Wireless Communications and Mobile Computing

Advances in Infrastructure Mobility for Future Networks

Lead Guest Editor: Lu Wang
Guest Editors: Ziming Zhao and Wei Wang

Advances in Infrastructure Mobility for Future Networks

Wireless Communications and Mobile Computing

Advances in Infrastructure Mobility for Future Networks

Lead Guest Editor: Lu Wang
Guest Editors: Ziming Zhao and Wei Wang

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 3962128, 1 page
https://doi.org/10.1155/2018/3962128

Editorial

Advances in Infrastructure Mobility for Future Networks

Lu Wang[1], **Ziming Zhao**[2] **and Wei Wang**[3]

[1]*Shenzhen University, Shenzhen, China*
[2]*Arizona State University, Tempe, USA*
[3]*Huazhong University of Science and Technology, Wuhan, China*

Correspondence should be addressed to Lu Wang; wanglu@szu.edu.cn

Received 30 August 2018; Accepted 30 August 2018; Published 17 September 2018

During the last decade, the pervasive use of mobile devices has triggered various changes in network usage and infrastructures. Remarkable advances of autonomous systems have made ubiquitous connectivity no longer a futuristic promise, but rather an attainable technology to meet the imminent communication demands in emergency situations, harsh environments, and remote areas. The growth of Autonomous Vehicles (AVs) and Unmanned Aerial Vehicles (UAVs) empowers the wireless infrastructure with the ability to move physically and thus provides on-demand network services and multiple grid connections, alleviating unpredictable problems such as sudden traffic hotspots, poor coverage, and natural disasters. Researchers are making efforts to benefit from infrastructure mobility. The next breakthrough in network performance will emerge from new ways of organizing networks by merging wireless networking and robotics technologies.

Realizing such a vision needs various pieces to come together, spanning from network architecture to protocol design to communication techniques. This special issue talks about related topics in wireless networks and paves the way for binging mobility as a new degree of freedom (DoF) to the network design. Particularly, it focuses on advances in infrastructure mobility to cover the most recent ideas and research based on wireless networks, including intelligent and effective coordination in the dynamic environments, mobility enabled accurate indoor localization algorithms, and security and privacy issues regarding the mobile data. These advanced technologies in communication and networking ensure the communication robustness, information reliability, and user-privacy protection against a wide range of cyber-attacks.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Lu Wang
Ziming Zhao
Wei Wang

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 4217432, 17 pages
https://doi.org/10.1155/2018/4217432

Research Article

Diverse Mobile System for Location-Based Mobile Data

Qing Liao,[1] Haoyu Tan,[2] Wuman Luo,[2] and Ye Ding[2]

[1]*Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China*
[2]*Guangzhou HKUST Fok Ying Tung Research Institute, Hong Kong University of Science and Technology, Hong Kong*

Correspondence should be addressed to Ye Ding; dingye@ust.hk

Received 23 March 2018; Accepted 10 July 2018; Published 1 August 2018

Academic Editor: Ziming Zhao

The value of large amount of location-based mobile data has received wide attention in many research fields including human behavior analysis, urban transportation planning, and various location-based services. Nowadays, both scientific and industrial communities are encouraged to collect as much location-based mobile data as possible, which brings two challenges: (1) how to efficiently process the queries of big location-based mobile data and (2) how to reduce the cost of storage services, because it is too expensive to store several exact data replicas for fault-tolerance. So far, several dedicated storage systems have been proposed to address these issues. However, they do not work well when the ranges of queries vary widely. In this work, we design a storage system based on diverse replica scheme which not only can improve the query processing efficiency but also can reduce the cost of storage space. To the best of our knowledge, this is the first work to investigate the data storage and processing in the context of big location-based mobile data. Specifically, we conduct in-depth theoretical and empirical analysis of the trade-offs between different spatial-temporal partitioning and data encoding schemes. Moreover, we propose an effective approach to select an appropriate set of diverse replicas, which is optimized for the expected query loads while conforming to the given storage space budget. The experiment results show that using diverse replicas can significantly improve the overall query performance and the proposed algorithms for the replica selection problem are both effective and efficient.

1. Introduction

With the development of data collection capabilities, it is much easier to collect a huge number of location-based mobile data of users or objects via billions of electronic devices such as mobile phones, tablet computers, vehicle GPS navigators, and a wide variety of sensors. For example, taxi companies monitor the mobility information of taxis; telecom operators continuously record the locations of active mobile phones; location-based service (LBS) providers keep the mobile information of the users whenever they use the services. Such large amount of location-based mobile data is valuable for many research fields such as human behavior analysis [1], urban transportation planning [2], customized routing recommendation [3, 4], and location-based advertising and marketing [5].

We called the datasets as *location-based mobile data* because they share the following three common characteristics. Firstly, all these datasets have at least three core attributes: object ID, timestamp, and location. They may as well contain other attributes which are called common attributes that can vary among datasets. Secondly, most queries on these datasets are associated with spatial and temporal ranges. Hence, efficient indexing schemes for range data filtering are required to improve overall query performance. Thirdly, mainstream big data storage and management systems (e.g., HDFS, parallel RDBMSs, and NoSQL databases) are not suitable for storing and processing these data. This is because these systems do not naturally lend themselves to dealing with spatial-temporal range queries, especially when the number of the result records is very large. The main reason is that they cannot physically co-cluster records according to spatial and temporal proximity, which leads to too many slow random disk accesses.

In recent years, several dedicated storage systems have been proposed to store big location-based mobile data, such as TrajStore [6], CloST [7], and Panda [8]. Data are partitioned in terms of spatial and temporal attributes in

the above system. The records in the same partition are physically stored together. To process a range query, we only need to sequentially scan the partitions whose range intersects with the query range. It is demonstrated that this approach is much more efficient than fetching a large number of nonconsecutive disk pages. In addition, these systems can achieve high data compression ratio by leveraging specialized storage structures and encoding schemes.

However, the above existing dedicated systems do not work well when the ranges of queries vary widely. The fundamental reason is that there is only one set of configuration parameters to organize (i.e., partition and compress) the data. It is obvious that we cannot find a single configuration that is optimized for all possible queries. For example, consider that data are partitioned into many small partitions (whose size, in the extreme case, can be as small as a disk page). On one hand, queries with small ranges can be processed efficiently because we can prune most of the partitions. On the other hand, queries with large ranges will incur high I/O costs because a large number of partitions will be involved and locating each of them will invoke a random page access. In this context, these systems have to choose the parameters optimized for the overall performance of the expected query workloads. Note that the expected query workloads can be either derived from historical queries [7] or known as a priori knowledge [6].

In this paper, we explore the use of *diverse replicas* in the context of storage systems for big location-based mobile data. In big data storage systems, e.g., Hadoop HDFS, replication is mainly used for data availability and durability, but not yet for optimizing the performance of query processing. Hence, the use of diverse replicas is a novel approach. The implications of diverse replicas are twofold. First, data are partitioned and compressed in multiple ways such that different queries can pick the best-fit configuration to minimize the processing time. Second, in spite of the diversity of physical data organizations, diverse replicas can recover each other when failures occur because they share the same logical view of the data. Since we can replace the exact replicas with diverse ones, the gain of query performance does not necessarily come at the cost of more storage space. Though the potential advantages of using diverse replicas are prominent, it is nontrivial to determine which replicas to use. Concretely, given a large location-based mobile dataset, a representative workload, and a constraint on storage space, we need to find an optimal or near-optimal set of diverse replicas in terms of overall query performance. To address this problem, we make the following contributions:

(i) We propose BLOT, a system abstraction that describes an important class of location-based mobile data storage systems. Based on the BLOT system abstraction, we conduct general discussions on how to integrate diverse replicas into existing systems.

(ii) We formally define the *replica selection problem* that finds the optimal set of diverse replicas in the context of BLOT systems. Besides, we prove that this problem is at least NP-complete.

(iii) We propose two solutions to the replica selection problem, including an exact algorithm based on integer programming and an approximation algorithm based on greedy strategy. In addition, we propose several practical approaches to reduce the input size of the problem.

(iv) We design a simple yet effective cost model to estimate the cost of an arbitrary query on an arbitrary replica configuration. The parameters of the cost model can be either calculated by closed-form formula or measured accurately by a few low-cost experiments.

(v) We evaluate our solutions using two typical deploy environments of BLOT systems. The experiment results confirm that using diverse replicas can significantly improve the overall query performance. The results also demonstrate that the proposed algorithms for the replica selection problem are both effective and efficient.

The rest of this paper is organized as follows. Section 2 briefly summarizes the related works and Section 3 presents the common designs of BLOT systems as well as the general use of diverse replicas. Section 4 defines the replica selection problem, proves its hardness, and describes the solutions. Section 5 presents the query cost estimation model for BLOT systems. Section 6 shows the experiment results and conducts analysis and Section 7 concludes the paper.

2. Related Work

There is a plethora of works on storing spatial-temporal data and efficient processing of range queries. Early studies, dating back to 1970s and 1980s, mainly focus on indexing individual points or trajectories. Representative works include k-d tree [9], quadtree [10], R-tree [11], and TB-tree [12]. These data structures incur many random reads which are inefficient when the number of records in the query result is large. To address this issue, TrajStore [6] and PIST [13] attempt to co-locate data according to spatial and temporal proximities and use relatively large partition size. Both TrajStore and PIST cannot scale to terabytes of data because they can only consider nondistributed environments. CloST [7] and SpatialHadoop [14] are two Hadoop-based systems which aim at providing scalable distributed storage and parallel query processing of big location-based mobile data. SATO [15] is a spatial data partitioning framework that can quickly analyze and partition spatial data with an optimal spatial partitioning strategy for scalable query processing. Note that TrajStore, PIST, CloST, SpatialHadoop, and SATO can be viewed as concrete instances of BLOT systems without using diverse replicas.

Recommending a physical configuration for a given workload has been widely studied since 1987 [16]. Most of the existing works [17–23] propose effective methods to estimate the cost of a given workload over candidate physical configurations. However, only a few of them consider the situations where data can be replicated [20, 21]. An earlier work introduces the technique of Fractured Mirrors [24]

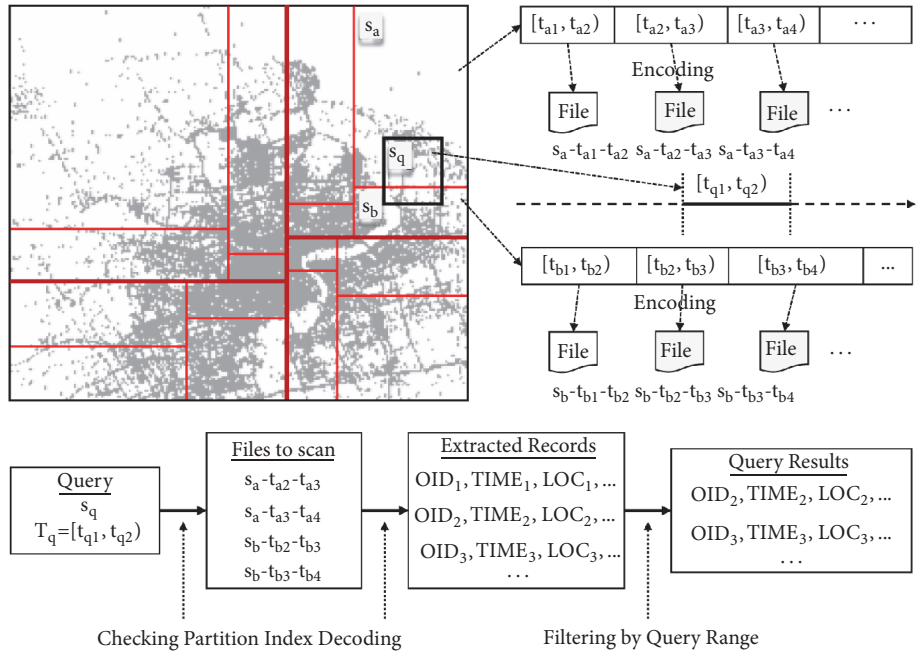


Figure 1: Overview of BLOT systems.

to store data in both row-fashion and column-fashion. For data partitioning, it has been proved in [25] that finding the optimal vertical partitioning is an NP-hard problem. Therefore, the are a number of works that focus on heuristic algorithms for vertical partitioning optimization [26–30]. For workload size reduction, the authors of [31] propose a workload compression method to reduce the size of SQL workloads. A more scalable workload grouping method is proposed in [20]. Most of the above works are based on the relational data model while our work is based on the BLOT data model which is more suitable for big location-based mobile data.

3. BLOT Systems and Diverse Replicas

In this section, we introduce *BLOT*, a system abstraction that reflects common designs of an important class of dedicated systems for storing big location-based mobile data. We refer to such systems as *BLOT systems*. Figure 1 shows an overview of how data are organized and queried in BLOT systems.

BLOT systems are primarily aimed at providing a storage layer that supports efficient data filtering by spatial-temporal ranges for high-level data analytical systems such as RDBMSs and Hadoop. They can be also used as standalone systems to dedicatedly answer range queries. The advantages of BLOT systems have been demonstrated by a number of existing works such as PIST [13], TrajStore [6], CloST [7], and Spatial Hadoop [14]. Compared with other solutions (e.g., using the original Hadoop or NoSQL databases), the speed of range queries in a BLOT system can be up to one to two orders of magnitude faster while using a much smaller storage space (typically 20% or less). In the rest of this section, we will first describe the general design of BLOT systems and then explain why using diverse replicas can significantly improve the overall system performance.

3.1. Data Model. A BLOT system stores a large number of location-based mobile records. Each record is in the form of (`OID`, `TIME`, `LOC`, $A_1, \ldots, A_m$), where `OID` is an object ID, `TIME` is a timestamp, `LOC` is the location of object `OID` at time `TIME`, and A_1 through A_m are other attributes that can vary among different datasets. We refer to the first three attributes as *core attributes* and the others as *common attributes*. Any dataset that naturally fits into this data model, i.e., containing and emphasizing core attributes, can be viewed as location-based mobile data.

3.2. Data Partitioning. Based on the data model, BLOT systems split a large dataset into relatively small partitions using core attributes. In TrajStore and CloST, for example, data are first partitioned by location (`LOC`) and then further partitioned by time (`TIME`). Records in the same partition are stored together in a storage unit which is optimized for sequential read. For instances, a storage unit can be an object stored in Amazon S3, a file on HDFS, a segment of a file on a local file system, etc. Typically, the size of a storage unit in BLOT systems is much larger than that of a disk page, ranging from hundreds of kilobytes to several megabytes. The advantages of using relatively large storage units are twofold. First, queries with large spatial-temporal ranges can be efficiently processed because data are mostly accessed sequentially. Second, it makes the number of storage units sufficiently small such that we can easily maintain the

Table 1: Comparison of involved partitions and estimated scans in Figure 2.

	Case 1	Case 2	Case 3
Involved partitions	4	3	8
Estimated data to scan	100%	30%	50%

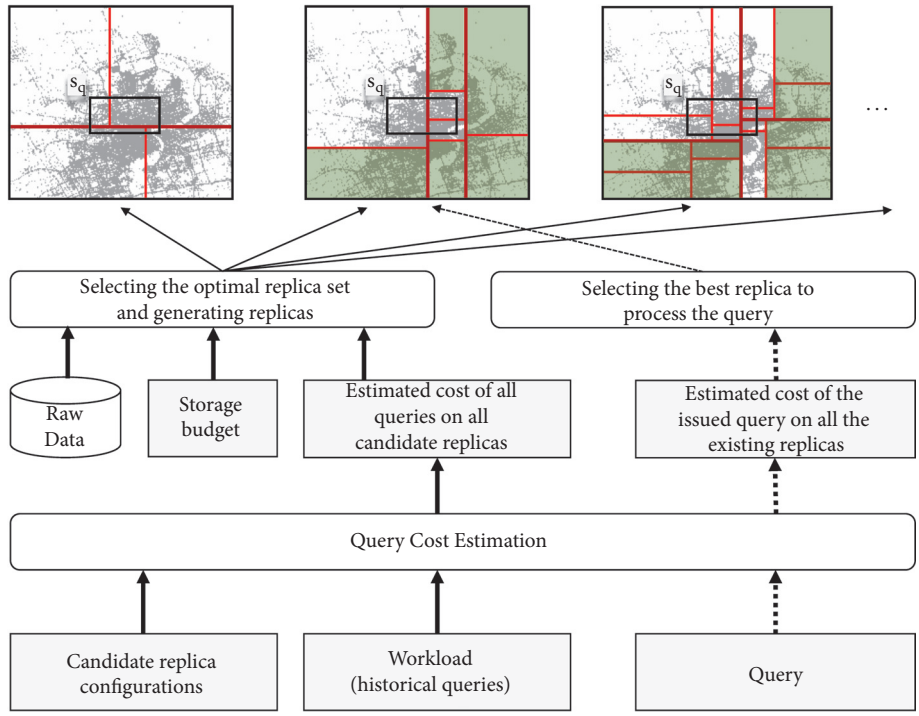


Figure 2: Using diverse replicas in BLOT systems.

partitioning index, a small global data structure to index the spatial-temporal ranges of all data partitions.

3.3. Data Encoding. A data partition can be stored in any format. A popular approach is to store each partition as a CSV file with each line specifying a record. While this format is easy to process, the storage utilization is low. It is therefore undesirable for huge datasets, especially when using cloud storage systems that charge for every bit stored. To reduce the storage size, a BLOT system usually uses various compression techniques to encode records in a partition. For example, we can

(1) use binary format instead of text format;

(2) apply a general compression algorithm to compress the entire partition;

(3) organize the data in column fashion and then apply column-wise encoding schemes (e.g., delta encoding and run-length encoding).

Moreover, we can use the combinations of the above techniques to further reduce the storage size. Note that higher compression ratio comes at the cost of longer decompression time which may degrade the performance of query processing.

3.4. Query Processing. To process range queries in a BLOT system, we first search for *involved partitions*, i.e., the partitions whose range intersect with the query range. Next, we read and decompress each involved partition to extract all the records. Finally, we check the extracted records and output the ones within the query range. Note that it is straightforward to conduct parallel query processing by scanning multiple partitions simultaneously.

In general, the cost of processing an involved partition consists of two parts: scan cost which includes the cost of extracting and filtering the records and extra cost which includes the cost of initializing the procedure, locating the partition, loading the decoder, cleaning up the procedure, etc. In a typical BLOT system, scan cost is usually proportional to the total number of records in the partition while extra cost is usually a constant decided by the corresponding encoding scheme. Therefore, for a specific query, the query cost is determined by the total amount of records to be scanned and the total number of involved partitions. Consider three partitioning schemes and a query shown in the upper part of Figure 2. For illustration purpose, we omit the temporal dimension and highlight the partitions that are not involved. Table 1 compares the number of involved partitions and the estimated percentage of data to scan among the three cases.

In this example, it is obvious that the middle case has the lowest query cost because both the scan cost and the extra cost are the lowest. However, it is unclear whether the query cost of the left case is higher or lower than that of the right case. To answer that question, we will develop an effective cost estimation model in Section 5.

3.5. Diverse Replicas. From Figure 2 we can see that the cost of a query may vary a lot with different partitioning schemes. Undoubtedly, encoding scheme also has a significant influence on query performance. Most existing BLOT systems can adaptively optimize the configuration of the physical storage organization, such as spatial and temporal partition sizes, based on analyzing the historical queries. However, in the cases when the range of queries has high variation, the optimal configuration may still be far from satisfactory in terms of overall query performance. It is intuitive that using multiple copies of data with different physical organizations can mitigate the "one-size-does-not-fit-all" problem. Traditionally, this is a typical performance tuning approach that trades space for time. However, in the context of big data storage systems where data are replicated for fault-tolerance, we can make better use of the storage space by replacing exact replicas with diverse ones. As a result, the overall query performance can be improved without necessarily using more storage space.

Figure 2 illustrates the use of diverse replicas in BLOT systems. There are two components that are key to the success of such systems. First, the system must be able to estimate the query cost both efficiently and effectively. Query cost estimation helps the system to determine which one of the existing replicas is supposed to have the least processing time for the issued query. For example, in Figure 2, the second replica is chosen to answer the given query. Besides, the estimated costs of all queries in the given workload on all candidate replicas are important inputs for the second component which selects a set of diverse replicas (and generates the actual replicas) that is optimized for a given workload under a storage constraint. The storage constraint is a hard constraint indicating the upper bound of the available storage space. It turns out that selecting the optimal set of diverse replicas in BLOT systems is a challenging problem. To the best of our knowledge, it has not been well investigated in the previous works. Therefore, we will elaborate on this problem in the next section.

4. Replica Selection Problem

Given a very large location-based mobile dataset D, we want to choose a set of diverse replicas which conforms to a storage size constraint and optimizes the overall performance for a given workload. In this section, we first formally define the replica selection problem and then propose several practical solutions.

4.1. Problem Definition. Before formalizing the replica selection problem, we first give the formal definitions of several important concepts mentioned in Sections 3 and 4.

Definition 1 (partitioning scheme). Let U denote the spatial-temporal bounding box of D. A spatial-temporal partitioning scheme $P = \{p_1, p_2, \ldots, p_n\}$ is a spatial-temporal partition of U, where

$$\bigcup_i p_i = U, \tag{1}$$

and

$$p_i \cap p_j = \emptyset, \quad \forall i, j \in \{1, 2, \ldots, n\}, \ i \neq j. \tag{2}$$

Particularly, p_i is called the i-th spatial-temporal partition of U.

Definition 2 (data partition). Given a partitioning scheme P, for any partition $p_i \in P$, the corresponding data partition d_i is the set of all records in D that are spatial-temporally contained by p_i. In addition, we define

(1) $D(p_i) = d_i$;
(2) $P(d_i) = p_i$;
(3) $D(P) = \{d_i \mid P(d_i) \in P\}$.

By Definition 1, we have

$$\bigcup_i d_i = D, \tag{3}$$

and

$$d_i \cap d_j = \emptyset, \quad \forall i, j \in \{1, 2, \ldots, n\}, \ i \neq j. \tag{4}$$

Since it is usually clear from the context, we often use the term *partition* to indicate both spatial-temporal partition (i.e., p_i) and data partition (i.e., d_i). In addition, we use $\mu(p)$ and $\mu(d)$ to denote the spatial-temporal range of a spatial partition p and that of a data partition d, respectively.

Definition 3 (encoding scheme). Given a data partition d, an encoding scheme E is an algorithm that generates a physical storage layout for d.

Definition 4 (replica and replica set). A replica $r = \langle D, P, E \rangle$ is a physical organization of all records in D in which records are partitioned by P and each partition is encoded by E. A replica set $R = \{r_1, r_2, \ldots, r_m\}$ is a set of diverse (i.e., unique) replicas.

We use $P(r)$ and $E(r)$ to indicate the partitioning scheme and the encoding scheme of r, respectively. Note that the above definition requires that all partitions are encoded by the same encoding scheme. Nevertheless, the essential theoretical analysis in the following can be easily generalized for BLOT systems that allow a separate encoding scheme for each partition.

Definition 5 (storage size). The storage size of a replica r, denoted by $\eta(r)$, is the size of storage space required to store all encoded partitions in r. The storage size of a replica set R, denoted by $\eta(R)$, is the total storage size of all replicas in R, i.e.,

$$\eta(R) = \sum_{r_j \in R} \eta(r_j). \tag{5}$$

Definition 6 (query and workload). A (range) query q is a process that extracts all records in D that are contained by a cuboid whose size is specified by $\langle \delta_x, \delta_y, \delta_t \rangle$ and centroid is specified by $\langle x, y, t \rangle$. A workload $W = \{(q_1, w_1), (q_2, w_2), \ldots, (q_n, w_n)\}$ is a set of unique queries with each query associated with a non-negative weight.

Similar to $\mu(p)$ and $\mu(d)$, we use $\mu(q)$ to denote the spatial-temporal range of q, i.e., $\langle x, y, t, \delta_x, \delta_y, \delta_t \rangle$. The weight of a query in a workload can be interpreted as the importance (frequency, priority, etc.) of the query. In some situations, the weights are normalized such that

$$\sum_{i=1}^{n} w_i = 1. \tag{6}$$

In particular, we use $Q(W)$ to denote the set of all queries in W, i.e., $Q(W) = \{q_1, q_2, \ldots, q_n\}$.

Below we define query cost and workload cost based on the query processing mechanism described in Section 3.4.

Definition 7 (query cost and workload cost). Given a replica $r \in R$ and a query $q \in Q(W)$, the query cost of q on r is denoted as $\rho(q, r)$. Therefore,

$$\rho(q, R) = \min_{r_j \in R} \rho(q, r_j), \tag{7}$$

and

$$\rho(W, R) = \sum_{(q_i, w_i) \in W} w_i \cdot \rho(q_i, R). \tag{8}$$

Now we can formally define the problem of finding an optimal set of diverse replicas.

Definition 8 (replica selection problem). Given a dataset D, a workload $W = \{(q_1, w_1), (q_2, w_2), \ldots, (q_n, w_n)\}$, a set of candidate replicas $R = \{r_1, r_2, \ldots, r_m\}$, and a storage budget b, find a replica set R^* such that

(1) $R^* \subseteq R$;

(2) $\eta(R^*) \leq b$;

(3) $\rho(W, R^*) \leq \rho(W, R')$ for all $R' \subseteq R$ such that $\eta(R') \leq b$.

In most situations, R contains all possible replicas, i.e., if we have m_P partitioning schemes and m_E encoding schemes, then $m = m_P * m_E$.

To find the optimal replica set R^*, we need to know the query cost $\rho(q_i, r_j)$ and the storage size $\eta(r_j)$ for all $q_i \in Q(W)$ and $r_j \in R$ in the first place. For $\eta(r_j)$, we can estimate it using the compression ratio of the corresponding encoding scheme $E(r_j)$. Since compression ratio is stable in most situations, it can be effectively measured with a small sample of D. For $\rho(q_i, r_j)$ and we will propose a highly accurate cost model in Section 5 to estimate query cost without generating actual replicas.

