Journal of
Sensor and Actuator Networks

Article

Blossom: Cluster-Based Routing for Preserving Privacy in Opportunistic Networks

Benedikt Kluss , Samaneh Rashidibajgan * and Thomas Hupperich

Department of Information Systems, University of Münster, 48149 Münster, Germany
* Correspondence: samaneh.rashidibajgan@wi.uni-muenster.de

Abstract: Opportunistic networks are an enabler technology for typologies without centralized infrastructure. Portable devices, such as wearable and embedded mobile systems, send relay messages to the communication range devices. One of the most critical challenges is to find the optimal route in these networks while at the same time preserving privacy for the participants of the network. Addressing this challenge, we presented a novel routing algorithm based on device clusters, reducing the overall message load and increasing network performance. At the same time, possibly identifying information of network nodes is eliminated by cloaking to meet privacy requirements. We evaluated our routing algorithm in terms of efficiency and privacy in opportunistic networks of traditional and structured cities, i.e., Venice and San Francisco by comparing our approach against the PRoPHET, First Contact, and Epidemic routing algorithms. In the San Francisco and Venice scenarios, Blossom improves messages delivery probability and outperforms PRoPHET, First Contact, and Epidemic by 46%, 100%, and 160% and by 67%, 78%, and 204%, respectively. In addition, the dropped messages probability in Blossom decreased 83% compared to PRoPHET and Epidemic in San Francisco and 91% compared to PRoPHET and Epidemic in Venice. Due to the small number of messages generated, the network overhead in this algorithm is close to zero. The network overhead can be significantly reduced by clustering while maintaining a reliable message delivery.

Keywords: opportunistic network; privacy-preserving routing; cluster-based routing; cloaking

check for updates

Citation: Kluss, B.; Rashidibajgan, S.; Hupperich, T. Blossom: Cluster-Based Routing for Preserving Privacy in Opportunistic Networks. *J. Sens. Actuator Netw.* **2022**, *11*, 75. https://doi.org/10.3390/jsan11040075

Academic Editor: Antonio Celesti

Received: 14 September 2022
Accepted: 10 November 2022
Published: 16 November 2022

1. Introduction

In recent years, smart and wearable devices have rapidly developed to gain higher processing power and storage space. In combination with the establishment of the Internet of Things (IoT) paradigm [1], the Internet of Wearable Things (IoWT) emerged [2].

Picture the concept of smart cities with numerous entities, including homes, vehicles, buildings, wearables, air stations monitoring, etc. These entities, utilized in various fields of applications such as healthcare, environmental, automation, industrial, and emergency care, have attracted millions of mobile, i.e., vehicles, smartphones, wearables, and portable devices and nonmobile, e.g., home and buildings, nodes spreading over the cities capable of data transition [3,4]. These nodes are equipped with telecommunication senders and receivers, connecting via short-range, e.g., Bluetooth, and long-range communication, e.g., Wi-Fi, and exchanging data to shape the network [5].

Due to their high popularity and degree of mobility, wearable devices are carried by individuals in daily routine activities, utilizing them as the nodes of a network for transmitting and exchanging data.

As an IoT enabler technology, opportunistic networks support a decentralized infrastructure and, therefore, are suitable for connecting such devices [3,6–8]. Opportunistic networks (OppNets) take advantage of the mobility of the nodes as well as the social communication network of individuals who carry these nodes to exchange messages [9].

OppNets, unlike traditional networks, do not need a direct path between the receiver and the sender to send the message [10]. In this mechanism, when nodes are in the communication range, they exchange messages and carry them in their buffers until another node gets in range and the message is forwarded. In each message transmission attempt, a message is delivered to a closer node or a node with a higher chance of delivering it to the final destination. In these networks, the store-carry-forward mechanism is used [11]. OppNets are self-organized, delay-tolerant, and dynamic with a flexible topology and frequent link disruptions [12].

Forwarding a message to all available neighbors is a straightforward method. Although it reduces message latency, it poses problems to network performance, such as increasing network overhead, nodes' buffer, and battery consumption [13]. Thus, different routing algorithms have been proposed, addressing these concerns to efficiently send messages to available and suitable nodes, i.e., neighbors. However, finding a trade-off between message delivery and network overhead is still an open issue.

Nevertheless, the participation of individuals in shaping an OppNet, knowing that there is a possibility of revealing the locations, communication linkages, and identity of itself as a node and the friends (neighboring nodes), may result in refusing [14]. Therefore, it is necessary to provide privacy for users in OppNets. To achieve this goal, the identities and locations of the network participants should remain anonymous. Due to the unstable structure of these networks, it is not practical for traditional methods in these networks [15].

To overcome the first issue and decrease the network overhead, we have used the advantages of clustering. As a result, a limited number of messages are sent to the network. Cluster analysis aims to find homogeneous groups of clusters.

This paper introduces a new privacy-preserving routing algorithm called Blossom that dynamically builds clusters based on the node's directions. Blossom is designed for domains of urban traffic control and Vehicular Networks based on OppNets, where pedestrians, cars, and public transport move in different directions. In such networks, smart devices (individuals' smartphones and embedded data transporter in vehicular) can connect and forward traffic messages. Therefore, the individuals and cars can move on the most optimal routes, and in case of an accident, the rescue team can be notified immediately and attend the accident site.

Nodes (vehicles and pedestrians) moving in the same direction are clustered and shape a group as a cluster. Consequently, the generated message from a node is sent to group-based receivers (nodes in a cluster) rather than node-based. Blossom focuses on hierarchical clustering. This agglomerative technique is used for Blossom to avoid the step of first merging all considered contributors in advance before performing the actual cluster analysis.

We have also utilized cloaking to provide privacy for both location and identity. The cloaking algorithm is designed to find a spatial or temporal space that satisfies a minimum of k users to decrease the accuracy of the location data to provide anonymity for the contributors [16]. In cloaking, anonymity is supported, and location information cannot be instrumented to re-identify a network contributor. The network's nodes take care of this task, remove identifiers, and apply the cloaking concept afterward.

We compared Blossom to the state-of-the-art; namely, First Contact [17], Epidemic [18], and PRoPHET [19] algorithms, in terms of message delivery probability, dropped messages probability, and network overhead. The simulation results prove that the proposed structure works better. Due to the scope of this paper, it will contain only privacy-preserving techniques. Other security issues for OppNets, such as trust management, cooperation, authentication and access control, confidentiality, and data integrity [20], are out of the scope of this paper.

In addition, we have analyzed the network performance while malicious nodes are present. We compared the network without malicious nodes with networks where 10%, 20%, and 50% of nodes are selfish and drop the messages in their buffer, e.g., due to malware.

The main contributions of this work are as follows:

- Proposing an innovative routing algorithm according to clustering nodes in order to improve the network performance in terms of messages delivery probability, dropped message probability, and network overhead;
- Comparing the proposed algorithm performance with the First Contact, Epidemic, and PRoPHET algorithms and results validation;
- Analyzing the network performance with the presence and absence of malicious nodes in the network;
- Preserving node's privacy by cloaking.

The rest of this paper is organized as follows: In Section 2, we provided a brief summary of related works in the literature. We then describe the proposed structure for Blossom in Section 3. Method, simulation result, and discussion are discussed in Section 4. In the end, the research is concluded in Section 5.

2. Related Work

In recent years, substantial research has addressed secure routing issues in OppNets. One of the most critical aspects of security is privacy. The main concerns in proposed structures regarding privacy in the literature are identity privacy and location privacy. Some existing protocols are listed in Table 1 with their advantages and disadvantages, and are as follows.

In [21], a privacy protection routing algorithm based on utility value was proposed. The authors have used Bloom Filter to obscure the friends' list and the node's utility values. This paper proposed a self-organized key management scheme that included an identity authentication technique based on the zero-knowledge proof of the elliptic curve and a key agreement scheme based on threshold cryptography. Each node contained a certificate library, including its authentication efficiency and success rate. Moreover, node identity was proved by other nodes.

