]The examples have been obtained directly from simulations, and the snapshots have been post-processed for the sake of the readability and the clarity of the figures.

knowledge about the terrain, may be discovered via the usage of a distributed secure position service like [**?**], or via the usage of an alternate communication channel.

This assumption is hard to accomplish in scenarios where the distances and latencies are small, because the nodes can move through all the scenario and it is hard to predict where a node will be in the next few moments. However, it is reasonable in big scale scenarios like the one presented in Section 2 (105Km^2), where the distances to travel and the latencies are big, because the movement of the nodes will usually be confined in one concrete part of the scenario, with only few and short occasional trips out of their usual surroundings. If the scenario has these features, it should be easy for the users to know some things like where are the bridges to cross a certain river, what mountainous terrain has to be avoided or what valley leads to the desired location. This is the knowledge needed to set the waypoints. These waypoints travel together with the message[7].

The reader should note that, even if it is impossible for the sender to set the waypoints, the message can be sent using an approximate destination's location as the only waypoint, and PrivHab will try to route the message directly towards it.

6.2. The routing algorithm

Given the definition of habitat, we assume that nodes spent most of the time inside the area defined by their habitats. For this reason, when two nodes' habitats do not enclose the next waypoint W, the node with the closest habitat is expected to bring the message nearer the waypoint than the other one. On the same line, when both habitats enclose W, the node with the smallest habitat is expected to remain closer, and to be more likely to pass by the waypoint.

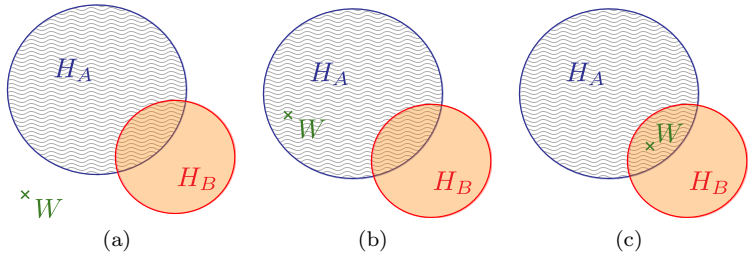


Figure 7: Three possible situations when comparing two habitats to select the best choice: (a) The next waypoint is located outside the two habitats; (b) Only one of the habitats encloses the location of the next waypoint; (c) The two habitats enclose the location of the next waypoint.

The routing algorithm uses this reasoning to compare two nodes and to decide who is the best choice to carry the message towards its destination. The

[7]Note that it is much easier to know the approximate physical path that the message has to travel to reach its destination, than to know what nodes have to carry it through this path.

algorithm chooses the nodes whose habitat's enclose the destination, prioritizing those nodes whose habitat is the smallest. If a waypoint is contained outside two habitats, then the algorithm chooses the node whose border is the closest to the next waypoint. Figure 7 show the different situations that can be faced. In (a) and (b) node A is chosen as the best option, because the waypoint W is closer to H_A or inside it. In (c) the best choice is B, because both habitats contain W, but H_B is smaller than H_A.

6.3. Nodes' privacy

At [**?** **?**], the authors recognize that privacy is an important issue in a routing protocol. In PrivHab, the habitat is used by the Carrier agent to select the next node of its itinerary, the best node to carry the message towards its destination. However, it can not be made public, since this will hurt the privacy of nodes. For this reason, nodes need PrivHab to be secure and do not reveal information about their habitats. On the other hand, waypoints are routing information that has to be known by the nodes that take custody of the message. Moreover, although they are not a private information, they must remain hidden to the nodes that do not need this information. Besides, the presented protocol is fully compatible[8] with pseudonym generator mechanisms as [**?**] that generate pseudonyms of the nodes using its public key, or [**?**] that uses a secret shared between the nodes and hashing functions. These mechanisms can be used in scenarios where the destination does not want the forwarders of the messages to associate its identity with a set of waypoints.

PrivHab uses techniques of secure multi-party computations to protect nodes' privacy. This way, the habitats and the waypoints are operated and compared while cryptographically protected in order to avoid revealing this private information to the other parts.

6.4. Background: homomorphic encryption

PrivHab requires the cryptosystem used to have a concrete property: to be additive homomorphic. An additive homomorphic cryptosystem is one in which, given two encrypted operands $E(a)$ and $E(b)$, $E(a+b)$ can be computed without separately decrypting each one.

The cryptosystem used by PrivHab is the Paillier [**?**]. In a communication between Alice and Bob, Alice selects two random primes p and q and computes $n = pq$; plaintext messages are elements of $\mathbb{Z}_n$; however, ciphertext messages are elements of $\mathbb{Z}_{n^2}$. Then Alice picks a random $g \in \mathbb{Z}^*_{n^2}$ such that $\gcd((L(g^\lambda \bmod n^2)), n) = 1$, where $\lambda = \text{lcm}(p-1, q-1)$ and $L(x) = (x-1)/n$. Alice's public key[9] is $Pk_A : (n, g)$ and her private key is $pk_A : (\lambda, p, q)$.

[8] A tuple with three values greater or equal than 0, sent in the third step of the protocol, does not reveals if the data has to be sent to B because it is a better carrier than A or because B is the destination.

[9] If Bob does not trust Alice when she generates her Paillier modulus, he can ask she to prove it is the product of exactly two nearly equal primes [**?**].

To encrypt a message m, Bob picks a random $r \in \mathbb{Z}_n^*$ and computes $c = E(m) = g^m \cdot r^n \bmod n^2$, the ciphertext of m. Then, Bob can easily compute $E(a+b) = E(a) \cdot E(b) \bmod n^2 = g^{a+b} \cdot (r_1 \cdot r_2)^n \bmod n^2)$ and $E(a \cdot s) = E(a)^s \bmod n^2 = g^{a \cdot s} \cdot (r_1^s)^n \bmod n^2)$.

