Improving Podcast Distribution on Gwanda using PrivHab: a Multiagent Secure Georouting Protocol

Adrián SÁNCHEZ-CARMONA[a,b]; **Sergi ROBLES**[a]; **Carlos BORREGO**[a]

[a]*Department of Information and Communications Engineering, Universitat Autónoma de Barcelona (UAB)*

[b]*corresponding author: adria.sanchez@deic.uab.cat*

KEYWORD

Routing Protocols; Privacy; Applications; Mobile agents

ABSTRACT

We present PrivHab, a multiagent secure georouting protocol that improves podcast distribution on Gwanda, Zimbabwe. PrivHab learns the whereabouts of the nodes of the network to select an itinerary for each agent carrying a piece of data. PrivHab makes use of cryptographic techniques to make the decisions while preserving nodes' privacy. PrivHab uses a waypoint-based georouting 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 based on mobile agents and is designed to operate in areas that lack network infrastructure. The PrivHab protocol is compared with a set of well-known delay-tolerant routing algorithms and shown to outperform them.

1. Introduction and Motivation

The last decade, many initiatives have been implemented in fields as e-health, e-government, e-education, e-commerce and e-agriculture in order to improve the life conditions of people living in developing countries. Universalize the access to knowledge and information is a requirement for all these applications to become useful.

These applications are constrained by the need of infrastructure and cannot operate in regions lacking it. Besides, e-agriculture applications, usually targeting rural areas, are very likely to deal with a lot of challenges. A sparse population, illiteracy, a bad, non-existent or expensive telephony coverage and, especially, a lack of data communication networks are the most common ones.

Delay Tolerant Networking (DTN) (Borrego et al., 2014), based on the store-carry-and-forward strategy, is designed to operate in challenged scenarios like the mentioned above. DTN's operation is based on the usage of mobile devices that opportunistically establish contact and exchange messages called bundles between them. Due to its design, the network's topology cannot be known beforehand in DTN because it changes quickly. Mobile Agent based Delay Tolerant Networking (MADTN) (Martínez et al., 2013) uses mobile agents to perform this store-carry-and-forward strategy, and it is designed to operate in scenarios where there are no simultaneous end-to-end paths.

We propose to use MADTN to reduce the digital divide in developing countries by distributing podcast

radio programs. We designed PrivHab to improve the itinerary decision-making of the data-carrier agents. PrivHab leverages the existence of life-cycles of the network users to learn about their usual whereabouts. Then, this information is used to find an itinerary to carry the data to its destination. PrivHab treats this information about the mobility habits of the network elements in a secure manner in order to protect node's privacy.

Our main contributions are summarized below:

- We present an e-agriculture application of podcast distribution, based on the real need of an NGO that operates in Gwanda, Zimbabwe.
- We introduce the concept of node's habitat, the area where a node is more likely to be found; and an algorithm for making routing decisions based on this information.
- We define PrivHab, the first geographical routing protocol that uses the information about the mobility habits of the nodes to make routing decisions.

The rest of this article is organized as follows. In Section 2, we review the state of the art. In Section 3, we present an e-agriculture application of podcast distribution that can be enhanced through the usage of PrivHab, and we discuss how to implement it using a multiagent system. In Section 4, we present the habitat, a model of nodes' whereabouts useful to make itinerary-selection decisions. In Section 5, we present PrivHab, a routing protocol that use the habitats of the nodes to route messages while preserving the privacy of the nodes of the network. In Section 6, we expose the results of the experiments made to measure PrivHab's performance. Finally, Section 7 concludes this paper.

2. Related work

In this Section, we provide the reader with a review of the related work. We present the state of the art of Geographical Routing Protocols. Later, we review some Social-based Routing Protocols that are related to our proposal.

2.1 Geographical Routing Protocols

Most Geographical Routing Protocols protocols only take into account the position of the nodes at the moment of the transmission, but not their movement pattern. GeoDTN+Nav (Cheng et al., 2010) is designed for routing in a network of streets. 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 drawback in scenarios where nodes are carried by people. MoVe (LeBrun et al., 2005) is a routing protocol designed to work in Vehicular Networks where nodes forward messages to a neighbour if the neighbour is expected to come closer to the destination. Nodes exchange their speed vectors to make routing decisions. This short-term predictions loose precision in when latencies are big due to a low level of connectivity. In (Li et al., 2006), GPSR (Karp and Kung, 2000) is modified to adapt it to DTN. However, messages are routed in the basis of a neighbourhood table. This planning approach does not adapt well to a scenarios where the topology of the network changes quickly and in an unpredictable manner. In (Kuiper and Nadjm-Tehrani, 2011a), a Location Service called LoDIS is presented to improve LAROD (Kuiper and Nadjm-Tehrani, 2011b) 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. However, between the sender and the destinations there could be barriers that nodes carrying the data can not cross, as a river, and there could be some locations that are crucial to overcome this obstacles, as a bridge. Therefore, data should try to follow paths that take advantage of this knowledge, even if this imply temporarily moving away the data from its destination. This is a constraint that make usual georouting protocols unusable (because they assume a plain world withotu obstacles).

Geographical Routing Protocols 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.

2.2 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. MobySpace (Leguay et al., 2007) 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. SANE (Mei et al., 2011) 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". The frameworks presented in (Musolesi and Mascolo, 2009) (Costa et al., 2008) go one step further and not only use the recent past to model the behaviour of a node, they use Kalman filters to predict the future values of their attributes. However, all predictions are finally condensed in a single value, the probability of delivery. This probability is the metric used to decide the node where every message is forwarded.

