An Approximate Set Membership Approach to Resilient Multi-Robot Communication

Nicholas Smith[1], Jen Jen Chung[2], and Graeme Best[1]

***Abstract*— Effective communication is critical for coordinating multi-robot teams, yet practical deployments often face severe bandwidth constraints and frequent message loss. This paper presents a communication protocol that leverages Bloom filters to enable efficient, approximate set membership queries in multi-robot systems. Bloom filters offer a tunable trade-off between false positive rate and memory footprint, making them well suited for bandwidth-limited communication. To mitigate the effects of false positives, we introduce a salting strategy that decorrelates Bloom filters and enables *stacking*— the combination of membership queries across multiple filters. These stacked results are incorporated into each robot's belief map, such that only sufficiently corroborated information influences frontier generation and exploration planning. We evaluate our proposed communication protocol in a multi-robot exploration task, where robots share information about their observed cells to enable efficient coverage. Our results demonstrate that compared to exact methods, our Bloom filter-based protocol reduces communication cost by up to 6× while maintaining team exploration performance, even under severe communication dropouts.**

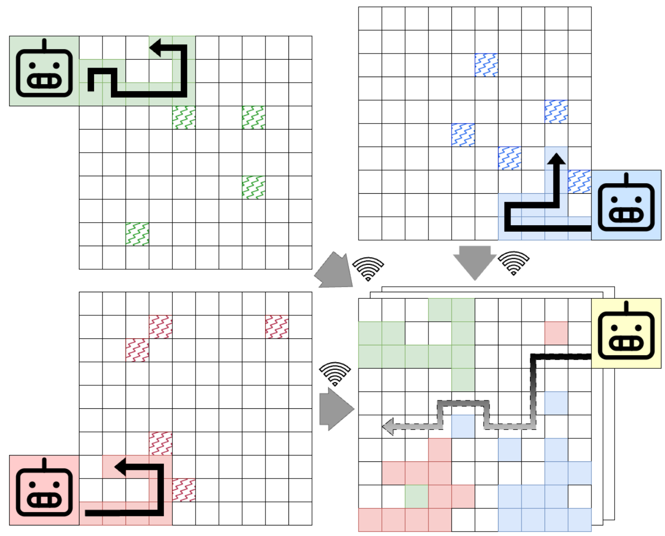

Fig. 1. Resilient communication allows multi-robot teams to coordinate even with packet loss and limited bandwidth. Three robots encode their explored areas as Bloom filters, which compactly represent observed cells but may include false positives (shaded). The fourth robot fuses the received filters using *stacking* and *decorrelation* to suppress false positives and plan efficiently, avoiding redundant exploration and sustaining performance under degraded network conditions.

I. Introduction

The ability of a *Multi-Robot System* (MRS) to coordinate effectively depends on reliable and timely communication between its constituent robots. Communication is critical for many reasons: it prevents redundant work, synchronises decision-making, and ensures that critical information, such as maps of previously unexplored environments, is shared to the entire team. In practice, however, robots deployed in the real world are typically subject to degraded communication, such as bandwidth constraints, latency, and packet loss. These challenges demand carefully designed communication protocols that maximise the amount of useful data that is shared across a team [1], [2].

A key difficulty in such scenarios is achieving *resilient communication*: maintaining team performance when messages are lost, while operating within bandwidth budgets. Many existing approaches focus on improving connectivity or routing to reduce message loss [3], but these methods cannot guarantee reliable delivery when even the best available network suffers high dropout rates. Moreover, wireless bandwidth is a shared and limited resources, often contended by multiple robots transmitting simultaneously and by other data streams such as telemetry and raw sensor feeds. Simply retransmitting lost data or increasing redundancy can quickly saturate the network, leading to congestion and degraded performance. What is needed is a communication scheme that is inherently robust to message loss, yet minimises the number of bytes transmitted, ensuring that the most critical information is delivered reliably and in a timely manner without overwhelming the network.

In this paper, we address this problem by formulating resilient communication as an *approximate set membership* problem (see Fig. 1). Many types of information exchanged between robots can be naturally represented as sets—for example, the collection of observed voxels during exploration [4], [5] or the set of landmarks discovered in a search task [6], [7]. Inspired by set-theoretic approaches to localisation [8], [9], we leverage Bloom filters to compactly encode these sets and transmit them efficiently. Because Bloom filters have a tunable size based on a desired false positive rate, they enable predictable, low-latency transmissions, which is a crucial property when data must be delivered promptly under bandwidth constraints.