For the rest of this section, we assume that all $\rho(q_i, r_j)$ and $\eta(r_j)$ are already given and focus on designing practical algorithms to solve the problem.

4.2. Exact Solution. Before presenting the exact solution, we first prove the following theorem.

Theorem 9 (NP-hard). *The replica selection problem is NP-Hard.*

Proof. We prove the theorem by reducing from the minimum weight set cover problem [32] to the replica selection problem. Specifically, given a set of n elements $A = \{a_1, a_2, \ldots, a_n\}$, and a set of m sets $S = \{s_1, s_2, \ldots, s_m\}$, where

$$s_i \subseteq A, \quad \forall s_i \in S, \tag{9}$$

and

$$\bigcup_{s_i \in S} s_i = A, \tag{10}$$

the minimum weight set cover problem is to find a set $S^* \subseteq S$ such that

$$\left|S^*\right| \leq |S|, \tag{11}$$

and

$$\bigcup_{s_i \in S^*} s_i = A, \tag{12}$$

and the cost of S^* is minimum where the cost of S^* is defined as

$$\sum_{i=1}^{|S^*|} c_i. \tag{13}$$

where c_i is the cost (weight) of set s_i.

The minimum weight set cover problem is a well-known NP-hard problem [32]. In this proof we will demonstrate that we can solve any instance of the minimum weight set cover problem by constructing and solving an instance of the replica selection problem.

In correspondence to A, we construct a workload $W = \{(q_1, 1), (q_2, 1), \ldots, (q_n, 1)\}$, where all weights are set to 1. In correspondence to S, we construct a set of candidate replicas $R = \{r_1, r_2, \ldots, r_m\}$ where all $\eta(r_j) \in R$ are set to 0. The query cost is set as follows:

(1) $\rho(q_i, r_j) = 0$ if $\rho(q_i, r_j) = \rho(q_i, R)$;

(2) $\rho(q_i, r_j) = +\infty$ if $\rho(q_i, r_j) \neq \rho(q_i, R)$.

According to Definition 7, we can interpret $\rho(q_i, r_j) = 0$ as that answering q_i on r_j requires the minimum query cost, and $\rho(q_i, r_j) = +\infty$ as that answering q_i on r_j requires more query cost than the minimum.

For the ease of presentation, we use problem α and problem β to denote the instance of the minimum weight set cover problem and the corresponding instance of the replica selection problem, respectively.

Suppose that we have found an optimal replica set R^* in problem β. We can then construct the corresponding set $S^* = \{s_j \mid \text{for all } j \text{ such that } r_j \in R^*\}$ in problem α. To decide whether problem α is feasible, we need to discuss

two cases. On one hand, if $\rho(W, R^*) = 0$ in problem β, then any query in $Q(W)$ can be answered instantly by some replica in R^*. According to our construction process from problem α to problem β, it follows that any element in A must be covered by some set in S^*. In this case, we can safely conclude that problem α is feasible. On the other hand, if $\rho(W, R^*) = +\infty$ in problem β, we prove that problem α is infeasible by contradiction. Assume S^{**} is a feasible solution to problem α. We can then construct a replica set $R^{**} = \{r_j \mid \text{for all } j \text{ such that } s_j \in S^{**}\}$ for problem β. We can easily verify that $\rho(W, R^{**}) = 0$, which follows that $\rho(W, R^{**}) < \rho(W, R^*)$. This contradicts with the fact that R^* is an optimal replica set in problem β.

Thus, we have proved that problem α is feasible if and only if the optimal workload cost in the corresponding problem β equals 0. We therefore conclude that the replica selection problem is equally hard to the set covering decision problem. This completes the proof. □

Though Theorem 9 eliminates the possibility of finding the optimal replica set in polynomial time, an exact solution is still useful when the input size is relatively small. Our exact solution is to model the original problem as a 0-1 Mixed Integer Programming (MIP) problem [33] and hand it over to a MIP solver. The challenge here is how to model the problem properly to ensure that the optimal solution of the 0-1 MIP problem is the optimal solution of the original problem.

Let $n = |W|$ and $m = |R|$. For any $i \in \{1, 2, \ldots, n\}$ and any $j \in \{1, 2, \ldots, m\}$, let x_j be a 0-1 variable indicating whether replica r_j is present in the replica set R and y_{ij} be a 0-1 variable indicating whether query q_i is processed on replica r_j. We first list the constraints as follows.

The constraint related to storage size is

$$\eta(R) = \sum_{j=1}^{m} \eta\left(r_j\right) \le b. \tag{14}$$

We use exactly one replica to process each query:

$$\sum_{j=1}^{m} y_{ij} = 1, \quad \forall i \in \{1, 2, \ldots, n\}. \tag{15}$$

Any replica that is chosen to process at least one query must be present in R:

$$y_{ij} \le x_j, \quad \forall i \in \{1, 2, \ldots, n\}, \ j \in \{1, 2, \ldots, m\}. \tag{16}$$

We can see that (16) specifies $n \times m$ constraints. Because an MIP problem may become extremely difficult in the presence of too many constraints, it is preferable to use fewer constraints. Therefore, we use the following m constraints instead (which are slightly relaxed but do not change the optimal solution):

$$\sum_{i=1}^{n} y_{ij} \le n \cdot x_j, \quad \forall j \in \{1, 2, \ldots, m\}. \tag{17}$$

Let $c_{ij} = \rho(q_i, r_j)$; we use the following objective function:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} w_i \cdot c_{ij} \cdot y_{ij}. \tag{18}$$

Putting them together, we need to minimize (18) subject to the constraints specified by (14), (15), and (17). The details are shown in the following:

$$\text{minimize:} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} w_i \cdot c_{ij} \cdot y_{ij}, \tag{19}$$

$$\text{subject to:} \quad \sum_{j=1}^{m} \eta\left(r_j\right) \le b,$$

$$\sum_{j=1}^{m} y_{ij} = 1, \quad \forall i \in \{1, 2, \ldots, n\}, \tag{20}$$

$$\sum_{i=1}^{n} y_{ij} \le n \cdot x_j, \quad \forall j \in \{1, 2, \ldots, m\}.$$

As x_j and y_{ij} are 0-1 variables, this is a well-formed 0-1 MIP problem that can be solved directly by MIP solvers.

4.3. Reducing the Problem Size. In general, the computation time of solving an MIP problem grows exponentially with the problem size, i.e., the number of decision variables. The total number of decision variables in our formulation is $m(n + 1)$ (all x_{ij} and y_j) which could be very large even though both m and n are relatively small. For example, there are more than 10^5 decision variables when we have 20 partitioning schemes, 5 encoding schemes, and 1000 queries in the given workload. Though this is a typical scenario in practice, it already makes the formulated MIP problem computationally infeasible (on up-to-date computers nowadays). Thus, to make the aforementioned solution more scalable, we propose several practical techniques that can significantly reduce the problem size.

4.3.1. Reducing the Workload Size. If we directly use all historical queries recorded in the query log to form the input workload, then m may increase too fast in a working system where new queries are issued frequently. To address this issue, we treat each $q \in Q(W)$ as a group of similar queries. Specifically, we use only one *grouped query*, denoted by Q^G, to represent all the queries with the same size of spatial-temporal range. Accordingly, we adjust the definition of query in Definition 6 by replacing $\mu(q) = \langle x, y, t, \delta_x, \delta_y, \delta_t \rangle$ with $\mu(Q^G) = \langle \delta_x, \delta_y, \delta_t \rangle$. This variation reflects the observation that queries with the same size of range often occurs many times in real situations. For example, it is common that users use an equal-sized grid to decompose the space and then conduct simple statistics for each grid cell. It is worth pointing out that estimating the cost of a grouped query is generally more difficult than estimating a single query. We will address this issue in Section 5. In addition, if the number of different range sizes is still large, we can use clustering algorithms such as K-means to cluster the range sizes and only use the cluster centers to construct the input workload. In this way, we have full control of the value of m by manipulating the number of clusters.

Input: W, R, b, $\rho(q_i, r_j)$ for all $q_i \in Q(W)$ and $r_j \in R$
Output: R^*
(1) **begin**
(2) $R^* \longleftarrow \emptyset$;
(3) **while** $\eta(R^*) < b$ **do**
(4) $r^* \longleftarrow$ null;
(5) $score^* \longleftarrow 0$;
(6) **for** $r \in R$ **do**
(7) $score \longleftarrow \dfrac{\rho(W, R^*) - \rho(W, R^* \cup \{r\})}{\eta(r)}$;
(8) **if** $score > score^*$ **then**
(9) $score^* \longleftarrow score$;
(10) $r^* \longleftarrow r$;
(11) **if** r^* *is null* **then**
(12) **break;**
(13) **else**
(14) $R^* \longleftarrow R^* \cup \{r^*\}$;
(15) $R \longleftarrow R \setminus \{r^*\}$;
(16) **return** R^*.

Algorithm 1: A greedy replica selection algorithm.

4.3.2. Reducing the Number of Candidate Replicas. Considering two replicas $r_1, r_2 \in R$ satisfying

$$\eta(r_1) \leq \eta(r_2), \tag{21}$$

and

$$\rho(q_i, r_1) \leq \rho(q_i, r_2), \quad \forall q_i \in Q(W), \tag{22}$$

we refer to this case as replica r_1 *dominates* replica r_2. Obviously, if we use $R\backslash\{r_2\}$ instead of R as the input candidate replicas, it will not change the optimal workload cost $\rho(W, R^*)$. Therefore, we can safely prune r_2 from R. In general, it is more common that a replica is dominated by a set of replicas. Concretely, given a replica $r \in R$ and a replica set $R^D \subseteq R$, we say that replica set R^D *dominates* replica r if

(1) $r \notin R^D$;

(2) $\eta(R) \leq \eta(r)$;

(3) $\rho(q_i, R^D) \leq \rho(q_i, r), \forall q_i \in Q(W)$.

Ideally, we want to find a minimum *dominant replica set* $R^D \subseteq R$ such that R^D dominates any replica $r \in R\backslash R^D$. However, as we can prove that the replica selection problem itself is NP-hard, we do not pursue a minimum R^D in practice. Instead, we use a rough yet effective heuristic algorithm to find a suboptimal dominant replica set.

4.4. Approximation Solution. In this section, we propose several approximate algorithms to select a near-optimal set of replicas based on the reduced problem size as illustrated in Section 4.3. Approximation algorithm is suitable in case that the number of candidate replicas is still large after pruning or the workload is changing rapidly so that the replica set should be reselected frequently.

4.4.1. Greedy Strategy. First we give a fast greedy algorithm to solve the replica selection problem. This algorithm is adopted and extended from the minimum weighted set cover algorithm. As shown in Algorithm 1, we add one replica at a time to the replica set R^* such that in each step the added replica r maximizes,

$$\frac{\rho(W, R^*) - \rho(W, R^* \cup \{r\})}{\eta(r)}, \tag{23}$$

until the storage budget is exhausted or the overall workload cost $\rho(W, R^*)$ cannot be further decreased by adding any one of the remaining replicas. Before the storage size is full, each time we add one replica into the replica set, in worst case we need iterate $|R|$ times until the storage space is full. In each iteration, we

(1) score all $|R\backslash R^*|$ replica candidates that are not added to R^* yet;

(2) add the replica with highest score into R^*.

The scoring step computes the gain of each replica candidates that may be added to $R*$; thus in this step all $q \in Q(W)$ are compared with the costs on the current replica and the candidate replicas. Hence, this step takes $O(|R||Q(W)|)$ time, and it will result in an $O(\log n)$ approximation ratio, where n is size of the set of all queries $q \in Q(W)$. The running time of this greedy algorithm is $O(nm^2)$, where m is size of the set of candidate replicas. In Section 6, we will see that the approximation ratio of the greedy algorithm is quite desirable (lower than 1.3 in most cases) in practice.

4.4.2. LP Rounding Strategy. Although the greedy strategy is simple to implement and achieves good approximate result in practice, the best we can hope for the greedy strategy is a logarithmic approximate ratio ($\log n$). When the

Input: $W, R, b, \rho(q_i, r_j)$ for all $q_i \in Q(W)$ and $r_j \in R$, N_i for all $q_i \in Q(W)$
Output: R^*
(1) **begin**
(2) $\Gamma \longleftarrow \emptyset$;
(3) sort $q_i \in Q(W)$ by C_i in ascend order;
(4) **while** $\eta(R^*) < b$ *and* $Q(W) \neq \emptyset$ **do**
(5) choose q_i from $Q(W)$ with smallest C_i;
(6) $\gamma \longleftarrow 0$;
(7) **foreach** $q_{i'} \in \Gamma$ **do**
(8) **if** $N_i \cap N_{i'} \neq \emptyset$ **then**
(9) $\gamma \longleftarrow 1$;
(10) **break;**
(11) **if** $\gamma = 0$ **then**
(12) $\Gamma \longleftarrow \Gamma \cup \{q_i\}$;
(13) $Q(W) \longleftarrow Q(W) \setminus \{q_i\}$;
(14) $R^* \longleftarrow \emptyset$;
(15) **foreach** $q_{i'} \in \Gamma$ **do**
(16) $R^* \longleftarrow r^*$ with $\rho(q_{i'}, r^*) = \rho(q_{i'}, N_{i'})$;
(17) **return** R^*.

Algorithm 2: A LP rounding based algorithm.

quantity of queries goes large, the performance guarantee will drop accordingly. In this section we introduce a constant-factor approximate algorithm based on linear programming rounding [34]. The linear programming rounding strategy consists of three stages:

(1) Formulating the problem to integer linear programming

(2) Relaxing the integral constraints and finding the optimal solution for the relaxed linear programming

(3) Rounding the fractional solution of the linear programming and producing an integral solution.

In the replica selection problem, the LP rounding strategy is based on the MIP proposed in Section 4.2; thus we already finished stage 1. In stage 2, we further relax the MIP by allowing $x_j \leq 1$ and $y_{ij} \geq 0$. Then we can solve the LP in polynomial time resulting fractional x_j and y_{ij}. In stage 3, since general rounding techniques cannot be directly adopted on the replica selection problem, we present the following rounding strategy.

Suppose we have found an optimal solution for the LP in stage 2. For any query $q_i \in Q(W)$, we define the *neighborhoods* of q_i as

$$N_i = \left\{ r_j \in R^* \mid y_{ij} > 0 \right\}. \tag{24}$$

All the replicas r_j that serve q_i fractionally are the neighborhoods of q_i. Further we define *cluster* as a set of queries and replicas with the center $q_i \in Q(W)$. In the LP, we denote

$$C_i = \sum_{r_j \in R^*} c_{ij} \cdot y_{ij}, \tag{25}$$

and thus the total query cost is

$$\rho\left(W, R^*\right) = \sum_{(q_i, w_i) \in W} w_i \cdot C_i. \tag{26}$$

Now we sort queries q_i by C_i in ascending order and then iteratively assign each query and replica to clusters until all queries and replicas are assigned to one cluster. In each iteration, we pick the query q_i with the smallest C_i. If $N_i \cap N_{i'} = \emptyset$ for any existing cluster center $q_{i'} \in \Gamma$, we open a new cluster i and add q_i into the new cluster and denote q_i as the cluster center. If $N_i \cap N_{i'} \neq \emptyset$, we add q_i to cluster i'. Then we can round the fractional solution: for each cluster, we select the cheapest replica r_i for each cluster center q_i in N_i and assign queries in this cluster to replica r_i. The overall constant-factor approximation algorithm is shown in Algorithm 2. Theorem 10 provides the approximate ratio of the LP rounding based strategy.

Theorem 10. *The proposed LP rounding strategy is a 3-factor approximation algorithm.*

Proof. Suppose the optimal solution of the MIP is Θ_0 and the optimal solution of the relaxed LP is Θ_1, since Θ_0 is a feasible solution of Θ_1, we can prove $\Theta_0 \leq \Theta_1$ [35]. In the rounding solution, we select replicas that have the cost at most $4\Theta_1$.

Assuming that $q_{i'}$ is the center of cluster k, we have selected replica r_{j_k} for any query q_i in cluster k. For q_i, there are three types of query cost $\rho(q_i, N_i)$ on replica r_{j_k}:

(1) q_i is in cluster k and $c_{ij_k} \leq c_{i'j_k}$.

(2) q_i is in cluster k but $c_{i'j_k} \leq c_{ij_k}$. Since $N_i \cap N_{i'} \neq \emptyset$, queries q_i and $q_{i'}$ share some replica in common. By triangle inequality we have $c_{i'j} \leq c_{ij_k} + c_{ij} + c_{i'j_k} \leq C_i + 2C_{i'} \leq 3C_{i'}$. The last inequality is because we sort C_i in ascending order and pick the query with smallest C_i each time.

(3) q_i is not in cluster k; in this case, we set c_{ij_k} to ∞ and q_i will not be queried on any replica in this cluster.

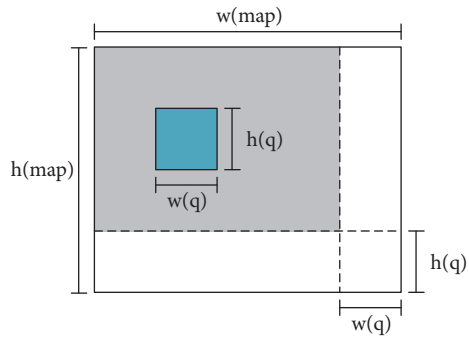


Figure 3: An example of the distribution of queries in this paper.

The total cost of the rounding solution is

$$\rho(W,R) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_i \cdot c_{ij} \cdot y_{ij} \le 3 \cdot \sum_{i=1}^{n} w_i C_i = 3 \cdot \Theta_1, \quad (27)$$

which is at most triple cost of the LP. □

5. Query Cost Estimation

In this section, we propose an effective model to estimate the query cost for the replica selection problem.

We estimate the cost of a query with respect to a replica via the expectation of the running time towards the replica. Since each partition p of a replica r consists of a spatial range $S(p)$ and a temporal range $T(p)$, we will show our estimations of the query cost in both spatial and temporal aspects.

As defined in Definition 6, in this paper, we consider q as a cuboid and we use $\mu(q)$ to denote the spatial-temporal range of q, i.e., $\langle x, y, t, \delta_x, \delta_y, \delta_t \rangle$. To clearly show the proof, in this section, we use $S(q)$ to denote the spatial range of q, where $S(q) = \langle x, y, w, h \rangle$, $\langle x, y \rangle$ is the top-left point of the rectangle and $w(q)$ and $h(q)$ are the width and the height of the rectangle, respectively. Similarly, for each partition $p \in P(r)$, we use $w(p)$ and $h(p)$ to denote the width and the height of the partition.

To clearly address the expected partitions that a query should scan, we consider the queries are uniformly distributed in the space, as shown in Figure 3. In Figure 3, $w(D)$ and $h(D)$ are the width and height of the map, respectively. The query is shown as a blue rectangle, and the top-left point of the query is only allowed to be generated in the gray area, because if a query exceeds the spatial range of the map, such query can be considered as another query with a smaller spatial range. The probabilities of the top-left point being anywhere of the gray area are the same, i.e., uniformly distributed.

5.1. Expected Spatial Partitions. In this paper, given a workload W, the probability of a spatial partition being scanned is clearly the quotient of the number of queries overlapped with the partition, being divided by the total number of queries in W. Since the queries are uniformly distributed, the probability can be written as the quotient of the area within which the queries may overlap with the partition (the orange rectangle in Figure 4), being divided by the entire area that all the queries belong to (the gray area in Figure 3).

Assuming the distances between a partition p and the boundary of the map are $west(p)$, $east(p)$, $north(p)$, and $south(p)$, we define the expected spatial partitions as follows.

Theorem 11 (expected spatial partitions). *Given query q and replica r with partitions $p \in P(r)$, the expected number of spatial partitions $\varepsilon_s(q,r)$ that the query should scan is*

$$\varepsilon_s(q,r) = \sum_{p \in P(r)} \varepsilon_s(q,p), \quad (28)$$

where

$$\varepsilon_s(q,p) = \frac{(w(q) + w(p) - w(\alpha)) \cdot (h(q) + h(p) - h(\alpha))}{(w(D) - w(q)) \cdot (h(D) - h(q))}, \quad (29)$$

where α is the offset of the query, and

$$\begin{aligned} w(\alpha) &= \max(0, w(q) - west(p)) \\ &\quad + \max(0, w(q) - east(p)), \\ h(\alpha) &= \max(0, h(q) - north(p)) \\ &\quad + \max(0, h(q) - south(p)). \end{aligned} \quad (30)$$

Proof. The proof of the denominator of $\varepsilon_s(q,p)$ is trivial; thus we only consider the numerator, denoted by S; i.e., the area within which the queries may overlap with the partition as shown in Figure 4. In Figure 4, the query is colored in blue, and the partition is colored in purple. The orange rectangle shows the area within which the queries may overlap with the partition.

(1) The area of the partition is smaller than the query, as shown in Figure 4(a). From observation, we have $S = (w(q) + w(p)) \cdot (h(q) + h(p))$, and $w(a) = h(a) = 0$. Hence Theorem 11 holds.

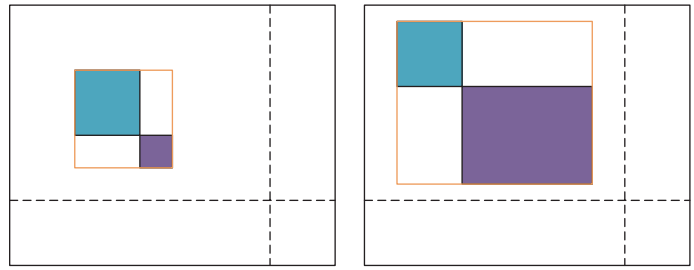
(2) The area of the partition is larger than the query, as shown in Figure 4(b). Similar to the previous situation, Theorem 11 holds.

(3) The partition is in the corner and exceeds the legal range of the query, as shown in Figure 4(c). From observation, we have $S = (w(q)+w(p)-w(a)) \cdot (h(q)+h(p)-h(a))$. Hence Theorem 11 holds.

(4) The partition is adjacent to the boundary, as shown in Figure 4(d). From observation, we have $S = (w(q)+w(p)) \cdot (h(q)+h(p)-h(q)) = (w(q)+w(p)) \cdot h(p)$, since $h(q)+h(p)-h(a) = h(q)+h(p)-0-(h(q)-0) = h(p)$; Theorem 11 holds.

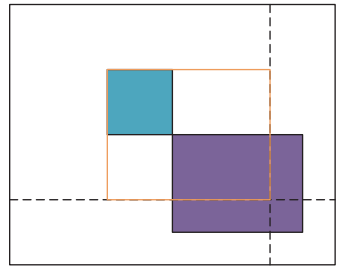
(5) The partition is adjacent to more than two boundaries. This is not possible based on the spatial partition scheme, because the number of partitions ≥ 4.

In conclusion, Theorem 11 holds. □

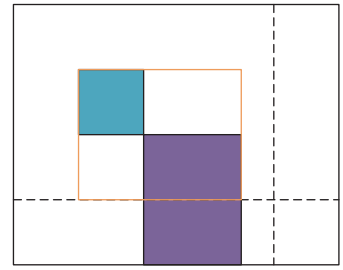


(a) Partition is smaller than the query

(b) Partition is larger than the query



(c) Partition exceeds the legal range of the query



(d) Partition is adjacent to the boundary

FIGURE 4: Different situations when calculating $\varepsilon_s(q, p)$.

5.2. Expected Temporal Partitions. Similar to the expected spatial partitions, the probability of a temporal partition being scanned is the quotient of the temporal range within which the queries may overlap with the partition, being divided by the temporal range that all the queries belong to. Assuming the intervals between a partition p and the temporal range of all the records $T(D)$ are $top(p)$ and $bot(p)$, we define the expected temporal partitions as follows.

Theorem 12 (expected spatial partitions). *Given query q and replica r with partitions $p \in P(r)$, the expected number of temporal partitions $\varepsilon_t(q, r)$ that the query should scan is*

$$\varepsilon_t(q, r) = \sum_{p \in P(r)} \varepsilon_t(q, p), \tag{31}$$

where

$$\varepsilon_t(q, p) = \frac{T(q) + T(p) - T(\alpha)}{T(D) - T(q)}, \tag{32}$$

where α is the offset of the query, and

$$\begin{aligned} T(\alpha) = {} & \max(0, T(q) - top(p)) \\ & + \max(0, T(q) - bot(p)). \end{aligned} \tag{33}$$

The proof of Theorem 12 is similar to Theorem 11.

5.3. Expected Query Cost. As described in Section 3, to answer query q on replica r, a BLOT system scans (the physically stored objects of) all partitions $p \in P(r)$ that satisfies $\mu(p) \cap \mu(q) \neq \emptyset$ and then filters each record by $\mu(q)$. Based on the expected number of spatial and temporal partitions that a query should scan, we can combine them as the expectation of desired partitions given query q:

$$\varepsilon(q, r) = \sum_{p \in P(r)} \varepsilon_s(q, p) \cdot \varepsilon_t(q, p). \tag{34}$$

Now given the number of spatial and temporal partitions n_s and n_t of $P(r)$, respectively, we have

$$\rho(q, r) = \varepsilon(q, r) \cdot \left(\frac{\eta(r)}{n_s \cdot n_t \cdot \zeta(r)} + \xi(r) \right) \tag{35}$$

where $\zeta(r)$ and $\xi(r)$ are the scanning speed in terms of number of records scanned per unit time and the time before and after the actual scan process of the replica given its encoding scheme $E(r)$, respectively. For example, if each partition is stored continuously as a regular file on a local disk, then $\xi(r)$ is the seek time of locating the beginning of the file and $\zeta(r)$ is the transfer rate of the disk (assuming that CPU always waits for I/O). As another example, if each partition is stored as an object on Amazon S3 and queries are processed on Amazon EMR (Elastic MapReduce), then $\xi(r)$ is the time initializing the map task plus the time locating the S3 object before scanning the partition. The value of $\zeta(r)$ depends on the encoding scheme $E(r)$. In real situations, a high compression ratio generally leads to a slow scan speed.