Finally, to decrypt a ciphertext c, Alice computes $D(c) = L(c^\lambda \bmod n^2) = m$.

6.5. Background: point inclusion

A point $P : (x_P, y_P)$ is contained inside a circular habitat with centre $C : (x_C, y_C)$ and radius R if and only if the distance $\sqrt{(x_C - x_P)^2 + (y_C - y_P)^2}$ between C and P is lesser than R. Equivalently, we can check the sign of $d = R^2 - ((x_C - x_P)^2 + (y_C - y_P)^2)$, P is contained inside the circle if $d > 0$. This way PrivHab can know if a waypoint is contained inside the habitat using only operations allowed by the Paillier cryptosystem.

6.6. Background: distance between a circle and a point

The distance between a point $P : (x_P, y_P)$ and a habitat H with centre $C : (x_C, y_C)$ and radius R is $d(H, P) = \sqrt{(x_C - x_P)^2 + (y_C - y_P)^2} - R$. Equivalently, we can compute $X : (a, b)$, the nearest point of H to P, with $a = x_C - R \cdot \cos\beta$ and $b = y_C - R \cdot \sin\beta$ being $\beta = \tan^{-1}(\frac{y_C - y_P}{x_C - x_P})$ the angle between the x axle and the segment joining P and C. Then, we calculate $d(H, P) = d(X, P) = \sqrt{(a - x_P)^2 + (b - y_P)^2}$. This way PrivHab can compare one node's distance with another's using only operations allowed by the Paillier cryptosystem[10]: by checking the sign of $d = d_1(X_1, P_1)^2 - d_2(X_2, P_2)^2$.

6.7. Background: mapping negatives

In order to calculate both the point inclusion and the distance between a circle and a point, PrivHab requires subtraction between encrypted values. To allow us to work with Paillier operations over encrypted data, we substitute the subtraction by the addition of a negative value. However, as there are no negative values in $\mathbb{Z}_n$, we map them in a way that they could still be added to other cyphered operands or multiplied by a plain operand.

We map positive integers lower than $n/2$ using the identity function and negative integers greater than $-n/2$ with its representation modulo n, as shown in Equation 3.

$$\text{Map}(x) = \begin{cases} x & x \in [0, n/2) \\ x + n & x \in (-n/2, 0) \end{cases} \tag{3}$$

This way, we use Paillier addition between a positive integer a and a negative integer $-b$ (mapped as $-b + n$) to obtain $(a - b) + n \bmod n$. Note that if $a > b$

[10]Note that $d(H, P)^2 = (\sqrt{(x_C - x_P)^2 + (y_C - y_P)^2} - R)^2$ cannot be computed without computing first the square root. While $d(X, P)^2 = (a - x_P)^2 + (b - y_P)^2$ can be computed without computing any square root.

then $(a-b)+n \bmod n = (a-b)$, and that if $a < b$ then $(a-b)+n \bmod n = (a-b)+n$. The same way, we can use the Paillier multiplication between a negative integer $-b$ (mapped as $-b+n$) and a plain operand s to obtain $(-b+n)\cdot s \bmod n = -b\cdot s + n\cdot s \bmod n = -b\cdot s + n$. Then, the result of the operation can be recovered using the inverse mapping function shown in Equation 4.

$$\text{Inverse Map}(x) = \begin{cases} x & x \in [0, n/2) \\ x-n & x \in (n/2, n-1] \end{cases} \tag{4}$$

In order to use this mapping, we have have to ensure that the operations used in our system never exceed the boundary of $n/2$, which means that encrypted computation results should never be a positive integer higher than $n/2$ nor a negative number lower than $-n/2$. For this reason, since PrivHab works with 32 bit GPS precision coordinates[11], the minimum key length (n value) allowed in PrivHab is 128 bits, since 32 bits are for positive integers, other 32 bits are for the results of multiplications between positive integers, 32 bits more allow the results of multiplications of a negative and a positive integer, and 32 bits more are accounted for negative integers. Finally, Figure 8 provides a scheme of this mapping.

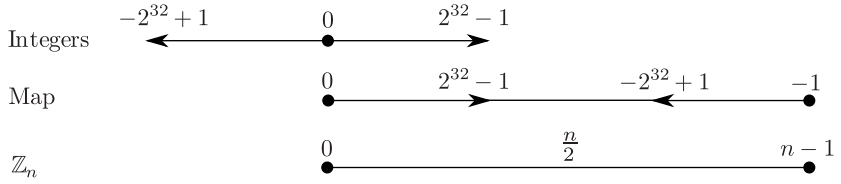


Figure 8: Positive integers are mapped to $\mathbb{Z}_n$ using the identity function. Negative integers are mapped to the higher part of $\mathbb{Z}_n$ using its representation modulo n. Positives and negatives are separated in $\mathbb{Z}_n$ by $n/2$.

6.8. Exchanged messages

We assume that every location can be mapped to two-dimensional coordinates with a mapping known to both A, the node that carries the data, and B, a candidate neighbour. Let A's habitat be $H_A : (C_A, R_A)$. Let $W[i] : (x_{W[i]}, y_{W[i]})$ be the next waypoint where the data has to be carried to. Let B's habitat be $H_B : (C_B, R_B)$. We denote $E_Y(m)$ as the Paillier additive homomorphic encryption of m using Y's public key. We denote a message sent by A to B with $A \rightarrow B : message$.

The PrivHab protocol, described below, requires the Interactor agents of the two nodes to exchange three messages.