In all proposals, nodes are expected to broadcast their information about the locations they visit or the details about their interests to the neighbours. The authors of (Boldrini et al., 2007) recognize that privacy is an important issue to consider in Social-based Routing Protocols, and that more work is needed to solve it. Unfortunately, as pointed in (Karlof and Wagner, 2003), most Secure Routing Protocols aim to protect the routing algorithm's performance against malicious behaviours, and there are little proposals of routing algorithms that respect or protect the privacy of the nodes of the network.

3. Application and involved Entities

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

3.1 Application: Podcast distribution in Gwanda

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 Gwanda, Zimbabwe (see Figure 1), the poor radio signal of the area leads the NGO *Practical Action*[1] to use a manpower of 60 cooperators to bring podcasts to the villagers. The cooperators, equipped with portable MP3 players and speakers, physically travel to the NGO office to obtain new podcasts that they play at their assigned villages. 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.

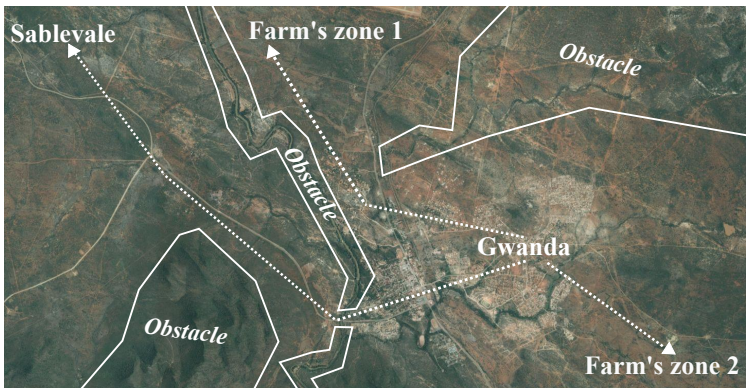

Figure 1: Map of a scenario of application located in a rural area of Gwanda (Zimbabwe). White lines are natural obstacles approximate limits. Podcasts sent from the village of Gwanda to the areas where the cooperators roam have to be routed through specific locations.

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. 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

[1] More information about this programme at http://practicalaction.org/podcasting-gwanda

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.

For these reasons, 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 local volunteers. We also propose to automate the podcast distribution using MADTN. The deployment's cost of the nodes should be low[2], and can be considered as an investment, since the NGO will not need to spend more resources on the podcast distribution.

3.2 Entities of the multi-agent system

PrivHab's goal is to build an intelligent system by improving the itinerary selection of the MADTN agents that carry the messages. The entities involved in this multiagent system (depicted in Figure 2) are listed and explained below.

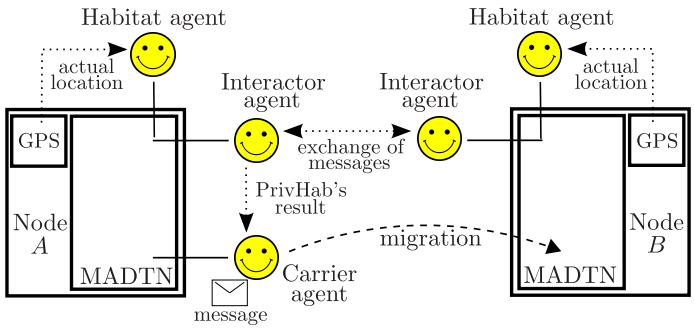

Figure 2: Schema of the multiagent system. Dotted lines depict the main interactions between entities, while slashed lines depict the movement of the agents. The Habitat agent updates the habitat using information from the navigation system (e.g. a GPS receiver). The Interactor agent exchanges PrivHab's 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.

- **Node**: It 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.
- **Message**: It contains the data (e.g. the podcast), it also contains the identifier of both the sender and the receiver, and a list of waypoints that the message has to pass by in order to reach its destination.
- **Habitat agent**: This agent learns and updates the whereabouts of the node (more details about this process in Section 4).
- **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 5).
- **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

[2] Small devices like Raspberry Pi can be acquired by less than 30$/unit.

to reach a location. 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.

4. The Habitat

In this section, we present the cornerstone of our georouting routing protocol: the habitat of a node. We define the concept and show how we model it. Then, we explain the parameters involved in the calculations.

4.1 The usual whereabouts of a node

In the described scenario, each node is a small device that may be carried by the personnel of the NGO, placed in one of their vehicles vehicle or fixed in some strategic place. Therefore, the future movements of every node will be strongly related[3] to the past movements of its carrier. A node carried by a person will probably spend much time in the vicinity of the carrier's home and near his workplace. A node placed in a vehicle will often be inside a particular area. 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 (Sánchez-Carmona et al., 2015).

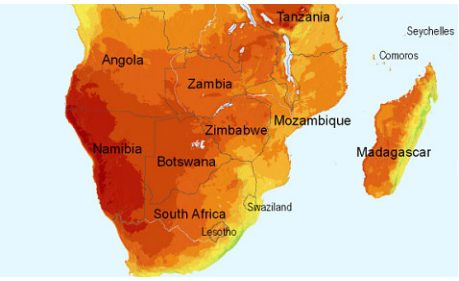

Figure 3: Example of a heatmap. The dark red area corresponds to the area where most time is spent, and the lighter orange and yellow areas correspond to the regions that are rarely visited.