PRIVO was introduced in [22]. Paillier Homomorphic Encryption had been used in this scheme to provide anonymization and attribute privacy for nodes in Delay-Tolerant Networks. The simulation results of the paper confirmed that, on average, cryptography costs for this algorithm are blowing 1%.

The authors in [23] introduced an approach that used an optimized version of Millionaire's Problem to provide node security. By this approach, nodes can trust each other without revealing sensitive information. They proposed four different privacy-preserving forwarding protocols for OppNets.

Privacy-Preserving Probabilistic Prediction-based Routing (4PR) protocol was introduced in [24]. It compared aggregated information about communities and calculated the probability that at least one node in a community can be the destination or not.

PIDGIN was introduced in [25] to provide privacy-preserving for network participants in OppNets. The messages' sender considered some policies, and the only authorized node that satisfies fine-grained policies can access messages.

A privacy-preserving protocol for utility-based routing (PPUR) was introduced in [26] for delay-tolerant networks. When nodes are in the communication range, each generated and collected information anonymously. Then, information is forwarded to a trusted authority, sending back the secure routing path. They have used bilinear mapping technology to generate bilinear parameters with the security parameter; hashing and symmetric cryptography are also used in this approach.

A privacy-preserving distance-based incentive method to avoid selfish behavior and provide location privacy was presented in [27]. This paper aimed to provide nodes' location confidentiality, message integrity, and accuracy of reputation computation. They have used multiparty computation and homomorphic encryption to achieve this goal.

PEON was proposed in [28] to provide privacy in OppNets. In this algorithm, anonymous communication and rerouting messages via peer nodes are used to conceal the link between the sender and receiver of a message.

The privacy-preserving history-based (PPHB) algorithm was introduced in [29]. In PPHB, nodes construct a nickname and a polynomial such that the nickname is the root of the polynomial. Each node's polynomial is multiplied by the frequently visited nodes' polynomials. To evaluate whether a node is suitable for carrying a message, check if the receiver's nickname can make the node polynomial zero. It will be zero, meaning the node probably visits the neighbor or can make the message closer to the destination.

ePRIVO was proposed in [30] to provide privacy for vehicular delay tolerant networks. ePRIVO produces a time-varying neighboring graph based on the vehicular network. The edge of the graph resembles the neighbor relationship in the network. Therefore, nodes can make routing decisions without knowledge of private information.

The privacy-preserving exchange-based routing protocol (PPERP) was proposed in [31]. This algorithm classifies nodes into delivery and non-delivery nodes. Each node can calculate its routing utility without depending on a trusted third party. The bilinear mapping technique is also used to guarantee security.

In [32], authors proposed to use a compelling lightweight cryptographic encryption algorithm to provide identity, data privacy, and anonymity for nodes in opportunistic mobile social networks. They have tried to preserve privacy for a group of nodes by sharing data in the form of a packet via Bluetooth-enabled smartphones. Their proposed algorithm has some hardware and software limitations.

Authors in [33] proposed using Blockchain-based routing algorithms to provide privacy and security for Opportunistic networks. In the proposed algorithm, every node stores the set of messages in a block and forwards them as the blockchain. Nodes merge their blocks to carry them.

Table 1. Compression of different privacy-preserving algorithms in OppNet. No comparison in the table indicates that the performance of the references has not been compared with the other routing algorithms. No malicious nodes indicates that the effect of malicious nodes has not been considered in the network.

Reference	Approach	Privacy Aspect	Advantage	Disadvantage
[21]	Bloom Filter to obscure the friends' list	Identity privacy	Identity authentication based on zero knowledge	No comparison No malicious nodes
Privo [22]	Paillier Homomorphic Encryption, Binary anonymization, and neighborhood randomization	Identity and attribute privacy	The cryptography costs are blowing 1%.	No comparison No malicious nodes
[23]	Optimized version of Millionaire's Problem	Attribute privacy	A good coverage level, high receivers' accuracy	No comparison No malicious nodes
4PR [24]	Community based routing to conceal nodes' mobility	Location privacy	Predicting routing path, preserving privacy	No malicious nodes.
PIDGIN [25]	Policy tree that represents access structure	Attribute privacy	It does not leakage information to untrusted nodes. It is implemented on a smartphone.	No comparison No malicious nodes
PPUR [26]	Bilinear mapping technology, Hashing, Symmetric encryption	Information privacy	It provides messages confidentiality and integrity	No malicious nodes

Table 1. *Cont.*

Reference	Approach	Privacy Aspect	Advantage	Disadvantage
[27]	Multiparty computation, homomorphic encryption	Location privacy	By authenticated encryption it provides mutual authentication, non-repudiation, and conditional privacy preserving.	No comparison No malicious nodes
PEON [28]	Layered cryptography	Information privacy	Anonymous communication and rerouting messages via peer nodes.	No comparison No malicious nodes
PPHB [29]	Nodes produce a polynomial and hide their identity in the polynomial.	Identity and location privacy	Zero knowledge	Complex calculation
ePRIVO [30]	A time-varying neighboring graph was used	Information privacy	Nodes can make routing decisions without knowledge of private information.	No malicious nodes
PPERP [31]	Bilinear mapping technology	Information privacy	Providing confidentiality, integrity, and nonrepudiation of encounter records	Focused on privacy preserving for only nondelivery nodes. No malicious nodes
[32]	Lightweight cryptographic encryption	Identity, data privacy, and anonymity for nodes	High reliability of packet notification forwarding, Validating the algorithm implementing	No comparison, No malicious nodes, hardware and software limitations
[33]	Blockchain	Information privacy	Do not need trust management. Protecting data from manipulation, eavesdropping, masquerading, and other passive attacks.	Block size increases quickly and there are buffer limitations.

3. Blossom Structure

For the routing algorithm, we used clustering features to decrease the number of messages forwarded in the network. We clustered the aligned nodes in a similar direction to shape a cluster; consequently, the homogeneous clusters merged in the network. Therefore, the messages are sent to the clusters rather than individual nodes. We described the procedure of nodes clustering and merging them in detail in this section.

3.1. Calculating Directions

Clustering the nodes per direction requires first identifying each node's geographical direction. The node's direction are represented as a vector $\vec{d}_N(x_1, x_2)$, where x_1 and x_2 are the movements in the XY plane for node N. The GPS coordinates defined with longitude and latitude can meet the requirement. Each node's direction in an instant by Equation (1):

$$\sphericalangle(\vec{d}(x_1, x_2)) = \begin{cases} \cos^{-1} \frac{x_1}{|\vec{d}(x_1,x_2)|} & \text{for } x_2 \geq 0 \text{ and } x_1 \vee x_2 \neq 0 \\ 0 & \text{for } x_1, x_2 = 0 \\ 360 - \cos^{-1} \frac{x_1}{|\vec{d}(x_1,x_2)|} & \text{for } x_2 < 0 \end{cases} \tag{1}$$

To overcome the challenge of frequently changing the directions of nodes, we considered a path history for nodes, which is represented as follows:

$$\sphericalangle(\vec{d}) = (1-\lambda)\cdot\sphericalangle(\vec{d}_{old}) + \lambda\cdot\sphericalangle(\vec{d}_{new}) \quad \text{with } 0.5 < \lambda \leq 1 \tag{2}$$

In addition to the nodes' directions, the distance between nodes should be calculated to cluster nodes. The distance between two nodes i and j is calculated from Equation (3) [34]. The minimum of the counter calculation considers both directions in a circle. The distance is computed and mapped between 0 and 1 because the maximum inner circle distance is 180°. Distance or dissimilarities in Equation (3) is 1 for the absolute dissimilarity and 0 for the total similarity [35]:

$$Dist_{ij} = \frac{min(|\sphericalangle(\vec{d})_i - \sphericalangle(\vec{d})_j|, (360 - |\sphericalangle(\vec{d})_i - \sphericalangle(\vec{d})_j|))}{180} \tag{3}$$

3.2. Clustering Analysis and Homogeneous Group of Clusters Merging

Setting up the distance matrix between all reachable nodes is followed by finding the smallest dissimilarity between two clusters. The clusters with the smallest dissimilarity are merged as a homogeneous group of clusters. Accordingly, the matrix is updated and holds the dissimilarity between the new cluster and all other clusters. To calculate the required distances between the originated merged cluster and the other clusters, the average group method of Sokal and Michener [36] is applicable [35]. Such an unsupervised learning technique contributes to constructing a homogeneous group of clusters to tackle the structuring problem in an unknown area [37,38]:

$$Dist(R,Q) = \frac{1}{|R|\cdot|Q|}\sum_{i\in Q, j\in R} d(i,j) \tag{4}$$

In Equation (4), R and Q represent the merged clusters, while $|R|$ and $|Q|$ are the number of nodes inside each cluster [35].

