applied sciences

MDPI

Article

Collaborative Caching for Implementing a Location-Privacy Aware LBS on a MANET

Rudyard Fuster [1], Patricio Galdames [2,*] and Claudio Gutierréz-Soto [1]

[1] Departamento de Sistemas de Infomación, Universidad del Bío-Bío, Concepción 4051381, Chile; rudyard.fuster1401@egresados.ubiobio.cl (R.F.); cogutier@ubiobio.cl (C.G.-S.)
[2] Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastián, Concepción 8340369, Chile
* Correspondence: patricio.galdames@uss.cl

Abstract: This paper addresses the challenge of preserving user privacy in location-based services (LBSs) by proposing a novel, complementary approach to existing privacy-preserving techniques such as *k-anonymity* and *l-diversity*. Our approach implements collaborative caching strategies within a mobile ad hoc network (MANET), exploiting the geographic of location-based queries (LBQs) to reduce data exposure to untrusted LBS servers. Unlike existing approaches that rely on centralized servers or stationary infrastructure, our solution facilitates direct data exchange between users' devices, providing an additional layer of privacy protection. We introduce a new privacy entropy-based metric called accumulated privacy loss (APL) to quantify the privacy loss incurred when accessing either the LBS or our proposed system. Our approach implements a two-tier caching strategy: local caching maintained by each user and neighbor caching based on proximity. This strategy not only reduces the number of queries to the LBS server but also significantly enhances user privacy by minimizing the exposure of location data to centralized entities. Empirical results demonstrate that while our collaborative caching system incurs some communication costs, it significantly mitigates redundant data among user caches and reduces the need to access potentially privacy-compromising LBS servers. Our findings show a 40% reduction in LBS queries, a 64% decrease in data redundancy within cells, and a 31% reduction in accumulated privacy loss compared to baseline methods. In addition, we analyze the impact of data obsolescence on cache performance and privacy loss, proposing mechanisms for maintaining the relevance and accuracy of cached data. This work contributes to the field of privacy-preserving LBSs by providing a decentralized, user-centric approach that improves both cache redundancy and privacy protection, particularly in scenarios where central infrastructure is unreachable or untrusted.

Keywords: location-privacy-aware LBS; privacy loss; collaborative caching; MANET

check for updates

Citation: Fuster, R.; Galdames, P.; Gutierréz-Soto, C. Collaborative Caching for Implementing a Location-Privacy Aware LBS on a MANET. *Appl. Sci.* **2024**, *14*, 10480. https://doi.org/10.3390/app142210480

Academic Editor: Reiner Creutzburg

Received: 28 August 2024
Revised: 3 November 2024
Accepted: 9 November 2024
Published: 14 November 2024

1. Introduction

Location-based services (LBSs) have become integral to modern mobile computing thanks to the ubiquity of smartphones and social networks. These services provide essential functionality from navigation assistance to location-aware recommendations. However, increasing reliance on LBSs raises significant privacy concerns, as each query potentially exposes sensitive user information. Recent studies have shown that even seemingly innocent requests for public data can lead to privacy breaches when aggregated over time [1,2]. For example, analyzing patterns of LBS queries can reveal sensitive details about a user's lifestyle, health conditions, or daily routines, which is information that could be exploited by malicious entities or commercial interests like insurance companies.

The evolution of privacy protection in LBSs has seen several major developments. Early research established fundamental techniques such as *k-anonymity*, which builds cloaking areas to hide the exact location of a user among *k-1* other possible locations [3]. This foundation was extended through *l-diversity* approaches that protect query privacy by generating multiple plausible queries [4,5]. More recent work has explored the combination of these techniques with advanced caching mechanisms [6] and continuous privacy protection schemes [7]. Although these approaches provide crucial privacy protections, they face a fundamental limitation. All are based on centralized LBS systems that accumulate significant amounts of user data over time. Even with privacy protections in place, this centralization enables comprehensive user profiling through long-term data analysis [8].

Researchers have attempted to address this centralization vulnerability through various distributed approaches using mobile ad hoc networks (MANETs). Some solutions leverage MANETs with cellular network infrastructure [9], while others focus on distributed query processing without considering caching efficiency [10]. Recent work has explored privacy protection through distributed storage [2] but still relies on stationary infrastructure in fixed locations. This gap between existing privacy-preserving techniques and the need for truly distributed LBS solutions motivates our work. Previous distributed approaches either depend too heavily on fixed infrastructure or fail to adequately address the unique challenges of maintaining privacy in a fully mobile environment.

To address these limitations, we propose a novel approach that fundamentally reimagines how location-based services can operate in a privacy-preserving manner. Our solution draws inspiration from a time-honored practice: seeking information from nearby individuals who are likely to have relevant local knowledge. Consider how travelers traditionally consulted locals about nearby amenities: Although the local person learns something about the inquirer, their limited capacity and different purpose make it unlikely that they would capture a comprehensive user profile, which requires storing user history over time. This observation guides our development of a distributed LBS system that prioritizes user privacy while maintaining service utility.

We implement this concept through a fully distributed LBS functionality operating directly between users in a MANET, minimizing reliance on centralized and potentially untrusted LBS servers. This approach complements existing privacy-preserving techniques by adding another layer of protection through peer-to-peer data sharing. By distributing queries among MANET peers before resorting to a centralized LBS, we can significantly reduce data exposure while maintaining service quality.

To quantify the privacy benefits of our approach, we have developed a novel mathematical model that measures the loss of location privacy when users resolve queries in the LBS versus the mobile ad hoc network. This *accumulated privacy loss* (APL) metric is, to our knowledge, the first proposed to evaluate privacy loss in the context of a distributed MANET-based LBS system. Our model considers the LBS as the main adversary seeking to compromise location privacy while acknowledging that users in the MANET could also potentially compromise privacy—albeit to a lesser extent due to their limited storage and computing capabilities compared to an LBS server.

Unlike existing privacy metrics that focus solely on k-anonymity levels or query diversity, our APL metric provides a comprehensive measure of privacy loss across both centralized and distributed components of the system. It explicitly accounts for the differential privacy implications of data exposure to limited-capacity MANET peers versus a centralized LBS server with extensive data aggregation capabilities. This allows us to quantitatively assess the privacy benefits of our hybrid approach and guide the development of effective privacy preservation strategies. Our mathematical foundation for APL builds on information theory principles while incorporating practical considerations such as node mobility, cache obsolescence, and query patterns.

Our approach has two main goals: (1) to develop a decentralized peer-to-peer LBS architecture for mobile users that complements traditional LBS servers and (2) to design a storage-efficient collaborative caching scheme to protect user privacy and conserve LBS resources.

To provide a framework for our approach, we consider the following aspects:

- The LBS is the main adversary, as it has all data, which compromises the information of users who access the services.

- The queries submitted by users underlie geographical proximity (i.e., these queries are well-known location-based queries, LBQs), and they are not confidential; queries such as "What is the nearest hotel?" or "What is the nearest major tourist center to my location?" correspond to queries of this type.
- In MANET, users are constantly moving into and out of service ranges. Some users within the service range act as caches, forming a collaborative caching scheme.
- Our model operates at the application layer of the OSI model; details about the operation of the physical layer are not considered in this paper.

Implementing distributed geographic caching and LBS functionality in MANET introduces several challenges.

1. Maintaining location-relevant data near a corresponding geographic area is difficult when mobile users are constantly moving. To address this, we propose a dynamic data exchange strategy that considers both geographic proximity and user mobility patterns.
2. Disconnections in ad hoc networks cause delays; therefore, users must have latency tolerance thresholds aligned with their privacy risk aversion profiles.
3. The efficient use of limited storage across nodes is critical, as excessive redundancy reduces the space for new data, eventually pushing users back to the LBS server when the collective storage capacity is exceeded.
4. More data exchange helps coordination and reduces redundancy, but it consumes energy and bandwidth. The solution must balance response time, energy efficiency, and location privacy.
5. Data obsolescence in caches can lead to undesired access to the LBS and unnecessary storage consumption. We study the effect of data becoming obsolete and propose strategies for the timely removal of outdated information from caches.
6. Balancing privacy protection with system usability requires a careful consideration of when to utilize MANET peers versus accessing the LBS server, particularly when dealing with time-sensitive queries or sparse network conditions.