To mitigate the effects of false positives, we introduce

[1]Nicholas Smith and Graeme Best are with the Robotics Institute, University of Technology Sydney, NSW, Australia `nicholas.j.smith-2@student.uts.edu.au`, `graeme.best@uts.edu.au`

[2]Jen Jen Chung is with the School of Electrical Engineering and Computer Science, The University of Queensland, Brisbane, QLD, Australia `jenjen.chung@uq.edu.au`

This research was supported by the Commonwealth through an Australian Government Research Training Program Scholarship [DOI: https://doi.org/10.82133/C42F-K220].

a *salting* strategy that decorrelates Bloom filters across transmissions and enabled *stacking*—the corroboration of membership queries across multiple filters, either from different robots or from repeated messages. This design significantly reduces the probability of persistent false positives and allows robots to make probabilistic decisions about set membership with high confidence, even when some packets are dropped.

We evaluate our approach in a multi-robot exploration scenario where a team of robots share their observed cells to achieve coverage under bandwidth limits and probabilistic message loss. Across a wide range of false-positive rates and observation filtering thresholds, our method maintains exploration efficiency close to that of exact map sharing while reducing communication cost by up to a factor of 6 compared to the compressed full-map baseline. Performance remains near-optimal even under severe network degradation, with only a small increase in exploration time at high packet-loss rates. These results demonstrate that approximate set membership with salting and stacking is an effective strategy for enabling bandwidth-efficient, resilient, and timely coordination in multi-robot teams.

II. Related Work

A. Resiliency in Multi-Robot Communication

Current methods for enabling robustness and resiliency in MRS communication focus on reducing the amount of data sent, while other methods aim to improve link reliability between robots by physically shifting their location. Compression based methods enable a reduction in data and have been used to communicate occupancy maps [10] but performance is limited by how well the data compresses. Highly structured data compresses well, while unstructured data may not. Other work optimises data exchange methods to send only new data generated since the last communication event [11]. However, this approach risks data loss if messages are dropped, as undelivered data is not retransmitted. Other work aims to improve link quality [12]–[14] between robots by physically shifting their location, resulting in a favorable network topology. However, when the best link available fails to meet reliability requirements, the data does not compress well, or when it is too difficult to decide what data to include and what to exclude, there are limited fallback options.

This is the problem that our work addresses. We propose a method that approximates the source data to within a high degree of accuracy and then uses that approximation as if it were the real data. Our approximation is based on approximate set membership and uses Bloom filters with additional decorrelation to achieve a data representation and query mechanism that is a small constant size, and accurate to within a tunable false-positive rate.

B. Approximate Set Membership Queries

Approximate Set Membership Queries (AMQ), are a data representation and query method that offers a tractable way to scale queries to millions or billions of elements, trading some correctness for a large reduction in the amount of space required [15]. The literature details a number of data structures designed for AMQ: Bloom filters, the seminal AMQ data structure, exhibit a small constant time and linear space overhead and feature zero false negatives [15], [16]. *Invertible Bloom Lookup Tables* are designed to allow fast computation of the symmetric set difference, elements that are in neither set, but exhibit a per-element space overhead that is large compared to Bloom filters [17].

C. Bloom Filters

A data structure commonly associated with approximate set membership queries is the *Bloom filter* [16]. This is a compressed probabilistic representation of a set, which can be queried to determine if a given member is in the set, with a tunable false positive rate, and notably, a zero false negative rate. They are commonly used for caching [18], [19] and database querying [20]. Of particular interest to this work are protocols that use Bloom filters for set reconciliation, often framed as joining two large databases [21]–[23]. Our method takes inspiration from these approaches and brings it to the robotics context. By salting Bloom filter keys with a unique random value per filter, we are able to treat Bloom filters as probabilistic evidence that a location on the map has been observed.

Bloom filters have seen limited use in robotics to date. The *counting Bloom filter* has been used to speed up volumetric mapping in a single robot case by reducing the voxel occupancy query times and the required space [24]. Furthermore, the Boolean satisfiability problem, which is fundamental to linear temporal logic robot motion planning, is improved by applying Bloom filters to perform common variable detection [25]. However, previous work stops short of directly applying Bloom filters to the problem of multi-robot communication.

III. Problem Description

We propose to apply approximate set membership queries for individual robots to efficiently and robustly recover the joint information across the team. In this work, we focus on the application of coordinated exploration where the goal of the multi-robot team is to maximise coverage over an a priori unknown map in the shortest amount of time. While we present our communication algorithm in the context of this application, we note that the method is application agnostic and can be used for general tasks requiring multiple agents to consolidate shared information, provided that the information can be represented as a set. We will discuss other promising application areas further in Section VI.

