This is the accepted versión of: Sánchez-Carmona, A.; Robles, S. and Borrego, C. "Podcast distribution on Gwanda using PrivHad : multiagent secure georouting protocol" in Trends in practical applications of agents, multi-agent system and sustainability (Advances in intelligent systems and computing ; 372). The final publication is available at Springer via DOI 10.1007/978-3-319-19629-9_4

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

Adrián Sánchez-Carmona, Sergi Robles, and Carlos Borrego

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

**Abstract.** We present PrivHab, a georouting protocol that improves multiagent systems itinerary decision-making. PrivHab uses the mobility habits 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. We have evaluated PrivHab under the scope of a realistic podcast distribution application in remote rural areas. The PrivHab protocol is compared with a set of well-known delay-tolerant routing algorithms and shown to outperform them.

## 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.

E-agriculture applications, usually targeting rural areas, are very likely to deal with challenges like a sparse population, with the receivers of the information far away from each other, a bad, non-existent or expensive telephony coverage and, especially, a lack of data communication networks are the most common ones.

We propose to use PrivHab to reduce the digital divide in developing countries by distributing podcast radio programs using Mobile Agent based Delay Tolerant Networking [4]. MADTN uses mobile agents to perform a store-carry-and-forward strategy, and it is designed to operate in absence of simultaneous end-to-end paths.

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

## 2 Scenario of application

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.

In Gwanda, Zimbabwe, the poor radio signal of the area leads the NGO *Practical Action*[2] 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. 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. The deployment's cost of this network 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.

Between the NGO and the local radio stations there could be barriers that nodes carrying the data can not cross, as the Mtshabezi River, and there could be some locations that are very likely to have a higher density of nodes, as the markets. Therefore, data should try to follow paths that take advantage of this knowledge. For this reason, we propose a geographical routing protocol where the sender defines a set of waypoints where the data has to pass by in order to reach its destination.

## 3 A habitat-based itinerary

A **habitat** is defined as the area where a node is more likely to be found. It is based on the assumption of social-based routing protocols that future mobility of a node will be related to its near past mobility [3]. The heatmap (Figure 1) is an extremely accurate habitat representation.


Fig. 1: Heatmap of a node. 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.

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

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

However, creating and maintaining this data is a resource consuming task that does not fit well with the small devices of the proposed network. Therefore, we propose to model each nodes' habitat using a simple geometric shape. 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.

### 3.1 Circular model of habitat

We model the habitat using a circle. Each habitat $H$ is characterized by two elements: a centre point $C = (x, y)$ and a radius $R$. 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 $\omega$ 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.

**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 $R = 0$.

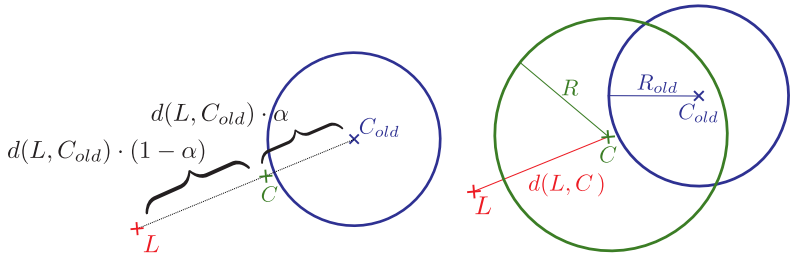


Fig. 2: Evolution of the habitat: (a) The new centre point $C$ is calculated averaging the old centre $C_{old}$ and the new location $L$; (b) The new radius $R$ is calculated averaging the old radius $R_{old}$ and the distance $d(L, C)$ that separates the new location $L$ from the centre point $C$.

**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 $\alpha$. This first step is depicted in Figure 2 (a).

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

**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 2 (b).

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

### 3.2 The motion common cycle

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 common motion cycle, and it has to be known 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%.