To our knowledge, no previous work has focused on implementing a privacy-aware location-based service directly in a MANET, particularly considering the challenges of data redundancy, privacy loss, and obsolescence in a distributed caching system. Moreover, the lack of a suitable metric to evaluate privacy loss in this context highlights the novelty of our approach. Although some research has explored MANETs with cellular network infrastructure [1,9] or distributed query processing [10], they have not addressed the critical aspects of cache management and privacy preservation in a fully distributed environment. Although recent studies have investigated distributed approaches to privacy protection [2,6], they continue to rely on stationary infrastructure at fixed locations, limiting their applicability in truly mobile scenarios.

The main contributions of this study are outlined below:

- We present a novel mathematical model and metric (APL) to quantify location privacy loss in both centralized LBSs and our proposed MANET-based system. Our theoretical framework provides a rigorous foundation for evaluating privacy benefits, incorporating factors such as data exposure patterns, node mobility, and cache dynamics.
- We propose a location-aware LBS architecture on MANETs that enables users to resolve spatial queries through peer collaboration before accessing centralized servers. Our experimental evaluation demonstrates significant reductions in LBS queries while maintaining acceptable response times and system overhead.
- We develop spatial query processing strategies based on two-tier caching: local caching maintained by each user and neighbor caching based on proximity. Our comprehensive comparison shows this approach substantially enhances user privacy by minimizing the exposure of location data to centralized entities.
- We introduce two location-based data storage strategies that maximize the proximity of stored data to their corresponding geographic locations. These strategies effectively

balance communication costs against redundancy, ensuring relevant data are cached only in users located within the referenced geographic area.

- We provide the first comprehensive analysis of data obsolescence impact on cache performance and privacy loss in distributed LBS systems, including mechanisms for identifying and removing outdated information to maintain cache relevance and accuracy.

Our approach is particularly valuable in scenarios where central infrastructure is unreachable or untrusted, such as during a disaster response, in crowded public gatherings, or in areas with limited network facilities. By giving users greater control over their information while increasing awareness of privacy risks, our method advances the development of privacy-respecting location-based services that maintain utility without compromising user privacy.

The remainder of this paper is structured as follows. Section 2 presents related work and positions our contributions within the existing literature. Section 3 defines key terms and introduces the system architecture. Section 4 details our implementation approach including data obsolescence handling and privacy loss metrics. Section 6 presents experimental results and performance evaluation, and Section 7 concludes with key findings and future work directions.

2. Related Works

Our work intersects three key research areas: collaborative caching systems, privacy protection mechanisms, and MANET-based solutions. Although these areas have been studied separately, our approach uniquely combines elements from each to create a distributed LBS system that preserves privacy. We build on collaborative caching techniques for efficient data distribution, privacy protection mechanisms for user security, and MANET architectures for infrastructure-independent operation. This section analyzes relevant work in each area and explains how our approach advances beyond their limitations.

2.1. Collaborative Caching in Continuous LBSs

Although our work focuses on sporadic queries, continuous LBSs research has developed several relevant caching concepts that we adapt to our privacy-preserving MANET environment. Jung et al. [11] proposed a P2P-based collaborative cache architecture that significantly reduces performance degradation by filtering candidate responses. We adapt this filtering concept for our MANET environment but distribute the filtering responsibility across mobile nodes rather than relying on fixed peers. Zhang et al. [12] combined Markov models with k-anonymity, using mobility predictions to improve cache efficiency when responses are not found locally. While effective for predicted trajectories, this approach's dependence on movement patterns limits its applicability in our more dynamic MANET context.

Peng et al. [13] advanced the field by developing a system that only requires the prediction of the next location, reducing the complexity of the prediction of the trajectory. However, their approach still depends on predetermined trajectories, unlike our solution, which accommodates arbitrary movements. Zhang et al. [14] introduced a particularly relevant two-level cache system with user caches supplemented by an anonymizer cache. Although this hierarchical approach influenced our design, we eliminated their centralized anonymizer, instead distributing this functionality across MANET nodes to enhance privacy protection and system resilience.

2.2. Privacy Protection in Sporadic LBSs

The privacy challenges of sporadic LBSs directly inform our approach, particularly in balancing privacy protection with system efficiency. The seminal work by Niu et al. [6] established the use of cell access points as data repositories, storing both direct and candidate responses. Although this approach demonstrated the value of distributed storage, its reliance on fixed access points creates potential privacy vulnerabilities that our fully distributed solution addresses. We extend their concept of geographical data distribution while eliminating the dependency on fixed infrastructure.

Recent approaches have explored various privacy protection mechanisms through three main directions: cryptographic techniques, mobile storage solutions, and hardware-based protection. A groundbreaking contribution by Ghinita et al. [15] demonstrated that trusted third-party anonymizers could be eliminated using private information retrieval (PIR) techniques. Their approach achieved strong privacy guarantees and protection against correlation attacks through cryptographic methods although at the cost of increased computational overhead. Although our work takes a different approach to eliminating trusted intermediaries through distribution rather than cryptography; both solutions demonstrate the feasibility of privacy preservation without centralized trusted parties.

On the mobile storage front, Jagarlapudi et al. [16] proposed using drones as external repositories for k-anonymity, demonstrating the potential of mobile caching points. Our work extends this concept by treating all network participants as potential mobile caches, creating a more robust and scalable system. Alsaawy et al. [17] developed a three-tier cache system combining user caches with cell access points, while Alrahhal et al. [18] introduced a leader-based collaborative scheme.

In the hardware security domain, recent systems like SecuDB [19] employ Trusted Execution Environments (TEEs) and secure enclaves to protect data during server-side processing. Although TEEs represent a significant advance in securing centralized systems, hardware vulnerabilities such as Meltdown [20] and Specter [21] have demonstrated inherent risks in hardware-based protection. Our MANET-based approach provides an orthogonal solution that could complement TEE-protected servers in hybrid deployments, combining the benefits of decentralized storage with hardware-protected processing when centralized computation is necessary.

Mu et al. [22] made significant advances by combining caching with k-anonymity while considering temporal query patterns. We build on their temporal awareness in our cache management strategy while extending it to handle the dynamic nature of MANET environments. Their work demonstrated the importance of considering time-based patterns in privacy preservation, which we incorporate into our cache obsolescence handling.

2.3. MANET-Based Approaches

MANET research has extensively explored challenges similar to those we address, particularly in vehicular networks (VANETs). Gupta et al. [23], Jin et al. [24], and Feng et al. [25] tackled connectivity and mobility challenges in VANET environments. However, our scenario presents different challenges as we consider human-carried mobile devices with inherent energy constraints, while VANET nodes have consistent vehicle power supply.

Several studies have advanced MANET caching techniques that we adapt for privacy preservation. Rathod et al. [26] presented collaborative caching strategies in wireless ad hoc networks (WANETs), where nodes share information stored in their cache. However, unlike in our study, information related to geographical location was not sought or maintained around a specific geographical location. Following this line, Liu et al. [27] developed a context-aware caching scheme focused on reducing server load, a goal we share, but we extend it to include privacy considerations. Sun et al. [28] optimized transmission distances in MANET caching, which is a concept that we incorporate into our geographic data distribution strategy. Ahmed Elfaki et al. [29] proposed query classification to reduce cache overhead, which we adapt to balance privacy protection with system efficiency.

2.4. Recent Developments in Edge Computing

Edge computing has emerged as a promising direction for improving location privacy, although it is still typically based on fixed infrastructure. Zhang et al. [2] proposed an innovative dual k-anonymity system using edge servers, demonstrating the potential of distributed privacy protection. Wu et al. [9] and Guizani et al. [30] further explored edge-based caching schemes, showing how distributed storage can enhance both performance and privacy. Although these approaches move away from fully centralized solutions, they maintain dependency on fixed edge infrastructure, which is a limitation our MANET-based approach overcomes.

Our work advances beyond these existing approaches by implementing the first fully distributed collaborative caching system in MANETs without external repositories. Although previous solutions like Niu et al. [6] relied on external access points and recent work has explored edge servers [2], our approach uniquely combines collaborative caching, privacy protection, and MANET capabilities without fixed infrastructure dependencies. This integration enables robust privacy protection while maintaining system efficiency through peer collaboration, addressing key limitations of previous approaches. By eliminating reliance on fixed infrastructure while preserving the benefits of distributed collaboration, our solution provides a more flexible and privacy-preserving framework for location-based services.