3.3. Clusters

Each node performs a cluster analysis independently to create a directional group of clusters. The maximum number of nodes should be located in a cluster. However, according to the analysis result, each node may meet the requirements to belong to more than one cluster. Therefore, the node must decide which cluster to join; for this purpose, we considered stop rules. While nodes in a cluster are moving, a maximum height of dissimilarity between the node and cluster is used to determine the most appropriate cluster in the stopping rule. In addition, Blossom uses agglomerative hierarchical clustering [35] to merge the most homogeneous clusters in each iteration. The clusters with the smallest dissimilarity become merged as homogeneous clusters.

The already allocated node to a cluster might conflict with the cluster analysis result due to the mobility of nodes over time.

We designed a mechanism to maintain sustainability by keeping the clusters alive and increasing the cluster members. Figure 1 illustrates creating clusters in Blossom. There are two methods to assign each node to a cluster as follows:

- If a node does not belong to any cluster, a new cluster analysis performs at the beginning of the simulation. The most appropriate cluster for node A is the cluster with a significant number of members. Under this premise, A is assignable to B's cluster when A's direction is not greater than the average direction of the cluster that the stopping rules allow. A new cluster is created if no suitable cluster is found or does not belong to a cluster yet. At the end of this assignment process, all new clusters get once-in-a-lifetime initialized with their current average directions.

- If the first method is invalid, the second method is performed. It attempts to assign a single node to an existing cluster with a significant number of members and fit with the node's direction. Therefore, if the node does not find a cluster, it will not create a new one but will look for an appropriate one while moving along. Hence, this method is only valid for reducing the injecting message into the network by increasing the size of existing clusters and preserving the node's privacy.

Blossom is a dynamic approach in which nodes frequently check the available suitable cluster and update their clusters. We have defined the stopping rule to allocate the nodes to the most appropriate clusters. Thus, when the distance between a node's direction and a cluster's direction is greater than the stopping rule, the node leaves the cluster and attempts to find another appropriate cluster based on the two methods described.

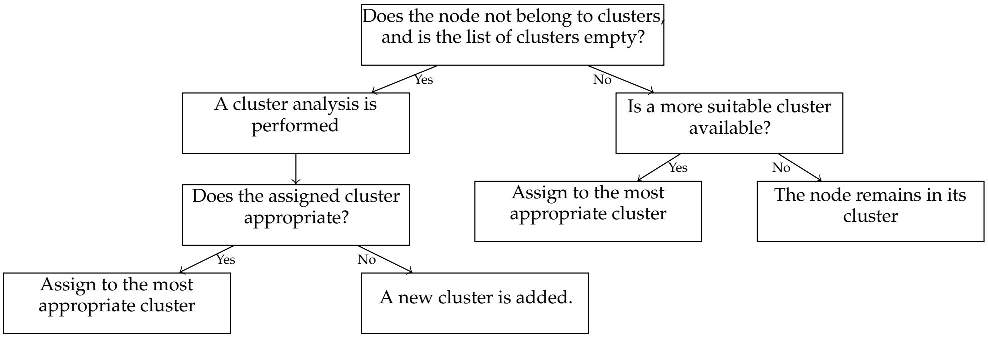

Does the node not belong to clusters, and is the list of clusters empty?
Yes
No
A cluster analysis is performed
Is a more suitable cluster available?
Yes
No
Does the assigned cluster appropriate?
Assign to the most appropriate cluster
The node remains in its cluster
Yes
No
Assign to the most appropriate cluster
A new cluster is added.

Figure 1. Conditions to be met for cluster assignments.

Figure 2 depicts the clustering in Blossom. In the left figure, nodes do not belong to a cluster yet, while in the right figure, nodes with similar directions shape different clusters (Each color represents a cluster).

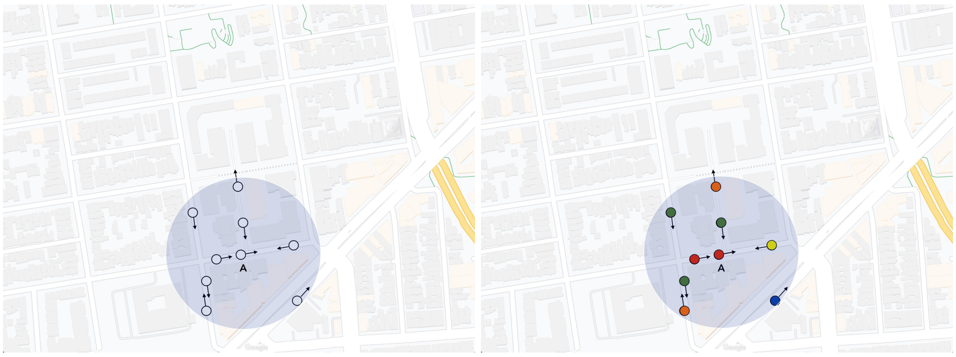


Figure 2. An example of clustering in Blossom.

3.4. Message Routing

Nodes do not generate any copy inside a cluster, reducing network overhead. Instead, the message is forwarded to all intermediate nodes in the cluster. If the receiving node is the destination, it holds the message and stops forwarding it. Suppose the message is sent to all intermediate nodes and does not reach its destination. In that case, the last traversed node initiates another cluster analysis to spread the message over other reachable nodes. We performed cluster analysis and grouped all connected neighbors of a node in a cluster. Each node transmits one replica to each cluster. Firstly, it serves the clusters with the average most extensive distance direction as they might be reachable for a shorter period. Secondly, the router goes to stand-by mode for a specified time. Nevertheless, if a node meets a message's destination, the node transmits the message to the neighbor despite stand-by.

3.5. Security Layer

In OppNets, the nodes' identity and location privacy should be protected to preserve privacy. The precise location, IDs, type of nodes (pedestrian, car, tram), and other information remain unknown. However, we designed a layer called Blossom Security (BlosSec) to preserve nodes' privacy. BlosSec eliminates nodes' identifiers and only considers the average node direction. Observing the direction of a node does not reveal its actual location. Concerning k-anonymity, the location remains secure in a significant context due to numerous ways of holding a specific direction. However, according to l-diversity [39], k-anonymity can be broken with certain background information. In OppNets, that information can be unique reachable directions. Blossom receives the directions with an accurate position through the small observed environment. Therefore, the security layer of Blossom ensures a node's privacy. BlosSec uses cloaking to provide privacy for users. BlosSec will eliminate all identifiers. In addition, if nodes are assigned to a cluster, they will hold the average direction of their dedicated clusters. Accordingly, the node remains anonymous via its cluster, which provides obfuscation using cloaking. By cloaking, the actual cluster size is hidden from Selfish nodes. Consequently, extracting the actual direction from the average direction is not feasible, providing k-anonymity with a context-dependent on k. The cluster creation time remains unknown, so the available information of the average direction does not lead to (or obtain) the current surrounding of a node. The creation time can be set in the past. Therefore, it is assumed that BlosSec provides l-diversity as all directions being well represented over time, adding a t-closeness on a larger scale.