In this paper, we assume that all candidate partitioning schemes will generate non-skewed data partitions. In other words, the number of records in each $D(p_i)$ is almost the same for all $p_i \in P(r)$. Non-skewed partitioning is a desirable property when partitions are processed in parallel (e.g., in MapReduce). An example of such partitioning schemes is using a k-d tree to partition the space where data are split equally each time the space is subdivided.

Table 2: Compression ratio of encoding schemes.

Uncompressed		Snappy		GZip		LZMA2	
Row	Col	Row	Col	Row	Col	Row	Col
1	0.557	0.485	0.312	0.283	0.179	0.213	0.156

Putting (34) and (35) together, we can compute the cost of any query on replica r in $O(|P(r)|)$ time. It follows that the time complexity of computing all query costs is

$$O\left(|W| \cdot |R| \cdot \max_{r_j \in R} \left|P\left(r_j\right)\right|\right). \tag{36}$$

6. Evaluation

In this section, we describe the experiment settings and present the evaluation results in detail.

6.1. Experiment Settings. We consider two typical execution environments for BLOT systems. The first one is a local Hadoop cluster where each partition is stored as a separate file on HDFS. The second one uses Amazon S3 to store partitions. To process a query, we lunch a map-only MapReduce job, either in local cluster or in Amazon EMR, with each mapper scanning exactly one of the involved partitions. The dataset we use is a sample of vehicle GPS log collected from more than 4,000 taxis in Shanghai during a month. Each record contains 8 attributes (including the 3 core attributes). The total number of records is around 65 million and the total storage size in uncompressed CSV format is 3.7 GB. The latitude ranges from 30 to 32, longitude from 120 to 122, and time from 11/01/2007 to 11/29/2007. It is worth pointing out that though the full dataset in our working system is more than 100 GB, we only need a small portion of the data to build the cost model and select diverse replicas for the whole dataset.

For data partitioning, we first partition the space and then the time to generate equal-sized (in terms of number of records) partitions. The space is partitioned according to a k-d tree [9] index which recursively decomposes the space by alternatively using each space dimension. The number of spatial partitions is chosen from 4^2, 4^3, 4^4, 4^5, 4^6 and the number of temporal partitions is chosen from $2^4, 2^5, 2^6, 2^7, 2^8$. Therefore, there are $5 \times 5 = 25$ candidate spatial-temporal partitioning schemes in total. For data encoding, we store data either by row or by column (with delta encoding), with an option of whether or not using a general compression method chosen from Gzip, Snappy, and LZMA2. Since uncompressed column-store has poor performance in terms of both compression ratio and scan speed, we do not use it as a candidate encoding scheme. Therefore, there are $2 \times 4 - 1 = 7$ candidate encoding schemes in total. The compression ratio of each encoding schemes measured on our dataset is listed in Table 2. Putting the above partitioning schemes and the encoding schemes together, the total number of candidate replicas is $25 \times 7 = 150$.

6.2. Measuring Scan Rate and Extra Time. Since ζ and ξ are constants with respect to encoding schemes, we conduct $7 \times 2 = 14$ measurements corresponding to 7 candidate encoding schemes in each execution environments, respectively.

For each measurement, we generate 5 sets of partitions with each set containing 20 partitions. The sizes of partitions within a partition set are the same while they are different across partition sets. We then launch a map-only MapReduce job with 20 mappers with each scanning a partition. After the job is finished, we compute the average processing time of all mappers and use it as the (measured) value of $\rho(q, p)$ in (34). Accordingly, we use the corresponding partition size (in terms of number of records) as $\eta(r)$. We therefore have 5 measured points for (35). In the last step, we perform linear regression to fit the measured points and use the fitted parameters as $1/\zeta$ and ξ.

In Figure 5, the left two subfigures show all the measurement results and the right two subfigures show the fitted lines for three measurements in each of the execution environments. In addition, the measured values of ζ and ξ are listed in Table 3. We can see that $\rho(q, p)$ is well-fitted by (35) especially when the size of partition is relatively large, which demonstrates the effectiveness of our cost model.

6.3. Performance of Replica Selection. To measure the effectiveness and the efficiency of our replica selection algorithms, we construct a synthetic workload containing 8 grouped queries with wildly varied range size. We conduct all the following experiments in the Amazon S3 and EMR execution environment.

Figure 6 compares the computation time via MIP upon different sizes of workload and candidate replicas. When the size of the given workload or candidate replica set increases, we can see that the computation time of the MIP solution increases exponentially. Hence, when the input workload or the candidate replica set is too large, it is desirable to switch to the greedy algorithm which runs in polynomial time.

Figure 7 compares the relative query performance for all the queries when the replica set is selected by different approaches. The storage budget is set to be the same as the storage size of 3 exact copies of the optimal single replica. The approximation ratio of each approach is shown in the brackets of the figure (ideal case is always 1.00). It is clear that when the size of data grows, the performance of the greedy algorithm and the MIP solution is closer to the ideal case than a single replica; thus the advantages of using diverse replicas become more and more prominent. Figure 8 shows the overall query performance relative to the ideal case when varying the storage budget. In this figure, the x-coordinate is the storage budget relative to the storage budget used in Figure 7. We can see that when the MIP solution is close to the ideal case regardless of the storage budget, which is faster than

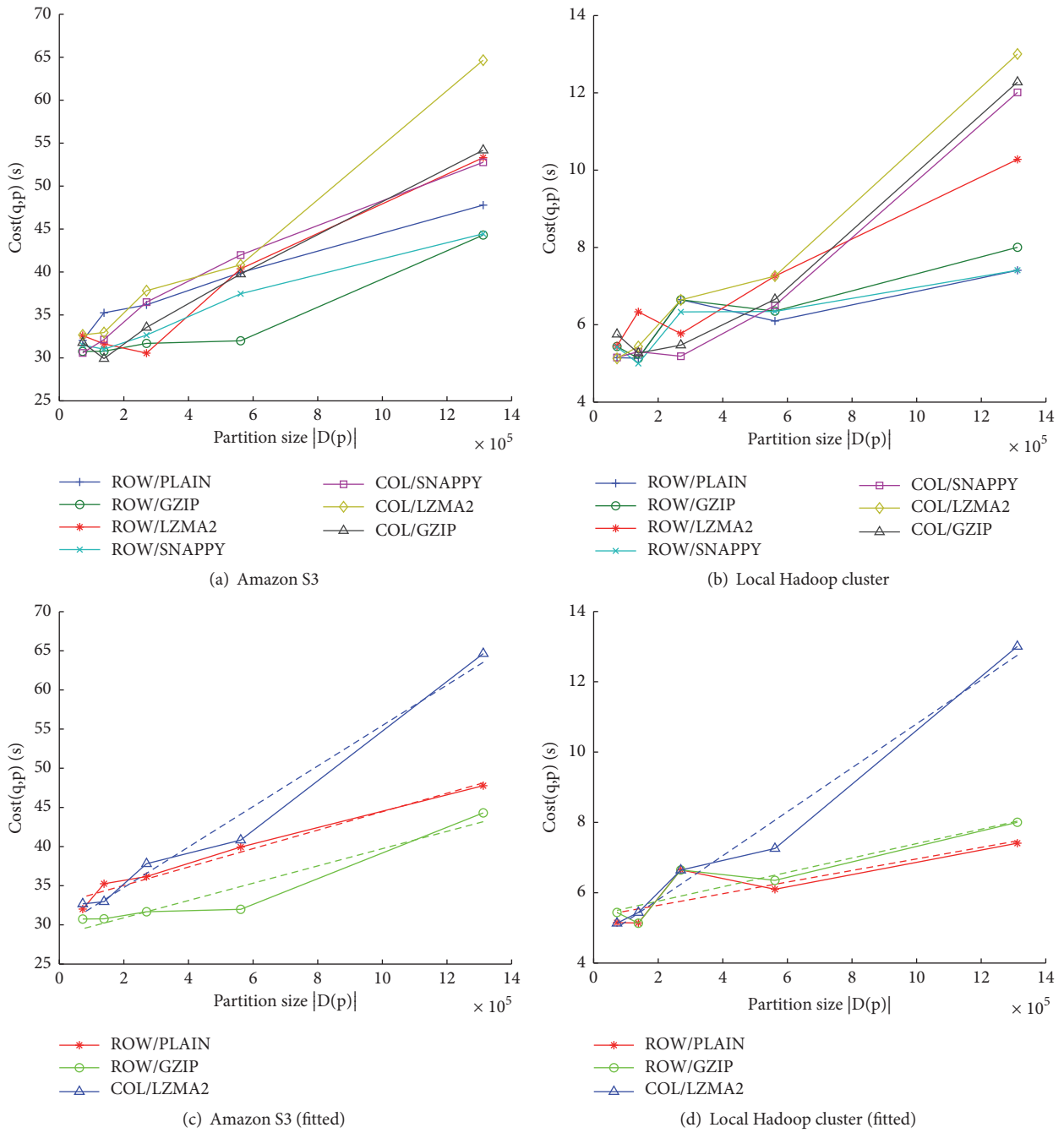


Figure 5: Measurement results of $\rho(q, p)$.

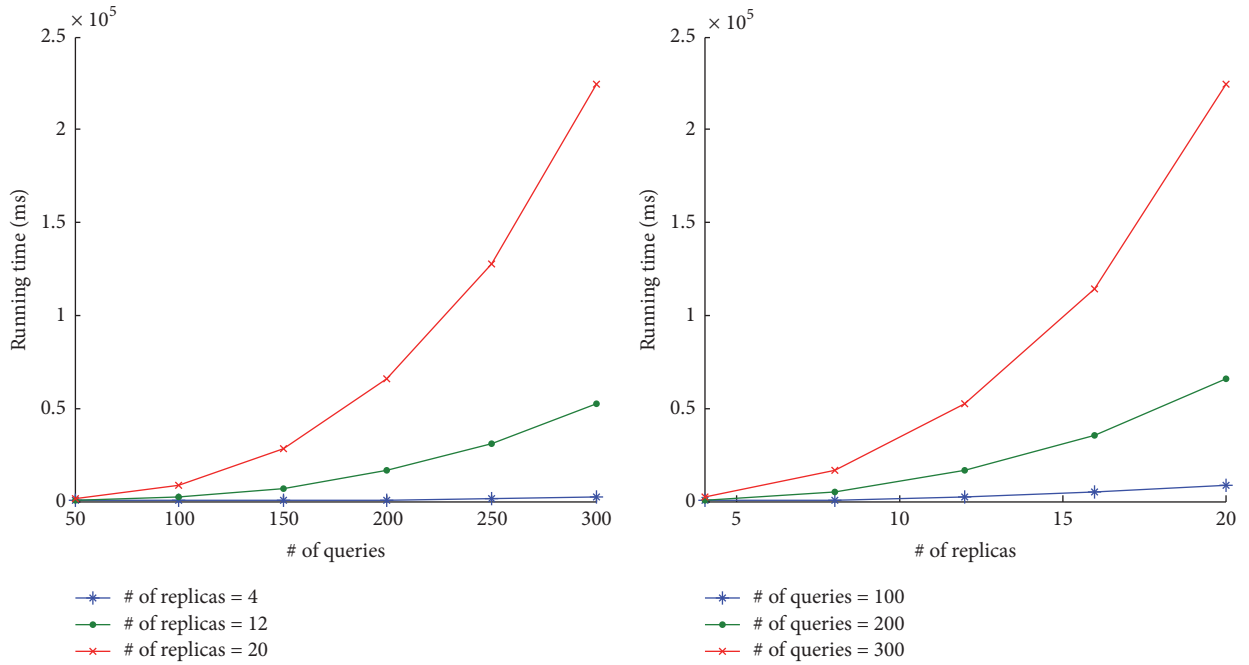
the single replica case by up to 80%, the approximation ratio of the greedy algorithm decreases dramatically as the storage budget increases. When the relative storage budget is greater than 1, the approximation ratio of the greedy algorithm is less than 1.2.

7. Conclusion

In this paper, we explore the use of diverse replicas in the context of storage systems for big location-based mobile data. Specifically, we propose BLOT, a system abstraction that describes an important class of location-based mobile data storage systems. Then, we formally define the replica selection problem that finds the optimal set of diverse replicas. We propose two solutions to address this problem, including an exact algorithm based on integer programming and an approximation algorithm based on greedy strategy. In addition, we propose several practical approaches to reduce the input size of the problem. We also design a simple yet effective cost model to estimate the cost of

Table 3: Measured ζ and ξ.

		Amazon S3 and EMR	
		$1/\zeta$ (ms)	ξ (ms)
Uncompressed	Row	85.02	32689
	Col	N/A	N/A
Snappy	Row	90.24	30187
	Col	56.98	30518
Gzip	Row	90.65	28698
	Col	51.72	28725
LZMA2	Row	54.39	29029
	Col	38.69	29609
		Local Hadoop Cluster	
		$1/\zeta$ (ms)	ξ (ms)
Uncompressed	Row	606.78	5312
	Col	N/A	N/A
Snappy	Row	598.84	5316
	Col	175.75	4150
Gzip	Row	488.32	5349
	Col	177.15	4427
LZMA2	Row	265.41	5244
	Col	159.98	4551



(a) Varying size of workload

(b) Varying size of candidate replicas

Figure 6: Comparison of the computation speed of MIP.

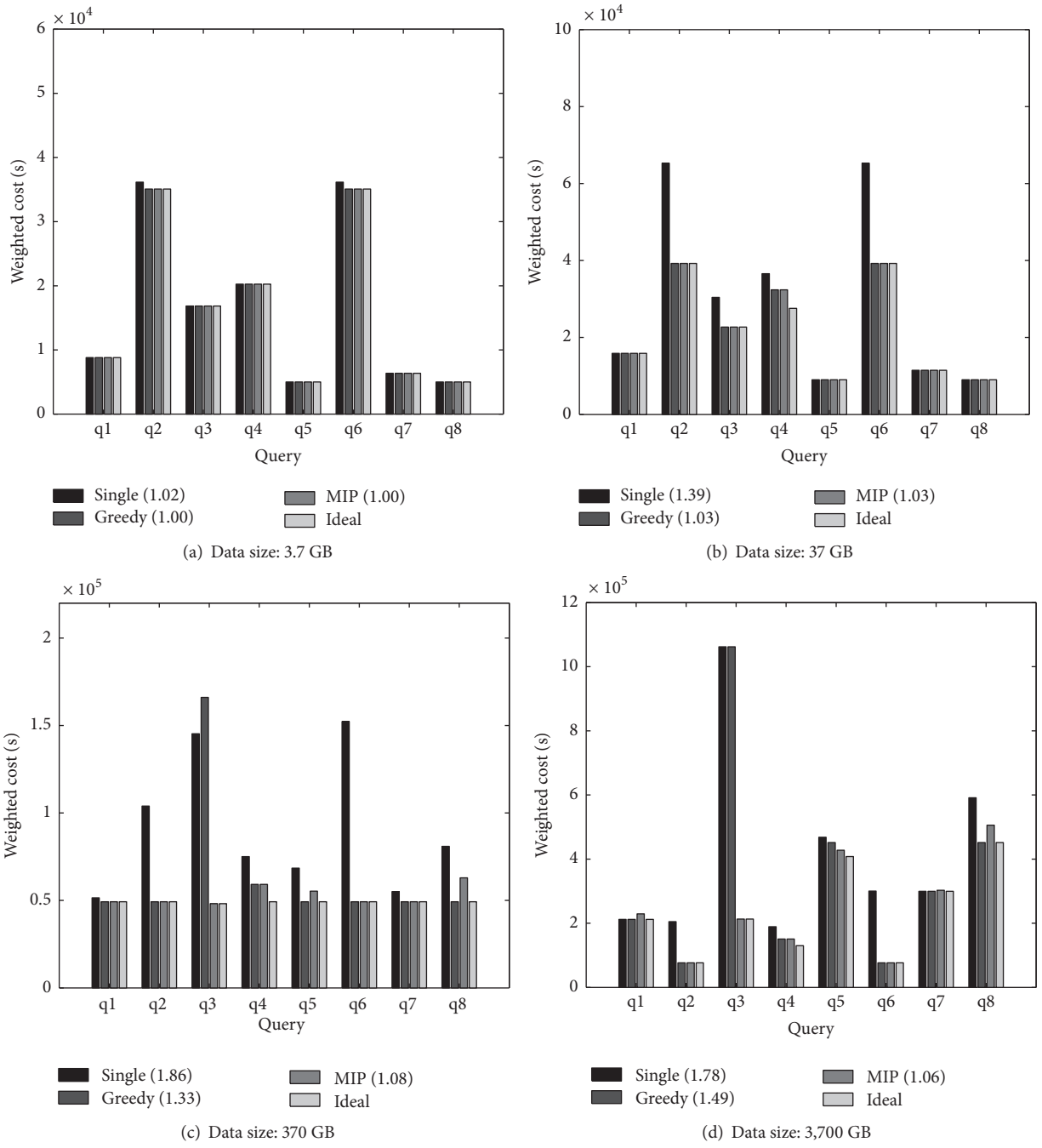


Figure 7: Relative overall query performance in Amazon S3 and EMR.

an arbitrary query on an arbitrary replica configuration. Finally, we evaluate our solutions using two typical execution environments including Amazon and local Hadoop cluster. The results demonstrate that the proposed algorithms for the replica selection problem is both effective and efficient. In this paper, we only consider full replication of the entire data. The use of partial replication, where only frequently accessed data ranges are replicated, is one of our future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

An earlier version of this work appeared in the Proceedings of IEEE ICDCS [36], June 2014.

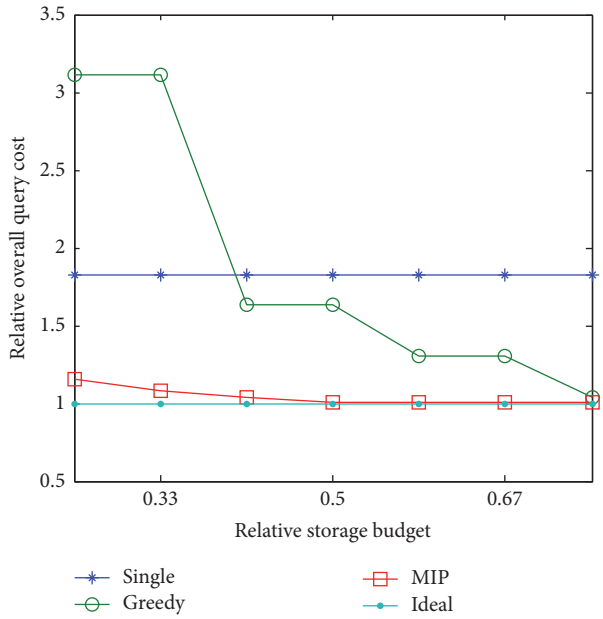


FIGURE 8: Relative overall query performance of different storage budgets.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China under Grant no. 2017YFB0202201 and National Natural Science Foundation of China under Grant no. U1711261.

References

[1] A. Doshi and M. M. Trivedi, "Tactical driver behavior prediction and intent inference: A review," in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems, ITSC 2011*, pp. 1892–1897, Washington, DC, USA, 2011.

[2] Y. Ding, J. Zheng, H. Tan, W. Luo, and L. M. Ni, "Inferring road type in crowdsourced map services," *DASFAA*, pp. 392–406, 2014.

[3] Y. Ding, S. Liu, J. Pu, and L. M. Ni, "Hunts: A trajectory recommendation system for effective and efficient hunting of taxi passengers," *MDM*, pp. 107–116, 2013.

[4] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," *SIGMOD*, pp. 713–724, 2013.

[5] C. S. Jensen, H. Lu, and B. Yang, "Geolife: A collaborative social networking service among user, location and trajectory," *IEEE Data Eng. Bull*, vol. 33, no. 2, pp. 12–17, 2010.

[6] P. Cudre-Mauroux, E. Wu, and S. Madden, "Trajstore: An adaptive storage system for very large trajectory data sets," *ICDE*, pp. 109–120, 2010.

[7] H. Tan, W. Luo, and L. M. Ni, "Clost: A hadoop-based storage system for big spatio-temporal data analytics," *CIKM*, pp. 2139–2143, 2012.

[8] A. M. Hendawi and M. F. Mokbel, "Panda: a predictive spatio-temporal query processor," in *Proceedings of the SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems (formerly known as GIS), SIGSPATIAL'12, Redondo Beach*, vol. 2012, pp. 13–22, CA, USA, 2012.

[9] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[10] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys*, vol. 16, no. 2, pp. 187–260, 1984.

[11] A. Guttman, "R-trees: a dynamic index structure for spatial searching," *SIGMOD*, pp. 47–57, 1984.

[12] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel approaches in query processing for moving object trajectories," *VLDB*, pp. 395–406, 2000.

[13] V. Botea, D. Mallett, M. A. Nascimento, and J. Sander, "PIST: An efficient and practical indexing technique for historical spatio-temporal point data," *GeoInformatica*, vol. 12, no. 2, pp. 143–168, 2008.

[14] A. Eldawy and M. F. Mokbel, "A demonstration of spatial-hadoop: An efficient mapreduce framework for spatial data," *PVLDB*, vol. 6, no. 2, 2013.

[15] H. Vo, A. Aji, and F. Wang, "SATO: a spatial data partitioning framework for scalable query processing," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Dallas/Fort Worth*, vol. 2014, pp. 545–548, TX, USA, 2014.

[16] S. Khoshafian, G. Copeland, T. Jagodits, H. Boral, and P. Valduriez, "A query processing strategy for the decomposed storage model," *ICDE*, pp. 636–643, 1987.

[17] N. Bruno and S. Chaudhuri, "Constrained physical design tuning," *PVLDB*, vol. 1, no. 1, pp. 4–15, 2008.

[18] D. Dash, N. Polyzotis, and A. Ailamaki, "Cophy: A scalable, portable, and interactive index advisor for large workloads," *PVLDB*, vol. 4, no. 6, pp. 362–372, 2011.

[19] H. Kimura, G. Huo, A. Rasin, S. Madden, and S. B. Zdonik, "Coradd: Correlation aware database designer for materialized views and indexes," *PVLDB*, vol. 3, no. 1, pp. 1103–1113, 2010.

[20] A. Jindal, J.-A. Quiané-Ruiz, and J. Dittrich, "Trojan data Layouts: Right shoes for a running elephant," *SOCC*, p. 21, 2011.

[21] M. P. Consens, K. Ioannidou, J. Lefevre, and N. Polyzotis, "Divergent physical design tuning for replicated databases," *SIGMOD*, pp. 49–60, 2012.

[22] J. Chen, K. Hu, Q. Wang, Y. Sun, Z. Shi, and S. He, "Narrowband Internet of Things: Implementations and Applications," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2309–2314, 2017.

[23] L. Wang, X. Qi, J. Xiao, K. Wu, M. Hamdi, and Q. Zhang, "Exploring Smart Pilot for Wireless Rate Adaptation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4571–4582, 2016.

[24] R. Ramamurthy, D. J. DeWitt, and Q. Su, "A case for fractured mirrors," *VLDB*, pp. 430–441, 2002.

[25] D. Sacca and G. Wiederhold, "Database Partitioning in a Cluster of Processors," *ACM Transactions on Database Systems*, vol. 10, no. 1, pp. 29–56, 1985.

[26] S. Agrawal, V. Narasayya, and B. Yang, "Integrating vertical and horizontal partitioning into automated physical database design," *SIGMOD*, pp. 359–370, 2004.

[27] M. Grund, J. Krüger, H. Plattner, A. Zeier, P. Cudre-Mauroux, and S. Madden, "Hyrise - a main memory hybrid storage engine," *PVLDB*, vol. 4, no. 2, pp. 105–116, 2010.

[28] R. A. Hankins and J. M. Patel, "Data morphing: An adaptive, cache-conscious storage technique," *VLDB*, pp. 417–428, 2003.

[29] S. Navathe, S. Ceri, G. Wiederhold, and J. Dou, "Vertical Partitioning Algorithms for Database Design," *ACM Transactions on Database Systems*, vol. 9, no. 4, pp. 680–710, 1984.

[30] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "Slade: a smart large-scale task decomposer in crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[31] S. Chaudhuri, A. K. Gupta, and V. Narasayya, "Compressing sql workloads," *SIGMOD*, pp. 488–499, 2002.

[32] J. M. Kleinberg and É. Tardos, *Algorithm Design*, Addison-Wesley, 2006.

[33] R. J. Dakin, "A tree-search algorithm for mixed integer programming problems," *The Computer Journal*, vol. 8, no. 3, pp. 250–255, 1965.

[34] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys, "A constant-factor approximation algorithm for the k-median problem," *Journal of Computer and System Sciences*, vol. 65, no. 1, pp. 129–149, 2002.

[35] J.-H. Lin and J. S. Vitter, "Approximation algorithms for geometric median problems," *Information Processing Letters*, vol. 44, no. 5, pp. 245–249, 1992.

[36] Y. Ding, H. Tan, W. Luo, and L. M. Ni, "Exploring the use of diverse replicas for big location tracking data," in *Proceedings of the IEEE 34th International Conference on Distributed Computing Systems, ICDCS 2014*, pp. 83–92, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 8956757, 8 pages
https://doi.org/10.1155/2018/8956757

Research Article

Enhance RSS-Based Indoor Localization Accuracy by Leveraging Environmental Physical Features

Peng Xiang, Peng Ji, and Dian Zhang

Guangdong Province Key Laboratory of Popular High Performance Computers, Shenzhen University, Shenzhen, China

Correspondence should be addressed to Dian Zhang; zhangd@szu.edu.cn

Received 25 February 2018; Accepted 4 May 2018; Published 9 July 2018

Academic Editor: Wei Wang

Indoor localization technologies based on Radio Signal Strength (RSS) attract many researchers' attentions, since RSS can be easily obtained by wireless devices without additional hardware. However, such technologies are apt to be affected by indoor environments and multipath phenomenon. Thus, the accuracy is very difficult to improve. In this paper, we put forward a method, which is able to leverage various other resources in localization. Besides the traditional RSS information, the environmental physical features, e.g., the light, temperature, and humidity information, are all utilized for localization. After building a comprehensive fingerprint map for the above information, we propose an algorithm to localize the target based on Naïve Bayesian. Experimental results show that the successful positioning accuracy can dramatically outperform traditional pure RSS-based indoor localization method by about 39%. Our method has the potential to improve all the radio frequency (RF) based localization approaches.