3. System Overview

We consider a system called M-LBS that consists of a fully distributed LBS implemented in a MANET, as shown in Figure 1.

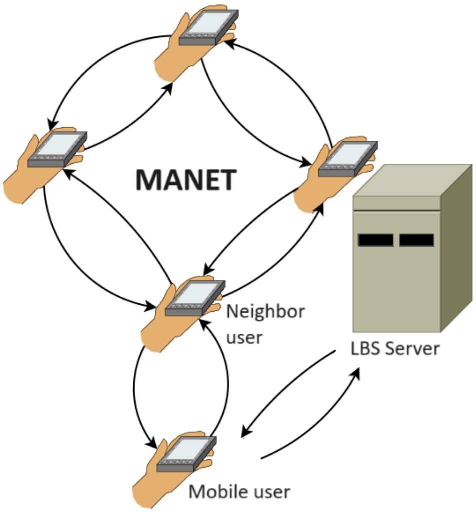

Figure 1. M-LBS system.

The system model is divided into three layers:

1. *Mobile user*: The end user is aware of its location and needs to solve a public location-based query (LBQ). Users can request details from restaurants or hotels near their location. A user or node moves in any direction in the service area without restrictions. It has storage, energy, and a limited transmission range to communicate with other users moving around. It can also communicate directly with the Internet using any existing cellular network if necessary. Without loss of generality, it is assumed that the hardware and software capabilities of the users are homogeneous.
2. *Neighbor users*: The geographical space where mobile users move is divided into disjoint square cells of $r \times r$, as shown in Figure 2. While a node-density-based division might seem intuitive, we chose this grid-based structure for several key advantages:
   - Predictable privacy guarantees independent of node density fluctuations;
   - Simplified and stable mapping between physical locations and cache responsibilities;
   - Reduced system overhead by eliminating the need for continuous boundary recalculation;
   - Direct compatibility with existing LBS spatial indexing systems and location cloaking approaches.

Within this structure, a group of users is identified as *neighbors* if their geographical location places them in the same cell. These neighbors collaborate to store and respond only to queries related to the cells that contain them. Neighbors are assumed to be honest, and the transmission model of a user is free-space and isotropic. The size of the diagonal of a cell is $r\sqrt{2}$ to ensure that when a mobile transmits, its neighbors hear it. See Figure 2, where the red circle, centered at a specific user's location, represents the user's transmission or coverage area. We assume that any neighbor within this region will receive the user's broadcasts. There are various link-layer protocols to regulate wireless communication within a cell. This study assumes no particular protocol because the distributed LBS implementation is performed at the application layer level.

3. *LBS server*: This server is located on the Internet and provides location-based information. It is considered the highest level of consultation that a user only wants to access as a last resort when it does not find a response from its neighbors. The LBS server provides reliable information but can compromise its users' location and query privacy.

Throughout this manuscript, we will use the terms "node" and "user" interchangeably; as in the jargon used in MANETs, the term "node" refers to a mobile user who carries a wireless device to communicate with others or with the cellular telephone network.

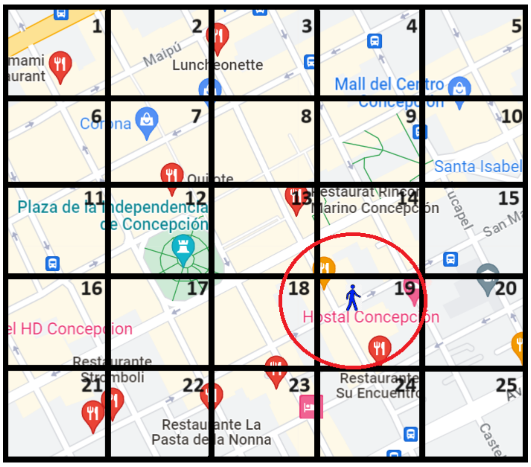

Figure 2. Network partition with the red circle indicating a user's coverage area.

When a mobile user needs to answer an LBQ, the node first queries its local cache. If the data in the local cache satisfy the user's query, we say that the user is successfully answered and that the service is considered complete. Otherwise, the user broadcasts its query to all its neighbors who search for answers in their respective caches. If at least one of the nodes has a response, then the response is transmitted to the querying user, assuming that the user is satisfied with it. If the node does not receive a response after a maximum waiting time, it may attempt to expand its query to neighbors in other nearby cells. As proposed in [10], if the network has disconnections between nodes due to low density, we propose that the user waits first some time to retry processing the query in the MANET (among neighboring nodes) and, when the MANET connection failed, can choose to access the LBS server on the Internet (compromising its privacy). In the latter case, the connection is assumed to have no delay.

4. Proposed M-LBS Scheme

In this section, we present details of the design of the M-LBS architecture. Then, we describe cell exit protocols that aim to reduce data redundancy within a cell. In addition, we explain the two-level caching management mechanism that stores the most relevant data within resident nodes. We also introduce the role of cell master along with privacy considerations for this role.

4.1. Design Details

Mobile users are always aware of their position and therefore know which cell they are in. When a user detects that it has crossed into a new cell, it broadcasts a greeting to all users in this cell to discover who the *cell master* is. The cell master is essentially a user who resides in the cell and collects statistics about the cell, such as the number of nodes present, the number of historical queries resolved in the cell, the frequency of query types, and the number of historical queries sent to the LBS. The new node can identify itself with the cell master by a pseudonym, which it maintains while moving within the cell.

Suppose that the cell master does not respond to a mobile user. In this case, the user concludes that the cell is empty, becomes the new cell master node, and attempts to locate the cell retention node (this is a former cell master node that is not present in the cell and maintains the cell statistics until the new master node is reached) using successive geocasts, following a protocol similar to that proposed in [10]. For our purposes, a geocast is a communication primitive that allows sending a message to all mobile users present in a geographical location, and it is not relevant to our system to determine the routing protocol under which it is implemented. To perform a geocast, we only need to identify the origin and destination cells of a message that is understood to be directed to all nodes located in the destination cell.

When a mobile user detects that they have left their current cell, a *cell exit protocol* is executed to distribute cache content among users who are within the previous cell. This data distribution maintains relevant information while minimizing redundancy within the cell. If the node is the latest node in the cell or the current nodes lack available cache storage, the user carries this information to the new cell. We propose two optimization-focused exit protocols in Section 4.4. Upon entering a new cell, the user sends a greeting message to establish their presence. If the mobile user was a cell master, they must first transfer this role to another node in their previous cell.

When a user moves within a cell, they can create an LBQ (location-based query), which is defined as a circular-range query centered on the user's position. This query seeks to determine the details of all public points of interest (POIs) in a specific category within the range. For example, a user looking for breakfast might query the menus and prices of restaurants in their neighborhood. The user determines which cells intersect with their query range, as shown in Figure 2 (where a red circle centered on the user intersects cells 13, 14, 18 and 19) and checks their cache for relevant data. For cells without cached data, they send a geocast requesting information. Users store all received data, implementing cache replacement policies as needed (detailed in Section 4.2).

If a user receives no response within the maximum waiting time, they prepare to access the LBS. This intention is communicated to the *cell master* for recording. Users concerned about privacy can also contact an anonymizer to obscure both their position and query [4,7]. When a node receives an LBQ, it searches its cache for responses. If found, it notifies its *cell master* and sends the grouped responses in a single geocast to the requesting cell.

4.2. Caching Management in M-LBS

In this work, we employ a two-level cache management system. The first level corresponds to the individual cache management performed by each mobile user. Since these users have limited cache memory available, we propose and evaluate various cache management techniques in our study. When the cache reaches its capacity and the user needs to insert new data, one of the following policies is implemented.

- *First-In-First-Out Elimination (FIFO)*: The element that was first added to the cache is removed, following an FIFO policy.
- *Random elimination (RAN)*: A random element is selected and deleted to free up space in the cache.
- *Distance Elimination (DIST)*: The distance between each element in the cache and the user's current location is calculated. The element with the longest distance is prioritized for deletion.

- *Popularity Elimination (POP)*: For each element in the cache of the node, a popularity score is calculated. This score is obtained by maintaining a counter that is incremented when an element is part of a response. The element with the lowest score is deleted.
- *Elimination of the minimum popularity/distance ratio (MINPD)*: For each element in the cache of a node, the popularity score is calculated and divided by the distance between the location of the element and the user. The element with the smallest ratio is deleted.
- *Reset of the cache when changing cells (RST)*: When the user changes cells, all the elements in the cache are deleted.