To have a heatmap of a node and its neighbours to decide who is the best choice would be ideal. For example, an agent would want to migrate to a node with a heatmap like the one shown in Figure 3 if it is carrying data destined to the west coast of Namibia, but would not if the data is destined to Mozambique. The heatmap is an extremely accurate representation of the whereabouts of a node. 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 use the habitat (the area where someone is more likely to be found) to model the whereabouts of the nodes by 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.

[3] 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 (Hsu et al., 2008).

4.2 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 $H = (C, R)$.

Every node's habitat has to be updated 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. 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[4] 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 $R = 0$.

$$H_0 = (L_0, 0) \tag{1}$$

First step. Update of the centre

The first step to updating 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 4.2.1). This first step is depicted in Figure 4 (a).

$$C = L * \alpha + C_{old} * (1 - \alpha) \tag{2}$$

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 4 (b).

$$R = d(L, C) * \alpha + R_{old} * (1 - \alpha) \tag{3}$$

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 will increase if L is out of H and will decrease if L is contained by H.

[4] Any cartographic projection can be used.

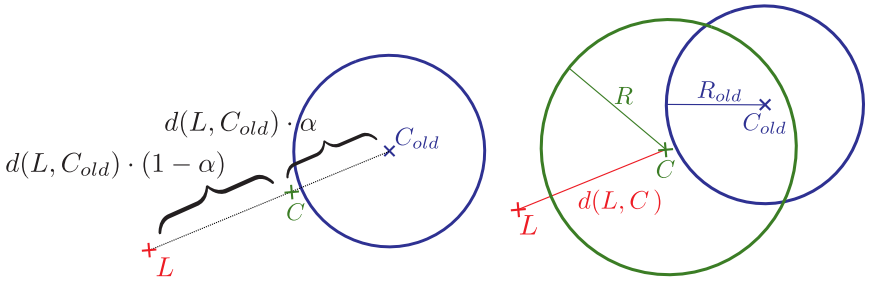

Figure 4: Evolution of the habitat: (a) The new centre point C is calculated averaging the old centre C_{old} and the new location L; (b) The new radius R is calculated averaging the old radius R_{old} and the distance $d(L, C)$ that separates the new location L from the centre point C.

4.2.1 The habitat's time span

The time span that a habitat considers is a very important parameter. For example, a node's habitat that considers only the last 2 hours is very likely to be a small circle around its current location. But if the habitat considers the last 3 days, it will probably be a bigger ellipse containing both its home and workplace.

When the time span of a habitat matches the life-cycle[5] of the nodes of the network, then it will become very useful to predict the areas that the nodes will visit again in the near future.

In order to perform meaningful comparisons between habitats that consider the same time span, PrivHab requires the nodes of the network to know it and to calculate the parameter α using Equation 4. Let ω be the frequency of update of the habitat in updates/hour, and let T be the time span that a habitat has to consider in hours[6]. During the rest of the article, we will assume that a habitat considers a time span of T hours if its parameter α has been calculated this way.

$$\alpha = \frac{2}{T\omega + 1} \tag{4}$$

5. The PrivHab protocol

In this section, we describe the PrivHab routing algorithm. Then, we explain how the usage of homomorphic encryption is crucial to protect nodes' privacy. Later, we introduce the background needed to fully understand the protocol. Following, every message that has to be exchanged during the execution of PrivHab are presented. Finally, we provide some security considerations.

[5] Usual life-cycles of people are a day or a week. People usually move very similarly to how they moved in the previous cycle.

[6] Using a parameter α calculated this way, due to the characteristics of EWMA, the last $T\omega$ locations added to the average tend to weight the $86,47\%$ of the total.

5.1 The PrivHab algorithm

We assume that the waypoints where the message has to pass to reach the destination can always be known or guessed by the NGO local station[7]. They may be known beforehand or may be inferred from the knowledge about the area.

The PrivHab routing algorithm compares two nodes and decides who is the best choice to carry the data towards its destination. The routing algorithm chooses the nodes whose habitat's border is closer to the next waypoint W, prioritizing those nodes whose habitat encloses it. If a waypoint is contained inside two different habitats, then the routing algorithm chooses the node with the smallest one, because the node with the smallest habitat is expected to remain closer, and to be more likely to pass by the waypoint. This algorithm is formalized in Algorithm 1.

Algorithm 1 *PrivHab itinerary selection algorithm*

Input:

H_A, r_A: Habitat of the node A carrying the message and its radius.
H_B, r_B: Habitat of the candidate node B and its radius.
P: Location where the message has to be carried to.

Output:

A or B: The best choice to carry the message towards P.

if $P \in H_B$ **and** $P \notin H_A$ **then**
  return B # P is inside habitat of B and outside habitat of A: select B
else if $P \in H_B$ **and** $P \in H_A$ **and** $r_A \geq r_B$ **then**
  return B # P is inside both habitats: select the smallest habitat
else if $d(P, H_A) > d(P, H_B)$ **then**
  return B # P is outside both habitats: select the nearest habitat to P
else
  return A # If B is not a better choice: by default select A
end if

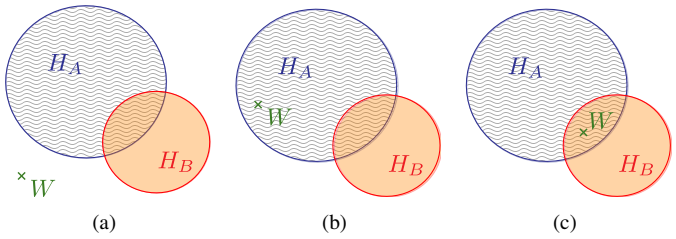

Figure 5: Three possible situations in habitat-based routing: (a) The next waypoint is located outside the two habitats; (b) Only one of the habitats encloses the location of the next waypoint; (c) The two habitats enclose the location of the next waypoint.