1. Introduction

It is widely accepted that indoor localization is essential to many service applications and attracts many researchers' attentions [1–6]. For example, indoor navigation is helpful for the customer to find the path to the destination in a large shopping mall. In a hospital building, the patients can be easily found and taken care if indoor localization technologies are applied. In the underground parking place, indoor localization is useful for people to find their vehicles.

Among various indoor localization technologies [7–22], the technologies based on the Radio Signal Strength (RSS) are popular, since RSS can be easily obtained by common wireless devices (e.g., wireless sensors, mobile phone) without additional hardware. However, RSS are apt to be affected by indoor environments, since radio signal is easily reflected, refracted, and scattered by various indoor objects [23]. Therefore, signal emitted from one transmitter will arrive at the receiver from many different propagation paths. Such phenomenon is called multipath phenomenon. As a result, although many improvement works have been proposed, the localization accuracy based on such technologies is very difficult to improve.

In order to overcome this drawback, we put forward an approach, which is able to leverage various other resources in localization. In detail, besides the traditional RSS information, the environmental physical features, e.g., the light, temperature, and humidity, are all comprehensively utilized in localization. At the same time, no much additional cost is introduced.

The basic idea is to construct a comprehensive map [5, 24–26] named RTHL (means the RSS, temperature, humidity, and light), which not only covers the fingerprint of RSS information, but also contains the fingerprint of temperature, humidity, and light information from different places in the target area. Figure 1 shows an example of our basic idea. There are some reference sensor nodes (marked as 1, 2, ..., 6) hung on the ceiling acting as the transmitter. The tracking node on the ground acts as the receiver. At the beginning of training phase, we will construct the RTHL map, where the RSS, light intensity, temperature, and humidity information by the receiver at different places are all collected. In the online phase, we may utilize our localization algorithm based on Naïve Bayesian to accurately localize the target object.

Our experiments are based on 7 telosB [27] sensors, which are placed randomly in an $8\times10m^2$ laboratory.

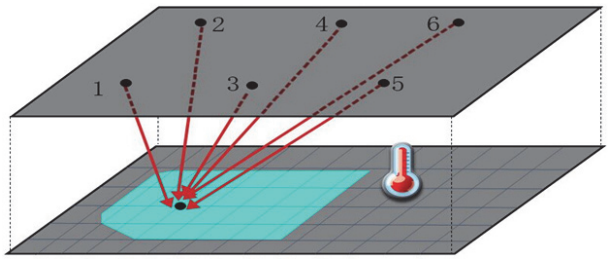


Figure 1: Basic idea.

Experimental results show us that the successful localization accuracy can dramatically outperform traditional indoor localization method based on just RSS by about 39%.

The main contributions of this paper are as follows. First, we introduce different environmental physical features to improve the localization accuracy of traditional pure RSS-based indoor localization. Second, we propose one localization algorithm which can accurately localize the target. Third, we comprehensively perform the experiments in different indoor environments with low cost. At last but not least, our method is a general method, which has the potential to improve all the RF-based localization approaches.

The rest of this paper is organized as follows. In the next section, we will introduce the related work about localization technologies. In the following, we will describe our methodology in detail. Section 4 will present our localization system implementation and evaluate its performance. Finally, we will conclude this work and point out some possible future work directions.

2. Related Work

There are a large number of indoor localization technologies utilizing RSS [1–3, 19–21] or CSI information [22, 28–32]. RSS information can be easily obtained from common wireless devices, e.g., wireless sensors and smart mobile phone, since it is a kind of free source and can be obtained without any cost. Channel State Information (CSI) can also be collected if physical layer information can be visited.

For RSS-based technologies, Bluetooth [16–18, 33, 34] is usually embedded in mobile phones, personal digital assistants (PDA), laptops, and other portable electronic devices, and the localization accuracy is limited. WLAN [1, 11, 26] have been deployed in many public places, such as hospitals and universities. A large number of fingerprinting technologies are proposed accordingly to improve the localization accuracy. ZigBee technologies [2, 24, 35] are able to localize the target through the mutual communication between sensors. Among these technologies, plenty of methods are adopted, including the machine learning algorithms and optimization algorithms, to improve the localization accuracy. However, since the radio signal is easily affected the by multipath phenomenon in indoor environments, the localization accuracy is difficult to be improved. Our work introduces other environmental features in localization, which is able to potentially improve all the above methods.

For CSI information [22, 31, 32], since it should visit the physical layer information, the device type is limited. WiFi devices are often selected in such technologies and special hardware interfaces are usually applied. However, in such technologies, CSI information is also easily affected by the multipath phenomenon in indoor environments. Our work can also improve the localization accuracy of such technologies.

There are also some other works introducing Inertial Measurement Unit (IMU) device. They utilize the accelerometer and Gyroscope information [36–40] from IMU in localization. However, such sensors are not available in all the devices. Furthermore, our work introduces more environment features, e.g., temperature, humidity, and light information, in localization. It can give an optional choice for some particular devices and improve the localization accuracy of those systems using IMU.

Luxapose [41] is explored for indoor positioning by using unmodified smartphones, it is required to slightly modify commercial LED luminaries. Moreover, it only utilizes one single environment feature (light intensity) in localization. If other environmental features are considered, the accuracy can be further enhanced. The SpyLoc [42] localization system leverages both acoustic and WiFi information in localization. However, if more environmental features such as ours are introduced, the localization accuracy can be further improved.

3. Methodology

In this section, we will first introduce the basic idea of our method. In the following, the detail algorithm to localize the target object will be described.

3.1. Basic Idea. In the subsection, we will show how the physical features can help in localization. We perform an initial experiment based on just one pair of transmitter and receiver (both TelosB sensors). The receiver is placed in different positions on the ground with different distance to the transmitter. Since the TelosB sensors are integrated with the light, temperature, and the humidity sensorsïijŇwe are able to collect the RSS plus the light intensity, temperature, and the humidity information from the receiver sensor.

As shown in Figure 2, we can observe that, at different positions on the ground, the RSS, light intensity, temperature, and the humidity information all have variance. Therefore, if we can leverage such variance of both RSS and physical features, we may improve the localization accuracy based on just RSS information.

3.2. Building the RTHL Map. Since we will utilize the RSS, temperature, humidity, and light intensity information (RTHL) together to localize the target, we should construct an RTHL map during the offline phase. At first, we hang a number of transmitters (e.g., 4, we will discuss how to set such value in the later subsection) on the ceiling. A receiver is placed at different places on the ground to collect its RTHL information. Such information will be transmitted back to the server. At each place on the ground, we can get a set of

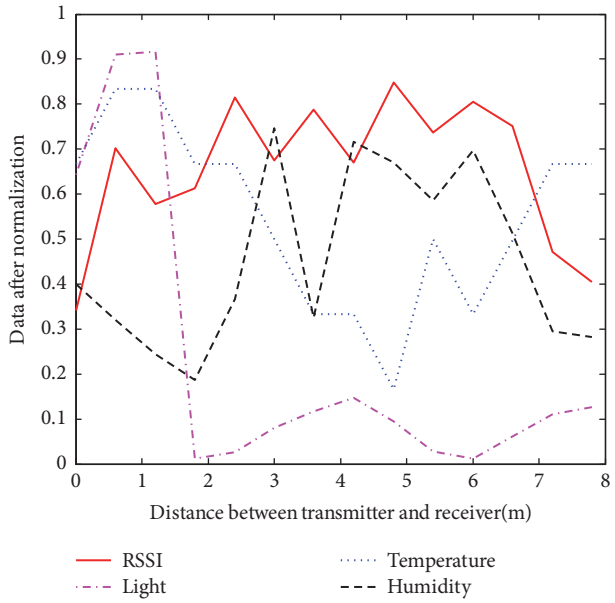


FIGURE 2: Observation based on one pair of transmitter and receiver.

vector $A_i = (A_{i1}, A_{i2}, \ldots, A_{in})$, $i \in (1, m)$, where the value of n depends on the number of reference nodes on the ceiling plus 3 physical feature values (the receiver's light, temperature, and humidity) and m is the number of known places in this environment. Considering the variance of different RSS and physical feature information, for each attribute A_{ij}, $i \in (1, m)$, $j \in (1, n)$, we run a Min-Max normalization to eliminate the variance; then RTHL map can be built.

3.3. Naïve Bayesian Algorithm. After the RTHL map is built, in the online phase, when the target acting as the receiver appears in the same environment, the localization algorithm is able to be performed based on the RTHL map. The detail algorithm will be introduced below.

In our system, using the Naïve Bayesian classification, we see the different locations in RTHL map which have known coordinates as one category C_i, supposing there are m locations, $i \in (1, m)$. When the tracking node enters the environment, it will receive the RSS information from different reference nodes, as well the sensed local temperature, humidity, and light intensity information. All this information is represented by a vector $X = (x_1, x_2, \ldots, x_n)$, where the value of n is the number of reference nodes on the ceiling plus 3 physical feature values. The localization algorithm basically can be divided into the following 2 steps.

Step 1. Calculate $P(C_1/X)$, $P(C_2/X), \ldots, P(C_i/X)$, $P(C_m/X)$, $i \in (1, m)$, where $P(C_i/X)$ is the probability of C_i given X.

Step 2. If $P(C_k/X) = \max\{P(C_1/X), P(C_2/X), \ldots, P(C_k/X), \ldots, P(C_m/X)\}$, $k \in (1, m)$, we regard that the category of X is C_k, which is the calculated target position.

The individual conditional probability in *Step 1* can be calculated by the following.

(1) First, the categories of all items in the training sample set are already known. Therefore, we may get the conditional probability of each feature property in different categories by statistical methods, which are represented as

$$\begin{aligned} &P(x_1/C_1), P(x_2/C_1), \ldots, P(x_n/C_1) \\ &P(x_1/C_2), P(x_2/C_2), \ldots, P(x_n/C_2) \\ &\cdots \\ &P(x_1/C_m), P(x_2/C_m), \ldots, P(x_n/C_m). \end{aligned} \tag{1}$$

(2) Since characteristic properties are independent of each in condition, we can obtain the following through the Naïve Bayesian theory:

$$P(C_i/X) = \frac{P(X/C_i)\,P(C_i)}{P(X)} \tag{2}$$

where $P(C_i)$ is the probability of category C_i and $P(X)$ is the probability of X.

In formula (2), since the denominator is a constant for all categories, we just need the molecule to be maximized. Each property is independent on the condition; according to formula (2), we can know that

$$\begin{aligned} &P(X/C_i)\,P(C_i) \\ &\quad = P(x_1/C_i)\,P(x_2/C_i)\cdots P(x_n/C_i)\,P(C_i) \\ &\quad = P(C_i)\prod_{j=1}^{n} P(x_j/C_i) \end{aligned} \tag{3}$$

Therefore, with the known training samples and its categories, we can calculate the results of formula (3). According to the maximum value, we may determine which category X belongs to. Since the signal intensity is continuously distributed in the indoor space, the value of each feature property obeys the Gauss distribution:

$$g(x, \eta, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\eta)^2/2\sigma^2} \tag{4}$$

Here, we know that

$$\begin{aligned} P(x_j/C_i) &= g(x_j, \eta_j, \sigma_j) \\ \eta_j &= \frac{1}{m}\sum_{i=1}^{m} x_{ji} \\ \sigma_j &= \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(x_{ji} - \eta_j)^2} \quad j \in (1, n) \end{aligned} \tag{5}$$

Therefore, using formulas (3), (4), and (5), we can get the category of X. Since we know the coordinates of each category in advance, the coordinates of the tracking node can be obtained.

Figure 3: Experimental environment.

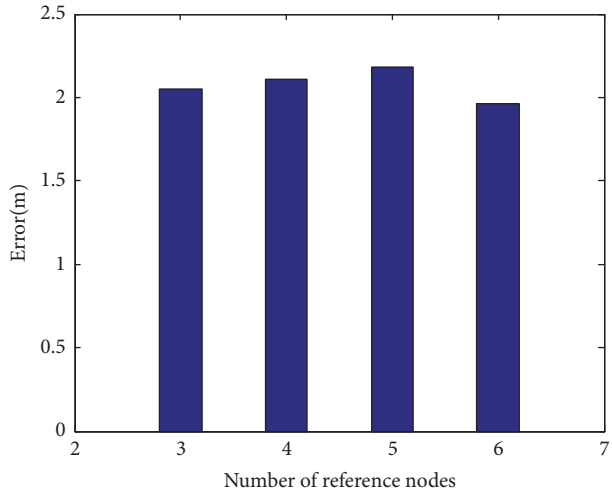


Figure 4: Comparison of different number of reference nodes.

4. Experiment and Evaluation

In this section, we first introduce the experimental environment of our system and then show the performance evaluation.

4.1. Experiment Setup. We carry out the experiments in our laboratory, whose area is 8×10 square meters, as shown in Figure 3. We utilize telosB sensor as the wireless node, which is composed of CC2420 radio chips and MSP430 microcontroller [43]. Such sensor is able to get the RSS information from other sensors, light intensity, humidity, and temperature from its local area. We program 6 sensors as reference nodes to broadcast beacons periodically. The reference nodes are hung on the ceiling. The default transmission power is 0 *dBm*. The default channel is 11. The tracking node on the ground acts as the receiver.

At first, in the offline phase, we build a RTHL fingerprint map consisting of RSS from the reference nodes, temperature, humidity, and the light intensity information of the whole laboratory environment. Later in the online phase, when the tracking node comes into the environment, it receives the RSS information from the reference nodes. It can sense its local temperature, humidity, and the light intensity information and transmit them back to the sink. The server runs our localization algorithm to calculate the position of the tracking node based on the RTHL map.

4.2. The Impact of the Number of Reference Nodes. At first, in order to investigate how the number of reference nodes will influence the localization results, we perform our experiments based on different number of reference nodes from 3 to 6. We choose 3 as the initial test number, since 3 is the minimum number of reference nodes in most fingerprint-based localization algorithms.

As Figure 4 shows, we can see that, when the reference node number is chosen as 3, 4, 5, and 6, the average errors are 2.05*m*, 2.11*m*, 2.18*m*, and 1.96*m*, respectively. We find that, in general, when the number of reference nodes increases, the localization error will decrease. When the number of reference nodes is set as 6, the localization error is the smallest.

Therefore, in our later experiments we use 6 reference nodes in our setting.

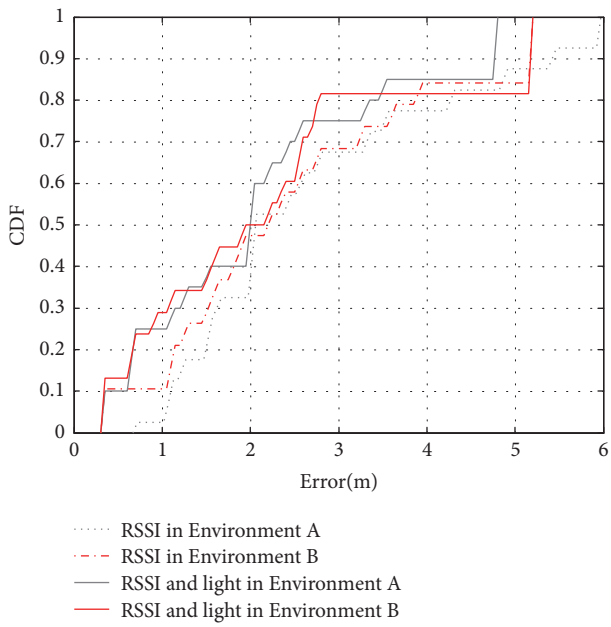


Figure 5: The impact of the light intensity.

4.3. The Impact of the Light Intensity. In this subsection, we will study how much the light intensity feature will improve the localization accuracy. Concerning it is a time-varying factor, and the light intensity information usually is different in the daytime and nighttime, we perform our experiments in such two scenarios. ***Environment* A** represents the daytime in our laboratory, while ***Environment* B** represents the nighttime in the same place. Based on 40 test samples, Figure 5 gives the experimental results. We find that the localization accuracy of leveraging light intensity information is much

better than just using RSS information. In the daytime, the localization accuracy is improved by more than 15%, which is possible due to the following reason. Since during the daytime the environment usually is filled with sunlight, the place in the lab close to the window usually has higher light intensity information. Therefore, the light intensity may vary a lot for different lab area. However, during the nighttime, the light from the lamp is usually uniformly distributed which results in the light intensity information having no big difference among different places. Usually larger variance in the RTHL fingerprint map will contribute more on the localization accuracy.

In summary, we find that, when leveraging light intensity information, the localization accuracy can be improved. If the environment has enough light intensity variance in different places, the localization accuracy can be further improved by about 30%.

4.4. The Impact of the Temperature. In this subsection, we will investigate how much the indoor temperature will improve the localization accuracy. We also perform our experiment in both day and nighttime. Based on 40 test samples, the experimental result is shown in Figure 6. We find that, with the help of temperature information, the localization accuracy can be improved by about 10% for both day and night environments. Also we observe that, no matter in the day or night environment, the improvement of localization accuracy has no big difference for the two environments. The reason is possible that indoor area usually has air conditioner. Therefore, there is little temperature difference among difference places. Only the places close to the air conditioner may have lower temperature than the other places.

In conclusion, we know that the temperature information can improve the localization accuracy by about 10%. If in other indoor area with higher temperature variance among different places, we believe the localization accuracy can be further improved.

4.5. The Impact of the Humidity. In this subsection, we will explore how much the indoor humidity will improve the localization accuracy. We also carry out our experiment in both day and nighttime. Based on 40 test samples, we can see the experimental result from Figure 7. It shows the humidity information can help to improve the localization accuracy by around 10%, no matter in *Environment* A or *Environment* B. The reason that why the humidity does not improve the localization accuracy dramatically may be that common indoor area has air conditioner, which results in the humidity information having no big difference in different lab area.

To sum up, the humidity information is able to improve the localization accuracy. We believe that, if in an area where the humidity has a big difference at different places, the localization accuracy can be further improved.

4.6. Localization Accuracy. In this subsection, we utilize RSS and all the environmental factors (light intensity, temperature, and humidity) in localization. The accuracy of traditional method using just RSS is 1.96m, while our system is 1.19m, whose accuracy comparison is shown in Figure 8. The algorithm running result is shown in Figure 9. The blue stars with number are the real positions of the tracking node, and the red stars with number are the calculated positions by using our system.

4.7. Latency. The latency of our system mainly depends on the beacon interval of each node. In our experiment, in order

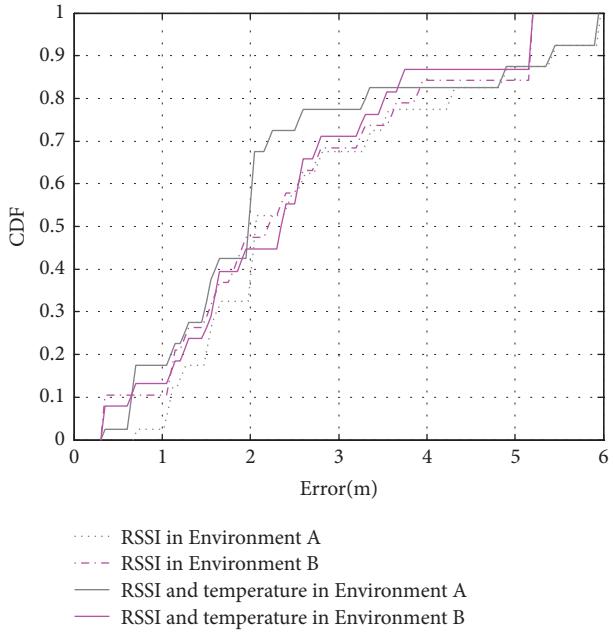


Figure 6: The impact of temperature.

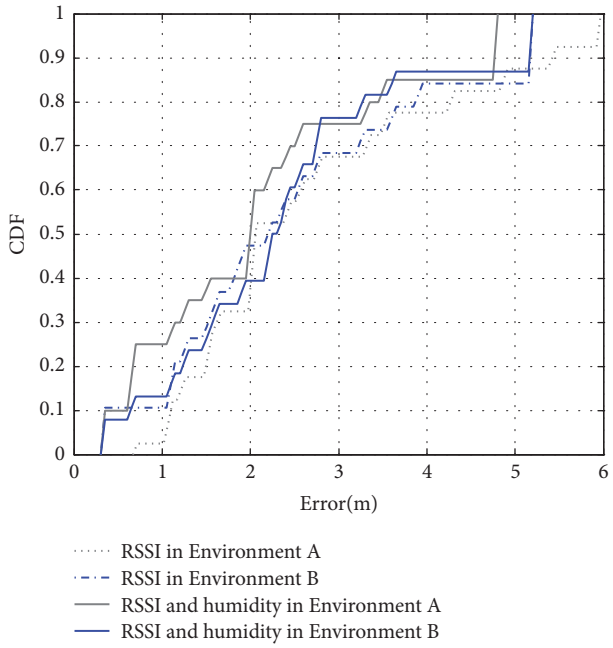


Figure 7: The impact of humidity.

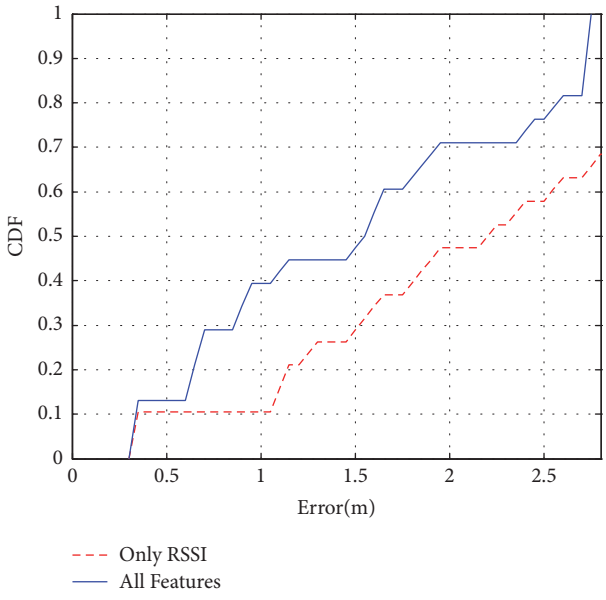


Figure 8: Accuracy comparison (CDF).

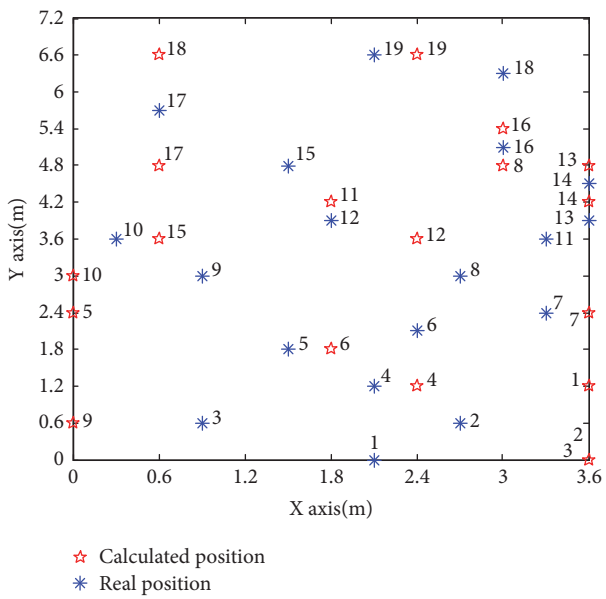


Figure 9: Localization result.

to avoid collision during data transmission, which causes data packets missing, we set the reference node's broadcast interval as 0.2*s* to transmit a packet with 51 bytes. The other time, e.g., algorithm running time in the server part, is neglected. Therefore, the latency is about 0.2*s*.

4.8. Discussion. Actually in our approach, how much the localization accuracy will be improved depends on different environments. Our experiments are performed in a relatively closed area. Regarding different factors, we have the following suggestions.

In such areas, the humidity of the whole area is relatively uniform, which results in the humidity feature having less impact on localization accuracy. We believe that if in a different large indoor complex environment, the improvement could be larger.

For the temperature feature, in our environment, the temperature close to the air conditioning is relatively low. We believe these features will contribute a lot to those closed environments where the air conditioning is always on.

Regarding light intensity feature during night in our environment, where artificial lights are used, the light intensity is different at different places. For example, a place close to a chair or under desk has a weak light intensity. Such feature will contribute a lot in an environment if the environment does not have relatively constant source of light. However, if in an open area which has changing light source, e.g., the sun light in the daytime, we suggest not using such feature in localization. It is also the reason why light intensity does not contribute too much during the daytime to our experiment, since our lab has many open windows.

To sum up, our approach gives a general solution to introducing more environment features in localization. How to choose these features in a specific environment depends on the detail condition in such environment. Users may choose those features which will contribute the most in localization. We believe our approach is able to potentially improve all the RSS-based localization technologies.

5. Conclusion and Future Work

In this paper, we have proposed a method, which can improve traditional RSS-based indoor localization accuracy by leveraging various environmental physical features, e.g., the light, temperature, and humidity information. By building a comprehensive fingerprint map for the above information, Naïve Bayesian algorithm is used to localize the target. We implement our system in two different environments based on wireless sensor networks. Experimental results show that, compared with the traditional indoor localization approach based on just RSS information, our method based on Naïve Bayesian can improve the localization accuracy by about 39%.