By implementing these cache management policies, we anticipate improved space utilization within the cache. The performance of these techniques is evaluated and compared in Section 6.

The second level of caching corresponds to the cell level. Here, we seek to maintain as much data as possible about POIs in the cell among the mobile users residing in that cell. Through managing the use of the cell exit protocol, we aim to reduce the search time for a response to a location-based query and increase the probability of finding a response in one or more user neighbors.

In this work, when measuring cache efficiency, we calculate the cache hit rate, which refers to the number of times the same location is queried, and its response is found either in the local cache memory of the user or in one of the neighbors (nodes in the querying user's cell) of the querying user. This idea is grounded on the observation that individuals' living habits are consistently regular, whereas their questioning habits frequently change at a steady rate.

One aspect is the data redundancy in a cell, which is measured in terms of how many replicas of metadata (extra data describing a location, for a restaurant, its menu, prices) about a POI are stored in the cache memory of the cell's users. While high redundancy allows one to reduce the search times for a response to a location-based query, it does not allow the storage of a wide variety of cell data, which could mean that the user has to access the LBS to resolve their query. Undoubtedly, a balance between response time and redundancy must be found to protect the location or query privacy of the mobile user.

4.3. Cache Freshness and Reset

A significant consideration in cache management is the period of validity or freshness of the cached data. This duration represents when the data are still considered valid before it should be removed from the cache. Different factors can influence the freshness such as dynamism in location-based services, user mobility, and data updates/maintenance from LBS providers, among others, also depending on what type of knowledge is being represented by a cache entry. For instance, details about a menu for a restaurant may only last a few hours, while availability/pricing information for rooms in hotels could be relevant up to several days with reference to particular times during the year; without any doubt, longer obsolescence periods equate higher hit rates. In order to deal with stale caches, we need some mechanism to reset them.

Commonly known as TTL or time-to-live (cache expiration time), past research works have shown (such as [2,11,13,14,29]) that cached items become stale or lose their freshness after staying in storage for a specific period of time after which they should be deemed no good and deleted either manually or automatically. If this does not happen, then subsequent requests will keep getting served outdated contents; hence, it such should never be allowed to happen.

The cache reset process can be implemented using various approaches, such as the following:

- *Time-based Reset:* Cached data are assigned a TTL value, and the cache is flushed or invalidated after the TTL expires. The TTL can be set based on the expected freshness requirements of the LBS data or the user mobility patterns [2].

- *Event-based Reset:* The cache is reset or invalidated in response to specific events, such as data updates from the LBS provider [13] or significant changes in user locations or context [14].
- *Hybrid Approach:* A combination of time- and event-based reset mechanisms can be employed, where the cache is reset based on which condition is first met (TTL expiration or relevant event occurrence) [11,29].

In this work, we use a hybrid approach that combines the cache expiration time and event-based scheme (when a user moves into a new cell).

4.4. Cell Exit Protocol

When a node leaves its current cell, it must execute a cell exit protocol to preserve data availability while minimizing redundancy. For example, when a user moves from cell 19 to cell 14 (Figure 2), the protocol ensures the efficient distribution of its cached data to the remaining nodes. Before describing the protocols, we establish their common elements:

Common Protocol Elements:

- *Goal:* Maintain data availability while minimizing redundancy;
- *Participants:* Departing node and remaining nodes (referred to as node *N*);
- *Prerequisites* List of current cell nodes from cell master;
- *Termination:* The protocol ends when either all the data are distributed or all the nodes are contacted.

We propose two protocols that prioritize different optimization goals:

Exit Protocol 1: Redundancy-Optimized

This protocol prioritizes minimizing data redundancy by checking for duplicate data before considering cache availability.

1. *Data Inventory:* Departing node sends its data list to node *N*;
2. *Redundancy Check:* Node *N* reports which data it already has;
3. *Duplicate* Removal: Departing node removes redundant data from transfer list;
4. *Availability Check:* Node *N* reports its available cache space;
5. *Data Transfer:* Departing node sends non-redundant data within availability limits;
6. *Confirmation:* Node *N* confirms successful data receipt.

Exit Protocol 2: Communication-Optimized

This protocol prioritizes minimizing message exchanges by checking cache availability before considering data redundancy.

1. *Availability Check:* Departing node requests node *N*'s cache space
2. *Space Verification:* Node *N* reports available cache space
3. *Data Inventory:* If space available, departing node sends its data list
4. *Redundancy Check:* Node *N* reports which data it already has
5. *Data Transfer:* Departing node sends non-redundant data within availability limits
6. *Confirmation:* Node *N* confirms successful data receipt

Protocols differ primarily in their optimization priorities. Protocol 1 ensures minimal redundancy but may require more messages when cache space is limited. Protocol 2 minimizes communication overhead but may result in greater redundancy when cache space is scarce.

Before executing any of these cell exit protocols, the departing node must verify whether it is the current *cell master* of the cell it leaves. If this is the case, the departing node sends a message to all nodes in the cell indicating that they must elect a new *cell master*. Each node interested in becoming a new *cell master* waits a time proportional to its distance from the center of the cell. The first node that exhausts its time informs the others that it will be the new *cell master* and receives all the information collected about cell statistics and the historical query frequency table for all cells.

4.5. Cell Master Privacy Considerations

The cell master role introduces potential privacy implications that must be carefully considered. While the cell master collects valuable system statistics, this information could be misused by an adversarial cell master. To mitigate this risk, we implement several privacy-preserving measures.

- *Limited Data Collection:* Cell masters only maintain aggregate statistics (node count, query frequencies) without individual user identifiers;
- *Temporal Restriction:* Historical data are periodically purged to prevent long-term pattern analysis;
- *Pseudonym Rotation:* Users regularly change their pseudonyms when entering new cells;
- *Distributed Trust:* Cell master role rotates among nodes to prevent prolonged data accumulation;
- *Query Anonymization:* Statistics record only query types not specific query content or user locations.

Even if a cell master becomes adversarial, their ability to compromise user privacy is limited because of the following:

- They cannot link queries across cells due to pseudonym changes;
- They only see aggregate patterns, not individual behaviors;
- Their view is temporally and spatially limited to their current cell;
- They cannot access the actual content of cached data in other nodes.

5. Data Security and Privacy Considerations

The LBS scheme we designed is distributed and has a dual purpose: protecting location privacy as well as increasing user consciousness on possible privacy infringement. This method ensures the security of private details while at the same time educating people about how their actions with the system can affect their privacy. In this part, we will discuss those important components of the scheme that need to be analyzed or considered so that effectiveness reliability and user friendliness are achieved. Our aim is to let users know more about their loss of privacy, making them have knowledge-based sharing decisions concerning data while appreciating features of our system that preserve anonymity.

In our model, MANET users are considered honest but curious, while LBS providers are viewed as potentially malicious. Mobile nodes face significant limitations in compromising user privacy due to the following:

- Limited storage and processing capacity;
- Restricted communication with users in other cells;
- Need to frequently erase cache for resource management.

These constraints make extensive user profiling difficult within the MANET. In contrast, centralized LBS providers can collect and analyze large amounts of data over time, posing greater risks to user privacy through potential data exploitation or vulnerability to hacking.

The queries considered in our scheme are range queries with respect to public places (e.g., restaurants, ATMs, banks, hotels, parks, hospitals). Although a single query might not reveal much information about an user, the collection of data over time could potentially reveal the user's lifestyle.

Our technique is intended to protect privacy at the application layer. To protect users' privacy, other techniques can add other protection layers, such as MAC address spoofing, to prevent tracking or identification of the user's device. The communication between users and the LBS server is protected by cryptographic techniques (e.g., TLS, transport layer), and sensitive data are stored in encrypted form. Our application layer privacy approach is orthogonal to other privacy techniques, such as VPNs (network layer), and so on.

Location Privacy Analysis

We consider a network system consisting of m disjoint spatial cells, each cell of size rxr as shown in Figure 2, containing n equally probable positions that represent the minimum location resolution. The probability of a cell being queried is proportional to its query frequency. To protect their privacy, users can select k cells to form a k-anonymity set when submitting location-based queries. We assume that a total of U users are moving within the network, and a fraction of them are located within each cell j.