In our multi-robot exploration task, the environment is discretised into a grid map, with $\mathcal{M}$ the set of all cells in the map. Each robot $r \in \mathcal{R}$ is initialised with its own copy of $\mathcal{M}$ and as it moves in the environment, it observes cells within its sensor field of view, which are then entered into its *observed* set $\mathcal{C}_r$. By sharing their observed sets with one another, robots in a team can explore more efficiently as they would have consensus on the unseen portions of the map, defined as $\mathcal{C}' = \mathcal{M} \setminus \bigcup_{r \in \mathcal{R}} \mathcal{C}_r$, preventing unnecessary re-observation of regions that have already been explored by another robot.

However, naïvely sharing sets can be infeasible as real-world communication constraints may be prohibitive (in time and bandwidth), and stochastic message dropouts can result in significant data loss if some level of redundancy is not encoded in the communication protocol. Specifically, we consider the case where communication is subject to message dropouts with probability P_{drop} and limited bandwidth B. In other words, it's infeasible for all robots to share their full set of information at each communication round (maximally redundant), while only sending the newly observed cells may result in lost information (no redundancy). Instead we offer a strategy that strikes a desirable middle ground between these two extremes through the use of approximate set membership query with Bloom filters.

IV. Multi-Robot Communication with Decorrelated Bloom Filters

Our proposed method requires each robot to construct a Bloom filter of its current observed set, which it then broadcasts to the team. When deciding where to explore next, each robot r performs a set membership query of all cells that are not in its observed set, i.e. $\mathcal{M} \setminus \mathcal{C}_r$, across all of its received Bloom filters in order to compute the approximate consensus set $\tilde{\mathcal{C}}'$ of unobserved cells across the entire team. For coordinated exploration contexts, the set $\tilde{\mathcal{C}}'$ is then used to compute a suitable frontier waypoint for the robot to visit. See Algorithm 1 for the pseudocode of our method.

Algorithm 1: Bloom Filter-based Coordinated Exploration (For Each Robot $r \in \mathcal{R}$)

input : $\mathcal{M}$ ▷ Set of all map cells
$\mathcal{C}_r$ ▷ Observed set of robot r
e_{FPR} ▷ Maximum false positive rate
$Bloom_{q \in \mathcal{R} \setminus r}$ ▷ Received Bloom filters
s ▷ Salt to decorrelate Bloom filters
output: $c_{frontier}$ ▷ Next frontier cell to visit

if *Communication phase* **then**
  $Bloom_r \leftarrow \text{CreateBloomFilter}\left(|\mathcal{C}_r|, e_{FPR}, s\right)$
  $\text{CommunicateBloom}\left(Bloom_r, s\right)$
if *Frontier finding* **then**
  $\tilde{\mathcal{C}}' \leftarrow \mathcal{M} \setminus \mathcal{C}_r$ ▷ Initialise unobserved set
  // Query unobserved cells against all received Bloom filters
  for $c \in \tilde{\mathcal{C}}'$ **do**
    for $q \in \mathcal{R} \setminus r$ **do**
      if $Bloom_q.\text{Query}\left(c, s\right)$ **then**
        $d_q \leftarrow d_q + 1$
        $P_{obs_q} \leftarrow 1 - (e_{FPR})^{d_q}$ ▷ (1)
      else
        $d_q \leftarrow 0$
        $P_{obs_q} \leftarrow 0$ ▷ (1)
    // Compute observation probability
    $P_{obs} \leftarrow 1 - \prod_{q \in \mathcal{R} \setminus r}(1 - P_{obs_q})$ ▷ (2)
    if $P_{obs} \geq \rho_{obs}$ **then**
      $\tilde{\mathcal{C}}' \leftarrow \tilde{\mathcal{C}}' \setminus c$ ▷ Cell considered observed
  $c_{frontier} \leftarrow \text{GetClosestFrontier}\left(\tilde{\mathcal{C}}'\right)$
  return $c_{frontier}$

A. Efficient Data Sharing via Bloom Filters

The Bloom filter allows the sender to encode their full observed set into a highly compressed representation. As described in Section II-C, the level of compression is controlled via the choice of a non-zero maximum false positive rate (FPR). Together with the set size n, the desired maximum FPR defines the number of hash functions k and bit array size m needed to guarantee that the total number of false positives remains below the specified rate. Thus, to maintain the desired FPR, the space requirement m is linear in n. A higher FPR will result in a smaller Bloom filter, however, this comes at the cost of a larger number of unobserved cells being incorrectly identified as belonging to the observed set associate with that Bloom filter. Thus, the FPR can be used as a tuning parameter to balance communication constraints against the mission's accuracy requirements.