[11] Note that latitude-longitude pairs have first to be converted into (x, y) coordinates using any cartographic projection, then these coordinates have to be converted into integers to operate with them. Finally, if needed, the resulting distances or radius must be mapped into negatives to allow subtractions.

1. Node A calculates $d_A = d(H_A, W[i])^2$, the square of the distance between its habitat and $W[i]$; $d_A = 0$ if $W[i] \in H_A$ and $d_A \geq 1$ otherwise. A knows both H_A and $W[i]$, so the calculation of d_A is very easy and can be performed quickly, without using homomorphic encryption.
2. Node B announces[12] to A the centre $C_B : (x_{C_B}, y_{C_B})$ of its habitat.

   $B \rightarrow A$: $\quad E_B(x_{C_B}), E_B(y_{C_B})$

3. Node A, using Equations 5 and 6, subtracts the coordinates of $W[i]$ to the coordinates of C. Then, A multiplies both results by the same *nonce* (a random one-use value).

$$E_B((x_{C_B} + (-x_{W[i]})) \cdot nonce) = (E_B(x_{C_B}) \cdot E_B(-x_{W[i]}))^{nonce} \tag{5}$$

$$E_B((y_{C_B} + (-y_{W[i]})) \cdot nonce) = (E_B(y_{C_B}) \cdot E_B(-y_{W[i]}))^{nonce} \tag{6}$$

   Following, A sends to B the results and the coordinates of $W[i]$, the distance d_A and the radius R_A.

$$\begin{array}{ll} A \rightarrow B: & E_B((x_{C_B} + (-x_{W[i]})) \cdot nonce), E_A(-x_{W[i]}^2), \\ & E_B((y_{C_B} + (-y_{W[i]})) \cdot nonce), E_A(-y_{W[i]}^2), \\ & E_A(-R_A), E_A(-d_A), E_A(-2x_{W[i]}), E_A(-2y_{W[i]}), \\ & E_A(-x_{W[i]}), E_A(-y_{W[i]}) \end{array}$$

4. B decrypts the received subtractions and uses the decrypted values to compute β using equation 7.

$$\beta = \tan^{-1}\left(\frac{(y_{C_B} + (-y_{W[i]})) \cdot nonce}{(x_{C_B} + (-x_{W[i]})) \cdot nonce}\right) \tag{7}$$

   Node B uses β to calculate X, the nearest point of H_B to $W[i]$, $X : (a = x_{C_B} - R_B \cdot \cos\beta, b = y_{C_B} - R_B \cdot \sin\beta)$. Then, B calculates the square of the distance $d(H_B, W[i])^2 = d(X, W[i])^2 = d_B$ using Equation 8.

$$\begin{gathered} E_A(d_B) = E_A((a - x_{W[i]})^2 + (b - y_{W[i]})^2) = \\ E_A(a^2 - 2ax_{W[i]} - x_{W[i]}^2 + b^2 - 2by_{W[i]} - y_{W[i]}^2) = \\ E_A(a^2) \cdot E_A(-2x_{W[i]})^a \cdot E_A(-x_{W[i]}^2) \cdot E_A(b^2) \cdot E_A(-2y_{W[i]})^b \cdot E_A(-y_{W[i]}^2) = \end{gathered} \tag{8}$$

   Following, B calculates the point inclusion of $W[i]$ in H_B using Equation 9, the comparison of distances using Equation 10, and the comparison of radius using Equation 11. This time, three different *nonce* values are

[12] This announcement can be made by adding this information to the messages exchanged during the neighbour discovery process.

used to randomize the results. The d_A factor is used to blur[13] the point inclusion test and the comparison of radius.

$$E_A((R_B^2 + d_B + (-d_A)) \cdot nonce) = (E_A(R_B^2) \cdot E_A(d_B) \cdot E_A(-d_A))^{nonce} \quad (9)$$

$$E_A((d_B + (-d_A)) \cdot nonce) = (E_A(d_B) \cdot E_A(-d_A))^{nonce} \quad (10)$$

$$E_A((R_B + (-d_A \cdot R_B) + (-R_A)) \cdot nonce) = (E_A(R_B) \cdot E_A(-d_A)^{R_B} \cdot E_A(-R_A))^{nonce} \quad (11)$$

Finally, B orders the results of the two comparisons and the point inclusion test in a random way and sends it to A.

$$B \rightarrow A: \quad \begin{array}{l} E_A((R_B + (-d_A \cdot R_B) - R_A) \cdot nonce), \\ E_A((R_B^2 + d_B + (-d_A)) \cdot nonce), \\ E_A((d_B + (-d_A)) \cdot nonce) \text{ randomly ordered.} \end{array}$$

5. Node A decrypts the three received values. B is considered a better choice if and only if the three decrypted values are negative or 0.

Figure 9 depicts the exchange of messages.

6.9. Security evaluation

In secure multi-party computations [**?**], a protocol is considered secure if it reveals only the result of the function and the inferences that can be deduced from this output with one or more input values. The presented protocol has been designed following these principles. On one hand, node A only knows if H_B is better or worse than H_A. Then, A can use this knowledge to infer about the relation between d_A and d_B, the relation between R_A and R_B, or to deduce if $W[i] \in H_B$. On the other hand, node B cannot even know the result of the execution, so it cannot learn anything about H_A. Maintaining $W[i]$ hidden to B (only β is revealed) when the data is not forwarded is crucial to avoid that B can calculate d_B and use it to infer information about H_A.

Anything learned by A about H_B is also learnable from the result alone. Moreover, when the Carrier agent migrates to B the waypoints are revealed to it, because waypoints will be needed in next steps of the routing. Otherwise, the only thing B learns about $W[i]$ is the angle[14] β where it is located in relation with H_B.