The entropy of a single cell, $H(C)$, is given by $H(C) = \log_2 n$, which is based on the uniform probability distribution of the location of a user within the cell [31]. Before a user enters a cell or submits a query, the cell's master has no knowledge of the user's location, and the entropy is $H(\text{before query}) = \log_2(m \times n)$. After the query is broadcast to the cell, the cell's master (and other users in the cell) learns that the user is within its cell, reducing the entropy to $H(\text{after query}) = \log_2 n$. The privacy loss for the cell's master is thus privacy loss (Cell) $= \log_2 m$.

Similarly, the *LBS server's entropy* before the query is $H(\text{before query}) = \log_2(m \times n)$. After receiving a query with *k-anonymity*, the LBS server learns that the user is within one of the k cells in the anonymity set, resulting in an approximate entropy of $H(\text{after query}) \approx \log_2(k \times n)$. The privacy loss for the LBS server is privacy loss (LBS) $\approx \log_2(m/k)$.

The accumulated privacy loss (APL) is proposed to capture the cumulative privacy loss across all users in the system. The formulation for APL is based on the following considerations.

- For cell-based queries, the APL is defined as

$$APL(cell_j) = V_j \times N_j \times \log_2 m \tag{1}$$

  where V_j is the number of queries based on cell j issued by users in cell j, and N_j is the total number of nodes in cell j. This formulation accounts for the fact that all nodes in the cell potentially receive the query information, not just the master node. The privacy loss is thus multiplied by the number of nodes in the cell, as each of these nodes could potentially compromise the privacy of the querying users.
- For LBS-based queries with k-anonymity, the APL is defined as

$$APL(\text{LBS}) \approx \alpha \times U_{\text{LBS}} \times \log_2(m/k) \tag{2}$$

  where U_{LBS} is the number of queries sent to the LBS server, and α is a weighting factor to account for the LBS server's ability to maintain and accumulate historical information from all cells over time. In contrast to the cells' master node, the LBS server is a centralized entity that collects information from all cells over time, potentially leading to a higher cumulative privacy loss. The weighting factor α ($\alpha > 1$) can be chosen based on the desired level of privacy protection and the relative importance of the comprehensive view of the LBS server compared to the limited view of cell nodes. For simplicity, in this work, we assume $\alpha = 1$.

  The average APL across all cells is given by

$$\begin{aligned} APL(cell) &= \frac{1}{m}\sum_{j=1}^{m} APL(cell_j) \\ &= \frac{1}{m}\sum_{j=1}^{m} V_j \times N_j \times \log_2 m \end{aligned} \tag{3}$$

For APL(LBS) to be greater than the average APL(cell), the following condition should be satisfied:

$$U_{\text{LBS}} \times \log_2(m/k) > \frac{1}{m}\sum_{j=1}^{m} V_j \times N_j \times \log_2 m \tag{4}$$

This condition depends on multiple factors: the number of queries sent to the LBS (U_{LBS}), the number of cells in the *k-anonymity* set (k), the number of queries issued within each cell (V_j), and the total number of nodes in each cell (N_j). The inclusion of N_j in the cell-based privacy loss calculation reflects the potential for all nodes in a cell to access query information, not just the cell master. This comprehensive approach provides a more accurate representation of the privacy implications in our distributed MANET-based LBS system.

The relative magnitude of privacy loss between LBS-based and cell-based scenarios will depend on the specific values of these parameters in a given implementation. For example, if cells typically contain many nodes (N_j is large) and a significant number of queries are issued within cells (V_j is large), the loss of privacy based on cells could be substantial. Conversely, if the LBS is frequently queried (U_{LBS} is large) or if the *k-anonymity* sets are small, the loss of privacy based on LBSs could dominate.

Therefore, we may say that the total privacy loss of the system is

$$APL(System) = APL(LBS) + APL(cell) \tag{5}$$

Thus, the complete equation for the system's APL is

$$\begin{aligned} APL(System) &= U_{\text{LBS}} \times \log_2(m/k) \\ &+ \frac{1}{m} \sum_{j=1}^{m} V_j \times N_j \times \log_2 m \end{aligned} \tag{6}$$

This formulation provides a comprehensive view of the privacy implications in the entire system, clearly showing how the total system privacy loss is composed of two components: the privacy loss from LBS queries and the privacy loss from cell-based queries. Importantly, it highlights that a user who needs to access the LBS experiences a cumulative privacy loss, encompassing both the cell-based loss (from interactions within their local MANET cell) and the LBS-based loss (from querying the centralized LBS). This loss of privacy in the dual layer reflects the reality of our hybrid system, where users first attempt to resolve queries within their local cell before resorting to using LBSs.

6. Performance Evaluation

In this section, we present our experimental evaluation methodology, setup, and the results of our comparative analysis. Our evaluation addresses both the efficiency of our collaborative caching approach and its effectiveness in preserving user privacy in MANETs. We follow established practices in distributed systems research while considering the unique challenges of privacy-preserving location-based services.

6.1. Experimental Setup

The experiments were conducted on an Intel(R) Core (TM) i7-9750H 2.60 GHz with 16 GB memory running Java SE 8. Each experiment was repeated five times to calculate the mean values, following statistical significance practices established in similar LBS privacy studies [2,6].

Simulation Parameters and Baseline Design

Our experimental design is grounded in established MANET and LBS research.

- *Spatial Configuration:* We model a city environment using a 12 × 12 grid of disjoint cells, each sized 10 × 10 steps (“*p*” units) to simulate walking distances. This cell granularity aligns with spatial partitioning approaches from previous LBS privacy studies [6,10].

- *Node Density:* We deploy 400 nodes by default, following density patterns established in [10] and validated in collaborative caching schemes [29]. This density ensures sufficient peer availability while maintaining realistic urban scenarios.
- *Mobility Model:* Nodes follow a random walk pattern with speeds from 2 to 6 steps/s, simulating pedestrian movement. As shown independently by [32,33], random walk effectively models individual movement behavior without constraints of physical structure. While more sophisticated models exist, random walk provides conservative performance estimates with minimal assumptions.
- *Cache Configuration:* Cache capacity is intentionally restricted to reflect the resources of mobile devices [34]. Although cache hit rates can improve with larger caches (up to 30 elements), we focus on low-capacity configurations (2–6 POIs) to realistically model resource-limited mobile devices. Our results demonstrate that even with this restricted range, we achieve high hit rates while maintaining reasonable communication costs.

We evaluated three variants of the system:

- *Baseline:* A non-collaborative MANET-LBS with individual caches and no cell exit protocols, representing the simplest distributed implementation and it serves as our lower bound.
- *M-LBSv1:* Our proposed system with redundancy-optimized cell exit protocol.
- *M-LBSv2:* Our proposed system with communication-optimized cell exit protocol.

The evaluation uses four key metrics established in distributed caching and privacy literature:

- *Communication cost (CC)*: Number of messages exchanged during simulation [10], measuring system overhead.
- *Redundancy (RR)*: Proportion of duplicate POIs in user caches [35], which is measured at both city and cell levels. For cell-level analysis, we calculate the mean redundancy across cells containing users.
- *Hit rate (HR)*: The cache hit rate [6], which is defined as

$$\text{Hit rate (HR)} = \frac{\text{NH}}{\text{NH} + \text{NM}} \tag{7}$$

  where NH represents successful cache hits (answers found in local/neighbor caches) and NM represents cache misses requiring LBS access.
- *Number of accesses to the LBS (ACC)*: Frequency of LBS server queries when MANET responses fail [7], directly indicating privacy exposure risk.

Our experiments fix the city size, simulation time (300 s), and cell size while varying key parameters identified in privacy-preserving LBS literature [2,6]. The default configuration follows established benchmarks:

- Simulation time = 300 s (steady-state behavior) [29];
- Nodes = 400 (urban density patterns) [10];
- Cache size = 4 (balancing storage and hit rate) [6];
- Speed = 4 p/s (average pedestrian movement).

6.2. Cache Management Policy Selection

Prior to our main experiments, we evaluated various cache management policies to identify the most effective approach. Following established strategies in distributed systems [29,35], we compared several techniques: FIFO (first input, first output), RAN (random elimination), DIST (distance elimination), POP (minimum popularity elimination), and MINPD (minimum popularity and distance ratio elimination). All evaluations used our default configuration parameters.