[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.

Figure 5 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.

5.2 Nodes' privacy protection: homomorphic encryption

As said in (Hsu et al., 2008), privacy is an important issue in a routing protocol that learns information about the users. In PrivHab, the habitat is used during the routing to decide the best node to carry the data towards its destination. For this reason, PrivHab needs to be secure and do not reveal the habitat information to any other part. On the other hand, waypoints are routing information that has to be known by the routers that take custody of the data. Moreover, although they are not a private information, they must remain hidden.

For this reason, PrivHab uses the Paillier (Zhong et al., 2007) additive homomorphic cryptography to protect nodes' privacy by comparing the habitats while cryptographically protected and avoiding revealing this private information to any other part. 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.

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 is $Pk_A : (n, g)$ and her private key is $pk_A : (\lambda, p, q)$.

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 $E(a+b) = E(a) \cdot E(b) \bmod n^2 = g^{a+b} \cdot (r_1 \cdot r_2)^n \bmod n^2)$, $E(a-b) = E(a)/E(b) \bmod n^2 = g^{a-b} \cdot (r_1/r_2)^n \bmod n^2)$, 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 obtain the message m, Alice computes $D(c) = L(c^\lambda \bmod n^2) = m$.

However, the operations that can be performed in the Paillier cryptosystem are restricted: only addition, subtraction and multiplication by a clear operand are allowed.

5.3 Background: distance comparison

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.

Finally, we calculate $d(H, P) = d(X, P) = \sqrt{(a - x_P)^2 + (b - y_P)^2}$. As $d(X, P)^2 = (a - x_P)^2 + (b - y_P)^2$ can be computed without computing any square root, PrivHab benefits from this to compare the square of the distances from habitats H_A and H_B to P by checking the sign of $d = d(X_A, P)^2 - d(X_B, P)^2$. The schema of this calculation is depicted in Figure 6.

5.4 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 $\sqrt{(x_C - x_P)^2 + (y_C - y_P)^2} < R$. Equivalently, we can check the sign of $d = R^2 - ((x_C - x_P)^2 +$

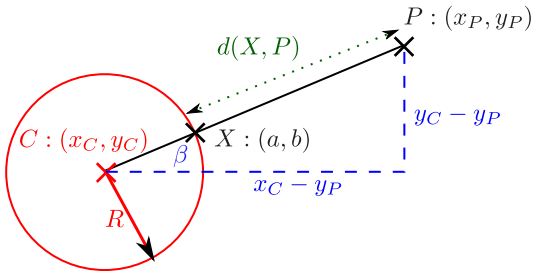

Figure 6: In slashed blue lines, the calculations needed to obtain β. In dotted green lines, distance $d(X, P)$ is calculated before calculating the location of $X : (a, b)$.

$(y_C - y_P)^2)$. P is contained inside the circle if and only if $d > 0$.

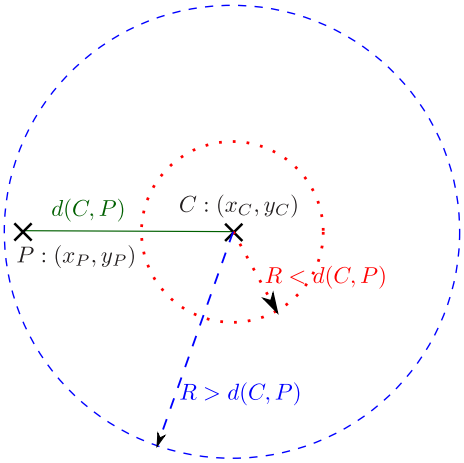

Figure 7: In slashed blue lines, P is located inside the habitat because R is bigger than $d(C, P)$. In dotted red lines, P is located outside the habitat because R is smaller than $d(C, P)$.

This way PrivHab can know if a waypoint is contained inside the habitat using only operations allowed by the Paillier cryptosystem[8]. The schema of this calculation is depicted in Figure 7.

5.5 PrivHab's exchange of messages

Let A be the node that carries a set of messages m_i, with a habitat $H_A : (C_A, R_A)$. Let $W_i : (x_{W_i}, y_{W_i})$ be the next waypoint where each message m_i wants to be carried to, and let B be a neighbour with a habitat $H_B : (C_B, R_B)$. 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 $A \rightarrow B : message$. By the previous definitions, A want to know if B is a better choice to carry each message m_i towards W_i.

The protocol consists of five steps. The first and the second prepare A and B to participate in the protocol, these actions can be done asynchronously. The third lets B obtain the information he needs to calculate the distance from $W[i]$ to the habitat H_B. The fourth performs the comparison between the habitats. And the

[8]Note that this way we avoid computing the square root of a cyphered operand, because this operation cannot be performed.

last step uses the information previously exchanged between the nodes to let A determine if B is a better choice and the message has to be forwarded.

1. Node A calculates $d_{Ai} = d(H_A, W_i)^2$, the square of the distance between its habitat and every W_i ($d_{Ai} = 0$ 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_{Ai} can be performed without using homomorphic encryption.