## 4 The PrivHab protocol

The PrivHab routing algorithm compares two nodes and decides who is the best choice to carry the data towards its destination[4]. The routing algorithm chooses the nodes whose habitat's border is closer to the next waypoint, 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.

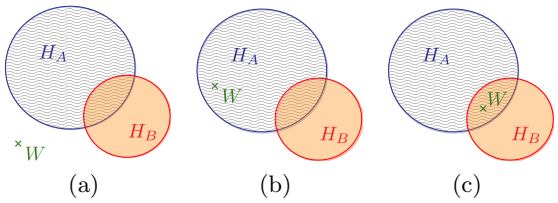


Fig. 3: Three possible situations in habitat-based routing: (a) The next waypoint is located outside the two habitats; (b) Only one of the habitats encloses the location of the next waypoint; (c) The two habitats enclose the location of the next waypoint.

Figure 3 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 $H_A$.

### 4.1 Nodes' privacy

At [1], Boldrini *et al.* recognize that privacy is an important issue in a routing protocol. Therefore, PrivHab needs to be secure and do not reveal the habitat

[4] We assume that the approximate locations the data has to pass to reach the destination can be known or guessed by the sender.

information to the other part. For this reason, PrivHab uses the Paillier [6] additive homomorphic cryptography 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.

### 4.2 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. 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.

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 $d_A$ can be performed without using homomorphic encryption.
2. Node $B$ announces to $A$ the centre $C_B : (x_{C_B}, y_{C_B})$ of its habitat.

   $B \rightarrow A$: $E_B(x_{C_B}), E_B(y_{C_B})$
3. Node $A$ 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).

$$(E_B(x_{C_B})/E_B(x_{W[i]}))^{nonce} = E_B((x_{C_B} - x_{W[i]}) \cdot nonce) \tag{3}$$

$$(E_B(y_{C_B})/E_B(y_{W[i]}))^{nonce} = E_B((y_{C_B} - y_{W[i]}) \cdot nonce) \tag{4}$$

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

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

4. $B$ decrypts the received subtractions and computes $\beta$.

$$\beta = \tan^{-1}(((y_{C_B} - y_{W[i]}) \cdot nonce)/((x_{C_B} - x_{W[i]}) \cdot nonce)) \tag{5}$$

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

$$(E_A(a^2) + E_A(b^2))/(E_A(2x_{W[i]})^a \cdot E_A(x_{W[i]}^2) \cdot E_A(2y_{W[i]})^b \cdot E_A(y_{W[i]}^2)) =$$

$$E_A(a^2 - 2ax_{W[i]} - x_{W[i]}^2 + b^2 - 2by_{W[i]} - y_{W[i]}^2) =$$

$$E_A((a - x_{W[i]})^2 + (b - y_{W[i]})^2) = E_A(d_B) \tag{6}$$

Following, $B$ calculates the point inclusion of $W[i]$ in $H_B$ using Equation 7, the comparison of distances using Equation 8, and the comparison of radius using Equation 9. This time, three different *nonce* values are used to randomize the results. The $d_A$ factor is used to blur[5] the point inclusion test and the comparison of radius.

$$(E_A(R_B^2) \cdot E_A(d_A))/(E_A(d_B))^{nonce} = E_A((R_B^2 + d_A - d_B) \cdot nonce) \quad (7)$$

$$(E_A(d_A))/(E_A(d_B))^{nonce} = E_A((d_A - d_B) \cdot nonce) \quad (8)$$

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

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

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

5. Node $A$ decrypts the three received values. $B$ is considered a better choice if the three decrypted values are equal or greater[6] than 0.

### 4.3 A Multiagent System

PrivHab is executed under the MADTN framework. The agents involved in this multiagent system are listed below.

- **Habitat agent**: The agent that performs the operations described in Section 3.1 to calculate and update the habitat of the node. This agent also periodically informs the Carrier agent of the current location to track if the node had approached enough the current waypoint.
- **Interactor agent**: The agent that performs the exchange of messages described in Section 4.2. This agent informs the Carrier agent with the result when the exchange of messages has finished.
- **Carrier agent**: A proactive agent that carries the data towards its destination. After the execution of PrivHab, it makes the decision of migrating, being cloned, or staying at the current node.