As shown in Table 1, DIST and FIFO emerge as the most effective strategies, generating significantly fewer LBS queries. DIST's superior performance (14,698 LBS accesses versus

FIFO's 15,336) stems from its geographical awareness, which is crucial for spatial queries. On the basis of these results, we adopt DIST for all subsequent experiments.

Table 1. Performance comparison of cache management policies.

Management Types	Number of Hits	Accesses to LBSs
FIFO	104,664	15,336
RAN	100,469	19,531
DIST	105,302	14,698
RST	77,580	42,420
POP	97,290	22,710
MINPD	100,717	19,283

6.3. Effect of Cache Memory Size

We evaluated system performance across our restricted cache capacity range (2–6 POIs) while maintaining constant node count (400) and speed (4 p/s). Cache size variation is particularly critical as it directly impacts both system efficiency and privacy protection [6]. Our analysis focuses on restricted cache capacities to realistically model resource constraints in mobile devices [34] while examining the performance implications within this constrained range.

An analysis of Tables 2–4 reveals several significant patterns consistent with distributed caching theory [29]:

1. *Redundancy Control:* The baseline system shows consistently high redundancy (24–27%) across all cache sizes due to uncoordinated caching. In contrast, our collaborative approaches maintain significantly lower redundancy levels (M-LBSv1: 4–10%, M-LBSv2: 6–12%), validating the effectiveness of our cell exit protocols in managing data distribution.
2. *Hit Rate Performance:* Even with restricted cache sizes, both collaborative systems achieve excellent hit rates (up to 0.95), significantly outperforming the baseline (0.91 maximum). This shows that effective coordination can compensate for the limited individual cache capacity.
3. *Communication Efficiency:* While larger caches reduce communication costs across all systems by increasing local hit probability, this effect is particularly pronounced in our collaborative approaches. M-LBSv1 shows the most significant improvement, reducing ACC from 26,495 to 5404 as the cache size increases from two to six POIs.

Table 2. Cache sizes, baseline.

Cache Size	ACC	NH	HR	Average RR per Cell	CC
2	29,194	90,806	0.76	24%	3.31
3	19,456	100,544	0.84	27%	2.58
4	14,398	105,602	0.88	25%	2.16
5	12,374	107,626	0.90	26%	1.94
6	10,459	112,541	0.91	24%	1.78

Table 3. Cache sizes, M-LBSv1.

Cache Size	ACC	NH	HR	Average RR per Cell	CC
2	26,495	93,505	0.78	4%	4.62
3	15,387	104,613	0.87	8%	4.03
4	8087	111,913	0.93	9%	3.69
5	6401	113,599	0.95	10%	3.63
6	5404	114,596	0.95	9%	3.56

Table 4. Cache sizes, M-LBSv2.

Cache Size	ACC	NH	HR	Average RR per Cell	CC
2	27,270	92,730	0.77	6%	4.68
3	16,046	103,954	0.87	10%	4.17
4	8436	111,564	0.93	10%	3.85
5	6606	113,394	0.94	11%	3.74
6	5681	114,319	0.95	12%	3.75

Figure 3 illustrates that while larger cache sizes improve performance across all systems, we observe diminishing returns beyond size 4, which is consistent with previous studies of LBS caching [6]. This validates our choice of restricted cache sizes, as additional capacity provides minimal benefit while increasing resource requirements. Both collaborative systems consistently outperform the baseline in reducing LBS queries with M-LBSv1 showing superior performance due to its redundancy-first optimization strategy. The results demonstrate that our approach achieves excellent performance even with limited cache capacity, supporting our focus on resource-constrained mobile environments.

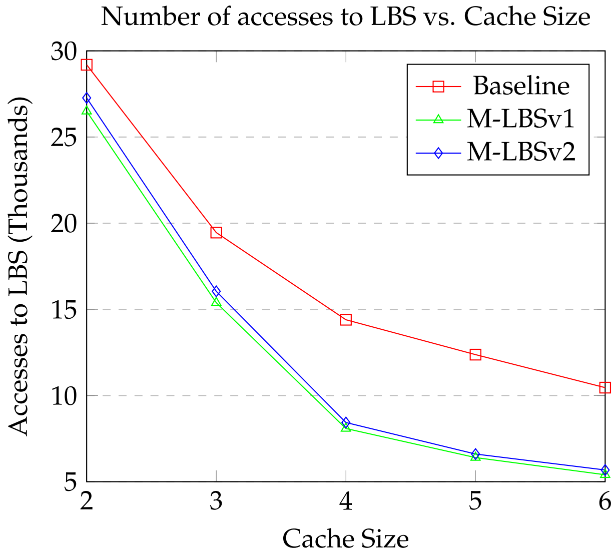

Figure 3. Number of LBS accesses when the cache size is varied.

6.4. *Effect of Speed*

The mobility of the nodes significantly affects the performance of the system in MANETs [10,33]. We analyzed this impact by varying node speeds while maintaining constant node count (400) and cache size (4). Our speed range (2–6 p/s) represents realistic pedestrian movement patterns in urban environments [2].

An analysis of Tables 5–7 reveals several critical insights:

1. *Redundancy Dynamics:* System-wide redundancy decreases with increasing speed across all approaches, which is consistent with expected MANET caching behavior [29]. The baseline shows the most dramatic reduction (34% to 21%), while our collaborative approaches maintain more stable, lower redundancy levels (M-LBSv1: 10% to 8%, M-LBSv2: 14% to 8%).
2. *Hit Rate Stability:* Despite increasing mobility, both collaborative approaches maintain high hit rates (M-LBSv1: 0.97 to 0.89, M-LBSv2: 0.97 to 0.89) compared to the higher degradation of the baseline (0.95 to 0.81). This demonstrates the resilience of our cell exit protocols to mobility effects.
3. *Communication Cost:* While increasing speed results in higher communication costs across all systems, our collaborative approaches show more controlled increases,

particularly at higher speeds. This suggests a better adaptability to dynamic network conditions.

Table 5. Speed, baseline.

Speed	ACC	NH	HR	Average RR per Cell	CC
2	6483	113,517	0.95	34%	1.2
3	10,461	109,539	0.91	32%	1.69
4	14,398	105,602	0.88	25%	2.16
5	19,485	100,515	0.84	23%	2.59
6	23,148	96,852	0.81	21%	2.93

Table 6. Speed, M-LBSv1.

Speed	ACC	NH	HR	Average RR per Cell	CC
2	3814	116,186	0.97	10%	2.87
3	5989	114,011	0.95	9%	3.28
4	8087	111,913	0.93	9%	3.69
5	10,997	109,003	0.91	9%	4.25
6	13,043	106,957	0.89	8%	4.67

Table 7. Speed, M-LBSv2.

Speed	ACC	NH	HR	Average RR per Cell	CC
2	4128	115,872	0.97	14%	2.77
3	6100	113,900	0.95	13%	3.34
4	8436	111,564	0.93	10%	3.85
5	11,430	108,570	0.90	9%	4.42
6	13,092	106,908	0.89	8%	4.91

Figure 4 shows that increased node mobility leads to more frequent LBS queries across all systems. However, this performance degradation is significantly less severe in our collaborative approaches with M-LBSv1 and M-LBSv2 maintaining considerably lower LBS access rates even at high speeds. Cell exit protocols effectively maintain data availability within cells despite increased node movement, resulting in enhanced privacy protection through a sustained reduction in LBS dependency. The results validate our protocol design's effectiveness in managing mobility-induced challenges while maintaining privacy protection.

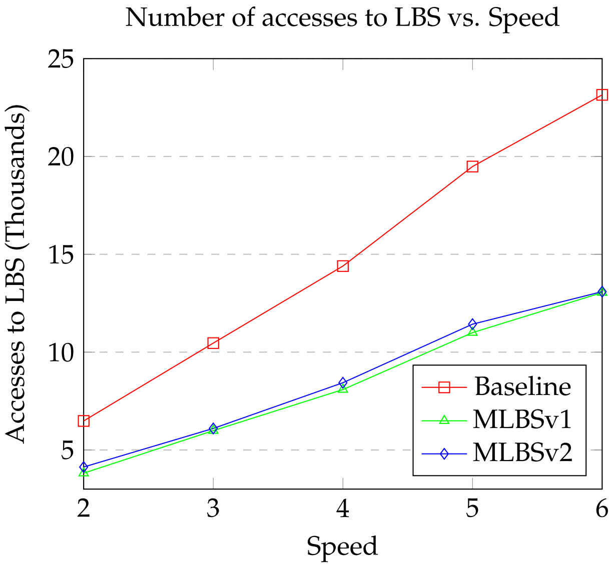

Figure 4. Number of LBS accesses when the speed of the nodes is varied.