In addition, the FPR of a Bloom filter is guaranteed up to the associated number n of set members. For our exploration task, the number of observed cells is initially small and grows over time. Therefore, we can achieve maximal communication efficiency by constructing a Bloom filter for the exact set size $|\mathcal{C}_r|$ and desired FPR at each communication instance. Construction of the Bloom filter is a linear time operation, $\mathcal{O}(kn')$, where n' is the number of members to be encoded in the filter. However, this also means that larger Bloom filters will take longer to create. Nevertheless, unlike node-based data structures, where the query time depends on the degree or branching factor of the structure, the time to query a Bloom filter is always constant $O(k)$. Depending on the compute vs. the communication requirements of the mission, one may decide whether initialising a larger Bloom filter that is incrementally updated is more suitable.

B. Mitigating False Positives

Given that applying membership queries to a Bloom filter will result in false positives, and that the rate at which those false positives occur is limited by the specified maximum FPR, we can design additional safeguards to manage the approximate consensus set $\tilde{\mathcal{C}}'$ of unobserved cells. In our multi-robot exploration task, $\tilde{\mathcal{C}}'$ is used to determine the next waypoint to send the robot. Since the exploration logic is not the main focus of this work, we simply compute this as the nearest frontier cell to the robot. Thus, false positives from our set membership queries will impact the accuracy of the computed frontiers. Note that since the Bloom filter never returns false negatives, the approximate consensus unobserved set will always be a subset of the true unobserved set; i.e., $\tilde{\mathcal{C}}' \subseteq \mathcal{C}'$.

1) Observation Probability for a Single Robot: From the perspective of robot r, given a maximum FPR of e_{FPR}, the probability that a cell has been observed by robot $q \neq r$ is given by

$$P_{obs_q} = \begin{cases} 0, & \text{if query returns negative,} \\ 1 - (e_{FPR})^{d_q}, & \text{otherwise,} \end{cases} \quad (1)$$

where d_q is a count of the most recent unbroken sequence of distinct Bloom filters received from robot r that returned positive. If varying FPRs are used to construct the consecutive Bloom filters, then the exponential term is instead computed as a product of those FPRs.

It is important to note that this result only holds true if the false positives are independent across successive Bloom filters. Due to its formulation, once an element is determined to be a member of the set, repeat queries of the same Bloom filter, regardless of any additional set members, will always return positive; hence, false positives are not independent. To account for this, the d_q should only be incremented if positive set membership is established for a new Bloom filter with a new set of hash functions, which we discuss further as follows.

2) Decorrelating Bloom Filters: We mitigate this complication of requiring independent false positives by constructing a new Bloom filter whenever new cells are observed between communication instances. To treat the event of a grid cell query returning *true* in each received Bloom filter as an independent probabilistic event, each Bloom filter must have a distinct probability distribution over its responses; querying a Bloom filter constructed with the same hash functions and containing the same keys does not provide new evidence. Solving this enables *stacking*, testing for set membership across multiple Bloom filters, to achieve a practical false positive rate (FPR) lower than that of any individual filter.

We achieve this decorrelation via a process called *salting*, where each Bloom filter is constructed by prefixing a unique 128-bit random value to each cell ID before insertion. This salt is transmitted alongside the Bloom filter, allowing recipients to query it correctly. The salt length can be tuned according to the expected number of Bloom filters and the number of historical filters retained, denoted n_{Bloom}.

For example, to query a Bloom filter salted with the string `abc`[1], a recipient performs the query using the salted key `abc,4,5` for grid cell $(4, 5)$. Agents retain the last $n_{\text{Bloom}} = 5$ Bloom filters from other agents and apply *stacking* by checking the salted key across all retained filters. Although decorrelation could also be achieved using different hash functions, salting allows reuse of the same hash function between filters and, by extension, robots.

3) Probabilistic Stacked Membership Queries: Given the individual robot observation probabilities P_{obs_q} for each other robot q, as computed through (1) and the salting process described above, we can now perform stacked membership queries. For coordinated exploration in particular, a common query would be to determine whether at least one robot has observed a cell. The probability that at least one robot has observed the cell can then be computed as

$$P_{obs} = 1 - \prod_{q \in \mathcal{R} \setminus r} \left(1 - P_{obs_q}\right). \quad (2)$$

This observation probability P_{obs} could be used in several different ways, depending on the task at hand. For frontier-based exploration, these probabilities could be used, for example, to prioritise the importance of frontiers during frontier selection.

Since the exploration approach is not the focus of this paper, we use a simpler method of thresholding the probability values to obtain a noise-filtered deterministic map, then compute frontiers on the resulting map. Specifically, we set a probability threshold ρ_{obs} to represent whether sufficient evidence has been provided to consider a particular cell as observed, and only those cells whose probability of observation satisfy this threshold are removed from $\tilde{\mathcal{C}}'$. Naturally, there is no need for a robot to evaluate cells that it has observed itself, the probability calculation is only required for cells not directly observed by the ego-robot but that returned a positive membership evaluation when queried against another robot's communicated Bloom filter.