As future work, at first, we will try our approach in a more complicated large indoor area. Thus, the environmental features may vary a lot at different places, which may increase the localization accuracy. Furthermore, we may use higher precision light, temperature, and humidity sensors to do experiments to achieve higher accuracy. At last, we will try our system in a 3*D* area.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported in part by Shenzhen Peacock Talent Grant 827-000175 and China NSFC Grants 61202377 and U1301251. Dian Zhang is the corresponding author.

References

[1] M. Kotaru, K. Joshi, D. Bharadia et al., "SpotFi: Decimeter Level Localization Using WiFi," *Acm Sigcomm Computer Communication Review*, vol. 45, no. 5, pp. 269–282, 2015.

[2] D. Zhang, J. Ma, Q. Chen, and L. M. Ni, "An RF-based system for tracking transceiver-free objects," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '07)*, pp. 135–144, March 2006.

[3] X. Guo, D. Zhang, K. Wu, and L. M. Ni, "MODLoc: Localizing multiple objects in dynamic indoor environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2969–2980, 2014.

[4] C. Wu, Z. Yang, Y. Liu et al., "WILL: Wireless Indoor Localization without Site Survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.

[5] S. Sorour, Y. Lostanlen, S. Valaee et al., "Joint Indoor Localization and Radio Map Construction with Limited Deployment Load," *Mobile Computing IEEE Transactions*, vol. 14, no. 5, pp. 1031–1043, 2013.

[6] S.-H. Jung, B.-C. Moon, and D. Han, "Unsupervised learning for crowdsourced indoor localization in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2892–2906, 2016.

[7] R. Mautz, "Indoor Positioning Technologies," *Südwestdeutscher Verlag für Hochschulschriften*, 2012.

[8] D. Yang, H. Gonzalez-Banos, and L. Guibas, "Counting people in crowds with a real-time network of simple image sensors," *IEEE International Conference on Computer Vision*, vol. 1, pp. 122–129, 2003.

[9] Q. Cai and J. Aggarwal, "Automatic tracking of human motion in indoor scenes across multiple synchronized video streams," in *Proceedings of the International Conference on Computer Vision*, pp. 356–362, 2002.

[10] T. Kivimäki, T. Vuorela, P. Peltola, and J. Vanhala, "A review on device-free passive indoor positioning methods," *International Journal of Smart Home*, vol. 8, no. 1, pp. 71–94, 2014.

[11] M. Huber, F. Kamangar, and I. Chlamtac, "Indoor location tracking using RSSI readings from a single Wi-Fi access point," *Wireless Networks*, vol. 13, no. 2, pp. 221–235, 2007.

[12] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures," in *Proceedings of the 20th ACM Annual International Conference on Mobile Computing and Networking, MobiCom 2014*, pp. 617–628, USA, September 2014.

[13] J. S. Lee, Y. W. Su, and C. C. Shen, "A comparative study of wireless protocols: bluetooth, UWB, ZigBee, and Wi-Fi," in *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON '07)*, pp. 46–51, 2008.

[14] C. S. Jensen, H. Lu, and B. Yang, "Graph model based indoor tracking," in *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM '09)*, pp. 122–131, May 2009.

[15] C. Jensen S, H. Lu, and B. Yang, "Indexing the trajectories of moving objects in symbolic indoor space," in *Advances in Spatial and Temporal Databases*, vol. 5644 of *Lecture Notes in Computer Science*, pp. 208–227, Springer, Berlin, Germany, 2009.

[16] S. Feldmann, K. Kyamakya, A. Zapater et al., "An Indoor Bluetooth-Based Positioning System: Concept, Implementation and Experimental Evaluation," in *Proceedings of the International Conference on Wireless Networks, Icwn '03*, pp. 109–113, 2003.

[17] Y. Gu, L. Quan, F. Ren, and J. Li, "Fast indoor localization of smart hand-held devices using bluetooth," in *Proceedings of the 10th IEEE International Conference on Mobile Ad-Hoc and Sensor Networks, MSN '14*, pp. 186–194, 2015.

[18] J. S. Lee, Y. W. Su, and C. C. Shen, "A comparative study of wireless protocols: bluetooth, UWB, ZigBee, and Wi-Fi," in *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON '07)*, pp. 46–51, Taipei, Taiwan, November 2007.

[19] X. Ye, Y. Wang, W. Hu, L. Song, Z. Gu, and D. Li, "WarpMap: Accurate and Efficient Indoor Location by Dynamic Warping in Sequence-Type Radio-Map," in *Proceedings of the 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, London, United Kingdom, June 2016.

[20] M. N. Husen and S. Lee, "High Performance Indoor Location Wi-Fi Fingerprinting using Invariant Received Signal Strength," Husen2016High, pp. 1-6, 2016.

[21] Y. Shu, Y. Huang, J. Zhang et al., "Gradient-based fingerprinting for indoor localization and tracking," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2424–2433, 2016.

[22] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2017.

[23] "various indoor objects," http://en.wikipedia.org/wiki/Indoor_positioning_system.

[24] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 775–784, IEEE, Tel Aviv, Israel, March 2000.

[25] Y. Shu, P. Coue, Y. Huang, J. Zhang, P. Cheng, and J. Chen, "G-Loc: Indoor localization leveraging gradient-based fingerprint map," in *Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 129-130, Toronto, ON, Canada, April 2014.

[26] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 205–218, ACM, June 2005.

[27] http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf.telosB.

[28] K. Wu, J. Xiao, Y. Yi et al., "CSI-Based Indoor Localization," *IEEE Transactions on Parallel Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2013.

[29] K. Wu, J. Xiao, Y. Yi et al., "FILA: Fine-grained indoor localization," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, vol. 131, pp. 2210–2218, March 2012.

[30] J. Xiao, K. Wu, Y. Yi et al., "Pilot: passive device-free indoor localization using channel state information," in *Proceedings of*

the IEEE 33rd International Conference on Distributed Computing Systems (ICDCS '13), pp. 236–245, July 2013.

[31] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using WiFi signals," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom 2015*, pp. 90–102, France, September 2015.

[32] W. Wang, A. X. Liu, M. Shahzad et al., "Understanding and modeling of WiFi signal based human activity recognition," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom 2015*, pp. 65–76, September 2015.

[33] M. Altini, D. Brunelli, E. Farella, and L. Benini, "Bluetooth indoor localization with multiple neural networks," in *Proceedings of the 5th International Symposium on Wireless Pervasive Computing (ISWPC '10)*, pp. 295–300, IEEE, Modena, Italy, May 2010.

[34] Y. Wang, Q. Ye, J. Cheng, and L. Wang, "RSSI-Based Bluetooth Indoor Localization," in *Proceedings of the 11th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2015*, pp. 165–171, December 2015.

[35] J. Larranaga, L. Muguira, J. Lopez-Garde et al., "An environment adaptive ZigBee-based indoor positioning algorithm," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–8, 2010.

[36] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, IEEE IV 2013*, pp. 719–724, aus, June 2013.

[37] Y. Nam, "Map-based indoor people localization using an inertial measurement unit," *Journal of Information Science and Engineering*, vol. 27, no. 4, pp. 1233–1248, 2011.

[38] C.-H. Hsu and C.-H. Yu, "An Accelerometer based approach for indoor localization," in *Proceedings of the Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing in Conjunction with the UIC'09 and ATC'09 Conferences, UIC-ATC 2009*, pp. 223–227, aus, July 2009.

[39] J. Bird and D. Arden, "Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors," *IEEE Wireless Communications Magazine*, vol. 18, no. 2, pp. 28–35, 2011.

[40] H. Myung, H. Lee, K. Choi et al., "Mobile robot localization with gyroscope and constrained Kalman filter," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 442–447, 2010.

[41] Y.-S. Kuo, P. Pannuto, K. Hsiao et al., *Luxapose: Indoor Positioning with Mobile Phones And Visible Light*, 2014.

[42] M. Uddin and T. Nadeem, "SpyLoc: a light weight localization system for smartphones," *Fuzzy Sets and Systems*, vol. 103, no. 2, pp. 303–315, 2013.

[43] A. Milenković, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: issues and an implementation," *Computer Communications*, vol. 29, no. 13-14, pp. 2521–2533, 2006.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9809345, 10 pages
https://doi.org/10.1155/2018/9809345

Research Article

An Efficient Security System for Mobile Data Monitoring

Likun Liu, Hongli Zhang, Xiangzhan Yu, Yi Xin, Muhammad Shafiq, and Mengmeng Ge

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Xiangzhan Yu; yuxiangzhan@hit.edu.cn

Received 13 February 2018; Accepted 8 May 2018; Published 11 June 2018

Academic Editor: Lu Wang

During the last decade, rapid development of mobile devices and applications has produced a large number of mobile data which hide numerous cyber-attacks. To monitor the mobile data and detect the attacks, NIDS/NIPS plays important role for ISP and enterprise, but now it still faces two challenges, high performance for super large patterns and detection of the latest attacks. High performance is dominated by Deep Packet Inspection (DPI) mechanism, which is the core of security devices. A new TTL attack is just put forward to escape detecting, such that the adversary inserts packet with short TTL to escape from NIDS/NIPS. To address the above-mentioned problems, in this paper, we design a security system to handle the two aspects. For efficient DPI, a new two-step partition of pattern set is demonstrated and discussed, which includes first set-partition and second set-partition. For resisting TTL attacks, we set reasonable TTL threshold and patch TCP protocol stack to detect the attack. Compared with recent produced algorithm, our experiments show better performance and the throughput increased 27% when the number of patterns is $\mathbf{10^6}$. Moreover, the success rate of detection is 100%, and while attack intensity increased, the throughput decreased.

1. Introduction

More and more mobile data, including bad along with good, emerged and congested the network, which brings challenges to improve system performance and attack detection capabilities. In order to improve this situation, ISPs constantly update the security devices and systems, such as NIDS or NIPS which are the front line of defense against cyber-attacks. A central component of NIDS/NIPS is Deep Packet Inspection (DPI) engine, in which the payload of the packets is inspected to detect predefined signatures of malicious information [1]. The vulnerability is exposed while the set of patterns is getting bigger and bigger over 10^6, and pattern matching algorithm can be taken down because of costing too many resources that lie at the center of most DPI engines. Therefore, efficient pattern matching algorithms are challenge for high performance.

In order to overcome the high performance barrier, multicore architectures provide an opportunity to achieve high performance at a relatively low cost. For each core of a conventional multicore with the traditional processors, it is favourable to adapt coarse-gained parallelism [2]. Previous research focused on the partitions of pattern set and mapping the subsets on parallel processors (cores); therefore, the problem is transformed into a scheduling problem. All the works divide the same length patterns as a minimal subset and suppose different combinational algorithms. However, the ideal result is based on the number of same length patterns uniformly distributed. Similar to buckets, scheduling problem depends on the most time-consuming core. If a subset costs too much time, the whole matching is still slow, and the advantages of multicore will be lost.

And on the other hand, emerging new attacks bring new challenges to the existing security system, such as the recent TTL attack that the adversary inserts a spurious packet with wrong sequence number and short TTL, while the short TTL makes maximum probability of the crafting packet reach NIDS, rather than service provider. After TCP control block (TCB) maintained by NIDS receives manipulating packet, the status will be out of synchronization and the TCP stream will be teardown. At the moment, attacker successfully evades NIDS and sends bad data to server.

Table 1: Performance of AC, WM, and SBOM.

Algorithm	Time complexity	Data structure
AC	$O(n)$	Trie
WM	$O\left(\frac{n}{(m-b+1)\times\left(1-\left((m-b+1)\times r\right)/\left(2\times\lvert\Sigma\rvert^{b}\right)\right)}\right)$	Hash Table
SBOM	$O\left(\frac{n\times\log_{\lvert\Sigma\rvert}mr}{m-\log_{\lvert\Sigma\rvert}mr}\right)$	Factor Oracle

Σ denotes an alphabet set, b is the block size in WM, m denotes the minimum length of pattern, and r denotes the number of pattern sets.

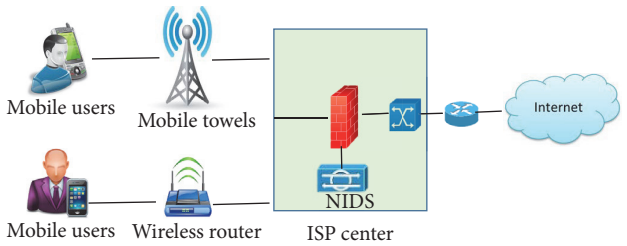


Figure 1: Security system deployment.

In this paper, the main contributions of this work are as follows.

(i) We designed an efficient security system for mobile data monitoring, which is deployed as a bypass to the mobile network, as shown in Figure 1. It is located at the ISP export. The system architecture will be introduced in Section 3.1.

(ii) We propose fined-gained parallel algorithm, a new two-step partition of pattern set. Firstly, we select a better partition from original set-partition and Tan's set-partition. Secondly, we make decision whether or not to split the uneven subsets, and standard deviation is taken as a measure. Finally, divide subsets and map all subsets on cores by AEA algorithm.

(iii) We design a table to record the hops between security system and TCP server, which contains TTL threshold and server hash (ip and port). The table is real-time update through learning TTL value from server packets. When $Server_{TTL} < Table_{TTL}$, the table will be updated. In case of $Client_{TTL} < Table_{TTL}$, and $Client_{seq} + datalen > TCB_{ack}$, drop packet and send alarm.

The paper is organized as follows: Section 2 presents the related work. Section 3 describes our system, fined-gained parallel algorithm, and a method to resist latest attack, followed by some experimental results in Section 4. Finally, we summarize the research in Section 5 with a discussion of the future work.

2. Related Work

For improving performance of NIDS, researchers endeavor to develop multi-core and many-core. Some researchers focused on the architecture, and other researchers devote themselves to the parallelization algorithm research of core component automata. The former pays more attention to system scheduling [3, 4] and cache optimization [5, 6]. The latter is dedicated to efficient pattern matching problem. In this paper, we mainly elaborate the research results of string matching parallelization algorithm.

Similarly, the previous work for string matching algorithm can be categorized into three classes: prefix searching, suffix searching, and factor searching. In prefix search methods, Aho-Corasick algorithm (AC) [7] is the most typified and efficient. The algorithm moved windows by computing the longest common prefix between the text and the patterns. In suffix search methods, the classic algorithm is Wu-Member algorithm (WM) [8], which features a backward searching from right to left within the window. In factor searching methods, the famous Set Backward Oracle Matching (SBOM) [9] is of this kind. It is treated as a combination of prefix searching and suffix searching. The performance of the above classical algorithms is determined by three factors: the size of the alphabet, the minimal length of the patterns, and the number of the patterns (see Table 1). AC performs well on short string, while WM is the opposite. SBOM is the most efficient in practice for long patterns.

The corresponding parallelization algorithm is subdivided into two directions: the text parallelization and pattern sets parallelization. The former cuts the text into multiple subtext which is sent to each core. Around cutting point, the combination of two adjacency subtext boundaries is a big nuisance. So far there is no better way, so the latter got more room for research. We divided the latter into two categories: bit-split and pattern-merge.

Bit-split is suitable for small-scale pattern sets. Hyun-Jin [10] proposed a memory-efficient bit-split deterministic finite automata- (DFAs-) based string matching scheme with multiple string matchers. The target is reducing memory requirements. He used the graph coloring of a unique graph to group iteratively patterns into multiple unique sets. Shervin et al. [11] introduced a pattern grouping algorithm for heterogeneous bit-split string matching architectures. The algorithm is composed of two phases: (1) a seed selection process, which uses a calculation to estimate the correlation between strings, and (2) a seed growing process for mapping strings onto subgroups.

Pattern-merge is applied to large-scale pattern sets; Liu et al. [12] proposed a shortest-path model for the optimal partition finding problem, which is suitable for filtering with the large-scale patterns set. In this approach, the patterns with same length would locate in one subset as a node. The

weight of the edge between any two nodes is the minimum runtime of AC, WM, and SBOM. The optimal partition is finding the shortest-path and consolidated subsets. Tan et al. [2] challenge Liu's optimal partition. They regarded processors as a factor of division and proved the optimal allocation of subsets to processors is NP-hard. $P_1, P_2 \dots P_l$ are subsets; $q_1, q_2 \dots q_s$ are processors. Tan considered two cases, $l \leq s$ and $l > s$, and designed two dynamic programming algorithms to get optimal partition. Two shortcomings are hidden in Tan's strategy. On the one hand, when $l > s$, a Greedy algorithm is applied to scheduling subsets into multicores, but it is easily trapped in the local optimization. A set-partition based genetic algorithm (GA) [13] is proposed to resolve the problem, but it is still easy to fall premature. On the other hand, if some subsets cost too much runtime, according to the strategy by Tan et al., there will be a phenomenon $\min \max_{i=1}^{q}\{T_i\} = \max_{i=1}^{q}\{T_i\}$; this means that the whole runtime always depends on a subset. In this case, neither shortest-path by Liu et al. nor dynamic programming by Tan et al. can achieve the optimal division.

We leverage the idea of Tan et al. [2] to develop a parallel multiple pattern matching algorithm. AEA algorithm, instead of Greedy algorithm, is adopted to schedule subsets and this algorithm can jump out of local optimization and avoid premature. Then standard deviation helps to make decision. On the assumption that standard deviation is greater than tolerable error θ, we further split subsets to satisfy uniformed distribution on each core. In this way, computation cost is minimized.

For attack-resistance, as reported by [14], attacker sent specially crafted packets, especially "insertion" packets. These insertion packets are crafted such that they are ignored by the intended server (nor never reach the server) but are accepted and processed by the NIDS. In insertion packet, the TTL and checksum are manipulated field—a packet with a short TTL value would never reach the intended server and a packet with wrong checksum would be discarded by the server. The basic principle is to destroy the TCB of NIDS. There are three attack points as follows.

(i) **TCB Creation**. A SYN packet, with fake sequence number and short TTL value, is injected while TCP builds a three-way handshake. The insertion packet was firstly sent, followed by a real SYN packet. The NIDS will ignore the real connection because of its "unexpected" sequence number.

(ii) **Data Reassembly**. There are two cases.

(1) Out-of-order data overlapping: for IP fragments, the only difference between insertion packet and real packet is that the former's data is filled with garbage. The insertion packet is sent prior to the real packet. For TCP segments, the order of sending is the opposite. For example, to be sent IP fragment packets are IP_1, IP_2, the sensitive words lie in IP_2, the attacker will insert IP_{2fake} with same offset and length as IP_2 and junk data, and the final order is IP_{2fake}, IP_2, IP_1.

(2) In-order data overlapping: before sending a real packet, a fake packet filled with same header except TTL and junk data is sent.

(iii) **TCB Teardown**. After finishing TCP three-way handshake, client can send a RST or FIN packet to NIDS; it will make TCB of NIDS teardown.

Zhang et al. [15] enhanced attack techniques and proposed two evasion strategies.

(iv) **Resync + Desync**. The sequence number of a SYN insertion packet is out of the expected receive window of the server. After three-way handshake, the client sends a SYN insertion packet. Subsequently, he sends a 1-byte data packet containing an out-of-window sequence number to desynchronize NIDS, and a real request followed.

(v) **TCB Reversal**. This situation has a prerequisite, which is connection established when NIDS receives SYN/ACK. The client will first send a SYN/ACK insertion packet. It will confuse client and server for NIDS.

Based on these technologies, Zhang et al. made further study and dug out a combination program of TCB Creation + Resync/Desync and TCB Teardown + TCB Reversal. Average successful evasion rate was 95.6% and 96.2%.

The above attack methods are mainly based on two aspects. (1) Insertion packet with small TTL or wrong checksum only reaches NIDS not intended server. (2) Some fields of insertion packet are filled with wrong data or garbage data to make TCB of NIDS teardown. The wrong checksum is always invalid because most NIDS will do check. Therefore, we study solutions to combat TTL attacks. The characteristic of the attack is small TTL value. We only need to find the right threshold. The choice of the threshold is crucial, but not easy because the proper threshold value varies from case to case and can be set by experience. We set it by learning from normal server packets. About the second aspect, we subtly design a small additional receive buffer to handle packets with same or unexpected sequence number. The corresponding TCP state machine will also be upgraded.

3. Efficient Security System

3.1. System Architecture. The security system consists of six modules: (1) traffic capture module, (2) protocol stack module, (3) DPI engine module, (4) Alarm/Log module, (5) set-partition module, and (6) attack detection. The first four modules are the original NIDS modules, while the last two are new modules. System architecture is shown as Figure 2. The analysis of each module is as follows.

(i) Traffic capture: traditional receiving packet is triggered by each packet interrupt; if the packet is too fast, the interrupt is too frequent; CPU always handles interrupt; thus other tasks cannot be scheduled, and the performance will reach the bottleneck. The DPDK technology we use in this system adopts the polling-mode drivers for networking, which makes them receive and send packets within the minimum number of CPU cycles (usually less than 80 cycles). Thus the throughput increases significantly.

(ii) Protocol stack module: this module decodes packets and analyzes protocols of transport and application layer.

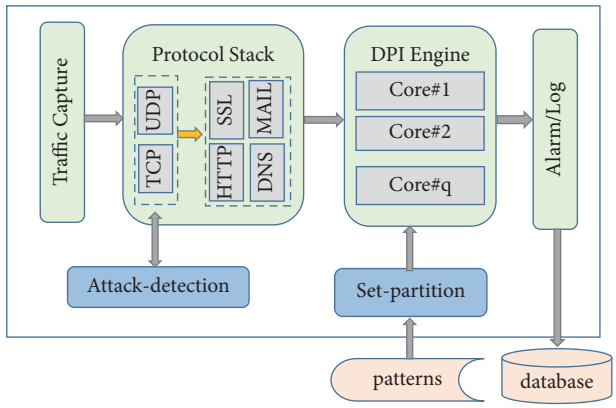


Figure 2: Security system architecture.

In this paper, we focus on TCP because it is the object of attack. TTL value of each TCP packet will be checked, that is, compared with the TTL record table (TRT) in attack detection module. An auxiliary small receive buffer is added to the TCP module for TCP reassembly attack detection.

(iii) DPI engine: DPI is the central component of security system. Its main function is to match the data from protocol stack module with the pattern set. Because this module consumes a large amount of CPU and memory, an efficient parallel matching algorithm is crucial. Each subset maps one automaton, and several automatons are assigned to the same core.

(iv) Alarm/Log: this is output module. If the matching is successful, the module will send an alarm message to the specific device and a log to database.

(v) Set-partition: this module is necessary for large-scale pattern set. An optimal partition algorithm is integrated in this module. Firstly, we get a partition by dynamic programming algorithm and calculate the average point of all the cores. Then AEA is used to schedule subsets to cores. Finally, if the runtime of each core is around the average point, the optimal partition is finished; else a new improved Greedy algorithm we proposed helps to subdivide a subset of the same length. See Section 3.2 for details.

(iv) Attack detection: this module provides common attack and the latest TTL attack detection interface. It can detect the following attacks: (1) common attack: ICMP flood, TCP SYN flood, TCP LAND, UDP flood, and ping of death; (2) the latest TTL attack: TCB Creation with SYN, reassembly out-of-order data, reassembly in-order data, TCB Teardown with RST, TCB Teardown with RST/ACK, TCB Teardown with FIN, TCB Creation + Resync/Desync, and TCB Teardown + TCB Reversal. A TTL table and a new TCP receive buffer aid in completing the detection. See Section 3.3 for details.

3.2. High Performance

3.2.1. Preliminary Knowledge. A set of patterns P, where P_i is a subset with same length, is

$$P = \{P_1, P_2, \ldots, P_n\} \tag{1}$$

We denote runtime of a subset as

$$T^i = \min\left\{T^i_{AC}, T^i_{WM}, T^i_{SBOM}\right\}, \quad 1 \le i \le n. \tag{2}$$

An optimal partition (Tan et al. [2]) using dynamic programming algorithm is

$$\begin{aligned} P_{dynamic} &= \{S_1, S_2, \ldots, S_k\}, \\ S_i &= \{P_r, \ldots P_s\}, \quad 1 \le r < s \le n \end{aligned} \tag{3}$$

Comparing runtime,

$$T_{P_{dynamic}} \le T_P \tag{4}$$

3.2.2. Optimal Patterns Set Decomposition and Schedule. The strategy for set-partition includes two steps:

(i) First choice of set-partition.

(ii) Second set-partition decision.

Step 1 (first choice of set-partition).

Problem 1. Given a set of patterns P and c cores, to find its decomposition $P_1, P_2, \ldots, P_l$, and scheduling of mapping subsets to cores,

$$\begin{aligned} \min \quad & \max_{i=1}^{c} \left\{ \sum_{j=1}^{n_i} T^i_j \right\} \\ \text{s.t.} \quad & P_i \cap P_j = \emptyset, \\ & \bigcup_{l}^{i=1} P_i = P, \\ & 1 \le i \ne j \le l. \end{aligned} \tag{5}$$

P_i is with same length, n_i is the number of pattern sets selected by core i, and the running time of each core is $\sum_{j=1}^{n_i} T^i_j$.

Problem 1 is a multiobjective combination problem, which is NP-hard. For the problem, we firstly divide the patterns into subsets and then schedule them. For first set-partition, we compare two methods: original set-partition and Tan's [2] dynamic programming algorithm. Before scheduling, $T_{P_{dynamic}} \le T_P$, but the result may be different after scheduling. We take an opposite example to prove it.