6.5. Privacy Loss

To evaluate privacy protection effectiveness, we analyze privacy loss patterns across different k-anonymity values for all three approaches. Our analysis is based on entropy-based privacy metrics established in information theory [31] and follows standard practices in k-anonymity evaluation [3,4].

Figure 5 demonstrates several key insights aligned with the privacy-preserving LBSs literature [2,8]:

1. *Anonymity–Privacy Relationship:* All approaches show a decrease in privacy loss as k increases, validating theoretical predictions about the effects of the size of the anonymity set [36]. This confirms that larger anonymity sets strengthen privacy protection even in distributed environments.
2. *Comparative Performance:* The baseline consistently shows a higher loss of privacy (110.5 K to 67.3 K) compared to our collaborative approaches (M-LBSv1: 72.5 K to 48.3 K; M-LBSv2: 75.7 K to 50.4 K). This substantial improvement comes from reduced centralized LBS queries through effective collaborative caching.
3. *Diminishing Returns:* Privacy improvement rates diminish at higher k-values, particularly beyond k = 6, which is in line with recent research on the preservation of privacy by LBSs [2]. This suggests that there is an optimal k-value that balances privacy protection with system overhead.

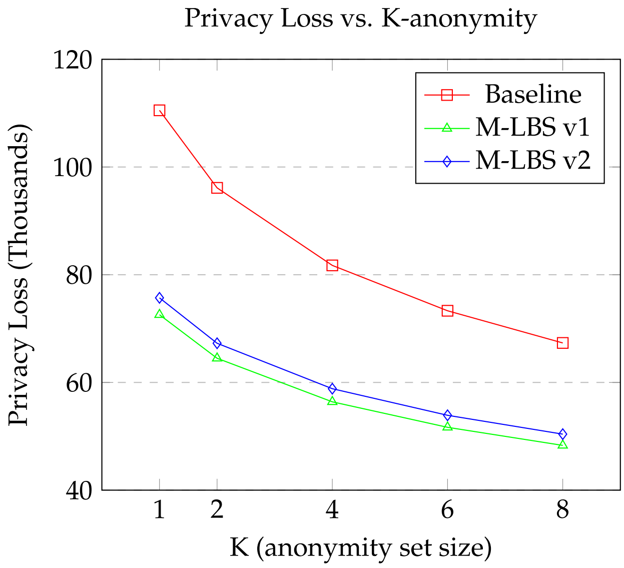

Figure 5. Privacy loss for different k-anonymity values across three approaches.

6.6. Effect of Obsolescence

Data obsolescence in cache-based systems significantly impacts both performance and privacy protection [11,29]. We analyze how expiration time affects query distribution between LBSs and MANET nodes.

Our analysis of Figure 6 reveals several critical patterns:

1. *LBS Query Reduction:* As the expiration time increases, LBS queries decrease across all systems:
    - Baseline: 2.97 to 1.74 × 10^4;
    - M-LBSv1: 2.09 to 1.04 × 10^4;
    - M-LBSv2: 2.09 to 1.11 × 10^4.

    This pattern aligns with cache efficiency studies in distributed environments [11], demonstrating that longer data validity periods effectively reduce external query requirements.
2. *MANET Response Evolution:* Local query resolution improves with longer expiration times:

- Baseline: 5.15 to 5.29 × 10^4;
- M-LBSv1: 6.78 to 7.23 × 10^4;
- M-LBSv2: 6.65 to 6.98 × 10^4.

This trend matches the expectations of collaborative caching research [13], showing improved local resolution capacity over time.

3. *Collaborative Advantage:* Both M-LBS variants consistently outperform the baseline to reduce LBS queries, the performance gap widening at longer expiration times. This demonstrates the robustness of our collaborative approach to data aging, which is a critical factor in mobile environments [14].
4. *Privacy Implications:* The reduced frequency of LBS queries with longer expiration times suggests enhanced privacy protection through decreased exposure to centralized servers. However, this benefit must be balanced against the requirements for data freshness, as noted in recent LBSs research preserving privacy [7].

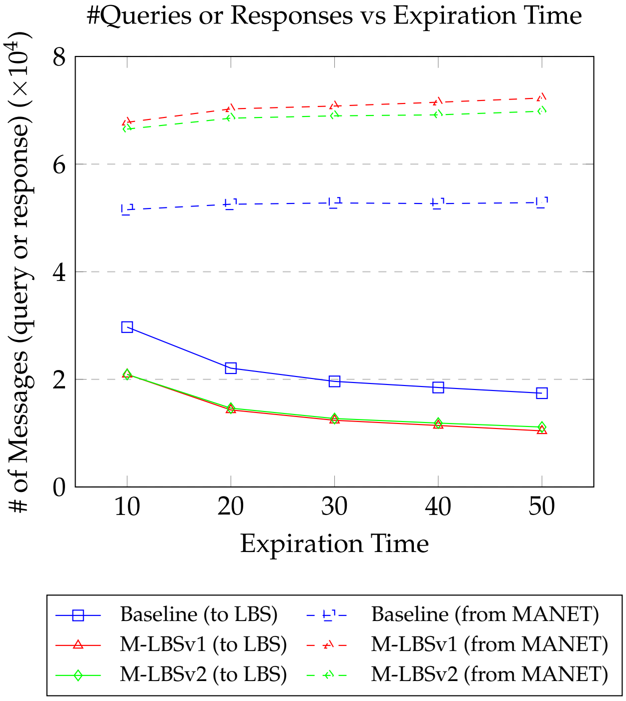

Figure 6. Number of queries sent to LBSs and number of query responses from MANET vs. expiration time. Solid lines represent queries sent to LBSs; dashed lines represent query responses from MANET.

The relationship between expiration time and query distribution demonstrates that careful cache management can significantly enhance privacy protection while maintaining service quality. Our results suggest that adaptive expiration policies, potentially based on data type and usage patterns [29], could further optimize this trade-off between data freshness and privacy protection.

7. Conclusions and Future Work

This study proposes a novel collaborative caching strategy to implement privacy-aware decentralized LBSs in MANETs. Our main contributions include (1) a peer-to-peer LBS architecture enabling direct query resolution within MANETs, (2) a two-tier caching system combining local and proximity-based storage, (3) location-aware data storage strategies optimizing geographical proximity, and (4) an accumulated privacy loss metric quantifying privacy implications in hybrid systems. Experimental results demonstrated that our collaborative approach effectively reduces LBS server queries and data redundancy while preserving privacy despite increased communication costs. The evaluation of data obsolescence showed that optimized expiration times enhance system effectiveness in promoting local query resolution.

Future work will focus on the following:

- Optimizing cache management for varying data validity requirements.
- Implementing decentralized cloaking mechanisms.
- Incorporating privacy budgets to enhance user awareness and control.

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

Funding: This work was supported by Vicerrectoría de Investigación y Doctorados de la Universidad San Sebastián—Fondo USS-FIN-24-APCS-30 and by Universidad del Bío-Bío, Chile, under grant INN I+D 23-53.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This research was funded by Universidad San Sebastián.