A node attempts to deliver the message directly when a destination is near a node. To enable this procedure without revealing identifiers, BlosSec will use Secure Hash Algorithms (SHA-256) [40]. The transmitting node sends the hashed message receiver to the cloaked receiver node in advance. When the receiver is the message's destination, its hashed name conforms with the received hashed value. Finally, it will send the plain name back to the sender for validation, approving the destination. A hash function is used to hold a decent security level in our algorithm.

4. Evaluation

In this paper, we utilized Opportunistic Network Environment (ONE) [41] for simulation. The obtained results from ONE were transferred to Matlab to generate the graphs and charts.

The routing decisions are made based on the node's directions in Blossom. Therefore, the performance can be affected due to the topology of the given directions. To evaluate the effect of the topology on the algorithm's performance, we have selected two different environments because of their different plurality of directions.

The first topology is based on the map of San Francisco with a grid-like street layout (see Figure 3a), and the second one is Venice, with random routes growing in a wide variety of directions (see Figure 3b).

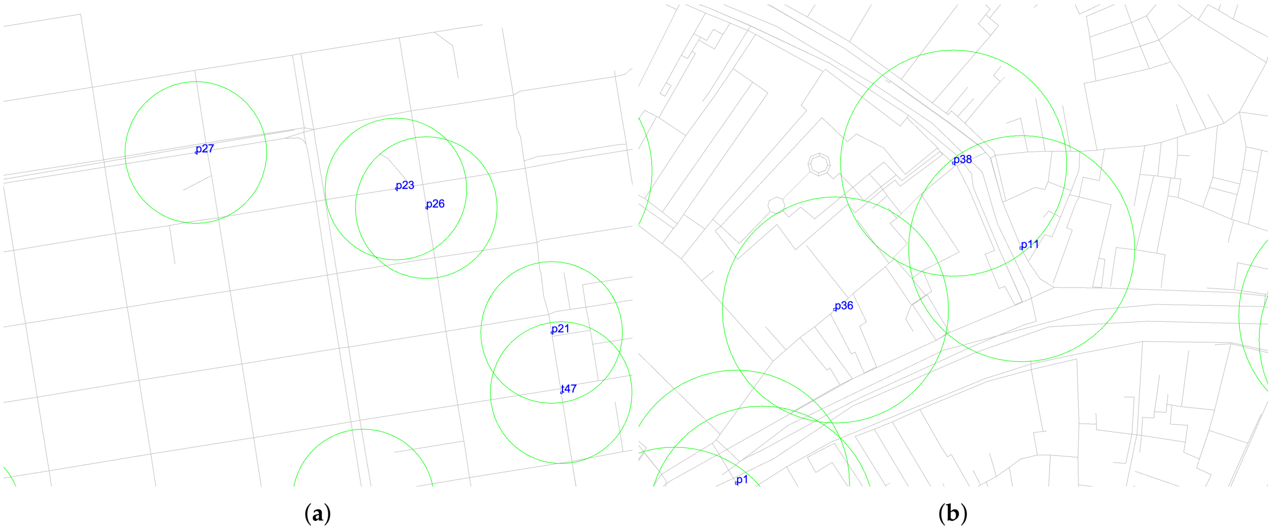


Figure 3. The city structures of San Francisco and Venice. (**a**) San Francisco; (**b**) Venice.

The same configurations are used for both environments. OppNets can have different applications depending on the area of usage. It can be constructed in a rural area (with a limited number of nodes) like a smart city. Thus, we adjusted the number of nodes and monitored the effect on the performance. The configuration starts from a low density of ten nodes to a density of 100 nodes with the step of 10. The movement model is the Shortest Path Map-Based Movement for each group in both environments to mimic a real-world application scenario during the simulation. In this movement model, each node follows the shortest path to a randomly set point of interest. In addition, messages are created every 25 to 35 s with a randomized size between 500 KB and 1 MB. The complete configuration is shown in Table 2. The simulation uses three different groups: Pedestrians, Cars, and Trams. The latter group is only added from a density greater than 50 nodes. Additionally, this group utilizes a higher transmitting speed and range. We assumed that transmission devices are embedded in cars and trams.

Table 2. Group configurations for running simulations for both cities.

Group	**Pedestrian**	**Car**	**Tram**
Number of Hosts	about 90%	about 10%	3 (if *host* $\geq$ 50)
Buffer Size	5 MB	5 MB	50 MB
Movement speed in m/s	0.5–1.5	2.7–13.9	7–10
Movement speed in km/h	1.8–5.5	10–50	26–37
Movement Waiting Time	0–120 s	0–120 s	10–30 s
Transmit Speed	250 KB/s	250 KB/s	10 MB/s
Transmit Range	100 m	100 m	1000 m

Bluetooth Low Energy (BLE) 4.2 is used during the evaluation to cluster as many directions as possible. Furthermore, it provides better privacy protection through an extensive range as nodes hide in larger clusters. We use an indoor-based Bluetooth study range (shorter) to ensure communication reliability, excluding the outdoor Bluetooth range. The outdoor Bluetooth range is typically more extensive than the indoor, unless under particular conditions. An example is BLE 5.0, which has a range of at least 200 m outdoors and low power consumption [42]. Several current high-end mobile phones are already equipped with that technology.

To examine the parameters, we have simulated and visualized all twelve combinations of the settings. The best combination settings of the delivery probability of nodes compared against the First Contact, Epidemic, and PRoPHET routing algorithms. If the delivery probability is the same for two or more settings, the network overhead is considered as the measure for the optimal setting.

4.1. Settings Investigation

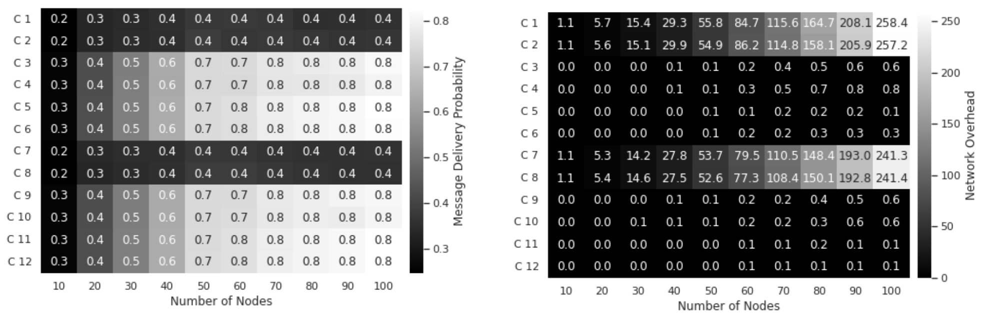
Blossom has three adjustable parameters: the maximum height of a cluster, the inclusion of historical directions, and sleep time (stand-by). We measured the different combinations of these parameters to make the best settings for the Blossom algorithm. The possible settings are demonstrated in Table 3. The values were chosen due to test simulations during the implementation. Those chosen settings might not provide the best possible performance for Blossom in each context. However, it was assumed to be sufficient to evaluate the parameters and the use of directions.

The Maximum Cluster Height (H) is considered in two different settings (see Table 3). For the cluster analysis, differences between directions are used. The values in Table 3 are percentages of the maximum distance of 180° that is possible within a circle. Thus, at 0.1, we have a cluster range of 18°. The Sleep Time (S) is in seconds. Adjustment of the Direction (A) is the third parameter in percentage. At the value of 1, it does not include the historical direction. With the value 0.8, the current direction is considered 80%, while the historical direction is 20%. Including the historical direction causes a delay in the current nodes' directions, which is considered to influence the cluster sizes.

Table 3. Investigated settings for the parameters of Blossom.

Max Cluster's Height (H)	Sleep Time (S)	Adjustment of the Direction (A)
0.1	0	1
0.2	10,000	0.8
-	20,000	-

The results were calculated and evaluated for both cities' message delivery probability and network overhead according to Table 3. There exist no significant differences between the settings of parameters for both cities. Therefore, only the results of Venice were presented in Figure 4 for message delivery probability and network overhead. C 1 to C 12 in these figures are the combination of setting parameters, and are described in Table 4.