(1) Initialization set $P = \{P_1, P_2, P_3, P_4, P_5\}$

(2) Runtime $T_P = \{1, 1, 6, 7, 8\}$

(3) Tan's method $P_{dynamic} = \{S_1, S_2\} = \{\{P_1, P_2\}, \{P_3, P_4, P_5\}\}$

$$T_{P_{dynamic}} = \{2, 14\} \tag{6}$$

(4) Cores=2, Greedy algorithm to schedule

$$\begin{aligned} SCH_P &= \{\{8,1,1\},\{7,6\}\}, \\ \max_P &= 13 \\ SCH_{P_{dynamic}} &= \{2,14\}, \\ \max_{P_{dynamic}} &= 14 \end{aligned} \tag{7}$$

(5) $\max_{P_{dynamic}} > \max_P$

The result shows that Tan's method is not optimal; the reason is that the partition $P_{dynamic}$ is actually based on one core to get, which is not suffice to show that the multicore scheduling result using Greedy algorithm is optimal. We leverage the result to choose $\min\{\max_{P_{dynamic}}, \max_P\}$ as the first partition.

Schedule. Since Greedy algorithm is easier to fall into local optimization, we design annealing evolution algorithm (AEA) instead.

We set some notations as follows:

Pop: population;

G_{max}: evolution generations;

F: the fitness function.

P_{cross}: cross probability;

$P_{mutation}$: mutation probability;

ε: premature decision sign;

C: temperature coefficient of cooling;

T: annealing initial temperature;

T_{min}: lower temperature limit.

Functions are defined as follows:

(i) The fitness function F: the optimal set-partition problem is to find the maximum; we set fitness function as

$$f(x) = \frac{1}{1 + c - g(x)}, \quad c \geq 0,\ c - g(x) \geq 0 \tag{8}$$

where c is an estimate of the boundary of the objective function.

(ii) P_{cross} and $P_{mutation}$ have a great influence on the convergence of algorithm; thus we use the following formula for adaptive tuning. Assume the average of fitness function is $f'_{avr} = (1/T)\sum_{i=1}^{T} f'_i$, where f'_i is fitness function for individual i at the Gth generation and T is population size at the Gth generation. The fitness of the best individual is f'_{best}. All individuals whose fitness is less than f'_{best} are averaged to get f'_{avr}. Let $\varepsilon = f'_{best} - f'_{avr}$, as ε increased, the algorithm has a premature trend; in order to prevent it, we use the following functions:

$$\begin{aligned} P_{cross} &= 1 + a\frac{-1}{1 + e^{-k_1 * \varepsilon}} \\ P_{mutation} &= b + b\frac{-1}{1 + e^{-k_2 * \varepsilon}} \end{aligned} \tag{9}$$

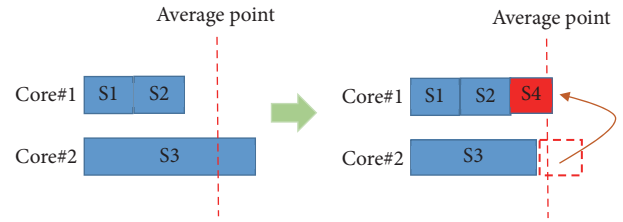


Figure 3: Second set-partition schematic.

where a = 0.4, b = 0.04, $k_1 > 0$, $k_2 > 0$, k_1 *and* k_2 are constant.

Pseudocode is shown in Algorithm 1.

AEA mixes GA algorithm and SA algorithm. GA is the main process. While GA converges, SA furthermore changes the initialization population to overcome the premature and GA runs again, until SA is convergent. Since GA can get optimal set-partition with high probabilities, SA should be quickly annealed, so the temperature coefficient of cooling should be chosen to be larger to speed up the convergence. Similarly, the parameter G_{max} of GA should be set larger too, because if it is small, SA may not be working or annealing times are few, the reasonable value should be near SA convergence point.

Step 2 (second set-partition decision). This part describes a metric that reflects the result of second set-partition decision and how to split subsets.

Measure: standard deviation: it is used to quantify the amount of variation or dispersion of a set of data values.

$$s = \sqrt{\frac{1}{c-1}\sum_{i=1}^{c}\left(T_{c_i} - \overline{T}\right)^2} \leq \theta, \quad 0 < i < c,\ 0 < \theta < 1 \tag{10}$$

where θ is tolerable error.

If $s > \theta$, it shows a large deviation; at this time, a second set-partition is necessary. According to the experiment, it is often caused by such subsets with long length, large size, and WM and SBOM algorithm. A Greedy strategy is applied to subdivide the subsets. As shown in Figure 3, in the core#2 with $\max_{i=1}^{c}\{T_i\}$, S_3 is cut into $\{S_3, S_4\}$ and the new S_4 is assigned to the core#1 with $\min_{i=1}^{c}\{T_i\}$. After the division and reorganization, we need to recalculate s and judge the convergence condition ($s \leq \theta$). If convergence condition is not satisfied, subdivision is continued.

For how to cut a subset, we choose the number of patterns as a measure and find it in relation to running time. We chose 20 sets of experimental data and adopt least-square method for data fitting. The result shows that the relation is linear; that is, the running time for WM and SBOM increases with the number increase, so we divide subset according to the ratio of time

$$N_{S_{new}} = \frac{T_{max} - \overline{T}}{T_{max}} N_S \tag{11}$$

(1) Initialize the $Pop_0 = N, P_{cross} = p_{c0}, P_{mutation} = p_{m0}$, $G = 1$
(2) FOR each generation t *in* M DO
(3) FOR each individual p_i *in* Pop_t DO
(4) calculate the fitness F_i
(5) ENDFOR
(6) IF $G \geq G_{max}$, THEN
(7) find the optimal set-partition.
(8) return
(9) ENDIF
(10) calculate ε_t, p_c, p_m
(11) IF $\varepsilon_t \geq \varepsilon$, THEN
(12) calculate p_{ct}, p_{mt}
(13) crossover operation with probability p_{ct}
(14) mutation operation with probability p_{mt}
(15) ELSE
(16) crossover operation with probability p_{c0}
(17) mutation operation with probability p_{m0}
(18) ENDIF
(19) generate next population Pop_{t+1}
(20) IF $pop_{t+1-n} = pop_{t+1}$, THEN
(21) switch (f'_{best}, f'_{best-1}) as new Pop_0
(22) annealing, $T = CT$, $t = 0$
(23) IF $T \leq T_{min}$
(24) stop and return set-partition.
(25) ENDIF
(26) GOTO 2
(27) ENDIF
(28) ENDFOR

Algorithm 1: Annealing evolution algorithm.

(1) calculate standard deviation s
(2) IF $s > \theta$ THEN
(3) find core i with $\max_{i=1}^{c}\{T_i\}$, core j with $\min_{j=1}^{c}\{T_j\}$
(4) find the subset S_k *with* $\max T_{c_i}^{k}$
(5) split $S_k = \{S_{k1}, S_{k2}\}$, move S_{k2} to core j
(6) recalculate T_i, T_j *and* $\overline{T}$
(7) GOTO 1
(8) ELSE
(9) find the optimal set-partition, return
(10) ENDIF

Algorithm 2: Greedy algorithm to split subsets.

where $N_{S_{new}}$ is the pattern number of new subset which will be moved to the core with T_{min} and N_S is the pattern number of the subset with $\max_{i=1}^{m}\{T_{max}^{i}\}$.

Pseudocode is shown in Algorithm 2.

3.3. Attack-Resistance. Attack detection module includes common attack detection and TTL attack detection. For common attack, which includes ICMP flood, TCP SYN flood, TCP LAND, UDP flood, and ping of death, we set thresholds to prevent attacks as other NIDS. TTL attack detection is the focus of our research. Many attacks against security system are borrowing TTL value and the package with short TTL will never reach the intended server but security system (see Figure 4).

Table 2

Server identify (hash with ip and port)	TTL threshold

Through the analysis of TTL attack characteristics, we assume that servers are credible and design a TTL table to record the minimum hops between security system and server. The record hops are TTL threshold to prevent attacks. The table fields are shown in Table 2.

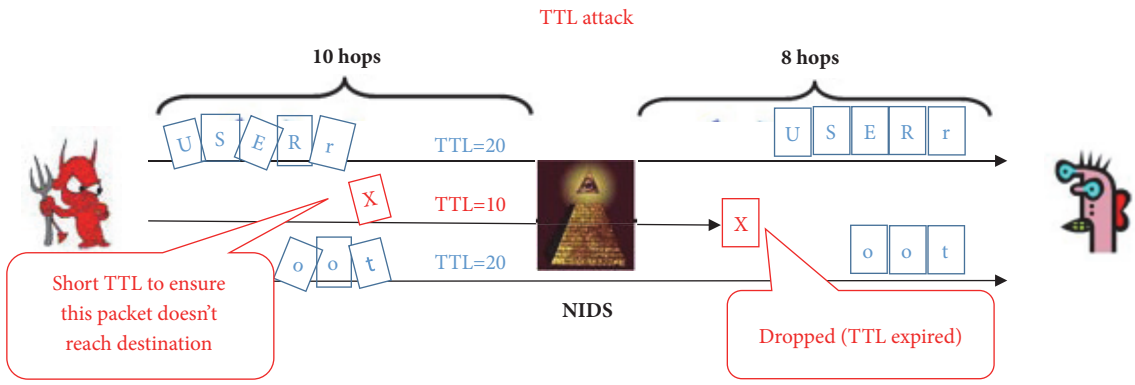


Figure 4: TTL attack.

Table 3: The relationship between defence type and attack method.

Control packet	TCB Creation, TCB Teardown, TCB Reversal, Resync + Desync
Data packet	Data Reassembly: out-of-order data overlapping and in-order data overlapping Data Desync

TTL threshold is learned from server packets. A packet is sent from different system, and the TTL value is different, but the common ground is that the initial value is 2^n, or $2^n - 1$, generally is one of $base[64, 128, 255]$. The security system computes hops as follows:

$$hops = \min_{i=1}^{3} \{|base_i - serverpkt_{TTL}|\} \quad (12)$$

All the IP packets from server need to check the table. If $hops < Table_{TTL}$, the table will be updated with $hops$. In case of $Client_{TTL} < Table_{TTL}$, we suspect that it is an attack packet, but this is not sufficient to prove it; there are two reasons:

(i) Dynamic changes in the network topology and the minimum hops between security system and server may be changed. If $hops_{real} > Table_{TTL}$, the attack packet with TTL value in $[Table_{TTL}, hops_{real}]$ will be missed.

(ii) In addition to setting short TTL, the attacks will operate other fields to complete attacks.

The latest TTL attack [15] combines short TTL with a variety of strategies. The heart of the strategies is to destroy or trick the TCP connection state of security system. To break the heart, we have defense in two aspects: control packet and data packet. Table 3 shows the relationship between the two aspects and attack methods.

Control packet: the attackers aim to destroy the TCB by sending insertion control packet. In order to tear the intention, we build a TCB link for same 4-tuple, so security system will not drop control packets while multiple SYNs or SYN/ACKs are coming. Each TCB has a timer which is updated by new packet. When time is out, the attacked TCB will be released, while the normal one is kept. In addition, a limited number of links are set to prevent continuous attacks. The TCB link will make TCB Creation, TCB Reversal, and Resync + Desync fail. For TCB Teardown, when insertion RST/FIN packet comes, the TCB will not be immediately released and reset the expiration time. During the time, if there is still coming packet, we can make sure it is an attack and keep the TCB.

Data packet: the essence of the attack is that the attacker generates junk data to protect sensitive data to escape security system. Attack techniques regard fraud as purpose, and attack features are forging data packet with same or incorrect control information. In this situation, security system can easily be misled and lose control of data. For same control information (Data Reassembly), we design an auxiliary small receive buffer for TCP reassembly attack detection. When the same sequence number packets come, the first is recorded in normal TCP receive buffer, and the second is recorded in the auxiliary buffer. We do not identify the authenticity of the data, because it costs too much time to parse protocol. TTL check should be a good helper to make decision that the one with short TTL will be abandoned. For incorrect control information (Data Desync), if the sequence number is out of window, there may be two situations: it is an attack packet or the system did not capture the previous packet. The packet is kept in auxiliary buffer and we determine which one by identifying the next sequence number from client and ACK number from server. If $Seq_{out} + Data_{out} = ACK_{server}$, this indicates that server has correctly received the packet, it is not an attack, and the window of security system will be updated. If $Seq_{next} + Data_{next} = ACK_{server}$, obviously, the next packet is right, and the one in auxiliary buffer is an attack packet.

In general, the security system detects the latest attack in two steps: (1) check TTL, (2) check control packet and control information in data packet. Experiments show that security system can effectively detect such attack.

Table 4: Contrast for original sets and Tan's subsets. Runtime is in milliseconds.

Thread Number	Original Greedy	Original AEA	Tan's Greedy	Tan's AEA
1	2955	2915	2935	2913
2	2902	2910	2890	2890
3	2893	2902	2833	2833
4	2870	2893	2810	2832

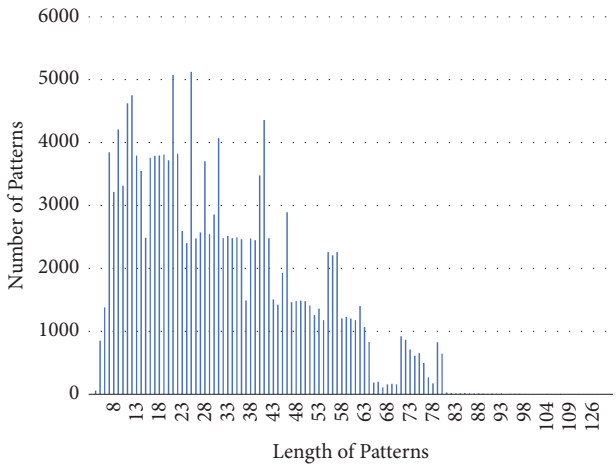


Figure 5: The number of patterns with different length.

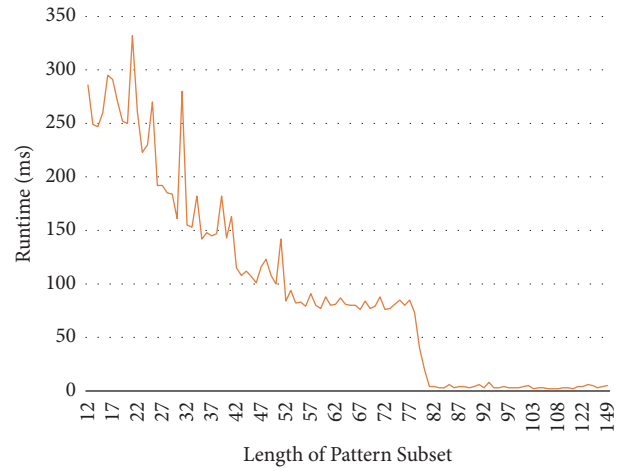


Figure 6: Runtime of each subset with the same length.

4. Experimental Results

4.1. Experimental Environment. We use a system with Intel Core(TM) i7-6700HQ CPU 2.60 GHz, quad-core, where each core has two hardware threads, 32KB L1 data cache (per core), 256KB L2 cache (per core), and 8MB L3 cache (shared among cores), 16G Memory. The system runs Linux CentOS 7, and DPDK is installed to capture packets. Our program runs with 8 threads in parallel, one thread is used to capture packet and preprocess, three threads are used to parse in parallel the application protocol, the remaining four threads work for parallel matching. In addition, we have prepared two windows computers for sending packets, which install CSNAS tool.

The pattern set consists of 10^6 URL strings; there are four sources: (1) extracting from the Snort rules, (2) selecting from URL blacklist, (3) parsing URL of traffic which is captured at the export of the HIT network center, and (4) generating URL randomly according to the length ratio from URL of traffic. Duplicate strings were eliminated. The input text consists of 5×10^5 strings; half is randomly extracted from the pattern set and another half is randomly generated. As Figure 5 shows, the number of patterns is not uniformly distributed. There are 115 different lengths, the shortest is 3, and the longest is 149.

4.2. High Performance Experiment. In this section, all the runtimes represent the algorithm execution time and the input text is introduced in Section 4.1. We first divide the pattern set into subsets by length and choose the best runtime from AC, WM, and SBOM for each subset. The experiment shows that when the length > 10, WM and SBOM performs more efficiently. Figure 6 shows the runtime of each subset with same length; while being combined with Figure 5, it can be drawn that the larger the subset, the longer the runtime.

After dynamic programming, the number of subsets is 21. Then we schedule them into four threads, respectively, by the Greedy algorithm and the AEA algorithm.

According to Table 4, we can see Tan's algorithm yielded better results. We analyze that there are two reasons: (1) in Tan's algorithm, the pattern with length less than 10 is merged into a subset, which is using AC. For small subset, the runtime of AC is not affected by the number of patterns. (2) The number is very small for length greater than 78, and after merging these subsets, the number of hash tables is reduced from 41 pairs to 1 pair that reduce memory usage. In contract to the Greedy and AEA, no matter original subsets or Tan's subsets, AEA is prior to Greedy.

After first set-partition, we will conduct second set-partition experiment. We choose Tan's AEA as input. The results are shown in Table 5; it is obvious that the second set-partition is relatively uniform, and the max runtime is less than the first.

Above we have tested the matching effect of parallel algorithm, then we will start the traffic test. The traffic is captured at the export of the HIT network centre, and we got two files in size 4.2G, 5.1G. Two windows computers are used to send packets. We chose CSNAS tool to send packets and the

Table 5: Contrast for first and second set-partition. Runtime is in milliseconds.

Thread Number	1	2	3	4
First	2913	2890	2833	2832
Second	2875	2868	2864	2861

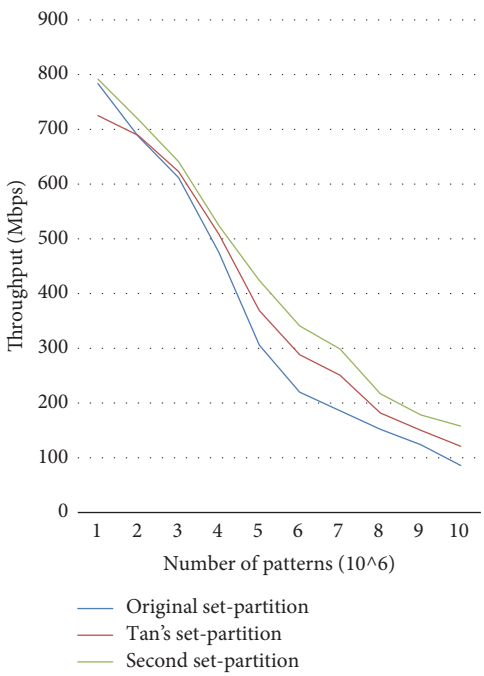


Figure 7: Throughput for different set-partitions.

Table 6: Throughput for different attack intensity.

%	10	20	30	50	80	100
Throughput (Mbps)	154	147	136	96	79	72

speed of sending is limited at 300-400Mbps, and the mode is set cycle.

The experiment compared the throughput of original set-partition, Tan's set-partition, and two-step set-partition, as shown in Figure 7. As the pattern set increases, the throughput of all methods decreases. The second set-partition we proposed performs best, such that the throughput increased 27% than Tan's when the number of patterns is 10^6.

4.3. Attack Detection Experiment. We constructed a total 700 of TCP attack streams; the method is randomly extracted from two capture files and inserted the attack packet. There are 100 for each attack type. The third computer is used to send attack packets with CSNAS tool. Attack intensity is adjustable by limiting the sending speed of three computers.

The result of attack detection experiment is shown in Table 6, and the success rate of detection is 100%, as attack intensity increases, the throughput decreases.

5. Conclusions

In order to monitor the mobile data in ISP or enterprise, we design a security system to detect malicious information and attacks. We demonstrate the architecture of the entire system and give solutions of high performance and anti-attack. For high performance, we propose a new two-step partition of pattern set. In first set-partition, the best set-partition is chosen from several algorithms, and AEA schedule algorithm replaces Greedy algorithm. In second set-partition, standard deviation is taken as a measure, in order to make each core with similar running time, we propose an equilibrium cut strategy to reorganize subsets. For anti-attack, the security system is based on TTL check, TCB link is added against control packet attacks, and an auxiliary small receive buffer is added against data packet fraud. In the final stage, we, respectively, perform our experiments from the above two aspects and the results show that the security system has high performance and anti-attack capacity.

In the future, we plan to push forward the work in two aspects: on one hand, we will focus on resilience to algorithmic complexity attacks. On the other hand, according to the detected attacks, we will focus on mining the intention of the attackers and classifying the attackers.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Key Research and Development Program of China with Grant no. 2016QY05X1000 and no. 2016QY01W0100.

References

[1] Y. Afek, A. Bremler-Barr, Y. Harchol, D. Hay, and Y. Koral, "Making DPI Engines Resilient to Algorithmic Complexity Attacks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3262–3275, 2016.

[2] G.-M. Tan, P. Liu, D.-B. Bu, and Y.-B. Liu, "Revisiting multiple pattern matching algorithms for multi-core architecture," *Journal of Computer Science and Technology*, vol. 26, no. 5, pp. 866–874, 2011.

[3] H. Jiang, G. Zhang, G. Xie, K. Salamatian, and L. Mathy, "Scalable high-performance parallel design for Network Intrusion Detection Systems on many-core processors," in *Proceedings of the 9th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS 2013*, pp. 137–146, San Jose, CA, USA, October 2013.

[4] J. Haiyang, X. Gaogang, and S. Kave, "Load Balancing by Ruleset Partition for Parallel IDS on Muti-Core Processors," in *Proceedings of the 22th International Conference on Computer Communications and Networks*, Nassau, Bahamas, 2013.

[5] M. Jamshed, J. Lee, S. Moon et al., "Kargus: A highly-scalable software-based intrusion detection system," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS 2012*, pp. 317–328, NC, USA, October 2012.

[6] J. Nam, M. Jamshed, B. Choi, D. Han, and K. Park, "Scaling the performance of network intrusion detection with many-core processors," in *Proceedings of the 2015 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 191-192, Oakland, CA, USA, May 2015.

[7] A. V. Aho and M. J. Corasick, "Efficient string matching: an aid to bibliographic search," *Communications of the ACM*, vol. 18, pp. 333–340, 1975.

[8] S. Wu and U. Manber, "A fast algorithm for multi-pattern searching," Technical Report TR-94-17, 1994.

[9] C. W. Beate, "A string matching algorithm fast on the average," in *Proceedings of the 6th Colloquium on Automata, Languages and Programmin*, pp. 118–132, Graz, Austria.

[10] H. Kim, "Memory-efficient parallel string matching scheme using distributed pattern grouping without matching vectors," *IEEE Electronics Letters*, vol. 52, no. 13, pp. 1124–1126, 2016.

[11] S. Vakili, J. P. Langlois, B. Boughzala, and Y. Savaria, "Memory-Efficient String Matching for Intrusion Detection Systems using a High-Precision Pattern Grouping Algorithm," in *Proceedings of the the 2016 Symposium*, pp. 37–42, Santa Clara, California, USA, March 2016.

[12] L. Ping, L. Yan, and T. Jian, "A partition-based efficient algorithm for large scale multiple-strings matching," in *In Proceedings of the 12th ACM Internet Conference. String Processing and Information Retrieval*, pp. 399–404, Buenos Aires,Argentina, 2005.

[13] J. Liu, F. Li, and G. Sun, "A parallel algorithm of multiple string matching based on set-partition in multi-core architecture," *International Journal of Security and Its Applications*, vol. 10, no. 4, pp. 267–278, 2016.

[14] K. Sheharbano, J. Mobin, and D. A. V. Philip, "Towards Illuminating a Censorship Monitors Model to Faicilitate Evasion," in *Proceedings of the In Proceedings of the 3th USENIX Workshop on Free and Open Communications on the Internet*, Washington, D.C. USA, 2013.

[15] W. Zhongjie, C. Yue, and Q. C. S. Zhiyun, "A Closer Look at Evading Stateful Internet Censorship," in *In Proceedings of the 17th ACM Internet Measurement Conference*, London, UK, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9517942, 12 pages
https://doi.org/10.1155/2018/9517942

Research Article

A Novel Indoor Localization Algorithm for Efficient Mobility Management in Wireless Networks

Yalong Xiao,[1] Shigeng Zhang ,[2] Jianxin Wang ,[2] and Chengzhang Zhu[1]

[1]*College of Literature and Journalism, Central South University, Changsha, China*
[2]*School of Information Science and Engineering, Central South University, Changsha, China*

Correspondence should be addressed to Shigeng Zhang; sgzhang@csu.edu.cn

Received 6 February 2018; Accepted 26 April 2018; Published 3 June 2018

Academic Editor: Wei Wang

Along with the penetration of smart devices and mobile applications in our daily life, how to effectively manage the mobility issues in wireless networks becomes a challenging task. The ability to continuously and accurately track the target object's position plays a vital role in mobility management. In this paper, we propose a novel indoor localization algorithm that fuses multiple signal features as the location fingerprints. The rationale that motivates our algorithm design stems from the following observation: although using one special signal feature (e.g., channel state information (CSI)) might achieve statistically higher accuracy than using another signal feature (e.g., received signal strength (RSS)), the accuracy for individual position estimations is usually diversified when only one signal feature is used in localization. For example, using RSS can obtain more accurate location estimation than using CSI for some individual positions. Thus, we propose a novel indoor localization algorithm that fuses multiple types of signal features as fingerprint of positions, which can effectively improve localization accuracy. We designed several fusion schemes and evaluated their performance. Experiments show that our algorithm achieves localization error below 0.5m and 1.1m in two typical indoor environments, about 30% lower than the accuracy of algorithms by fusing multiple signal features.

1. Introduction

With the penetration of mobile devices and smart applications [1], how to provide efficient mobility infrastructure and how to effectively manage mobility issues in future networks become more and more challenging [2–8]. For example, we need to provide flexible mobility infrastructure to support real-time multimedia applications and on-demand services in vehicular networks. The current Long Term Evaluation (LTE) 4G networks [9] and the future 5G [10] networks require smart mobility management schemes to handle the mobility of even millions of mobile terminals. Mobility infrastructure and mobility management are also critical in the new emerging computing paradigms like mobile cloud computing [6] and fog computing [11].