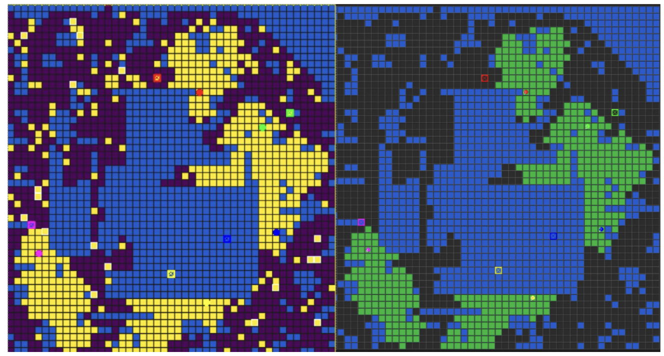

Fig. 2. Communicated vs Ground Truth. Five robots coordinate to explore the Intel office map with $\rho_{obs} = 0.9999$ and $e_{FPR} = 0.01$. Left: reported observations of free space are in yellow. Right: ground truth observations are shown in green. For both, obstacles are in blue while the darker color is unobserved free space. Notice that some false positives are present in the communicated map on the left, but no false negatives are present.

V. Experiments

We evaluate our proposed communication method[2] in a simulated multi-robot exploration task designed to stress both bandwidth usage and message reliability. A team of robots explore an unknown world while exchanging information about observed cells. These experiments evaluate the efficacy of our method in comparison to standard methods that require high bandwidth, are not resilient to message loss, or disregard the noise properties of the Bloom filter.

A. Experimental Setup

1) Exploration Scenario: We consider a team of five robots tasked with collaboratively exploring an unknown grid world, illustrated in Fig. 2. The environment is modelled as

[1] For brevity, `abc` represents a 128-bit value

[2] Source code can be accessed at the UTS Robotics Institute GitHub page: https://github.com/UTS-RI

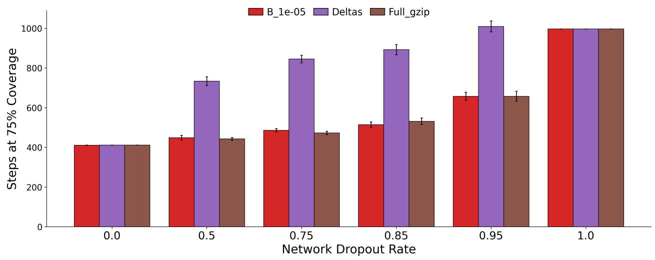

Fig. 3. Steps to reach 75% coverage under varying dropout rates (lower is better). Our method (red) performs comparably to the exact method (brown) across all dropout levels and unlike *Deltas* shows robustness to dropped messages. Bloom filter–based approximations (*B*) achieve similar exploration speed to the exact method while incurring much lower communication cost (see Fig. 6). Error bars indicate standard error across 40 trials.

a grid of discrete cells, each of which is either occupied or free. Each robot maintains a belief map of the world (initially fully unknown) and updates this map based on its local observations. At each time step, a robot observes the occupancy state of all cells within a fixed sensing radius, subject to occlusion by occupied cells. The team's objective is to minimise the time required for every reachable free cell in the map to be observed by at least one robot.

Robots perform exploration by repeatedly selecting the closest frontier cell according to a combined belief map computed locally by each robot. The manner in which this combined map is constructed varies between the comparison methods (discussed below). Robots can periodically share their belief maps with one another; however, each communication message is subject to a probability of loss, such that it may not be received by any other robot. The experiments sweep through a range of drop-out probabilities.

2) Comparison Methods: We compare our proposed Bloom Filter communication protocol (Algorithm 1) against two baseline methods, chosen to isolate different aspects of communication efficiency and robustness:

- *Deltas*: Each robot keeps track of cells it has previously transmitted, $\hat{\mathcal{C}}_r \subseteq \mathcal{C}_r$, and sends only the newly observed cells $\mathcal{C}_r \setminus \hat{\mathcal{C}}_r$ in uncompressed form. This baseline measures the effect of eliminating redundant transmissions.
- *Full Gzip*: Each robot transmits its entire set of observed cells $\mathcal{C}_r$ at every communication step, compressed using Gzip/Deflate [26] with the best compression setting (level 9). This serves as a high-fidelity reference point, providing complete information sharing at the cost of high bandwidth usage.