(**a**) (**b**)

Figure 4. Comparison of Blossom's settings for the message delivery probability and the network overhead ratio on the map based on Venice. (**a**) message delivery probability; (**b**) network overhead.

Table 4. Different possible combinations of H, S, and A.

C 1	0.1 H	0.0 S	0.8 A
C 2	0.1 H	0.0 S	1 A
C 3	0.1 H	10,000.0 S	0.8 A
C 4	0.1 H	10,000.0 S	1 A
C 5	0.1 H	20,000.0 S	0.8 A
C 6	0.1 H	20,000.0 S	1 A
C 7	0.2 H	0.0 S	0.8 A
C 8	0.2 H	0.0 S	1 A
C 9	0.2 H	10,000.0 S	0.8 A
C 10	0.2 H	10,000.0 S	1 A
C 11	0.2 H	20,000.0 S	0.8 A
C 12	0.2 H	20,000.0 S	1 A

We calculated the arithmetic mean of parameters for Venice, shown in Figure 4, and also for San Francisco, and they are presented in Table 5. According to Table 5, the cluster′s height and direction adjustment did not influence the delivery probability of the nodes in the cities.

Table 5. Measures for messages' delivery probability and network overhead ratio for the parameters of Blossom in San Francisco and Venice.

Parameter	Message Delivery Probability		Network Overhead	
	San Francisco	Venice	San Francisco	Venice
0.1 H	0.46	0.55	19.22	31.26
0.2 H	0.46	0.55	18.53	29.19
0 S	0.3	0.34	56.06	90.34
10,000 S	0.53	0.65	0.47	0.25
20,000 S	0.55	0.66	0.1	0.09
0.8 A	0.46	0.55	18.90	30.33
1 A	0.46	0.55	18.84	30.13

In contrast, there is a considerable difference in the Sleep Time values in both cities. In particular, the Sleep Time 0 S has half of the average delivery probability of 10,000 S and 20,000 S. Therefore, during the sleep time 10,000 S and 20,000 S, there is a slight difference in both cities. The results indicate the convergence parameter to a specific value.

According to Figure 4, increasing the Maximum Cluster Height causes a slight reduction in network overhead. According to Table 5, by increasing the Maximum Cluster Height from 0.1 to 0.2, the network overhead for San Francisco decreases by nearly 0.7 and for Venice by more than 2.

With the increase in the Sleep Time of the router in Figure 4, the network overhead is dramatically reduced. This leads to deleting fewer messages and improving message delivery. In addition, this also shows that most of the forwarded messages have been delivered.

The results show that the direction adjustment almost does not influence the delivery probability and the network overhead. For San Francisco and Venice, it reflects a deterioration of the arithmetic mean.

4.2. Comparison against Other Routing Algorithms

This section compared Blossom with First Contact, Epidemic, and PRoPHET algorithms regarding Message Delivery Probability, Dropped Message Probability, and Network Overhead. Table 6 listed the used settings of Blossom for the simulations. Parameters for each node density were set based on the previous output.

Table 6. The used settings of H, S, and A in Blossom for the simulations.

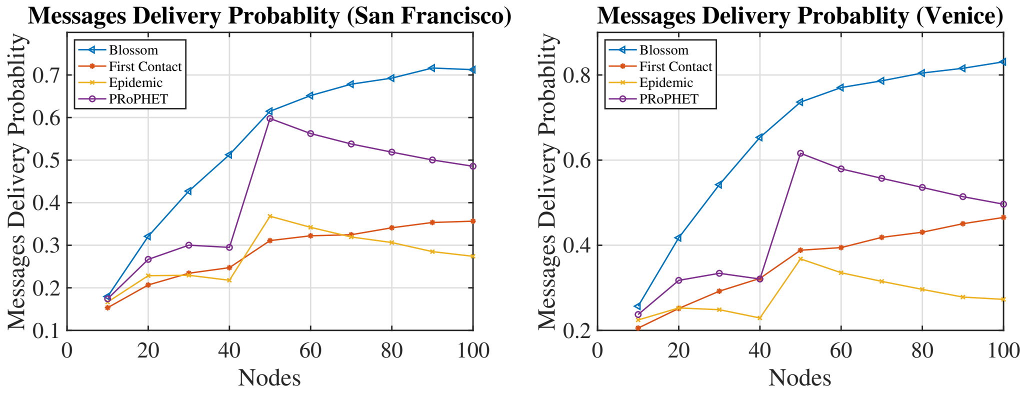
	San Francisco			Venice		
Nr. of Nodes	**Max Cluster's Height (H)**	**Sleep Time (S)**	**Adjustment of the Direction (A)**	**Max Cluster's Cluster's Height (H)**	**Sleep Time (S)**	**Adjustment of the Direction (A)**
10	0.1	10,000	0.8	0.1	10,000	0.8
20	0.2	20,000	0.8	0.1	20,000	0.8
30	0.1	20,000	0.8	0.2	20,000	0.8
40	0.1	20,000	0.8	0.1	20,000	0.8
50	0.2	20,000	0.8	0.2	20,000	1.0
60	0.1	20,000	0.8	0.2	20,000	0.8
70	0.1	20,000	0.8	0.2	20,000	1.0
80	0.2	20,000	0.8	0.2	20,000	1.0
90	0.1	20,000	0.8	0.2	20,000	0.8
100	0.1	20,000	0.8	0.2	20,000	0.8

To be precise, we have used a 95% confidence interval for results over five different pseudo-random simulations for each city as follows:

$$x_o = \bar{x} + 1.96\frac{s_x}{\sqrt{n}} \tag{5}$$

where $\bar{x}$ is the arithmetic means, s_x is the standard error, and n is the sample size of $n = 10$.

Figure 5 shows that Blossom's delivery probabilities perform significantly better than PRoPHET, Epidemic, and First Contact algorithms. The kink of all algorithms in Figure 5, at the level of 50 nodes, is due to the introduction of the tram group with a more extensive Bluetooth range and more storage (see Table 2). In all algorithms, the message delivery probability increases with the number of nodes. With 100 nodes in the network of San Francisco, Blossom improves the Message Delivery Probability rate by 46%, 100%, and 160% compared to PRoPHET, First Contact, and Epidemic, respectively. In addition, with the same amount of nodes in Venice, Blossom improves the Message Delivery rate by 67%, 78%, and 204% compared to PRoPHET, First Contact, and Epidemic, respectively.

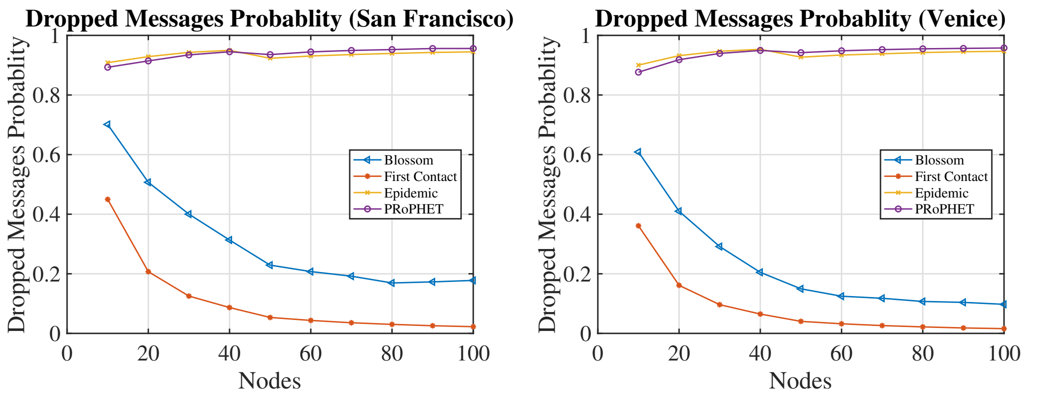


(**a**) (**b**)

Figure 5. Comparison of routing algorithms for messages' delivery probability. (**a**) San Francisco; (**b**) Venice.