How to accurately estimate the position of a target object (e.g., mobile device or a person) plays a vital, important role in efficient mobility management. In recent years, along with the rapid development of location-based services (LBSs), accurate positioning techniques are required in many fields including travel guidance, mobile advertising, and urban computing. Many LBSs require knowing the positions of target objects in indoor environments. Indoor localization has received much research attention in recent years [12–17]. Among others, due to the pervasive penetration of wireless local area networks (WLANs) and Wi-Fi enabled mobile terminals, the fingerprint-based indoor positioning technology [18] has attracted much research attention in both academic and industry communities. Most existing works utilize received signal strength (RSS) [14] or channel state information (CSI) [12, 19] as the fingerprint of a particular position. Unlike RSS that is an aggregated value of the all subcarriers' amplitudes, CSI estimates the channel on each subcarrier in the frequency domain. Thus it can depict the multipath propagation to some extent and provide more stable and fine-grained signature in distinguishing different positions [20].

Many localization methods utilize single type of signal feature as location fingerprint, which may not well deal with the instability of signal features caused by mobility. We need to combine multiple types of signal features (e.g., RSS and CSI) to enhance the robustness and reliability of positioning and support more flexible mobility management. Moreover, we find that it is not the case that for all positions CSI-based approaches generate more accurate results than RSS-based approaches, although the overall statistical positioning performance of CSI-based method is usually better than that of RSS-based approaches [13, 19]. We conduct some experiments and find that, for a nonnegligible fraction of positions, the Horus algorithm [14], which is a representative RSS-based fingerprint approach, outputs more accurate location estimations than FIFS [19], a representative CSI-based fingerprint approach.

The above observation inspired us to design new localization algorithms to achieve higher localization accuracy by fusing different signal features as the position's fingerprint. In particular, we combine the results of Horus [14], FIFS [19], and D-CSI [15], which use different features as fingerprints of positions, to achieve higher accuracy than using each single feature. D-CSI [15] is a new positioning approach which uses the distribution of CSI amplitude as location fingerprints called Hp. It achieves higher accuracy than FIFS because it uses distribution of CSI amplitude as position fingerprints, which contains both spacial-diversity and frequency diversity of the signal, while FIFS only simply adds up all the subcarriers' amplitudes and uses them as position fingerprints called He.

In this paper, based on our previous work [21], we propose a hybrid method with multiple feature fusion (MFF) to counteract the positioning error of single feature based approaches. We first propose Fusion 1, a weighted fusion localization method simply combining the results of Horus and FIFS which can quickly obtain the location of mobile user. In order to further improve the positioning accuracy, we propose Fusion 2 which merges three different features as position fingerprints, namely, RSS, He, and Hp. Specifically, we first get a set of reference points called alternative reference points by running Horus, FIFS, and D-CSI, respectively. Secondly, we select three most possible candidate positions from the generated reference points of the three approaches. When there are more than three reference points, we use a minimal-triangle principle to select three out of them. Finally, we calculate a weighted centroid of the three reference points and take it as the target location. Experiments show that the proposed localization approach achieves median error of 0.5m and 1.1m in two typical indoor environments, significantly less than that of the best single feature based approach whose corresponding error is 0.7m and 1.3m, respectively. In average, our approach can reduce localization error by a factor of around 30% by feature fusing.

The rest of this paper is organized as follows. Section 2 reviews related work. In Section 3 the proposed method is described in detail with analysis. The simulation results are given and discussed in Section 4. Finally, Section 5 concludes this paper.

2. Related Work

2.1. Network Mobility Management. In recent years, mobility management has been studied in many types of networks, including IP and future networks [3, 22–25], 4G and 5G networks [2, 4, 10, 26], wireless sensor networks [5, 27–32], and mobile cloud computing [6, 11]. In [3], the authors investigate handover and mobility issues for IP-based multimedia services and applications. In [22] the authors gave a comprehensive analysis and comparison of different distributed mobility management (DMM) schemes. Evolution of sink mobility in wireless sensor networks is investigated in depth in [28]. The authors classified mobility management schemes in WSNs into four categories: uncontrollable mobility (UMM), path-restricted mobility (PRM), location-restricted mobility (LRR), and unrestricted mobility (URM). The authors analyze advantages of different schemes and compare their performance via simulation. In [22], the authors propose a self-organizing and adaptive Dynamic Clustering algorithm to perform efficient data gathering in WSNs with a mobile element.

The mobility infrastructure and mobility management in LTE 4G networks and 5G networks have also been widely studied recently. In [4] the authors discussed how to implement a decentralized LTE network and design a novel mobility management approach to support mobile IP. In [2] the authors discuss how to efficiently manage resource and handle mobility issues in dense 5G networks. In [10], the authors propose a memory-full context-aware mobility management algorithm that can predict the mobility of devices to achieve fast network handover. In [6, 11] the authors summarized latest research progress in new emerging computing paradigms like cloud computing and fog computing.

2.2. Indoor Localization. Indoor localization has attracted a growing research attention and various techniques have been proposed, including Wi-Fi [14, 33, 34], Bluetooth [35], radiofrequency identification (RFID) [7, 36], FM radio [37], acoustic signals [38], magnetic field [39], UWB [40], and light [41]. Among these signals, the use of Wi-Fi signal has attracted continuous attention due to the pervasive deployment of WLANs and Wi-Fi enabled mobile devices. Many efforts have been done to improve accuracy of WLAN based localization approaches.

The indoor localization system RADAR [33] is a pioneer work in WLAN fingerprinting, which used the K-nearest neighbor method to get the location of a person and achieved a median error of about 5m. Horus [14] used a maximum likelihood based approach to infer the target position and achieves higher accuracy than RADAR. Besides these two typical RSS-based fingerprint location approaches, there are many improvements over them. More RSS-based indoor localization approaches can be found in the literature review [42].

The signals of WLAN based localization contain RSSI and CSI. Compared to using RSSI as fingerprints, channel state information is considered as a finer grained signature to improve the localization accuracy. Wu et al. proposed a

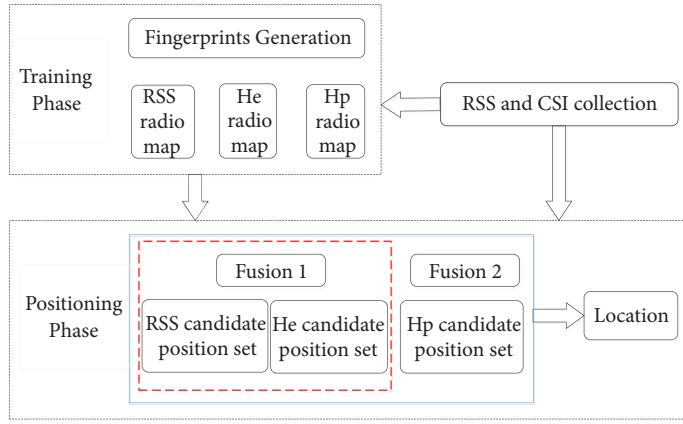


Figure 1: System architecture.

fingerprint-based indoor location system called FIFS [13], which used the CSI values obtained from the physical layer. The authors collected and proceeded with the CSI values as fingerprint by leveraging both the frequency diversity and spatial diversity to generate the radio map. Specifically, FIFS used the aggregated CSI amplitude values over all the subcarriers and leverages the spatial diversity to improve the performance of the RSSI-based method. The location fingerprint, which is aggregated over the all subcarriers, is a coarse metric and may not effectively distinguish the target position among different locations.

Xiao et al. [15] proposed a D-CSI system that makes better use of the frequency diversity with different subcarriers and the spatial diversity with multiple antennas and thus effectively improves the localization accuracy. The Kullback-Leibler divergence is used to calculate the similarity between different fingerprints, based on which the best matched position is calculated in the localization phase. Sen et al. [43] proposed a PinLoc system that utilizes the per-subcarrier frequency response as the feature of a location and relies on the machine learning algorithms to classify a device measurement to one of the trained locations. It leverages the frequency diversity but does not consider the spatial diversity. Wang et al. [17] presented DeepFi which is a deep learning based indoor fingerprinting scheme using CSI information. Although these techniques achieve a high localization precision, they require intensive computations and more training samples to localize the mobile users via machine learning or deep learning.

3. System Description

The CSI-based approaches usually achieve higher accuracy than the RSS-based approaches, but we find that the former may have lower accuracy than the latter in some positions. Therefore, the localization accuracy can be improved by integrating the methods that are based on different physical measurement.

In order to ensure the timeliness of localization results, we first give a method that leverages the weighted fusion of RSS and He. Then, adding the Hp feature, a localization method of fusing three features is proposed to further improve the positioning accuracy.

3.1. System Framework. There are two phases in fingerprint-based positioning system: the training phase and the positioning phase. Figure 1 shows the system architecture. In the training phase, we acquire physical measurements of wireless signals for calibration points, including both RSS and CSI. Then the features of He [19] and Hp [15] are extracted from the CSIs, together with the RSS as the location feature. Finally, we use the position information and the location feature of the calibration points to construct the fingerprint databases of RSS, He, and Hp, respectively.

In the positioning phase, RSS and CSI values are firstly collected at an unknown location. They are sent to the positioning engine and processed using the method as in the training phase to extract the fingerprint. In terms of positioning timeliness and accuracy, two methods are proposed to get the location of the mobile user. One fuses RSS and He features, called Fusion 1; the other one fuses RSS, He, and Hp features, called Fusion 2.

For each fingerprint feature, Fusion 1 finds the location result of the mobile user. Specifically, we feed out the RSS and He to Horus and FIFS, to calculate the candidate position for each approach. The position of the mobile user is obtained by weighted fusion of the two candidate results.

Adding the Hp feature, Fusion 2 finds three candidate reference points whose fingerprints are mostly close to the fingerprint of the unknown position. Specifically, we feed out the RSS, He, and Hp to Horus, FIFS, and D-CSI, respectively, and calculate the candidate position set for each approach. Three most possible candidate points are then selected from these positions according to a newly defined metric called confidence degree. Finally, the weighted centroid of the three selected candidate points is calculated and used as the position estimation.

3.2. Algorithm Details. The framework of MFF is given in Algorithm 1. Assume that there are N APs and M reference points in the region. The positions of the reference points

Input: The RSS and CSI values of the test point. The coordinates, RSS and CSI values of the calibration points.
Output: The coordinates of the test point
(1) According to the coordinate, RSS and CSI values of the calibration points to build RSS, He and Hp fingerprint database. //Step 1
(2) In the case of timeliness of positioning results, according to RSS and CSI values collected at the test point and the fingerprint databases built in step 1, Fusion1 feeds out the RSS and He to Horus and FIFS, then calculates the candidate position for each approach. The position of the test point is obtained by weighted fusion of the two candidate results and the coordinates of the test point is returned. //Step 2
(3) In the case of high positioning accuracy scenario, Fusion2 uses KL distance as the similarity metric and calculate candidate calibration point set under D-CSI. Merge the candidate reference points which are obtained by Horus, FIFS and D-CSI approaches and calculate the degree of each candidate calibration point. //Step 3
(4) Depending on the size of each calibration point degree obtained in step 3, Fusion2 optimizes the final location of the test point and returns the coordinates of the test point. //Step 4

Algorithm 1: The algorithm of MFF.

are denoted as $L = \{L_1, L_2, \cdots, L_M\}$. The algorithm contains 4 steps and the detailed operations in each step will be described as follows.

Step 1. Establish the RSS, He, and Hp fingerprint database.

Given the coordinates of the reference points and corresponding RSS and CSI values, we build the RSS, He, and Hp fingerprint database. Taking RSS as an example, the process of the fingerprint database construction is as follows. We denote the RSS vector of the M reference points as $F\_RSS = [F\_RSS_1, F\_RSS_2, \cdots, F\_RSS_M]$. The RSS vector of the ith reference point is $F\_RSS_i = [RSS_1^i, RSS_2^i, \cdots, RSS_N^i]$, where RSS_N^i represents the RSS of the ith reference point from the Nth AP. Then, F_RSS and L constitute the RSS fingerprint database. The features He and Hp can be obtained by FIFS and D-CSI approaches. Thus, F_He, F_Hp, and L constitute the He and Hp fingerprint database.

Step 2. Fusion 1 calculates the coordinates of the test point.

We denote $T\_RSS = [RSS_1, RSS_2, \cdots, RSS_N]$ as the RSS obtained by the mobile device at an unknown location. Then the distance between the point and the reference point is $D\_RSS_i = sqrt(\sum_{j=1}^{N} |RSS_j - RSS_j^i|^2)$, where i=1, 2,$\cdots$, M. A smaller D_RSS_i represents a shorter distance between the unknown position and the reference point. We select three minimum values from D_RSS_i and calculate the location of the test point under Horus method, denoted as $l_{rss} = (x_{rss}, y_{rss})$.

We denoted $T\_He = [He_1, He_2, \cdots, He_N]$ as He preprocessed under the CSI. The distance between the test point and the reference points can also be measured by the Euclidean distance, $D\_He_i = sqrt(\sum_{j=1}^{N} |He_j - He_j^i|^2)$, where i=1, 2,$\cdots$, M. We also select three minimum values from D_He_i and calculate the location of the test point under FIFS method, denoted as $l_{he} = (x_{he}, y_{he})$.

The position of the test point is calculated by $L = w_{rss} * l_{rss} + w_{he} * l_{he}$, $w_{rss} + w_{he} = 1$, where w_{rss} and w_{he} are the weight of the location result under Horus and FIFS.

Step 3. Fusion 2 calculates and fuses the candidate reference points.

We denoted $T\_Hp = [Hp_1, Hp_2, \cdots, Hp_N]$ as the Hp preprocessed under the CSI. Using symmetric KL distance, the distance D_Hp between each reference point and the test point is calculated under D-CSI method. We sort D_RSS, D_He, and D_Hp in the ascending order and select the first three reference points as the candidate positions for RSS, He, and Hp, denoted by $Node\_RSS = [n\_RSS_1, n\_RSS_2, n\_RSS_3]$, $Node\_He = [n\_He_1, n\_He_2, n\_He_3]$, and $Node\_Hp = [n\_Hp_1, n\_Hp_2, n\_Hp_3]$, respectively.

We put all points in $Node\_RSS$, $Node\_He$, and $Node\_Hp$ into a position set called $Node\_all$ and calculate the degree of each candidate point as follows. For each candidate point, we draw a circle with radius R (whose optimal value will be determined in Section 4) and count the number of reference points in $Node\_all$ that fall in the circle. The number is defined as the degree of that candidate point. When calculating the degree, if the reference point is at the area boundary, its degree is increased by 0.5 to compensate for the boundary effect. After calculating degree for all points in $Node\_all$, we sort them in descending order according to their degree and denote the sorted set as $sort\_D$.

Step 4. Fusion 2 performs position calculation.

The final position calculation and optimization are shown in Algorithm 2.

3.3. Algorithm Analysis. Assume that the reference points are evenly distributed in the monitoring area in a grid pattern with grid space a. When R is less than a, the degree of each reference point will be zero because no other reference point will fall within the circle. When $a \le R < \sqrt{2}a$, the alternative reference point adjacent to the reference point will fall into a circle with radius R. When $\sqrt{2}a \le R < 2a$, an alternative reference point that is either adjacent to the reference point or diagonally opposite to the rectangle falls within a circle of radius R. Therefore, based on the above analysis, in the next

Input: *Node_all*, *sort_D*, *R*, *offset*
Output: The coordinates of the target point (x_r, y_r)
(1) Calculate the number of the reference points with the degree of *sort_D*(1) and *sort_D*(2), denoted as *N*1 and *N*2, respectively.
(2) **if** *N*1=1, **then**
(3) **if** $N2 \geq 2$, **then**
(4) Find the set $R = \{r_1, r_2, \cdots, r_m\}$, where r_i is a reference point of degree *sort_D*(2) and *m* is the number of the reference points of degree *sort_D*(2) in *Node_all*.
(5) Obtain the all candidate set $C = \{c_1, c_2, \cdots, c_l\}$, where c_i is a candidate set which contains three reference points. One is the reference point of degree *sort_D*(1), the others are any two points in the *R*.
(6) **else**
(7) Select the reference points with degree *sort_D*(1), *sort_D*(2) and *sort_D*(3) as the candidate set *C*.
(8) **else**
(9) **if** $N1 = 2$, **then**
(10) Find the set $R = \{r_1, r_2, \cdots, r_m\}$, where r_i is a reference point of degree *sort_D*(3) and *m* is the number of the reference points of degree *sort_D*(3) in *Node_all*.
(11) Obtain the all candidate set $C = \{c_1, c_2, \cdots, c_l\}$, where $c_i$is a candidate set which contains three reference points. One is the reference point of degree *sort_D*(3) in *R*, the others are two points with degree of *sort_D*(1).
(12) **else**
(13) The $R + offset$ will be used as the radius and recalculate the degrees of each reference point according to the procedure in step 3, the repeat the operation of steps 1 to 11.
(14) **if** $N1 > 2$, **then**
(15) Find all the reference points of *sort_D*(1), three of three combinations as the candidate set *C*.
(16) According to all the candidate set *C*, calculate the perimeter of the triangle enclosed by the reference point in the set, select the set of reference points with the shortest perimeter, and find the center of the triangle as the coordinate of the point to be measured.
(17) return result.

Algorithm 2: Optimization of the final location of the test point.

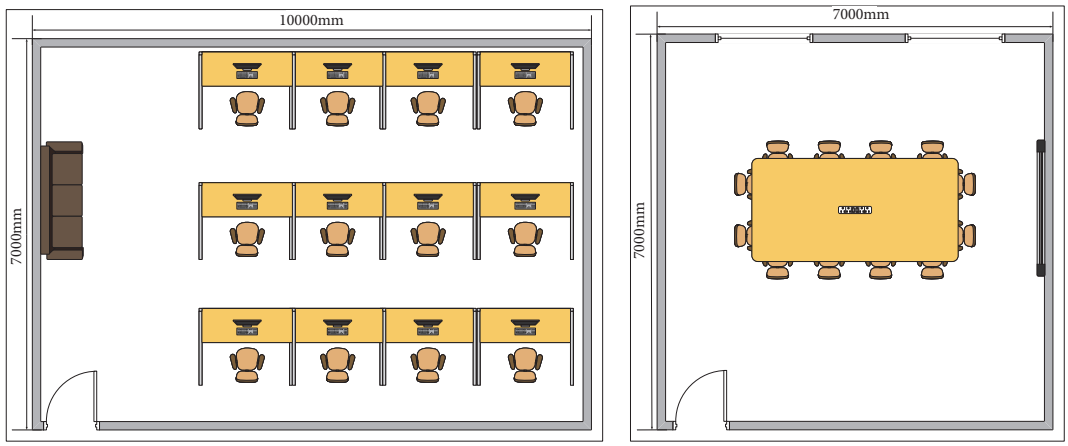
section of performance analysis, we compare the positioning error of Fusion 2 under the different *R*.

In Algorithm 2, another parameter *offset* is set for the following reasons. When there are more than three reference points with the same degree of maximum degree in the candidate reference points, they cannot be distinguished to be used as the final candidates of the test point, which suggests the granularity of the selected point to be measured is too small, and it is necessary to increase the length of the radius, filter out those reference points with no further increase, and then select the reference point set which is close to the point to be measured. Combining the above analysis of *R*, *offset* sets the following principles. When $a \leq R < \sqrt{2}a$, *offset* should be set as $\sqrt{2}a \leq R + offset < 2a$; when $\sqrt{2}a \leq R < 2a$, *offset* should be set as $2a \leq R + offset < \sqrt{5}a$; when $2a \leq R < \sqrt{5}a$, *offset* should be set as $\sqrt{5}a \leq R + offset < 3a$.

The algorithm proposed in this chapter can improve the positioning accuracy, mainly for the following two reasons. One is the set of reference points under three characteristic fingerprints. As a single feature fingerprints have limitations in positioning; a set of reference points obtained by merging multiple fingerprints can make up for its shortcomings. The second is to use a reference point set with a moderate reference point. The higher the reference point is, the closer to the test point it is.

3.4. Overhead Analysis. The proposed approach needs to store three fingerprint metrics (RSS, He, and Hp) for every training position. Assume that there are *N* APs in the environment and each AP has *L* antennas. Denote by *U* the number of intervals for counting the CSI amplitude distribution, and denote by *M* the number of antennas at the mobile device. In the Horus approach, it needs to record *N* values, each value being a received signal strength of AP. In the FIFS approach, for each position it only needs to record $N \times L \times M$ values, each representing an aggregated metric over 30 subcarriers. Then for each training position of D-CSI, it needs to record $N \times L \times M \times 30U$ values, each value being a float representing the possibility of the amplitude falls in the corresponding interval. Thus, the storage cost of our approach is higher than that in each single fingerprint method. However, because the values of *N, L, U* are usually smaller, the total storage cost of our approach can be afforded.

For the time efficiency, in our approach we need to calculate the Euclidean distance between two RSS metrics, the Euclidean distance between two CSI aggregated metrics, and



(a) The layout of the computer laboratory (b) The layout of the meeting room

Figure 2: The layout of two environments.

the KL distance between two CSI amplitude distributions, which incurs slightly higher computation cost than each approach. However, compared with the time used in calculation, the time used to collect data is the major factor that determines the time efficiency. Our approach uses similar number of packets as in FIFS or D-CSI, and thus the time efficiency is similar as in previous approaches.

4. Experiment Results

For the methods proposed in this paper, we have collected the signal strength value and channel state information of the corresponding locations and compared them with those of the single feature. This section first introduces the experimental setup and the data acquisition process. Then the performance analysis of the two methods is given, respectively. Specifically, the performance of Fusion 1 under different weights is compared, and we compare the performance of the fusion location algorithm with different radius R, as well as the performance of the location method with the single feature fingerprint and the localization method with the two or three characteristic fingerprints. In the acquisition of RSS and CSI, we use of a fixed device to prevent human error caused by jitter data.

4.1. Experiment Scenarios. In our experiment, the training spots are evenly distributed in the entire room and the test spots are randomly chosen. TL-WR742N routers work as the transmitters, while a Dell E6410 equipped with an Intel Wi-Fi Link 5300 NIC is used as the mobile device in both the training and the positioning phases. We also modify the driver as in [19] to collect the raw CSI values. We evaluate the system in two typical indoor environments: a computer laboratory and a meeting room.

4.1.1. Computer Laboratory. The floor plan of the computer laboratory is shown in Figure 2(a). Four APs are deployed at the four corners. In the training phase, the RSS and the CSI values are collected at 42 locations with 1.2m spacing in the room for constructing the fingerprint database. During the test, we randomly select 30 locations as the test positions. At each spot, we collect the raw CSI values of 60 packets. We collect 20 RSS samples at each position and select the RSS that appears most of the time as the fingerprint of the corresponding position.

4.1.2. Meeting Room. The floor plan of the meeting room is shown in Figure 2(b) and the four APs are also placed on the four corners. We collect 49 different calibration positions which are 1m apart in this scenario. 10 test locations are randomly selected. At each location, we also collect the RSS and the CSI data of 60 packets. We also collect 20 RSS samples at each position and select the RSS that appears most of the time as the fingerprint of the corresponding position.

4.2. Optimal Weight. We first consider the impact of the weight of Fusion 1 in two experiment environments. We know that the sum of the two weights equals 1. Without loss of generality, we choose the weight w_c as the system parameter to evaluate the localization performance. Through the experiment, we obtain the optimal weight so as to get the minimum positioning error.

The results are plotted in Figure 3. We can see that when $0.5< w_c <0.8$, the localization error can reach the minimum. w_c is too large or too small, and the localization accuracy decreases. Specifically, when w_c is too large, the position where the Horus is obtained with higher accuracy may not be selected. In extreme cases, the final position of the test point is the result of FIFS. When w_c is too small, it is not conductive to use the stability of CSI, which can make the localization accuracy increase.

It also can be observed that when $w_r < w_c$, the overall positioning performance is high. This also indicates that the overall positioning effect of FIFS is better than that of the Horus method. When the proportion of FIFS is higher than that of the Horus method, method one can achieve

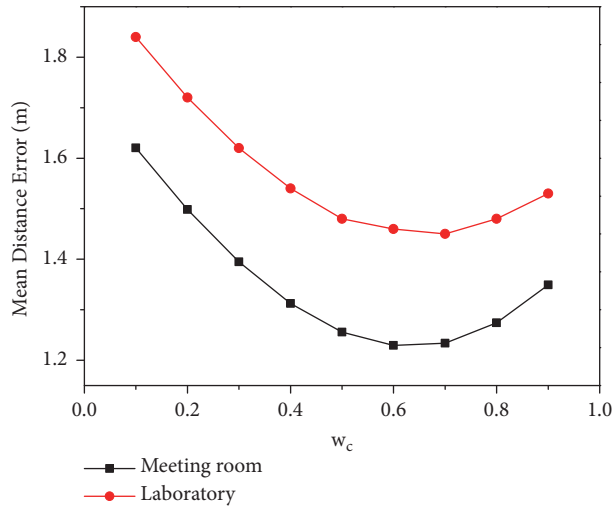


FIGURE 3: The mean distance error of different w_c.