2. Node B announces[9] 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 subtracts the coordinates of every W_i to the coordinates of C_B. Then, A multiplies both results by the same *nonce* (a random one-use value) using Equations 5 and 6.

$$(E_B(x_{C_B}) / E_B(x_{W_i}))^{nonce} = E_B((x_{C_B} - x_{W_i}) \cdot nonce) \tag{5}$$

$$(E_B(y_{C_B}) / E_B(y_{W_i}))^{nonce} = E_B((y_{C_B} - y_{W_i}) \cdot nonce) \tag{6}$$

Following, A sends to B the results[10] and the coordinates of W_i, the distances d_{Ai}, the radius R_A, and the information B needs to calculate $d_{Bi} = d(H_B, W_i)^2$.

$$A \rightarrow B: \quad \begin{array}{l} \{E_B((x_{C_B} - x_{W_i}) \cdot nonce), E_A(d_{Ai}), E_A(x_{W_i}^2), E_A(2y_{W_i}), E_A(2x_{W_i}), \\ E_B((y_{C_B} - y_{W_i}) \cdot nonce), E_A(y_{W_i}^2), E_A(x_{W_i}), E_A(y_{W_i})\}^i, E_A(R_A) \end{array}$$

4. B decrypts the received subtractions and, for each W_i, computes β_i using Equation 7.

$$\beta_i = \tan^{-1}(((y_{C_B} - y_{W_i}) \cdot nonce) / ((x_{C_B} - x_{W_i}) \cdot nonce)) \tag{7}$$

Node B uses β_i to calculate is the nearest point of H_B to W_i, called $X_i : (a_i = x_{C_B} - R_B \cdot \cos\beta, b_i = y_{C_B} - R_B \cdot \sin\beta)$. Then, B calculates $d(H_B, W_i)^2 = d_{Bi}$, the square of the distance between W_i and X_i using Equation 8.

$$\begin{array}{c} (E_A(a^2) + E_A(b^2)) / (E_A(2x_{W_i})^a \cdot E_A(x_{W_i}^2) \cdot E_A(2y_{W_i})^b \cdot E_A(y_{W_i}^2)) = \\ E_A(a^2 - 2ax_{W_i} - x_{W_i}^2 + b^2 - 2by_{W_i} - y_{W_i}^2) = \\ E_A((a - x_{W_i})^2 + (b - y_{W_i})^2) = E_A(d_{Bi}) \end{array} \tag{8}$$

Following, B calculates the point inclusion of each 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

[9] This announcement can be made by adding this information to the messages exchanged during the neighbour discovery process.
[10] We have used "{" and "}^{i}" to enclose the part of the information that is repeated one time for each message m_i.

nonce values are used to randomize the results. The d_{Ai} factor is used to blur[11] the point inclusion test and the comparison of radius.

$$(E_A(R_B^2) \cdot E_A(d_{Ai}))/(E_A(d_{Bi}))^{nonce} = E_A((R_B^2 + d_{Ai} - d_{Bi}) \cdot nonce) \tag{9}$$

$$(E_A(d_{Ai}))/(E_A(d_{Bi}))^{nonce} = E_A((d_{Ai} - d_{Bi}) \cdot nonce) \tag{10}$$

$$(E_A(R_A) \cdot E_A(d_{Ai})^{R_{Bi}})/(E_A(R_{Bi}))^{nonce} = E_A((R_A + d_{Ai} \cdot R_B - R_B) \cdot nonce) \tag{11}$$

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

$$B \rightarrow A: \quad \{Random\_order(E_A((R_A + d_{Ai} \cdot R_B - R_B) \cdot nonce), E_A((d_{Ai} - d_{Bi}) \cdot nonce), E_A((R_B^2 + d_{Ai} - d_{Bi}) \cdot nonce))\}^i$$

5. Node A decrypts the three received values to learn the result of PrivHab's execution. B is considered a better choice to carry m_i towards W_i if the three decrypted values are equal or greater[12] than 0.

5.6 Security considerations

A secure multi-party computation (Goldreich, 1998) consists in computing a function on any input, on a network where different participants hold each input. A protocol is considered secure if it ensures that no more information is revealed to a participant than what can be inferred from that participant's input and the computed output. PrivHab has been designed to reveal only the result of the comparison and the inferences that can be deduced from this output. The next paragraphs consider a passive adversary situation, where one participant executes the protocol and then makes inferences to obtain knowledge about the other participant's inputs.

On one hand, node A only knows if H_B is better or worse than H_A. Then, A can 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 $W_i \in H_B$. The information that A can learn and infer is presented in Table 1.

On the other hand, node B cannot even know the result of the execution, because it is computed while encrypted. Note that A uses the result of the execution of PrivHab to decide if a message m_i has to be forwarded to B or not, but A may use too any other information to make this decision, so receiving or m_i is not enough for B to infer the result of the comparison. For this reason, B cannot infer anything about H_A. Besides, maintaining W_i hidden to B 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. The only information that is revealed to B during the execution of the protocol is β. Note that 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 if he does not receives the message, and the protocol will not be executed again between the same participants. Therefore, B can not

[11] If $d_{Ai} > d_{Bi}$, then the best choice is B, and the result of the point inclusion test and the comparison of radius are not needed.

[12] PrivHab checks several times if an operand ρ is negative. As ρ is an element of $\mathbb{Z}_n$, to check this condition, we ensure that n is sufficiently large and that all values ρ we will use are $\rho \leq n/2$. Then, we can consider that $\rho > n/2 \iff \rho < 0$.