[13] If $d_A > d_B$, then the best choice is B, and the result of the point inclusion test and the comparison of radius are not needed.

[14] The angle β is a less accurate information than the coordinates of $W[i]$ or the distance between $W[i]$ and H_B. Moreover, B does not even know who is the destination, and the protocol will not be executed again between the same participants. Therefore, B can not relate $W[i]$ with any node neither triangulate its location.

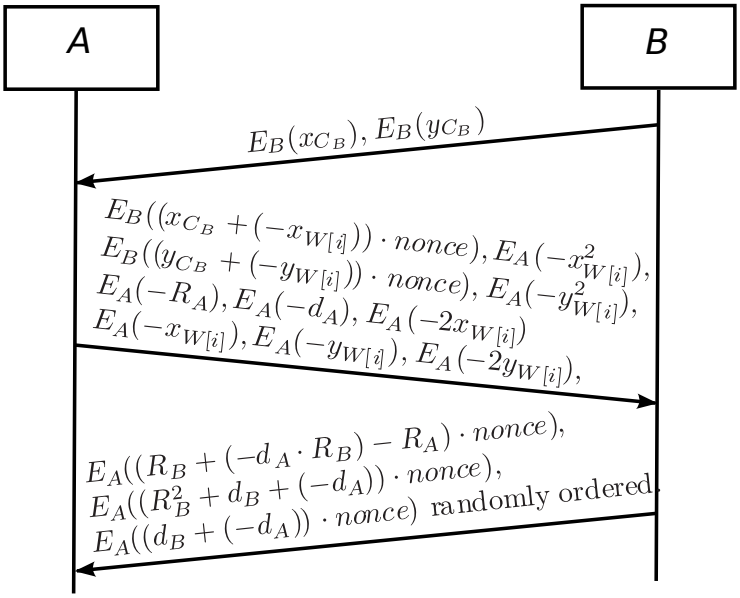


Figure 9: Sequence of the messages exchanged by the Interactor agents during the execution of PrivHab.

On the other hand, an active attacker can try to learn things about the other part's habitat by producing chosen-destination arbitrary messages and repeatedly executing PrivHab. In any case, the information obtained by the attacker is the same information that he can infer from a truthful execution of the protocol. As A is the node that starts the transaction and the only one that knows the number of messages he carries, he can determine how many times to execute PrivHab+. If A executes PrivHab enough times, he can try to uncover the area covered by H_B. Given that nodes always operate with encrypted data, there is no way for one part to tell apart a truthful execution of PrivHab from an untruthful one. However, B can decrease the effectiveness of these attacks by limiting the amount of interactions per unit of time with every other node and forcing A to send him at once the information needed to perform all the executions before sending any response. Besides, the information protected by PrivHab, the habitat, changes periodically. For this reason, slowing enough an attack is equivalent to avoiding it, because when time passes the habitats change and the first things learned by the attacker become obsolete.

7. Experiments and Results

In this section, we study the computational and communication overhead introduced by PrivHab. Then, we explain the scenario we have chosen to evaluate PrivHab's and other well known DTN routing protocols, and how we have

modelled and simulated it. Finally, we provide the obtained results, and we compare PrivHab with a set of popular DTN routing algorithms.

7.1. Physical implementation

As a proof-of-concept we have deployed an implementation of the presented protocol on three Raspberry Pi boards[15]. These are very cheap low-end devices that fit very well with the characteristics of the proposed application, and they are ideals to deploy a prototype network that will allow us to run field experiments in the near future. We have used them to measure the overhead that PrivHab adds to every transaction.

We have used our proof-of-concept implementation, using Paillier's length keys of 512, 1024 and 2048 bits, to forward 600 podcasts of sizes between 10MB and 20MB[16]. We have repeated the tests twenty times. We have measured the average time needed by the Interactor agent to make the calculations and to exchange all the messages. The obtained results are shown in Table 1 and have been incorporated to the simulations.

As can be seen in Table 1, PrivHab execution time depends heavily on the key length used. When using keys of 512 bits, PrivHab can be executed by a low-end device in less than half a second. Meaning an overhead of less than 3% when sending messages larger than 10MB. The execution time increases to 2.5 seconds when using keys of 1024 bits. Given the average length of connectivity windows in remote village scenarios presented in [**?**], this overhead is acceptable. When using keys of 2048 bits, the execution time is high. The key length should be chosen keeping in mind the duration of the connectivity windows and the security requirements of the scenario. In the presented application, the overhead of 2.5s using a 1024 bits key is efficient and secure enough[17].

Key length	**Time (ms)**	**Overhead 10MB (%)**	**Overhead 20MB (%)**
512 bits	401.94 ± 0.5	2.44	1.22
1024 bits	$2,585.05 \pm 23.1$	15.69	7.84
2048 bits	$15,018.9 \pm 38.8$	91.13	45.57

Table 1: Average execution time of PrivHab using different key lengths. The overhead is the extra amount of time needed to send a message of 10MB or 20MB.

[15]Raspberry Pi Broadcom BCM2835 SoC full HD, 700MHz Low Power ARM1176JZ-F, 512MB SDRAM, 256MB SD with Raspbian, Wi-Pi Wireless Adapter (802.11n up to 150Mbps), GPS receiver NL-302U (baud rate: 4800 bauds) and a dual output 5000mAh battery.

[16]This is the size of an audio file with ID3 version 2.4.0, extended header, containing: MPEG ADTS, layer III, v1, 128 kbps, 44.1 kHz, stereo, with a duration between 10 and 20 minutes.

[17]The effort needed to break the provided security is equivalent to the effort needed to factor a 1024 bits RSA key.