The dropped messages Probability in Figure 6 supports the results of the Delivery Probability. While PRoPHET and Epidemic drop messages at 97% (including copies of messages), Blossom and First Contact are cautious about dropping messages. In the First Contact algorithm, the Drop Probability is less due to the non-copying of messages, while Blossom is economical with the number of replicas produced. Blossom decreases Dropped

Messages Probability by 83% compared to PRoPHET and Epidemic in San Francisco having 100 nodes. The same parameter in Venice is decreased by 91% compared to PRoPHET and Epidemic.

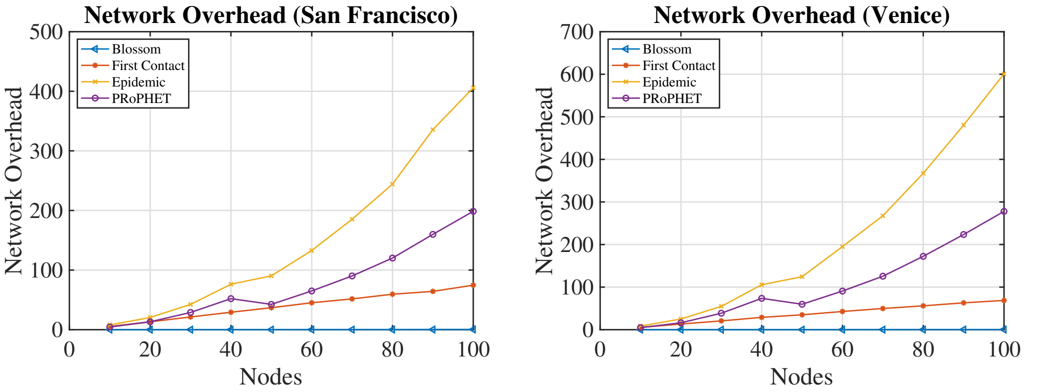


(**a**) (**b**)

Figure 6. Comparison of routing algorithms for dropped messages' probability. (**a**) San Francisco; (**b**) Venice.

The network overhead again shows the strength of Blossom in Figure 7, which is close to zero. As is known, Epidemic has a very high network overhead, while the First Contact algorithm is in the lower midfield. PRoPHET, on the other hand, has a considerable overhead, which also substantially impacts the delivery probability.

Due to a small number of messages in the network, message delivery improves in Blossom.



(**a**) (**b**)

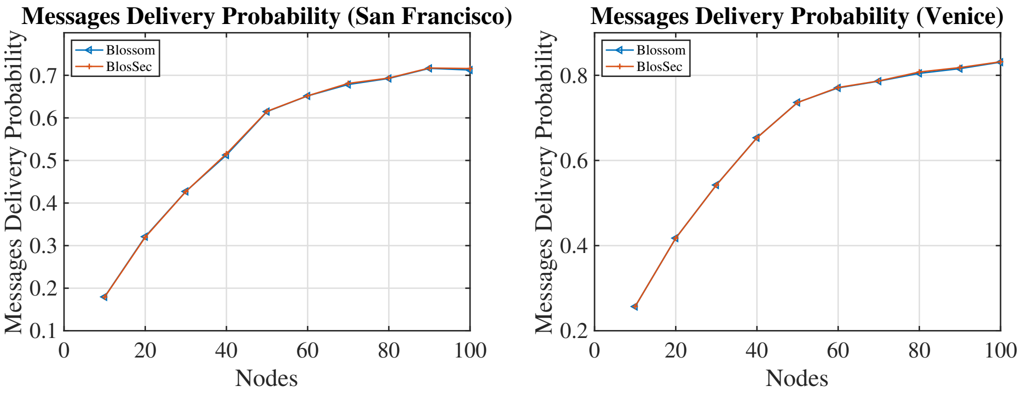
Figure 7. Comparison of routing algorithms for network overhead ratio. (**a**) San Francisco; (**b**) Venice.

The simulation results are in a positive impact of direction on the performance of the algorithm for OppNets. In addition, considering the lowest arithmetic mean, Blossom performs better than the routing algorithms First Contact and Epidemic even without Sleep Time in most cases. These settings only consider cluster-based routing over directions.

4.3. Investigation of the Security Layer's Impact

Applying the concept of BlosSec based on cloaking uses the average cluster direction. The cluster members using the same direction are protected, which results from the average direction during the initialization of a cluster. We intended to examine the impact of this procedure on the Delivery Probability and the average of the cluster's maximum size. The latter phenomena are divided into the maximum average cluster size over the whole simulation and the maximum average cluster size of all clusters that still exist at the end of a simulation. These average cluster sizes should provide information about the average k-anonymity. As the cluster size frequently changes and varies, the maximum cluster size was considered during the evaluation. Thus, the average sums up the maximum size of all clusters divided by the number of clusters.

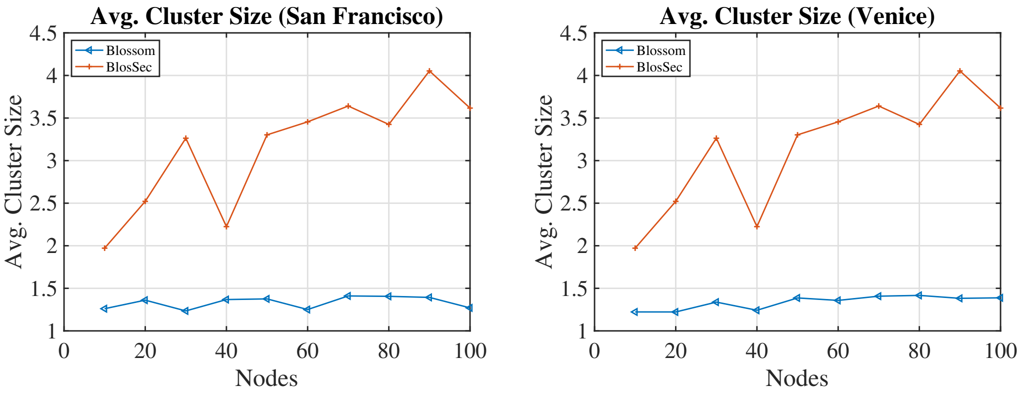
Figure 8 clearly shows that BlosSec has almost no influence on the Delivery Probability in both cities, even though, under the highest node density, a slight improvement of the delivery probability is seen but is negligible. As a result, the security layer does not affect the Delivery Probability inversely. Thus, privacy does not have to be weighed against Delivery Probability in each application domain.



(**a**) (**b**)

Figure 8. Comparison of Blossom against BlosSec in terms of message delivery probability. (**a**) San Francisco; (**b**) Venice.

Figure 9 shows a significant different size of Blossom and BlosSec, in order to get the users through cloaking. Therefore, the number of users in the same direction and assigned to the same cluster increases during the cluster analysis (see Section 3.2). BlosSec observed a minimum average cluster size of two, which grows to a size of four. Thus, BlosSec provides a k-anonymity with k between two and four during simulation.

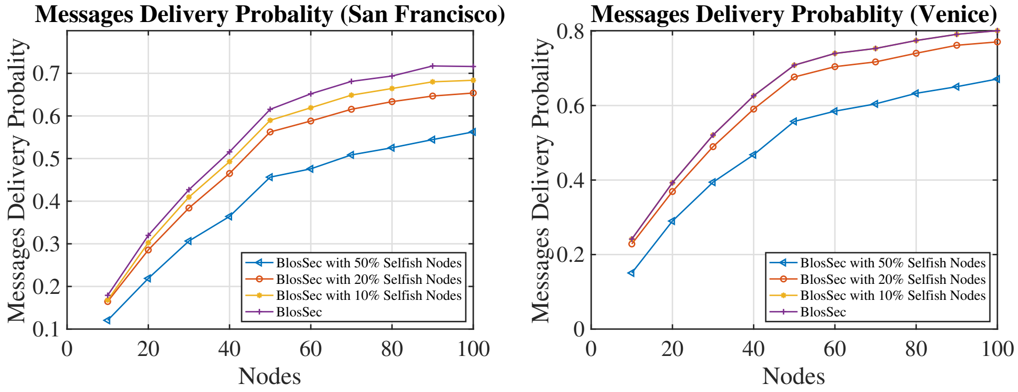


(**a**) (**b**)

Figure 9. Comparison of Blossom against BlosSec on avg. maximum cluster size. (**a**) San Francisco; (**b**) Venice.