## 5 Experiments and Results

As a proof-of-concept we have deployed an implementation of the presented protocol on three Raspberry Pi boards. We have used them to measure the overhead that PrivHab adds to every transaction.

[5] 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.

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

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[7]. 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.

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 0.57 seconds. Meaning an overhead of less than 3.48% when sending messages larger than 10MB. The execution time increases to $3.97 \pm 0.03$ seconds when using keys of 1024 bits. Given the average length of connectivity windows in remote village scenarios presented in [2], this overhead is acceptable. When using keys of 2048 bits, the execution time is too high ($25,031.5 \pm 69.8$ seconds).

### 5.1 Modelling and simulations

The scenario we have used in all the simulations is the one presented in Section 2. We have compared the performance of PrivHab with a bench-mark of well-known routing protocols used in [5]: 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), and have been repeated twenty times using different random seeds.

The performance of all the compared protocols is presented in Figure 4. 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, and 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. 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 takes into account both the pathway to the destination and the mobility patterns of the neighbours, and obtains the lowest network overhead and latency latency of the single-copy protocols because the spread is directed towards the destination.

[7] 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.

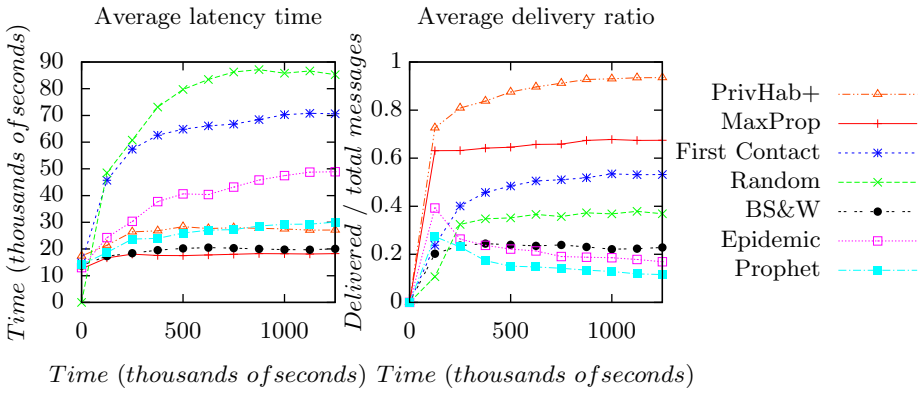


Fig. 4: Results of the simulations. Latency and delivery ratio.

## 6 Conclusions

The habitat models node's whereabouts based on the common motion cycle. It is used to decide what nodes are good choices to carry the data towards its destination. PrivHab uses homomorphic cryptography to preserve nodes' privacy.

## 7 Acknowledgment

This work has been partially funded by the Ministry of Science and Innovation of Spain, under the reference project TIN2010-15764 and by the Catalan Government under the reference project 2014SGR691.

## References

1. 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*, pages 1–12, June 2007.
2. S. Grasic and A. Lindgren. Revisiting a remote village scenario and its dtn routing objective. *Computer Communications*, 48:133140, 2014.
3. P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *Mobile Computing, IEEE Transactions on*, 10(11):1576–1589, Nov 2011.
4. 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 *10th International Symposium on Distributed Computing and Artificial Intelligence*, May 2013.
5. M. Musolesi and C. Mascolo. Car: Context-aware adaptive routing for delay-tolerant mobile networks. *Mobile Computing, IEEE Transactions on*, 8(2):246–260, Feb 2009.
6. G. Zhong, I. Goldberg, and U. Hengartner. Louis, lester and pierre: Three protocols for location privacy. In N. Borisov and P. Golle, editors, *Privacy Enhancing Technologies*, volume 4776 of *Lecture Notes in Computer Science*, pages 62–76. 2007.