7.2. Modelling and simulations

The scenario we have used in all the simulations is the one presented in Section 2. Nodes implement a mobility pattern that takes into account their *hotspots* [**?**] (home's and work's location). Agents carrying podcasts are injected in the network by the NGO office, who knows the exact location and the necessary waypoints to reach every destination. Nodes use PrivHab to make routing decisions. Carrier agents always chose to migrate to nodes that are considered better choices by the PrivHab algorithm. Table 2 provides the simulation parameters that have been used.

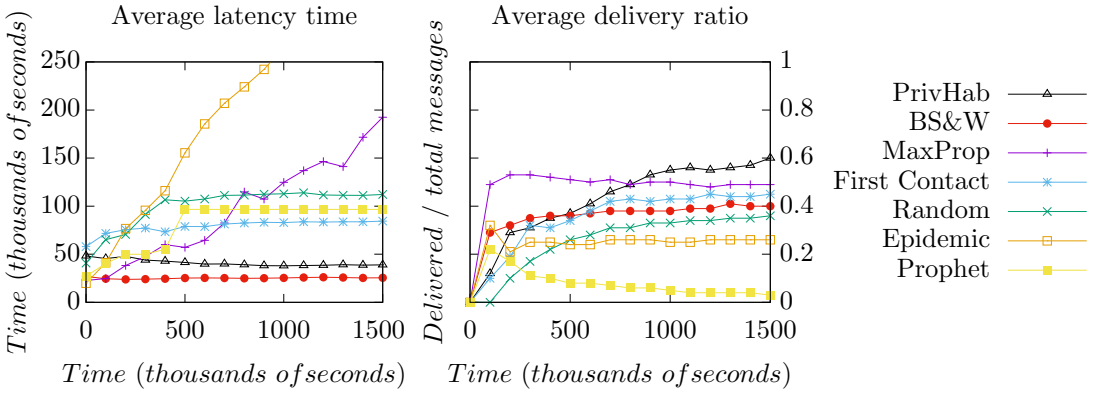
Parameter	**Value**
Total nodes	95
Source nodes	1 static
Destination nodes	2 static
Other nodes	92 mobile
Message size	$10 - 20$ MB
Buffer size	200 MB
Scenario size	15x7 Km
Simulated time	2.5 weeks
PrivHab's overhead	2.5 seconds
Habitat update frequency (ω)	2 updates / hour
Message generation ratio (messages / hour)	**High:** $0.5 - 1$
	Medium: $0.25 - 0.5$
	Low: $0.125 - 0.25$

Table 2: Parameters used at the simulations.

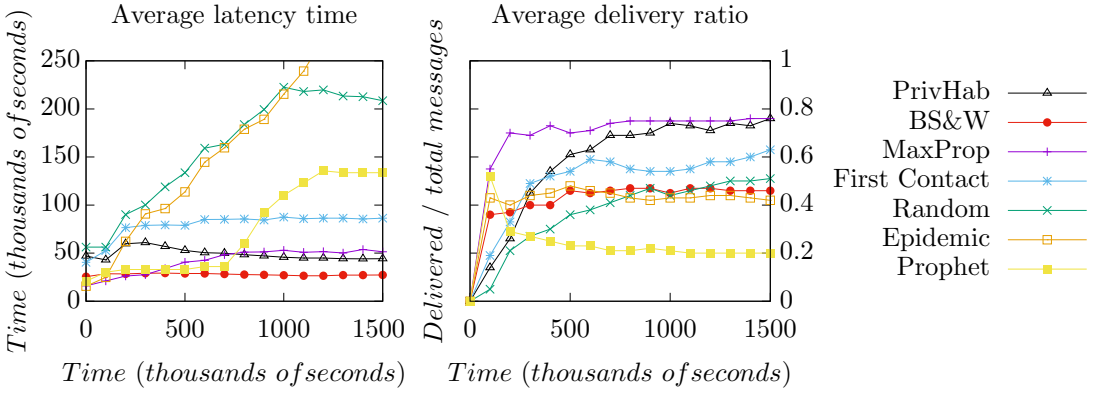
We have compared the performance of PrivHab with a bench-mark of well-known DTN routing protocols used in [**?**]: Prophet [**?**], Binary Spray & Wait (L=40) [**?**], Epidemic [**?**] and Random [**?**]. We have added two routing protocols to this set: MaxProp [**?**] and First Contact[18]. Random and First Contact are traditionally considered to achieve the lower bound of single-copy routing performance. Prophet and MaxProp are representatives in contacts-based prediction routing algorithms, the most common type of routing in privacy preserving protocols. Finally, BS&W and Epidemic are representatives of flooding-based algorithms. All simulations have been performed using *The Opportunistic Network Simulator* (The ONE) [**?**], and have been repeated twenty times using different random seeds.

The performance of all the compared protocols in terms of delivery ratio and latency while using different message generation ratios is depicted in Figure 10. Flooding-based protocols, as Epidemic and Prophet, fill the buffers early and perform badly with a high or medium message generation ratio. Therefore,