A **knows**		A **infers**		
Input	**Output**	$d_A \leftrightarrow d_B$	$P \leftrightarrow H_B$	$r_A \leftrightarrow r_B$
$P \in H_A$	B	$d_A = d_B = 0$	$P \in H_B$	$r_A \geq r_B$
	A	$d_A \leq d_B$	$P \notin H_B$ or $r_A < r_B$	
$P \notin H_A$	B	$d_A \geq d_B$	Nothing	Nothing
	A	$d_A < d_B$	$P \notin H_B$	Nothing

Table 1: Knowledge obtained by A. If B a better choice, then A infers that H_B is closer to location P than H_A. Node A also infers that H_B is smaller than H_A if P is contained inside H_A. When B is a worse choice, then A infers d_B is larger than d_A, but cannot know if H_B is bigger or smaller than H_A.

relate W_i with any node neither use β to triangulate its location. The information that B can learn and infer is presented in Table 2.

B **knows**	B **learns**		B **infers**
Output	About P	About d_B	$d_A \leftrightarrow d_B$
Message received	$P : (P_x, P_y)$	d_B	Nothing
Message do not received	β	Nothing	Nothing

Table 2: Knowledge obtained by B at the end of the protocol. Node B cannot infer if it is a better candidate than A. B receives the coordinates of P with the message. If the message is not sent, B only learns β.

Finally, an active attacker can try to learn things about the other part's habitat by producing chosen-destination arbitrary messages and repeatedly executing PrivHab. 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. An attacker A can try to uncover the area covered by H_B by executing PrivHab repeatedly, benefiting from the fact that there is no way for a node B to tell apart a truthful execution of PrivHab from an untruthful one because nodes always operate with encrypted data. 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. Depending on the configuration of the network, 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 and unuseful.

6. Experiments and Results

In this section, we study the computational and communication overhead introduced by PrivHab. Then, we explain how we have modelled the scenario we have chosen to evaluate PrivHab. Finally, we provide the obtained results.

6.1 Implementation measurements

As a proof-of-concept we have developed and deployed an implementation of the presented protocol on three Raspberry Pi boards as the one depicted in Figure 8.

Figure 8: 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.

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[13]. We have repeated the tests five times. We have measured the average time needed to make the calculations and to exchange all the messages. The obtained results have been incorporated to the simulations.

Key length	Time (ms)	Overhead 10MB (%)	Overhead 20MB (%)
512 bits	574.2 ± 0.7	3.48	1.77
1024 bits	$3,977.0 \pm 31.7$	24.07	12.28
2048 bits	$25,031.5 \pm 69.8$	151.49	77.29

Table 3: Average execution time of PrivHab using different key lengths. The overhead is the extra amount of time needed to send 10MB or 20MB.

As can be seen in Table 3, 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 half a second. Meaning an overhead of less than 4% when sending messages larger than 10MB. The execution time increases to 3.9 seconds when using

[13] 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.

keys of 1024 bits[14]. Given the average length of connectivity windows in remote village scenarios presented in (Grasic and Lindgren, 2014), this overhead is acceptable. When using keys of 2048 bits, the execution time is high.

6.2 Modelling

We have modelled the scenario presented in Section 3. In our model, 60 cooperator nodes implement a mobility pattern that takes into account home's and work's locations and a daily life-cicle. Agents carrying podcasts of 15-20MB are injected in the network by the NGO office, located in the village of Gwanda. Every podcast is sent to one cooperator chosen randomly. We assume that the NGO office knows the area assigned to each cooperator and the necessary waypoints to reach each one. After executing PrivHab, the Carrier agents always migrate to nodes that are considered better choices. We have modelled the overhead introduced by PrivHab considering that the Interactor agents needs 3.9s to perform the exchange of messages.

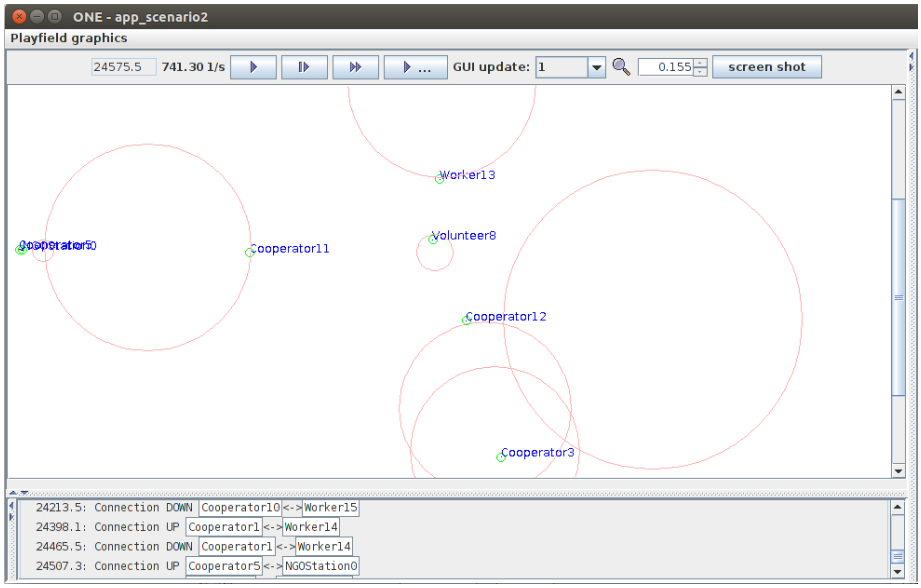

Figure 9: Snapshot of one of the simulation using The ONE. Nodes are depicted using a green circle with a blue label, habitats are depicted with red circles. The background map of Gwanda has been disabled to improve the visibility of the figure.