For our Bloom Filter approach, we additionally sweep key design parameters to study their effect on performance:

- *Bloom filter false-positive rate (FPR):* $0.00001, 0.01, 0.1, 0.3$ — controls the trade-off between filter size and membership accuracy.
- *Observation threshold ρ_{obs}*: $0.0, 0.99, 0.9999$ — sets the probability threshold associated with the minimum number of corroborations (via stacking) required before a cell is considered observed. The $\rho_{obs} = 0$ case effectively disables the stacking and salting, and corresponds to accepting a cell after a single positive membership query.

These variations allow us to examine how filter accuracy and corroboration requirements jointly influence communication cost, robustness to message loss, and overall exploration efficiency.

3) Evaluation Metrics: We focus on two key evaluation metrics, with a sample size of 40 trials for each parameter configuration with randomised start locations:

- *Time to coverage*: The number of simulation steps needed to cover the map. We present the time to 75% coverage of the map.
- *Bytes sent*: The number of bytes communicated over the network. It is assumed that the set elements are 32-bit floating point values representing the (x, y) coordinates of a grid cell, totalling 8 bytes per cell.

B. Results

1) Coordinated Coverage Performance: Fig. 4 provides the number of steps taken to achieve 75% coverage for each of our comparison methods across varying dropout rates and with different ρ thresholds. Our results show that while all methods require more steps to reach the same degree of coverage when dropout rates increase, the performance of *Deltas* deteriorates the fastest. Indeed, at the 75% dropout rate, some instances of *Deltas* already required 1000 steps to reach the coverage threshold, which is the same number of steps as in the 100% dropout rate (all robots effectively acting independently). In comparison, Fig. 3 shows that our proposed Bloom filter methods are robust to dropout, achieving 75% coverage around 650 steps for all dropout rates up to and including 95% dropout, which is on par with the *Full Gzip* baseline.

For our Bloom filter methods, the combination of the max FPR and probability threshold ρ_{obs} can have a large impact on the performance of the team, as seen in the $\rho_{obs} = 0.0$ and $\rho_{obs} = 0.99$ experiments that failed to reach 75% map coverage. With $\rho_{obs} = 0.0$, only a single positive set membership query is needed to satisfy the probability threshold to consider a cell as observed. Thus, when the false positive rate is non-negligible (i.e., $e_{FPR} \geq 0.1$), the number of cells wrongly assumed to already be observed accumulates, and thus the team does not effectively explore the map. Interestingly, under these circumstances, a high ρ_{obs} results in a longer time to accumulate sufficient evidence for a cell, trading communication fidelity for integration latency, which negatively impacts this exploration task, but may be a suitable trade off for others.

Our results also demonstrate that increasing the ρ_{obs} threshold can mitigate the effects of higher FPRs. For both of the other tested ρ_{obs} values, all Bloom filter methods, regardless of max FPR and dropout rate, were able to achieve 75% map coverage with times comparable to the *Full Gzip* method. Fig. 5 shows the effect of a higher ρ_{obs} value comparing the cells marked as observed by each agent with our proposed *stacking* method ($\rho_{obs} = 0.9999$) and without ($\rho_{obs} = 0.0$).