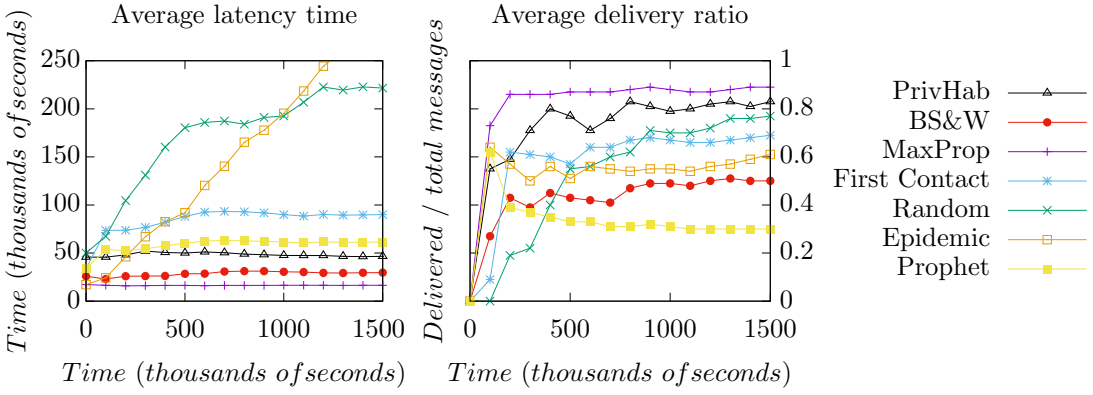
[18]When a neighbour is met, each podcast that has not been carried previously by the new neighbour is forwarded to him.



(a) Latency and delivery ratio obtained using a message generation ratio of: $0.5 - 1$ messages/hour.



(b) Latency and delivery ratio obtained using a message generation ratio of: $0.25 - 0.5$ messages/hour.



(c) Latency and delivery ratio obtained using a message generation ratio of: $0.125 - 0.25$ messages/hour.

Figure 10: Performance's comparative using different message generation ratio. MaxProp equals PrivHab's performance with a medium ratio, and outperforms it with a low one. PrivHab also benefits from a lower ratio to increase the amount of messages it delivers.

they obtain high latencies and low delivery ratios because nodes are forced to drop podcasts. When the message generation ratio is low, their latencies improve, but as most of the opportunistic contacts end before nodes had been able to forward all the carried messages, their delivery ratio continues to be low because podcasts lose opportunities to advance through their destination. MaxProp performs better because of his dropping policy based on probabilities of delivery, and improves vastly as the message generation ratio decreases. This way, MaxProp performs badly in terms of latency with a high message generation ratio, but achieves a good performance with a medium one, and outperforms the other protocols with the low one. Binary Spray & Wait performs well both in terms of latency and delivery ratio in all cases, but does not improves much its perfomance when the message generation ratio changes because of his depth-style spread. Therefore, it is a good choice (it obtains the lowest latency) with a high message generation ratio, but a bad one with a medium or low ratio. Besides, even in its best case scenario, its delivery ratio is not as good as PrivHab's because in BS&W the spread is not directed towards the destination. Finally, First Contact, Random and specially PrivHab obtain a high delivery ratio in all cases because they do not face the problems related to the size of the buffers and the connectivity windows. However, the performance of these protocols in terms of latency depends on the quality of their decision-making protocol. Random is the worst because it is equally likely to make a bad or a good choice. First Contact performs better because it forces podcasts to move away from their origin. These two protocols become worse by comparison as the message generation ratio decreases and the flooding protocols improve their results. Finally, PrivHab, that takes the best decisions because it takes into account both the pathway to the destination and the mobility patterns of the neighbours, obtains the best performance with a high message generation ratio, and perfroms slightly better, or worse, than MaxProp with a medium and low one.

Protocol	**Dropped messages**		**Network overhead (%)**	
Message generation	**High**	**Low**	**High**	**Low**
Epidemic	1,041,105.3	742,610.4	86,636.4	6,157.2
Prophet	628,897.4	329,756.3	89,705.5	35,357.7
Maxprop	206,372.7	8,145.9	7,682.2	162.1
BS&W	2,105.1	4,645.0	86.8	64.6
Random	86.9	7,3	40,582.5	2,557.6
First Contact	75.8	5.2	137.9	85.5
PrivHab	75.5	19.1	20.3	12.7

Table 3: Obtained results in terms of network overhead and number of dropped messages. PrivHab and First Contact waste fewer network resources.

Table 3 shows the average number of dropped messages and the network overhead, calculated as the relation between the number of the relays done and the number of delivered podcasts. Both low and high message generation ratio

cases have been considered. Low network overhead is desirable because reducing relays saves battery and increases the amount of time nodes are operational. Epidemic, Prophet, MaxProp and Random generate an enormous overhead of several thousand percent when the message generation ratio is high. This means that almost all nodes effort while forwarding podcasts is wasted, either because the podcasts are dropped or because the majority of the relays are bad choices. However, MaxProp improves its results and obtains a lower network overhead when the message generation ratio is low. This means that MaxProp generates copies that fill the buffers and consume energy because multiplies the number of relays done, but it makes use of this effort to deliver the podcasts to their destination. BS&W has a small amount of dropped podcasts, in comparison with the other multi-copy protocols, and a low network overhead. BS&W tries to limit the amount of resources used and obtains the second lowest network overhead with a low message generation ratio, but its perfomance in terms of latency and delivery ratio is not as good as others'. First Contact and PrivHab have generated a small amount of dropped messages, but the lowest network overhead of PrivHab means that his routing decisions are much better. The small network overhead produced by PrivHab could even allow users to use the same devices to run other applications because the main application does not congests either the device or the network.

Following, Table 4 finishes the comparison, regarding the Cajamarca scenario. In addition to those metrics that had been studied in previous paragraphs, delivery ratio, latency and network overhead; we also take into consideration the type of routing used, the nodes' privacy and the protocol's complexity.

Protocol	Type of routing	Nodes' privacy	Protocol's Complexity
PrivHab	Geographic	Preserved	Constant
MaxProp	Contacts-based	Violated	Linear
Prophet	Contacts-based	Violated	Linear
BS&W	Flooding	Not considered	Constant
Epidemic	Flooding	Not considered	Constant
First Contact	One-copy	Not considered	Constant
Random	One-copy	Not considered	Constant

Table 4: Feature comparison of the protocols. Contacts-based routing algorithm tend to violate nodes privacy and to have at least a linear complexity.