In order to obtain conclusive results, we have compared the performance of PrivHab with a bench-mark of well-known routing protocols used in (Musolesi and Mascolo, 2009): Prophet, Binary Spray & Wait (L=40), Epidemic and Random. We have added two routing protocols to this set: MaxProp and First Contact. All simulations have been performed using *The Opportunistic Network Simulator* (The ONE, Figure 9) (Keränen et al., 2009), and have been repeated twenty times using different random seeds.

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

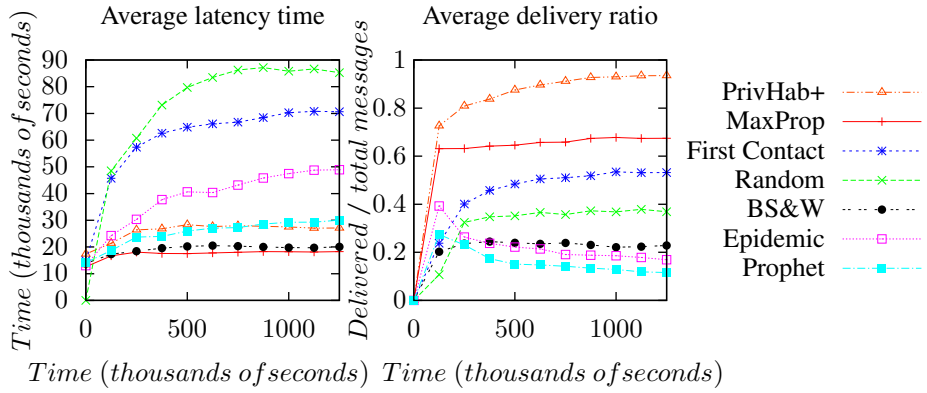

Figure 10: Results of the simulations. Latency and delivery ratio.

6.3 Simulation results

The performance of all the compared protocols is presented in Figure 10, that shows the latency and delivery ratio, and in Table 4, that shows the average number of aborted relays and dropped podcasts, and the network overhead, calculated as the relation between the number of the relays done and the number of delivered podcasts.

Single-copy protocols, as Random and First Contact, do not fill up the buffers. Therefore, they obtain medium delivery ratios because nodes are not forced to drop podcasts. However, their decision making is poor, podcasts are usually forwarded to nodes that could not bring the podcasts closer to their destination, so podcasts last longer on the network. For this reason, their latency is high and they produce an enormous amount of aborted relays. Flooding-based protocols, as Epidemic and Prophet, generate en enormous network overhead that fill the buffers early. Therefore, they obtain medium latencies but low delivery ratios because almost all nodes effort while forwarding podcasts is wasted, usually because the podcasts are dropped before approaching their destination. BS&W and MaxProp perform well in terms of latency. But their performance in terms of delivery ratio is totally opposed. Binary Spray & Wait, performs poor in terms of delivery ratio because of his epidemic-style spread, while MaxProp obtains a high delivery ratio because his dropping policy based on probabilities of delivery manages to drop less messages. PrivHab takes the best decisions because it is the only protocol that takes into account both the pathway to the destination and the mobility patterns of the neighbours, and manages to spread the podcasts towards their destination. For this reason, PrivHab obtains the lowest network overhead and latency latency of the single-copy protocols.

PrivHab delivers more data to its destination. Besides, it does it faster than all other protocols except BS&W and MaxProp, and it consumes fewer network resources to do so. We can state that PrivHab is the protocol that suits better to any scenario with characteristics like the presented one.

7. Conclusions

The habitat models node's whereabouts based on the idea of the time span. It is useful to decide what node is a better choice to carry the data towards its destination. In this paper, we present PrivHab, a privacy-respectful

Protocol	Dropped podcasts	Overhead	Aborted relays
Epidemic	197,030	964.66%	114,380
Prophet	130,647	855.96%	382,557
Maxprop	9,929	65.91%	252,023
BS&W-40	33,373	36.66%	114,380
Random	396	112.40%	375,200
First Contact	75	46.73%	217,280
PrivHab+	128	9.68%	51,343

Table 4: Obtained results in terms of network overhead and number of dropped podcasts. Network overhead is calculated as the relation between the number of the relays done and the number of delivered podcasts

multiagent system for itinerary-selection based on MADTN that uses the habitats to make routing decisions. PrivHab also makes use of homomorphic cryptography techniques to preserve nodes' privacy.

We have presented a podcast distribution application in Gwanda, based on the real work of the NGO *Practical Action*. PrivHab's characteristics make him ideal to operate in scenarios where nodes mobility patterns are complex, but non-random, where lots of hops are needed to reach the destination of the messages from their source because of the long distances, 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 improve the circular model of habitat, to study the best strategies when using PrivHab, to make PrivHab compatible with the usage of pseudonym generator mechanisms and to enhance PrivHab to compare simultaneously three or more habitats. We also plan to study the performance of PrivHab in different scenarios based on real applications.

8. Acknowledgment

This work has been partially funded by the Ministry of Science and Innovation of Spain, under the reference project TIN2014-55243-P, by the Catalan Government under the reference project 2014SGR691 and by the Autonomous University of Barcelona under the reference number 472-03-01/2012.