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

References

1. Liu, Y.; Mao, Y.; Shang, X.; Liu, Z.; Yang, Y. Distributed Cooperative Caching in Unreliable Edge Environments. In Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer Communications, Virtual, 2–5 May 2022; pp. 1049–1058.
2. Zhang, S.; Hu, B.; Liang, W.; Li, K.C.; Gupta, B.B. A Caching-Based Dual K-Anonymous Location Privacy-Preserving Scheme for Edge Computing. *IEEE Internet Things J.* **2023**, *10*, 9768–9781. [CrossRef]
3. Gruteser, M.; Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In Proceedings of the ACM MobiSys, San Francisco, CA, USA, 5–8 May 2003.
4. Niu, B.; Gao, S.; Li, F.; Li, H.; Lu, Z. Protection of Location Privacy in Continuous LBSs against Adversaries with Background Information. In Proceedings of the International Conference on Computing Networking and Communications and Information Security ICNC, Kauai, HI, USA, 15–18 February 2016.
5. Tobar, G.; Galdames, P.; Gutierrez-Soto, C.; Rodriguez-Moreno, P. A Batching Location Cloaking algorithm for Location Privacy. In Proceedings of the Workshop on Collaborative Technologies and Data Science in Smart City Applications CODASSCA, Yerevan, Armenia, 12–15 September 2018; pp.26–36.
6. Niu, B.; Li, Q.; Zhu, X.; Cao, G.; Li, H. Enhancing Privacy through Caching in Location-Based Services. In Proceedings of the IEEE INFOCOM, Kowloon, Hong Kong, 26 April–1 May 2015.
7. Gutierrez-Soto, C.; Galdames, P.; Faúndez, C.; Durán-Faúndez, C. Location-Query-Privacy and Safety Cloaking Schemes for Continuous Location-Based Services. *Mob. Inf. Syst.* **2022**, *2022*, 22. [CrossRef]
8. Galdames, P.; Gutierrez-Soto, C.; Curiel, A. Batching Location Cloaking Techniques for Location Privacy and Safety Protection. *Mob. Inf. Syst.* **2019**, *2019*, 11. [CrossRef]
9. Wu, H.; Fan, Y.; Wang, Y.; Ma, H.; Xing, L. A Comprehensive Review on Edge Caching from the Perspective of Total Process: Placement, Policy and Delivery. *Sensors* **2021**, *21*, 5033 . [CrossRef] [PubMed]
10. Galdames, P.; Kim, K.; Cai, Y. A Generic Platform for Efficient Processing of Spatial Monitoring Queries in Mobile Peer-to-Peer Networks. In Proceedings of the Eleventh International Conference on Mobile Data Management MDM, Kansas City, MO, USA, 23–26 May 2010.
11. Jung, K.; Park, S. Collaborative caching techniques for privacy-preserving location-based services in peer-to-peer environments. In Proceedings of the 2017 IEEE International Conference on Big Data, Big Data 2017, Boston, MA, USA, 11–14 December 2017; Volume 2018-January; pp. 4497–4506.
12. Zhang, S.; Liu, Q.; Wang, G. A caching-based privacy-preserving scheme for continuous location-based services. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 10067 LNCS, pp. 73–82.
13. Peng, T.; Liu, Q.; Meng, D.; Wang, G. Collaborative trajectory privacy preserving scheme in location-based services. *Inf. Sci.* **2017**, *387*, 165–179. [CrossRef]

14. Zhang, S.; Li, X.; Tan, Z.; Peng, T.; Wang, G. A caching and spatial [Formula presented]-anonymity driven privacy enhancement scheme in continuous location-based services. *Future Gener. Comput. Syst.* **2019**, *94*, 40–50. [CrossRef]
15. Ghinita, G.; Kalnis, P.; Khoshgozaran, A.; Shahabi, C.; Tan, K.L. Private Queries in Location Based Services: Anonymizers Are Not Necessary. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 9–12 June 2008; SIGMOD '08; pp. 121–132. [CrossRef]
16. Jagarlapudi, H.N.S.S.; Lim, S.; Chae, J.; Choi, G.S.; Pu, C. Drone Helps Privacy: Sky Caching Assisted k-Anonymity in Spatial Querying. *IEEE Syst. J.* **2022**, *16*, 6360–6370. [CrossRef]
17. Alsaawy, Y.; Alkhodre, A.; Eassa, F.A.; Sen, A.A.A. Triple cache approach for preserving privacy and enhancing performance of LBS. In Proceedings of the 2019 6th International Conference on Computing for Sustainable Global Development, INDIACom 2019, New Delhi, India, 13–15 March 2019; pp. 1277–1281.
18. Alrahhal, M.S.; Khemakhem, M.; Jambi, K. A survey on privacy of location-based services: Classification, inference attacks, and challenges. *J. Theor. Appl. Inf. Technol.* **2017**, *95*, 6719–6740.
19. Yang, X.; Yue, C.; Zhang, W.; Liu, Y.; Ooi, B.C.; Chen, J. SecuDB: An In-enclave Privacy-Preserving and Tamper-resistant Relational Database. *Proc. Vldb Endow.* **2024**, *17*, 3906–3919. [CrossRef]
20. Lipp, M.; Schwarz, M.; Gruss, D.; Prescher, T.; Haas, W.; Fogh, A.; Horn, J.; Mangard, S.; Kocher, P.; Genkin, D.; et al. Meltdown: Reading Kernel Memory from User Space. In Proceedings of the 27th USENIX Security Symposium, Baltimore, MD, USA, 15–17 August 2018; pp. 973–990.
21. Kocher, P.; Horn, J.; Fogh, A.; Genkin, D.; Gruss, D.; Haas, W.; Hamburg, M.; Lipp, M.; Mangard, S.; Prescher, T.; et al. Spectre Attacks: Exploiting Speculative Execution. In Proceedings of the 40th IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–22 May 2019; pp. 1–19.
22. Mu, X.; Shen, H.; Lu, Z. A temporal caching-aware dummy selection location Algorithm. In Proceedings of the 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2019), Gold Coast, Australia, 5–7 December 2019; IEEE: New York, NY, USA, 2019; pp. 501–504.
23. Gupta, R.; Rao, U.P. Achieving location privacy through CAST in location based services. *J. Commun. Netw.* **2017**, *19*, 239–249. [CrossRef]
24. Jin, H.; Papadimitratos, P. Resilient privacy protection for location-based services through decentralization. In Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, Boston, MA, USA, 18–20 July 2017; pp. 253–258.
25. Feng, B.; Feng, C.; Feng, D.; Wu, Y.; Xia, X.G. Proactive Content Caching Scheme in Urban Vehicular Networks. *IEEE Trans. Commun.* **2023**, *71*, 4165–4180. [CrossRef]
26. Rathod, B.R.; Kumar, C.A.; Narsimha, G. Distributed Cooperative Caching Strategies in Wireless Ad Hoc Networks. *Int. J. Comput. Appl.* **2012**, *57*, 20–25
27. Liu, Y.; Zhu, D.; Ma, W. A novel cooperative caching scheme for Content Centric Mobile Ad Hoc Networks. In Proceedings of the IEEE Symposium on Computers and Communications, Messina, Italy, 27–30 June 2016; pp. 824–829.
28. Sun, J.; Xu, W.; Wang, C.; Chen, X. Response Time-Delay Analysis of Cooperative Cache in MANETs. In Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, Beijing, China, 18–20 October 2019; pp. 637–643.
29. Ahmed Elfaki, M.; Ibrahim, H.; Mamat, A.; Othman, M.; Safa, H. Collaborative caching priority for processing requests in MANETs. *J. Netw. Comput. Appl.* **2014**, *40*, 85–96. [CrossRef]
30. Guizani, M. Diversity-Driven Proactive Caching for Mobile Networks. *IEEE Trans. Mob. Comput.* **2023**, 23, 7878–7894.
31. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 2006. [CrossRef]
32. Bettstetter, C. Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects. In Proceedings of the 4th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), Rome, Italy, 21 July 2001; pp. 55–62. [CrossRef]
33. Kim, K.; Cai, Y.; Tavanapong, W. Safe-Time: Distributed Real-Time Monitoring of cKNN in Mobile Peer-to-Peer Networks. In Proceedings of the 9th International Conference on Mobile Data Management (MDM), Ames, IA, USA, 27–30 April 2008; pp. 124–133. [CrossRef]
34. Du, Y.; Gupta, S.K.; Varsamopoulos, G. Improving on-demand data access efficiency in MANETs with cooperative caching. *Ad Hoc Netw.* **2009**, *7*, 579–598. [CrossRef]
35. Naz, S.; Naveed Bin Rais, R.; Shah, P.A.; Yasmin, S.; Qayyum, A.; Rho, S.; Nam, Y. A dynamic caching strategy for CCN-based MANETs. *Comput. Netw.* **2018**, *142*, 93–107. [CrossRef]
36. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. L-Diversity: Privacy beyond k-anonimity. In Proceedings of the IEEE International Conference on Data Engineering ICDE, Atlanta, GA, USA, 3–8 April 2006.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.