Nodes' privacy is preserved by PrivHab, which is the only one that uses private information in a secure manner. Privacy is obviously not considered by the protocols that do not use node-related information to make choices, but it is heavily violated by Prophet and MaxProp while nodes exchange their likelihood to contact others. However, their privacy preserving counterparts do not have this limitation, but they route the messages similarly, using a contacts-based prediction, so they perform similarly in this scenario. PrivHab, BS&W,

Epidemic, First Contact and Random need a constant number of operations to make a routing decision. Contacts-based algorithms need to update and compare an amount of probabilities that grow linear with the number of nodes of the network. When operating in networks with lots of nodes, probabilistic protocols have to limit the amount of encounter probabilities they store. This limitation decreases their performance because this reduces the value of their heuristics.

PrivHab delivers more messages to its destination. Besides, it does it faster than all other protocols except MaxProp with a low message generatio ratio, and it consumes fewer network resources to do so. Moreover, it preserves nodes' privacy and performs well in scenarios where the number of nodes is high and the destinations of the messages are hop-distant. Taking into account all these aspects, we can state that PrivHab is the protocol that suits better to any scenario with characteristics like the presented one.

8. Conclusions

The habitat models node's whereabouts during the habitat's time span. It is useful to compare nodes to decide who is a better choice to carry the data towards its destination. In this paper, we present PrivHab, a privacy preserving multiagent geographical routing protocol based on MADTN that uses the habitats to make decisions. PrivHab also makes use of homomorphic cryptography techniques to preserve nodes' privacy. We have presented a podcasts distribution application in rural areas lacking communication networks that could benefit from the characteristics and the performance of PrivHab.

PrivHab's characteristics make him ideal to operate not only in this concrete scenario of application, but also in any other DTN scenario with similar characteristics: scenarios where nodes mobility patterns are complex, but routinary, where lots of hops are needed to reach the destination of the messages from their source, and where nodes are so related, directly or indirectly, to a person that their privacy needs to be protected.

As future lines of research, we plan to study different behaviours for the Carrier agent, to improve the circular model of habitat using a more complex representation, and to develop an enhanced version of PrivHab that compares simultaneously three or more habitats. We also plan to study the performance of PrivHab in different scenarios based on real applications that could benefit from a geographic routing approach.

Acknowledgment

This work has been partially funded by the Ministry of Economy and Competitivity of Spain, under the reference project TIN2014-55243-P and 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.

The Authors

Figure 11: Adrián Sánchez-Carmona

Mr. Adrián Sánchez-Carmona Born in Terrassa, Barcelona. He received his degree in Computer Science (5 year programme) at the Universitat Autnoma de Barcelona (UAB). In 2013 he obtained the Master Degree on Security on Information Technology and Communications (UOC-UAB-URV). After finishing his studies he started his PhD. He started an internship at the CASA team of the Laboratoire IRISA at Université de Bretagne-Sud on 2016. He is actually a PhD student at the Department of Information and Communications Engineering (dEIC).

Figure 12: Sergi Robles

Dr. Sergi Robles received his PhD in Computer Science from Universitat Autnoma de Barcelona. He is an associate professor in the Department of Information and Communications Engineering at the Universitat Autnoma de Barcelona, where he leads the Security of Networks and Distributed Applications (SeNDA) research group. His latest research interests include mobile agents and security, and routing in Delay Tolerant Networks.

Figure 13: Carlos Borrego

Dr. Carlos Borrego Born in Madrid. He received his degree in Computer Science (6 year programme) at the Faculty of Computer Science at the Polytechnic University of Madrid. After finishing his studies he moved to work for CERN (Geneva, Switzerland). In 2001 moved to CASPUR, University La Sapienza (Rome, Italy) and stayed there for four years. In 2005 moved to the Autonomous University of Barcelona (Barcelona, Spain) where he finished his PhD and worked for Pic and Ifae research centers. He is actually researcher and adjunct professor at the Department of Information and Communications Engineering dEIC. He gives lectures on computer networks and cryptography.

References

[1] A. Singh, "Bridging the Rural Digital Divide. Case Study: Institution Based Information Systems, India.," tech. rep., Indian Agribusiness Systems Pvt. Ltd., Nov. 2005.

[2] R. Martínez, S. Castillo, S. Robles, A. Sánchez, J. Borrell, M. Cordero, A. Viguria, and N. Giuditta, "Mobile-agent based delay-tolerant network architecture for non-critical aeronautical data communications," in *In 10th International Symposium on Distributed Computing and Artificial Intelligence* (Springer, ed.), May 2013.

[3] C. Borrego, S. Castillo, and S. Robles, "Striving for sensing: Taming your mobile code to share a robot sensor network," *Information Sciences*, no. 0, 2014.

[4] A. P. Silva, S. Burleigh, C. M. Hirata, and K. Obraczka, "A survey on congestion control for delay and disruption tolerant networks," *Ad Hoc Networks*, vol. 25, Part B, pp. 480 – 494, 2015. New Research Challenges in Mobile, Opportunistic and Delay-Tolerant NetworksEnergy-Aware Data Centers: Architecture, Infrastructure, and Communication.

[5] E. Kuiper and S. Nadjm-Tehrani, "Geographical routing with location service in intermittently connected manets," *Vehicular Technology, IEEE Transactions on*, vol. 60, pp. 592–604, Feb 2011.

[6] J. Miao, O. Hasan, S. B. Mokhtar, L. Brunie, and K. Yim, "An investigation on the unwillingness of nodes to participate in mobile delay tolerant network

routing," *International Journal of Information Management*, vol. 33, no. 2, pp. 252 – 262, 2013.