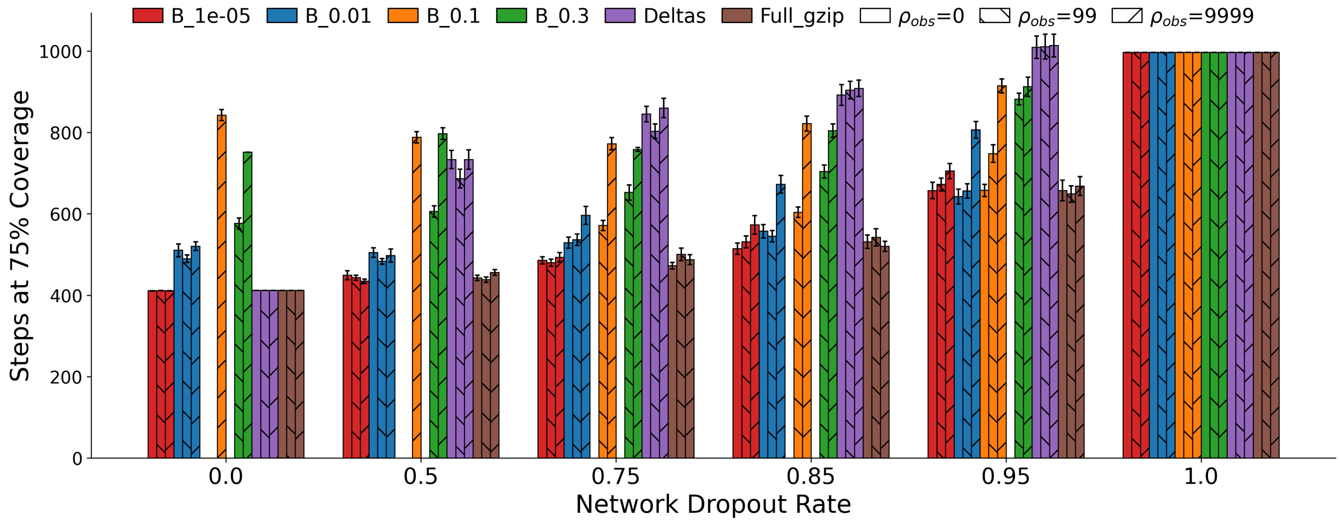

Fig. 4. By varying the observation probability threshold ρ_{obs}, Bloom filter methods (prefixed *B*) using a higher false positive rate are able to perform comparably to the exact compression-based method, shown in brown. Comparison of number of steps taken to achieve 75% coverage when using the different communication protocols across varying dropout rates and with varying ρ_{obs} values. Error bars indicate standard error across 40 trials.

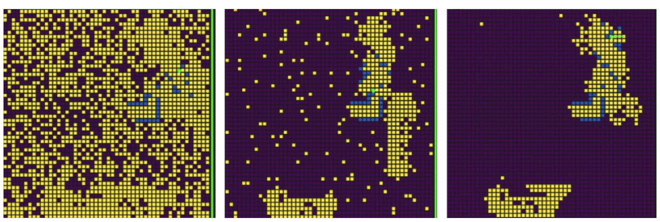

Fig. 5. Effect of *stacking* on belief maps with varying observation acceptance thresholds, shown from left ($\rho_{obs} = 0.0$) to right ($\rho_{obs} = 0.999999$). A ρ_{obs} filters more false positives because more filters must report a positive. Right requires a positive to appear in three consecutive Bloom filters. Bloom filters use a false positive rate of $e_{FPR} = 0.01$.

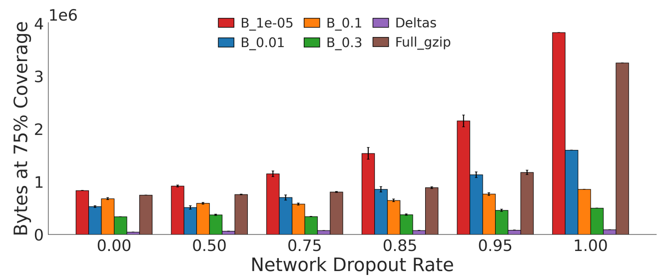

Fig. 6. Bytes sent per method under varying message dropout rates. Approximate methods using Bloom filters (labelled *B* with false positive rate) require fewer bytes than the exact method (brown) but more than the Deltas method, offering a trade-off between cost and robustness to dropped messages. Bloom filter results assume $\rho_{obs} = 0.9999$. Error bars indicate standard error across 40 trials.

2) Communication Cost: Fig. 6 presents the total number of bytes sent over the network to reach 75% coverage. These results were consistent across all ρ_{obs} values and so we only show the numbers for $\rho_{obs} = 0.9999$. As expected, *Deltas* communicates the fewest bytes, and the amount of data sent is not affected by the dropout rate. All other methods show an increase in bytes sent as dropout rates rise, and this is due to the robot team requiring more steps (resulting in more communication rounds) to reach the same degree of map coverage. Nevertheless, for even modest FPRs, the Bloom filter methods result in a similarly low amount of communicated data as *Deltas*. In particular, when compared to the *Full Gzip* method, only a max FPR of 0.00001 results in more bytes being sent. Taken together, the results show how our proposed Bloom filter-based method is robust to communication dropouts and can achieve efficient, coordinated exploration while reducing the amount of data communicated between robots.

VI. Conclusion

This paper introduced a communication method using *approximate set membership queries* via Bloom filters, reducing data transmission compared to exact approaches while maintaining task performance under unreliable networks. By varying the Bloom filter false positive rate and observation threshold, we showed that communication fidelity can approach that of exact methods with far lower bandwidth, making it suitable for constrained networks where occasional false positives are acceptable. Although demonstrated for map sharing in coordinated exploration, the approach generalises to any task involving set-structured data in a metric space—such as occupancy grids, pose graphs, object databases, visited locations, or task lists—suggesting savings for distributed SLAM and task allocation. Future work will extend the method to heterogeneous teams and broader coordination tasks, including collaborative planning and resource allocation, to further assess scalability and generality.