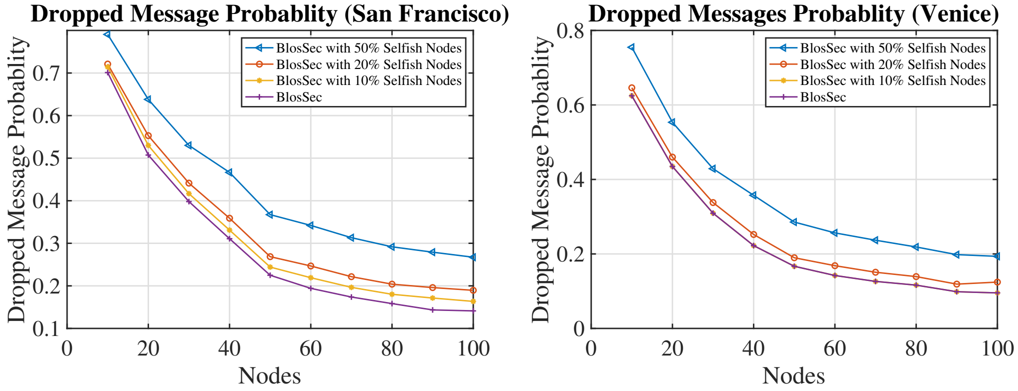
Clusters are created when messages are sent. Therefore, the cluster sizes are relatively small at the beginning of a simulation. Hence, there are relatively more minor clusters in the beginning, and they further expand over time. It is also essential to consider the maximum cluster size of the most recent existing clusters. Figure 9 demonstrates how these maximum sizes of the clusters affect the network. BlosSec has a significantly higher peak than Blossom, ranging from two to nearly ten members for both cities. Thus, prolonging the network runs with BlosSec results in larger clusters and consequently improves the members' privacy. This is because, during the simulations, the existing final clusters grew to a maximum size of up to ten members on average. Therefore, on average, up to ten members are indistinguishable since no identifiers are used, and all cluster members have the same direction. In this respect, identity and location privacy are protected with a k-anonymity with $k = 10$. Due to the many possible directions, and the obfuscation of directions by the clusters, an l-diversity is assumed with $l = k$ as discussed earlier. Thus, the use of directions provides a method to build privacy-protecting clusters.

Finally, we investigated the delivery probability of BlosSec in networks where a certain percentage of selfish nodes exist. The investigated values are 10%, 20%, and 50%, and selfish nodes only drop 20% of messages in their buffer. The results in Figures 10 and 11 show that BlosSec is affected by selfish nodes. As expected, as the number of selfish nodes increases, the message delivery is reduced, and the messages dropping on the network also increases.



(**a**) (**b**)

Figure 10. Comparison of the messages delivery probability of BlosSec in networks with selfish nodes dropping 20% of their messages. (**a**) San Francisco; (**b**) Venice.



(**a**) (**b**)

Figure 11. Comparison of the dropped messages probability of BlosSec in networks with selfish nodes dropping 20% of their messages. (**a**) San Francisco; (**b**) Venice.

4.4. Restrictions of the Study

We proposed a new privacy-preserving routing algorithm for Opportunistic networks. We also estimated the influence of malicious nodes on network performance. Although two different cities with different urban structures have been used for the simulation, our work is limited to the number of simulation scenarios (restricted to two cities), the number of nodes in each city (maximum 100 nodes), and the lack of automatic adjustment of the settings to H, S, and A. In addition, implementing the algorithm on the embedded systems and devices, for example wearable devices, is not within this work's scope but would be significant for validating the algorithm and results under the actual scenario.

5. Conclusions

In this work, we proposed an algorithm called Blossom for clustering nodes in OppNets and forwarding messages to clusters of nodes. This approach reduced the number of messages produced in the network and significantly decreased the network overhead. Furthermore, the proposed algorithm improved the overall message delivery probability and decreased message dropping. Compared to similar algorithms, such as PRoPHET, First Contact, and Epidemic, our approach showed a higher performance and efficiency. Blossom was analyzed in two cities, San Francisco and Venice, in ONE simulator. It improved messages delivery by 46%, 100%, and 160% compared to PRoPHET, First Contact, and Epidemic in San Francisco, and by 67%, 78%, and 204% compared to PRoPHET, First Contact, and Epidemic in Venice, respectively. Blossom could reduce the dropped message by 83% compared to PRoPHET and Epidemic in San Francisco and 91% compared to PRoPHET and Epidemic in Venice. Due to the small number of messages generated, the network overhead in this algorithm is close to zero.

Cloaking and hashing were also used to provide privacy for network participants. The security measures do not negatively affect network performance and provide k-anonymity for network participants. The simulation results prove that the security layer has almost no influence on the network performance. Additionally, an evaluation of network performance was conducted for scenarios where 10%, 20%, and 50% of nodes drop messages frequently. Our approach remains resilient for up to 20% of all participating nodes that are compromised, keeping a steady network performance and delivering messages to their final destinations. The simulation results show how the presence of malicious nodes in the network can affect the network performance.

In future work, we plan to implement Blossom in some wearable devices carried with individuals in order to evaluate the algorithm's efficiency and performance in concrete scenarios. The wearable devices must perform according to the proposed algorithm and will shape clusters and forward messages securely. However, the network performance has to be analyzed and validated in concrete scenarios. Comparing the simulation results with the actual output of the wearable devices should enable us to compare, assess, and improve the algorithm performance and tackle the possible shortcomings. Furthermore, we plan to regulate the automatic adjustment of the settings for H, S, and A either with machine learning or with evaluated regularities.

Author Contributions: Conceptualization, B.K., S.R.; methodology, B.K., S.R. and T.H.; software, B.K.; validation, S.R. and T.H.; formal analysis, B.K., S.R. and T.H.; investigation, B.K. and S.R.; resources, B.K., S.R.; data curation, B.K., S.R.; writing—original draft preparation, B.K., S.R. and T.H.; writing—review and editing, B.K., S.R. and T.H.; visualization, B.K., S.R.; supervision, T.H.; project administration, T.H.; funding acquisition, T.H. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge support from the Open Access Publishing Fund of the University of Muenster.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OppNet	Opportunistic Network
H	Max cluster's height
S	Sleep time
A	Adjustment of the direction
BlosSec	Blossom Security