[7] A. Boukerche and A. Darehshoorzadeh, "Opportunistic routing in wireless networks: Models, algorithms, and classifications," *ACM Comput. Surv.*, vol. 47, pp. 22:1–22:36, Nov. 2014.

[8] E. Kuiper and S. Nadjm-Tehrani, "Geographical routing with location service in intermittently connected manets," *Vehicular Technology, IEEE Transactions on*, vol. 60, pp. 592–604, Feb 2011.

[9] P.-C. Cheng, K. Lee, M. Gerla, and J. Hrri, "Geodtn+nav: Geographic dtn routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2010.

[10] X. Cai, Y. He, C. Zhao, L. Zhu, and C. Li, "Lsgo: Link state aware geographic opportunistic routing protocol for vanets," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1–10, 2014.

[11] T.-Y. Wu, Y.-B. Wang, and W.-T. Lee, "Mixing greedy and predictive approaches to improve geographic routing for vanet," *Wireless Communications and Mobile Computing*, vol. 12, no. 4, pp. 367–378, 2012.

[12] C. Si-Ho, L. Keun-Wang, , and C. Hyun-Seob, "Grid-based predictive geographical routing for inter-vehicle communication in urban areas," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[13] X. Lu, P. Hui, D. Towsley, J. Pu, and Z. Xiong, "Anti-localization anonymous routing for delay tolerant network," *Computer Networks*, vol. 54, no. 11, pp. 1899 – 1910, 2010.

[14] A. Kate, G. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," in *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pp. 504–513, Sept 2007.

[15] R. Lu, X. Lin, and X. Shen, "Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, March 2010.

[16] X. Lin, X. Sun, P. H. Ho, and X. Shen, "Gsis: A secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3442–3456, Nov 2007.

[17] A. Vahdat, D. Becker, *et al.*, "Epidemic routing for partially connected ad hoc networks," tech. rep., Technical Report CS-200006, Duke University, 2000.

[18] L. Zhang, J. Song, and J. Pan, "Towards privacy-preserving and secure opportunistic routings in vanets," in *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 627–635, June 2014.

[19] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, (Washington, DC, USA), pp. 160–164, IEEE Computer Society, 1982.

[20] O. Hasan, J. Miao, S. B. Mokhtar, and L. Brunie, "A privacy preserving prediction-based routing protocol for mobile delay tolerant networks," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, pp. 546–553, March 2013.

[21] E. Papapetrou, V. F. Bourgos, and A. G. Voyiatzis, "Privacy-preserving routing in delay tolerant networks based on bloom filters," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*, pp. 1–9, June 2015.

[22] I. Parris, G. Bigwood, and T. Henderson, "Privacy-enhanced social network routing in opportunistic networks," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 624–629, March 2010.

[23] K. E. Defrawy and G. Tsudik, "Privacy-preserving location-based on-demand routing in manets," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 1926–1934, December 2011.

[24] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *Mobile Computing, IEEE Transactions on*, vol. 10, pp. 1576–1589, Nov 2011.

[25] J. Leguay, T. Friedman, and V. Conan, "Evaluating mobyspace-based routing strategies in delay-tolerant networks," *Wireless Communications and Mobile Computing*, vol. 7, no. 10, pp. 1171–1182, 2007.

[26] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless forwarding in pocket switched networks," in *INFOCOM, 2011 Proceedings IEEE*, pp. 251–255, April 2011.

[27] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, "Hibop: a history based routing protocol for opportunistic networks," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pp. 1–12, June 2007.

[28] W. Hsu, D. Dutta, and A. Helmy, "CSI: A paradigm for behavior-oriented delivery services in mobile human networks," *CoRR*, vol. abs/0807.1153, 2008.

[29] J.-H. Song, V. W. Wong, and V. C. Leung, "Secure position-based routing protocol for mobile ad hoc networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 76 – 85, 2007. Security Issues in Sensor and Ad Hoc Networks.

[30] R. Jiang and Y. Xing, "Anonymous on-demand routing and secure checking of traffic forwarding for mobile ad hoc networks," in *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium*, pp. 406–411, Oct 2012.

[31] M. Mahmoud and X. Shen, "Lightweight privacy-preserving routing and incentive protocol for hybrid ad hoc wireless network," in *Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE*, pp. 1006–1011, Apr. 2011.

[32] G. Zhong, I. Goldberg, and U. Hengartner, "Louis, lester and pierre: Three protocols for location privacy," in *Privacy Enhancing Technologies* (N. Borisov and P. Golle, eds.), vol. 4776 of *Lecture Notes in Computer Science*, pp. 62–76, 2007.

[33] M. Liskov and R. Silverman, "A statistical limited-knowledge proof for secure rsa keys," tech. rep., IEE P1363 working group, 1998.

[34] O. Goldreich, "Secure multi-party computation," 1998.

[35] S. Grasic and A. Lindgren, "Revisiting a remote village scenario and its dtn routing objective," *Computer Communications*, vol. 48, p. 133140, 2014.

[36] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *Communications Magazine, IEEE*, vol. 52, pp. 29–35, Aug 2014.

[37] M. Musolesi and C. Mascolo, "Car: Context-aware adaptive routing for delay-tolerant mobile networks," *Mobile Computing, IEEE Transactions on*, vol. 8, pp. 246–260, Feb 2009.

[38] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, pp. 19–20, July 2003.

[39] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, p. 259, ACM, 2005.

[40] T. Spyropoulos, R. N. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: Taxonomy and design," *Wireless Networks*, vol. 16, pp. 2349–2370, Nov. 2010.

[41] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–11, 2006.

[42] A. Keränen, J. Ott, and T. 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), ICST, 2009.