Acknowledgement

GitHub Copilot was used to generate boilerplate design patterns used in the simulation code. We also thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] J. Gielis, A. Shankar, and A. Prorok, "A critical review of communications in multi-robot systems," *Current Robotics Reports*, vol. 3, no. 4, pp. 213–225, Aug. 2022.

[2] S. Li, S. He, Y. Zhang, X. Shi, G. He, and T. Jiang, "Edge intelligence enabled heterogeneous multi-robot networks: Hybrid framework, communication issues, and potential solutions," *IEEE Network*, vol. 36, no. 6, pp. 108–115, Nov. 2022.

[3] R. J. Marcotte, X. Wang, D. Mehta, and E. Olson, "Optimizing multi-robot communication under bandwidth constraints," *Autonomous Robots*, vol. 44, no. 1, pp. 43–55, Jan. 2020.

[4] J. Jessup, S. N. Givigi, and A. Beaulieu, "Robust and efficient multi-robot 3D mapping with octree based occupancy grids," in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 3996–4001.

[5] S. Macenski, D. Tsai, and M. Feinberg, "Spatio-temporal voxel layer: A view on robot perception for the dynamic world," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, p. 1729881420910530, 2020.

[6] U. Dah-Achinanon, S. E. Marjani Bajestani, P.-Y. Lajoie, and G. Beltrame, "Search and rescue with sparsely connected swarms," *Autonomous Robots*, vol. 47, no. 7, pp. 849–863, 2023.

[7] G. Best, R. Garg, J. Keller, G. A. Hollinger, and S. Scherer, "Multi-robot, multi-sensor exploration of multifarious environments with full mission aerial autonomy," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 485–512, 2024.

[8] X. Li, Y. Li, N. Li, A. Girard, and I. Kolmanovsky, "Set-theoretic localization for mobile robots with infrastructure-based sensing," *Advanced Control for Applications*, vol. 5, no. 1, p. e117, 2023.

[9] M. Di Marco, A. Garulli, S. Lacroix, and A. Vicino, "Set membership localization and mapping for autonomous navigation," *International Journal of Robust and Nonlinear Control*, vol. 11, no. 7, pp. 709–734, 2001.

[10] E. Psomiadis, D. Maity, and P. Tsiotras, "Communication-aware map compression for online path-planning," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 12 368–12 374.

[11] L. Carlone, M. Kaouk Ng, J. Du, B. Bona, and M. Indri, "Rao-Blackwellized Particle Filters multi robot SLAM with unknown initial correspondences and limited communication," in *Proc. of IEEE International Conference on Robotics and Automation*, May 2010, pp. 243–249.

[12] D. Mox, K. Garg, A. Ribeiro, and V. Kumar, "Opportunistic communication in robot teams," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 12 090–12 096.

[13] M. Zoula and J. Faigl, "Wireless communication infrastructure building for mobile robot search and inspection missions," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 5970–5976.

[14] M. Lindhé and K. H. Johansson, "Exploiting multipath fading with a mobile robot," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1363–1380, Oct. 2013.

[15] E. Szabo-Wexler, "Approximate membership of sets: A survey," technical report.

[16] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[17] M. T. Goodrich and M. Mitzenmacher, "Invertible Bloom Lookup Tables," Oct. 2015. [Online]. Available: http://arxiv.org/abs/1101.2245

[18] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: a scalable wide-area web cache sharing protocol," *IEEE/ACM transactions on networking*, vol. 8, no. 3, pp. 281–293, 2000.

[19] Squid Software Foundation, "SQUID web proxy cache," 2025, `https://github.com/squid-cache/squid`.

[20] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, pp. 1–26, 2008.

[21] L. F. Mackert and G. M. Lohman, "R* optimizer validation and performance evaluation for local queries," in *Proc. of ACM SIGMOD International Conference on Management of Data*, 1986, pp. 84–95.

[22] J. K. Mullin, "Optimal semijoins for distributed database systems," *IEEE Transactions on Software Engineering*, vol. 16, no. 5, pp. 558–560, 1990.

[23] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost, "Informed content delivery across adaptive overlay networks," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 47–60, 2002.

[24] J. Ryde and J. J. Corso, "Fast voxel maps with counting bloom filters," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4413–4418.

[25] Z. Wang, K. Liu, and C. Xu, "A bloom filter-based algorithm for fast detection of common variables," in *Proc. of International Conference on Frontiers of Robotics and Software Engineering (FRSE)*, 2023, pp. 50–56.

[26] L. P. Deutsch, "GZIP file format specification version 4.3," Internet Engineering Task Force, Tech. Rep. RFC 1952, May 1996. [Online]. Available: https://datatracker.ietf.org/doc/rfc1952