8.1 Vitae

Mr. Adrián Sánchez Born in Terrassa, Barcelona. He received his degree in Computer Science (5 year programme) at the Autonomous University of 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 is actually a PhD student at the Department of Information and Communications Engineering (dEIC).

Dr. Sergi Robles received his PhD in Computer Science from the Autonomous University of Barcelona. He is an associate professor in the Department of Information and Communications Engineering at the Autonomous University of 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.

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.

(a) (b) (c)

Figure 11: The authors: (a) Adrián Sánchez-Carmona (b) Sergi Robles; (c) Carlos Borrego.

9. References

Boldrini, C., Conti, M., Jacopini, J., and Passarella, A., 2007. 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*, pages 1–12. doi:10.1109/WOWMOM.2007.4351716.

Borrego, C., Castillo, S., and Robles, S., 2014. Striving for sensing: Taming your mobile code to share a robot sensor network. *Information Sciences*, (0). ISSN 0020-0255. doi:http://dx.doi.org/10.1016/j.ins.2014.02.072.

Cheng, P.-C., Lee, K., Gerla, M., and HÃd'rri, J., 2010. GeoDTN+Nav: Geographic DTN Routing with Navigator Prediction for Urban Vehicular Environments. *Mobile Networks and Applications*, 15(1):61–82. ISSN 1383-469X. doi:10.1007/s11036-009-0181-6.

Costa, P., Mascolo, C., Musolesi, M., and Picco, G., 2008. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 26(5):748–760. ISSN 0733-8716. doi:10.1109/JSAC.2008.080602.

Goldreich, O., 1998. Secure Multi-Party Computation.

Grasic, S. and Lindgren, A., 2014. Revisiting a remote village scenario and its DTN routing objective. *Computer Communications*, 48:133–140. ISSN 0140-3664. doi:10.1016/j.comcom.2014.04.003.

Hsu, W., Dutta, D., and Helmy, A., 2008. CSI: A Paradigm for Behavior-oriented Delivery Services in Mobile Human Networks. *CoRR*, abs/0807.1153.

Karlof, C. and Wagner, D., 2003. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures.

Karp, B. and Kung, H. T., 2000. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 243–254. ACM, New York, NY, USA. ISBN 1-58113-197-6. doi:10.1145/345910.345953.

Keränen, A., Ott, J., and Kärkkäinen, T., 2009. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST, New York, NY, USA. ISBN 978-963-9799-45-5.

Kuiper, E. and Nadjm-Tehrani, S., 2011a. Geographical Routing With Location Service in Intermittently Connected MANETs. *Vehicular Technology, IEEE Transactions on*, 60(2):592–604. ISSN 0018-9545. doi:10.1109/TVT.2010.2091658.

Kuiper, E. and Nadjm-Tehrani, S., 2011b. Geographical Routing With Location Service in Intermittently Connected MANETs. *Vehicular Technology, IEEE Transactions on*, 60(2):592–604. ISSN 0018-9545. doi:10.1109/TVT.2010.2091658.

LeBrun, J., Chuah, C.-N., Ghosal, D., and Zhang, M., 2005. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, volume 4, pages 2289–2293 Vol. 4. ISSN 1550-2252. doi:10.1109/VETECS.2005.1543743.

Leguay, J., Friedman, T., and Conan, V., 2007. Evaluating MobySpace-based routing strategies in delay-tolerant networks. *Wireless Communications and Mobile Computing*, 7(10):1171–1182. ISSN 1530-8677. doi:10.1002/wcm.520.

Li, Y., Lai, T.-H., Liu, M. T., Sun, M.-T., and Yang, J., 2006. DTGR: Disruption-Tolerant Geographic Routing for Wireless Ad Hoc Networks. *Simulation*, 82(6):399–411.

Martínez, R., Castillo, S., Robles, S., Sánchez, A., Borrell, J., Cordero, M., Viguria, A., and Giuditta, N., 2013. Mobile-Agent Based Delay-Tolerant Network Architecture for Non-Critical Aeronautical Data Communications. In Springer, editor, *In 10th International Symposium on Distributed Computing and Artificial Intelligence*.

Mei, A., Morabito, G., Santi, P., and Stefa, J., 2011. Social-aware stateless forwarding in pocket switched networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 251–255. ISSN 0743-166X. doi:10.1109/INFCOM.2011.5935076.

Musolesi, M. and Mascolo, C., 2009. CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks. *Mobile Computing, IEEE Transactions on*, 8(2):246–260. ISSN 1536-1233. doi:10.1109/TMC.2008.107.

Sánchez-Carmona, A., Robles, S., and Borrego, C., 2015. Podcast Distribution on Gwanda Using PrivHab: A Multiagent Secure Georouting Protocol. In Bajo, J., Hernández, J. Z., Mathieu, P., Campbell, A., Fernández-Caballero, A., Moreno, M. N., Julián, V., Alonso-Betanzos, A., Jiménez-López, M. D., and Botti, V., 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. ISBN 978-3-319-19628-2. doi:10.1007/978-3-319-19629-9_4.

Zhong, G., Goldberg, I., and Hengartner, U., 2007. Louis, Lester and Pierre: Three Protocols for Location Privacy. In Borisov, N. and Golle, P., editors, *Privacy Enhancing Technologies*, volume 4776 of *Lecture Notes in Computer Science*, pages 62–76. ISBN 978-3-540-75550-0. doi:10.1007/978-3-540-75551-7_5.