References

1. Weber, R.H.; Weber, R. *Internet of Things*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 12.
2. Cha, H.; Lee, W.; Jeon, J. Standardization strategy for the Internet of wearable things. In Proceedings of the 2015 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 28–30 October 2015; pp. 1138–1142.
3. Pozza, R.; Nati, M.; Georgoulas, S.; Moessner, K.; Gluhak, A. Neighbor discovery for opportunistic networking in internet of things scenarios: A survey. *IEEE Access* **2015**, *3*, 1101–1131. [CrossRef]
4. Chourabi, H.; Nam, T.; Walker, S.; Gil-Garcia, J.R.; Mellouli, S.; Nahon, K.; Pardo, T.A.; Scholl, H.J. Understanding smart cities: An integrative framework. In Proceedings of the 2012 45th Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2012; pp. 2289–2297.
5. Woungang, I.; Dhurandher, S.K.; Anpalagan, A.; Vasilakos, A.V. *Routing in Opportunistic Networks*; Springer: New York, NY, USA, 2013.
6. Denko, M.K. *Mobile Opportunistic Networks: Architectures, Protocols and Applications*; CRC Press: Boca Raton, FL, USA, 2016.
7. Guo, B.; Yu, Z.; Zhou, X.; Zhang, D. Opportunistic IoT: Exploring the social side of the internet of things. In Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Wuhan, China, 23–25 May 2012; pp. 925–929.
8. Guo, B.; Zhang, D.; Wang, Z.; Yu, Z.; Zhou, X. Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things. *J. Netw. Comput. Appl.* **2013**, *36*, 1531–1539. [CrossRef]
9. Chilipirea, C.; Petre, A.C.; Dobre, C. Energy-aware social-based routing in opportunistic networks. In Proceedings of the 2013 27th International Conference on Advanced Information Networking and Applications Workshops, Barcelona, Spain, 25–28 March 2013; pp. 791–796.
10. Huang, C.M.; Lan, K.c.; Tsai, C.Z. A survey of opportunistic networks. In Proceedings of the 22nd International Conference on Advanced Information Networking and Applications-Workshops (Aina Workshops 2008), Washington, DC, USA, 25–28 March 2008; pp. 1672–1677.
11. Pan, D.; Zhang, H.; Chen, W.; Lu, K. Transmission of multimedia contents in opportunistic networks with social selfish nodes. *Multimed. Syst.* **2015**, *21*, 277–288. [CrossRef]
12. Kumar, P.; Chauhan, N.; Chand, N.; Awasthi, L.K. SF-APP: A secure framework for authentication and privacy preservation in opportunistic networks. *Int. J. Web Serv. Res.* **2018**, *15*, 47–66. [CrossRef]
13. Amah, T.E.; Kamat, M.; Bakar, K.; Moreira, W.; Oliveira, A., Jr.; Batista, M.A. Preparing opportunistic networks for smart cities: Collecting sensed data with minimal knowledge. *J. Parallel Distrib. Comput.* **2020**, *135*, 21–55. [CrossRef]
14. Zakhary, S.; Benslimane, A. On location-privacy in opportunistic mobile networks, a survey. *J. Netw. Comput. Appl.* **2018**, *103*, 157–170. [CrossRef]
15. Chen, D.; Borrego, C.; Navarro-Arribas, G. A Privacy-Preserving Routing Protocol Using Mix Networks in Opportunistic Networks. *Electronics* **2020**, *9*, 1754. [CrossRef]
16. Gruteser, M.; Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, San Francisco, CA, USA, 5–8 May 2003; ACM: New York, NY, USA, 2003; pp. 31–42.
17. Jain, S.; Fall, K.; Patra, R. Routing in a delay tolerant network. In Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Portland, OR, USA, 30 August–3 September 2004; ACM: New York, NY, USA, 2004; pp. 145–158.
18. Vahdat, A.; Becker, D. *Epidemic Routing for Partially Connected Ad Hoc Networks*; Duke University: Durham, NC, USA, 2000.
19. Lindgren, A.; Doria, A.; Schelén, O. Probabilistic Routing in Intermittently Connected Networks. In *Service Assurance with Partial and Intermittent Resources*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3126, pp. 239–254.
20. Rashidibajgan, S.; Hupperich, T.; Doss, R.; Förster, A. NSecure and privacy-preserving structure in opportunistic networks. *Comput. Secur.* **2021**, *104*, 102208. [CrossRef]
21. Qin, Y.; Zhang, T.; Li, M. Privacy Protection Routing and a Self-organized Key Management Scheme in Opportunistic Networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*; Springer: Cham, Switzerland, 2019; Volume 300, pp. 252–268.

22. Magaia, N.; Borrego, C.; Pereira, P.; Correia, M. PRIVO: A privacy-preserving opportunistic routing protocol for delay tolerant networks. In Proceedings of the 2017 IFIP Networking Conference (IFIP Networking) and Workshops, Stockholm, Sweden, 12–15 June 2017; pp. 1–9.
23. Costantino, G.; Martinelli, F.; Santi, P. Privacy-preserving interest-casting in opportunistic networks. In Proceedings of the 2017 IFIP Networking Conference (IFIP Networking) and Workshops, Paris, France, 1–4 April 2012; pp. 2829–2834.
24. Miao, J.; Hasan, O.; Mokhtar, S.B.; Brunie, L.; Hasan, A. 4PR: Privacy preserving routing in mobile delay tolerant networks. *Comput. Netw.* **2016**, *111*, 17–28. [CrossRef]
25. Asghar, M.R.; Gehani, A.; Crispo, B.; Russello, G. PIDGIN: Privacy-preserving interest and content sharing in opportunistic networks. In Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, Kyoto, Japan, 4–6 June 2014; ACM: New York, NY, USA, 2014; pp. 135–146.
26. Jiang, Q.; Deng, K.; Zhang, L.; Liu, C. A privacy-preserving protocol for utility-based routing in DTNs. *Information* **2019**, *10*, 128. [CrossRef]
27. Song, J.; He, C.; Yang, F.; Zhang, H. A privacy-preserving distance-based incentive scheme in opportunistic VANETs. *Secur. Commun. Netw.* **2016**, *9*, 2789–2801. [CrossRef]
28. Song, J.; He, C.; Yang, F.; Zhang, H. PEON: Privacy-enhanced opportunistic networks with applications in assistive environments. In Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments, Corfu, Greece, 9–13 June 2009; ACM: New York, NY, USA, 2009; pp. 1–8.
29. Rashidibajgan, S.; Doss, R. Privacy-preserving history-based routing in Opportunistic Networks. *Comput. Secur.* **2019**, *84*, 244–255. [CrossRef]
30. Magaia, N.; Borrego, C.; Pereira, P.R.; Correia, M. ePRIVO: An enhanced privacy-preserving opportunistic routing protocol for vehicular delay-tolerant networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 11154–11168. [CrossRef]
31. Jiang, Q. A Privacy-Preserving Exchange-Based Routing Protocol for Opportunistic Networks. *Secur. Commun. Netw.* **2022**, *2022*, 6815911. [CrossRef]
32. Adu-Gyamfi, D.; Zhang, F.; Takyi, A. Anonymising group data sharing in opportunistic mobile social networks. *Wirel. Netw.* **2021**, *27*, 1477–1490. [CrossRef]
33. Dhurandher, S.K.; Singh, J.; Nicopolitidis, P.; Kumar, R.; Gupta, G. A blockchain-based secure routing protocol for opportunistic networks. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 2191–2203. [CrossRef]
34. Will, G. *Visualizing and Clustering Data that Includes Circular Variables*; Writing Project; Montana State University: Bozeman, MT, USA, 2016.
35. Kaufman, L.; Rousseeuw, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 344.
36. Sokal, R.R.; Michener, C.D. A Statistical Method for Evaluating Systematic Relationships. *Univ. Kansas Sci. Bull.* **1958**, *38*, 1409–1438.
37. Xu, D.; Tian, Y. A comprehensive survey of clustering algorithms. *Ann. Data Sci.* **2015**, *2*, 165–193. [CrossRef]
38. Ghavami, P. *Big Data Analytics Methods*; De Gruyte: Boston, MA, USA; Berlin, Germany, 2019.
39. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. *L-Diversity: Privacy beyond k-Anonymity*; ACM: New York, NY, USA, 2007; Volume 1, pp. 1–52.
40. Dangs, Q. *Secure Hash Standard*; NIST FIPS: Gaithersburg, MD, USA, 2015.
41. Keränen, A.; Ott, J.; Kärkkäinen, T. The ONE simulator for DTN protocol evaluation. In Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Rome, Italy, 2–6 March 2009; ACM: New York, NY, USA, 2009; pp. 1–10.
42. Collotta, M.; Pau, G.; Talty, T.; Tongu O.K. Bluetooth 5: A Concrete Step Forward toward the IoT. *IEEE Commun. Mag.* **2018**, *56*, 125–131. [CrossRef]