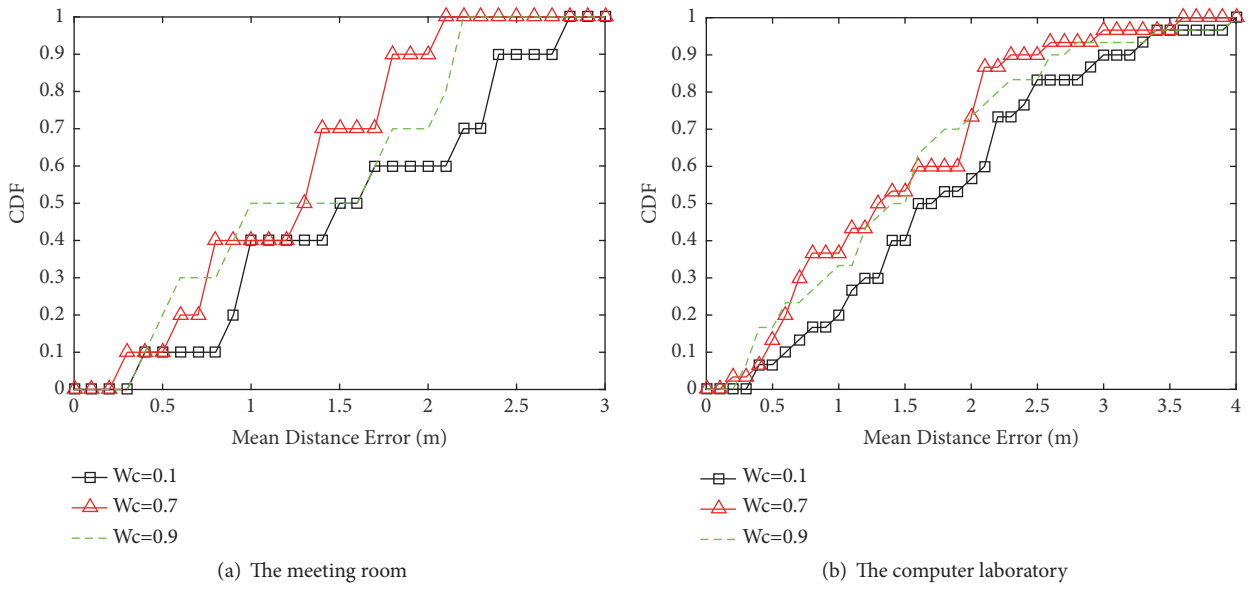


FIGURE 4: CDF of localization error using different w_c.

better positioning accuracy. Figure 4 plots the CDFs of the localization errors under different w_c in two experiment environments. As shown in Figure 4(a), when $w_c = 0.7$, 80% of the test spots have an error under 1.6m. However, when $w_c = 0.9$ and 0.1, the distance errors of the 80% spots enlarge to 2m and 2.3m, respectively. Similar trend can be observed in Figure 4(b). We can conclude that $w_c = 0.7$ is beneficial to improving the positioning accuracy.

4.3. Optimal Radius. In this section, we consider the influence of system parameter R, which is set in Fusion 2 method.

R is a system parameter which is used in calculating each alternative reference point degree. The setting of R has a great influence on the size of the reference point degree and affects the selection of the reference point degree, which determines the positioning performance. Through the experiments, we determine the interval that makes the positioning performance of fusion localization algorithm the best.

Figure 5 shows the mean distance error for the fusion localization method along with the curve of the R change in the two experimental scenarios. We can see when $a \leq R < \sqrt{2}a$, the positioning error can reach the minimum. When R is too large, centered on an alternative reference point with a radius R, this will increase the degree of reference points that are themselves at the edges, so that reference points that should not be selected may be used as the location of the point to be measured. In extreme cases, when R is the length of the entire region boundary, all alternative reference points will be three for a group to be the final reference point set of points

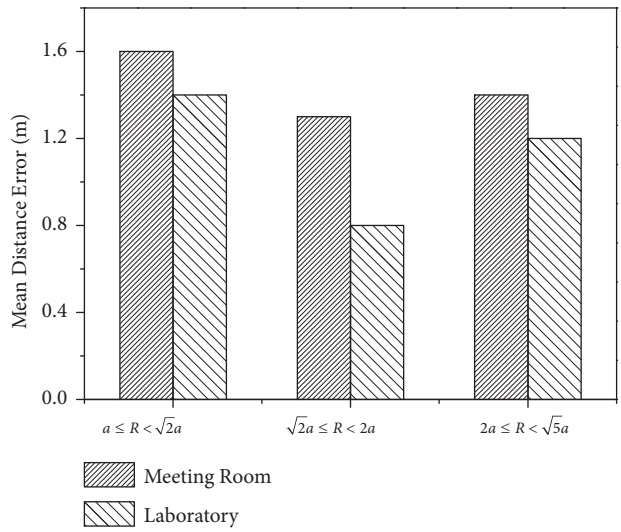


Figure 5: The mean distance error under the different indoor environments.

to be measured, so it cannot make full use of the feature that filters the reference points by calculating the size of degree, thus affecting the positioning accuracy.

In addition, when R is too small, it will make the degree size of the reference points with no difference, so that it is not easy to distinguish which alternative reference points should be selected for calculating the position of points to be measured. In the extreme case, as R is too large, it cannot make full use of the feature that filters the reference points by calculating the size of degree, thus affecting the positioning accuracy. So, whether too large or too small, the positioning effect will have a negative impact.

Figure 6 shows the boxplots of Fusion 2 localization errors under the different R. As shown in the figure, when $\sqrt{2}a \le R < 2a$, the positioning errors of more than 75% test points are less than 1.2m, but when $2a \le R < \sqrt{5}a$ and $a \le R < \sqrt{2}a$, in case of having the same number of test points, the positioning error-free range was expanded to 1.5m and 2m. The same trend is seen in Figure 6(b). These results show that R is not too large and too small in actual deployment. Based on our results in two experimental scenarios, Fusion 2 can achieve better positioning performance when R is set to $\sqrt{2}a \le R < 2a$.

4.4. Localization Performance of Fusion 1. In this subsection, we conduct the performance comparison tests among Horus, FIFS, and Fusion 1. We consider the situation of $w_c = 0.7$ and give the mean distance error and the cumulative distribution (CDF) of the localization error, respectively.

4.4.1. Mean Distance Error. Figure 7 gives the mean distance error obtained by three methods in the two representative scenarios. As shown in the figure, in the meeting room, Fusion 1 achieves the median accuracy of 1.2m, which outperforms FIFS and Horus by more than 0.2m and 0.4m, and the gain is about 14% and 25%. Moreover, in the computer laboratory scenario, the mean accuracy of our approach is 1.4m, which is about 12.5% and 23% gain compared with FIFS and Horus, respectively.

Table 1: Mean distance error.

Method	Horus	FIFS	D-CSI	**Fusion 2**
Meeting room	1.50(±0.9)	1.30(±0.8)	0.90(±0.4)	**0.80(±0.4)**
Laboratory	1.90(±0.9)	1.60(±0.7)	1.42(±0.6)	**1.30(±0.6)**

In addition, the standard deviation of the localization error of each method is given in Figure 7. From the two experimental scenarios, we find that Horus has the largest standard deviation, which shows that there is a big difference among the test spots. Fusion 1 has the minimum standard deviation. This shows that the difference of the localization error becomes smaller and more stable after fusing RSS and He.

4.4.2. CDF of Localization Error. Figure 8 plots the CDFs of the localization errors in the meeting room and the computer laboratory, respectively. As shown in Figure 8(a), for over 80% of the test spots, the errors of Fusion 1 are less than 1.6m. However, FIFS and Horus have the localization errors of 2.1m and 2.4m under the same experimental conditions.

As shown in Figure 8(b), in this more complex wireless signal environment, the median error of three methods is under 1.8m. However, for over 80% of the test spots, the error of Fusion 1 is 1m and 0.5m lower than Horus and FIFS, respectively.

4.5. Localization Performance of Fusion 2. In this subsection, we conduct the performance comparison tests between Fusion 2 and the existing works that used a single feature, such as Horus, FIFS, and D-CSI. We consider the radius varying from $\sqrt{2}a$ to 2a. We give the mean distance error and the cumulative distribution function of localization error of the four localization schemes in the two representative scenarios.

4.5.1. Mean Distance Error. Table 1 gives the mean distance error obtained by Fusion 2, Horus, FIFS, and D-CSI. As shown in the table, in the meeting room, Fusion 2 achieves the median accuracy of 0.8m, which outperforms D-CSI, FIFS, and Horus by more than 0.1m, 0.5m, and 0.7m, and the gain is about 11%, 38%, and 47%, respectively. Moreover, in the computer laboratory scenario, where there exists abundant multipath, the mean accuracy of our approach is 1.3m, which is with about 7%, 18.5%, and 31% gain compared with D-CSI, FIFS, and Horus, respectively.

In addition, the standard deviation of the localization error of each method is given in Table 1. From the two experimental scenarios, we find that Horus has the largest standard deviation, which shows that there is a big difference among the test spots. Fusion 2 has the minimum standard deviation. This shows that the difference of the localization error becomes smaller and more stable after fusing RSS, He, and Hp. Therefore, Fusion 2 is suitable

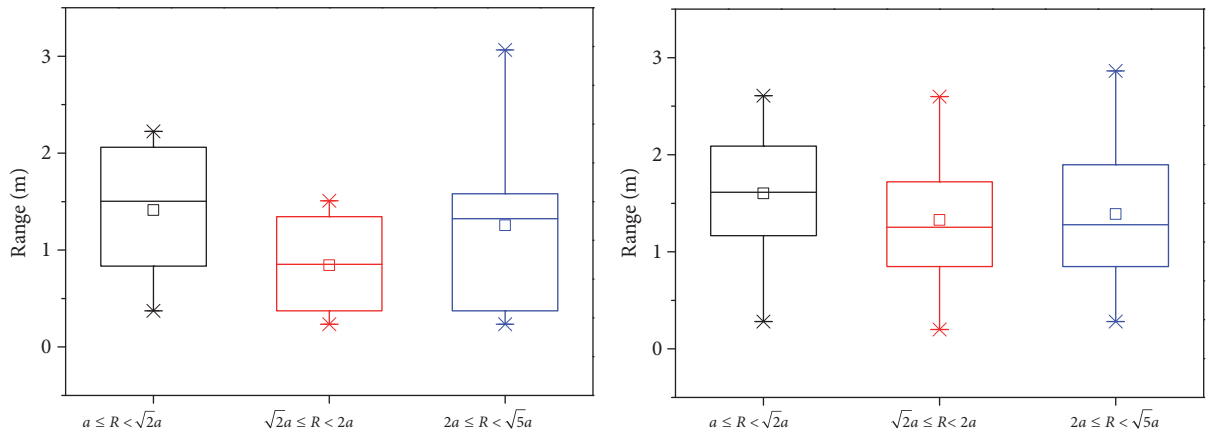


FIGURE 6: The boxplots under the different R.

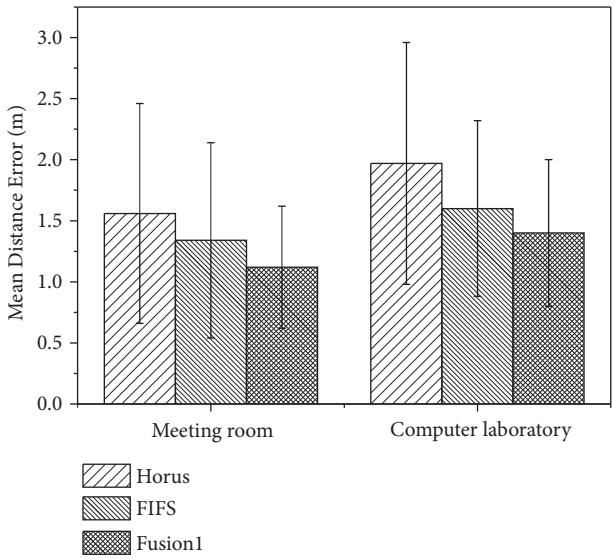


FIGURE 7: Mean distance error.

for applications that require smaller jitter of localization results.

4.5.2. The Cumulative Distribution Function of Localization Error. Figure 9 plots the boxplots of the localization errors in the meeting room and the computer laboratory, respectively. As shown in Figure 9(a), for over 80% of the test spots, the errors of Fusion 2 are less than 1.1m. However, D-CSI, FIFS, and Horus have the localization errors of 1.2m, 2.1m, and 2.4m under the same experimental conditions. All of the Fusion 2 test spots have an error under 1.3m, while only about 85%, 60%, and 40% of the D-CSI, FIFS, and Horus test spots have the same localization accuracy.

As shown in Figure 9(b), in this more complex wireless signal environment, Fusion 2 and D-CSI have a distance error of 1.4m for 50% of the test spots. Meanwhile, the median error of FIFS and Horus is 1.6m. However, for over 80% of the test spots, the error of Fusion 2 is 0.5m lower than D-CSI; the error of FIFS is 0.3m lower than Horus.

5. Conclusion

This paper proposes a multiple feature based indoor localization scheme that uses RSS and CSI information extracted from the commercial off-the-shelf Wi-Fi NICs. Firstly, we propose a weighted fusion scheme that can quickly obtain the location of mobile user while improving the localization accuracy. In order to further improve the positioning accuracy, we propose a three-feature fusion method that first gets the summation of the received signals (He) and the distribution of the subcarriers (Hp) from the CSI value. Then, a set of candidate reference points is obtained using the localization method of each fingerprint feature. Finally, according to the degree of each reference and the principle of triangle minimization, we find the best three reference points and take the centroid as the target location. The proposed method is evaluated in two typical indoor environments and is compared with the existing work. The results show that our scheme achieves better performance under the different scenarios.

Data Availability

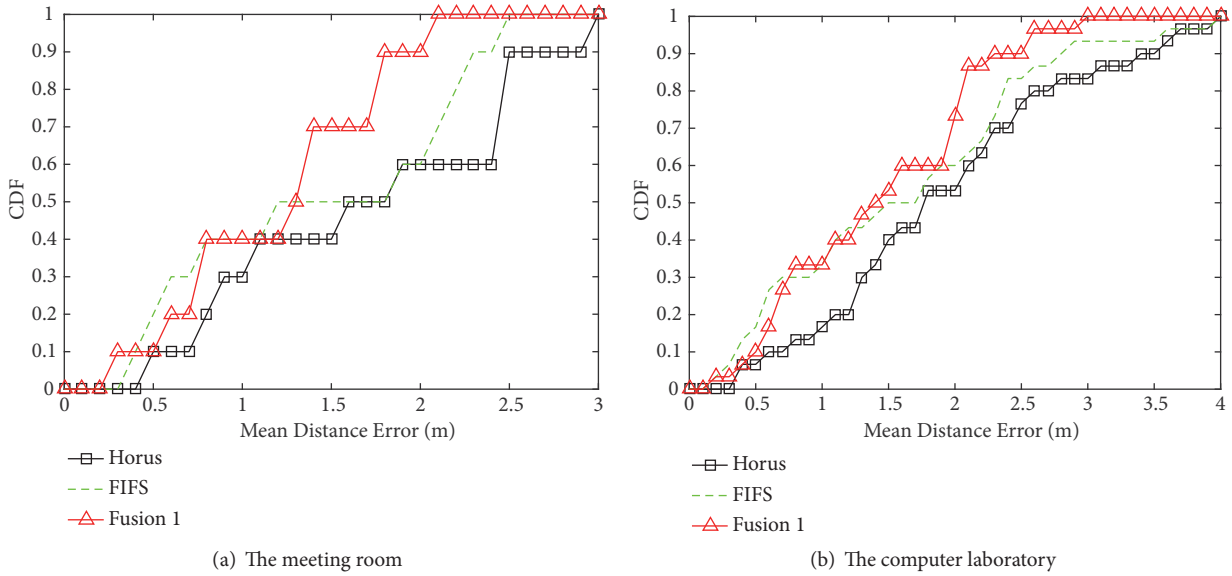
The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

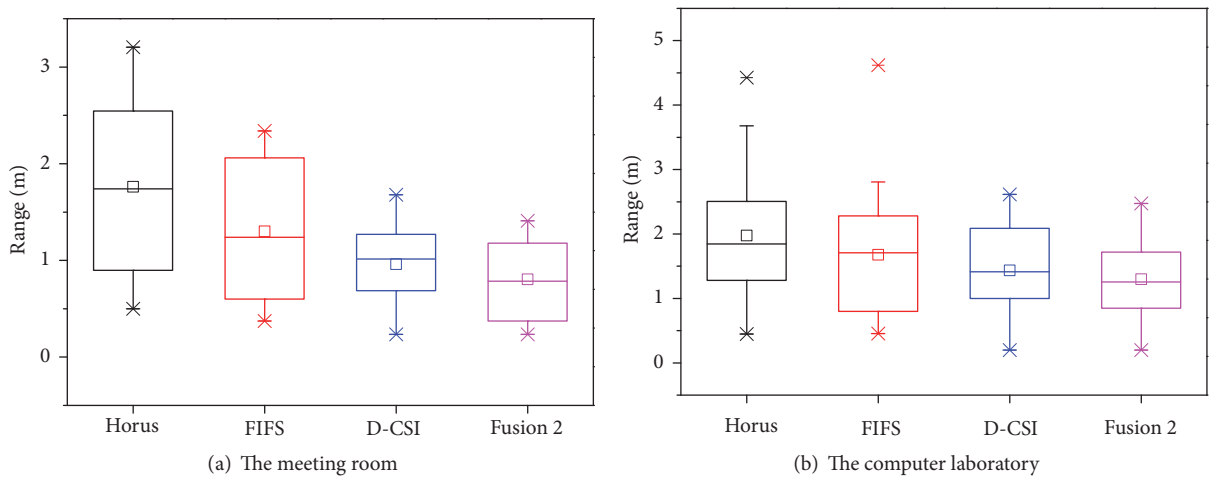
Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 61772559 and



(a) The meeting room (b) The computer laboratory

Figure 8: The CDFs under the different indoor environments.



(a) The meeting room (b) The computer laboratory

Figure 9: The localization accuracy under the different indoor environments.

61702559 and the Hunan Provincial Natural Science Foundation of China under Grant no. 2017JJ3413.

References

[1] M. A. Rahman and M. S. Hossain, "A location-based mobile crowdsensing framework supporting a massive Ad hoc social network environment," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 76–85, 2017.

[2] D. Calabuig, S. Barmpounakis, S. Gimenez et al., "Resource and Mobility Management in the Network Layer of 5G Cellular Ultra-Dense Networks," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 162–169, 2017.

[3] J. Wozniak, "Mobility management solutions for current IP and future networks," *Telecommunication Systems*, vol. 61, no. 2, pp. 257–275, 2016.

[4] M. Karimzadeh, L. Valtulina, A. Pras et al., "Double-NAT based mobility management for future LTE networks," in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference, WCNC 2017*, usa, March 2017.

[5] A. Abuarqoub, M. Hammoudeh, B. Adebisi, S. Jabbar, A. Bounceur, and H. Al-Bashar, "Dynamic clustering and management of mobile wireless sensor networks," *Computer Networks*, vol. 117, pp. 62–75, 2017.

[6] B. Zhou and R. Buyya, "Augmentation Techniques for Mobile Cloud Computing," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1–38, 2018.

[7] X. Liu, S. Zhang, and K. Bu, "A locality-based range-free localization algorithm for anisotropic wireless sensor networks," *Telecommunication Systems*, vol. 62, no. 1, pp. 3–13, 2016.

[8] S. Zhang, X. Liu, J. Wang, J. Cao, and G. Min, "Accurate range-free localization for anisotropic wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 11, no. 3, 2015.

[9] M. M. Hasan, S. Kwon, and J. Na, "Adaptive Mobility Load Balancing Algorithm for LTE Small-Cell Networks," *IEEE*

Transactions on Wireless Communications, vol. 17, no. 4, pp. 2205–2217, 2018.

[10] A. Mohamed, M. A. Imran, P. Xiao, and R. Tafazolli, "Memory-Full Context-Aware Predictive Mobility Management in Dual Connectivity 5G Networks," *IEEE Access*, vol. 6, pp. 9655–9666, 2018.

[11] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-Aware Application Scheduling in Fog Computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, 2017.

[12] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: indoor localization via channel response," *ACM Computing Surveys*, vol. 46, no. 2, article 25, 2013.

[13] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "CSI-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2013.

[14] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 205–218, ACM, June 2005.

[15] Y. Xiao, S. Zhang, J. Cao, H. Wang, and J. Wang, "Exploiting distribution of channel state information for accurate wireless indoor localization," *Computer Communications*, vol. 114, pp. 73–83, 2017.

[16] C. Luo, L. Cheng, M. C. Chan, Y. Gu, J. Li, and Z. Ming, "Pallas: Self-Bootstrapping Fine-Grained Passive Indoor Localization Using WiFi Monitors," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 466–481, 2017.

[17] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2017.

[18] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation techniques: Principles and applications*, Springer Science & Business Media, New York, NY, USA, 2012.

[19] J. Xiao, K. Wu, Y. Yi, and L. M. Ni, "FIFS: Fine-grained indoor fingerprinting system," in *Proceedings of the 2012 21st International Conference on Computer Communications and Networks, ICCCN 2012*, deu, August 2012.

[20] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of RSS fingerprinting based indoor localization," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '15)*, pp. 2479–2487, Hong Kong, May 2015.

[21] Y. Xiao, J. Wang, S. Zhang, H. Wang, and J. Cao, "Accurate Indoor Localization with Multiple Feature Fusion," in *Wireless Algorithms, Systems, and Applications*, vol. 10251 of *Lecture Notes in Computer Science*, pp. 522–533, Springer International Publishing, Cham, 2017.

[22] S. Jeon, S. Figueiredo, R. L. Aguiar, and H. Choo, "Distributed Mobility Management for the Future Mobile Networks: A Comprehensive Analysis of Key Design Options," *IEEE Access*, vol. 5, pp. 11423–11436, 2017.

[23] X. Kui, Y. Sun, S. Zhang, and Y. Li, "Characterizing the Capability of Vehicular Fog Computing in Large-scale Urban Environment," *Mobile Networks and Applications*.

[24] M. Z. A. Bhuiyan, J. Wu, G. Wang, T. Wang, and M. M. Hassan, "e-Sampling: Event-sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 12, no. 1, article no. 1, 2017.

[25] E. Luo, Q. Liu, J. H. Abawajy, and G. Wang, "Privacy-preserving multi-hop profile-matching protocol for proximity mobile social networks," *Future Generation Computer Systems*, vol. 68, pp. 222–233, 2017.

[26] S. Chen, F. Qin, B. Hu, X. Li, and Z. Chen, "User-centric ultra-dense networks for 5G: Challenges, methodologies, and directions," *IEEE Wireless Communications Magazine*, vol. 23, no. 2, pp. 78–85, 2016.

[27] X. Xing, G. Wang, and J. Li, "Collaborative target tracking in wireless sensor networks," *Ad-Hoc & Sensor Wireless Networks*, vol. 23, no. 1-2, pp. 117–135, 2014.

[28] Y. Gu, F. Ren, Y. Ji, and J. Li, "The evolution of sink mobility management in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 507–524, 2016.

[29] L. Wang, X. Qi, J. Xiao, K. Wu, M. Hamdi, and Q. Zhang, "Exploring Smart Pilot for Wireless Rate Adaptation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4571–4582, 2016.

[30] M. Z. A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, "Dependable Structural Health Monitoring Using Wireless Sensor Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 363–376, 2017.

[31] Y. Liu, K. Ota, K. Zhang et al., "QTSAC: An Energy-Efficient MAC Protocol for Delay Minimization in Wireless Sensor Networks," *IEEE Access*, vol. 6, pp. 8273–8291, 2018.

[32] A. Liu, M. Huang, M. Zhao, and T. Wang, "A Smart High-Speed Backbone Path Construction Approach for Energy and Delay Optimization in WSNs," *IEEE Access*, vol. 6, pp. 13836–13854, 2018.

[33] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, vol. 2, pp. 775–784, Tel Aviv, Israel, March 2000.

[34] X. Chen, C. Ma, M. Allegue, and X. Liu, "Taming the inconsistency of Wi-Fi fingerprints for device-free passive indoor localization," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.

[35] D. Contreras, M. Castro, and D. S. de la Torre, "Performance evaluation of bluetooth low energy in indoor positioning systems," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, 2017.

[36] Z. Gao, Y. Ma, K. Liu, X. Miao, and Y. Zhao, "An Indoor Multi-Tag Cooperative Localization Algorithm Based on NMDS for RFID," *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2120–2128, 2017.

[37] S. Yoon, K. Lee, and I. Rhee, "FM-based indoor localization via automatic fingerprint DB construction and matching," in *Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*, pp. 207–219, Taipei, Taiwan, June 2013.

[38] L. Zhang, D. Huang, X. Wang, C. Schindelhauer, and Z. Wang, "Acoustic NLOS identification using acoustic channel characteristics for smartphone indoor localization," *Sensors*, vol. 17, no. 4, 2017.

[39] N. Lee and D. Han, "Magnetic indoor positioning system using deep neural network," in *Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, Sapporo, September 2017.

[40] B. Kempke, P. Pannuto, B. Campbell, and P. Dutta, "Sure point: Exploiting ultra wideband flooding and diversity to provide

robust, scalable, high-fidelity indoor localization," in *Proceedings of the 14th ACM Conference on Embedded Networked Sensor Systems, SenSys 2016*, pp. 137–149, usa, November 2016.

[41] Z. Zhao, J. Wang, X. Zhao, C. Peng, Q. Guo, and B. Wu, "NaviLight: Indoor localization and navigation under arbitrary lights," in *Proceedings of the 2017 IEEE Conference on Computer Communications, INFOCOM 2017*, usa, May 2017.

[42] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[43] S. Sen, B. Radunović, R. R. Choudhury, and T. Minka, "You are facing the Mona Lisa: Spot localization using PHY layer information," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, MobiSys'12*, pp. 183–196, gbr, June 2012.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5967194, 16 pages
https://doi.org/10.1155/2018/5967194

Review Article

Revisiting of Channel Access Mechanisms in Mobile Wireless Networks through Exploiting Physical Layer Technologies

Junmei Yao,[1] Jun Xu,[1] Yue Ling Che,[1] Kaishun Wu,[1] and Wei Lou[2]

[1]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China*
[2]*Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Correspondence should be addressed to Kaishun Wu; wu@szu.edu.cn

Received 16 January 2018; Accepted 4 April 2018; Published 15 May 2018

Academic Editor: Ziming Zhao

The wireless local area networks (WLANs) have been widely deployed with the rapid development of mobile devices and have further been brought into new applications with infrastructure mobility due to the growth of unmanned aerial vehicles (UAVs). However, the WLANs still face persistent challenge on increasing the network throughput to meet the customer's requirement and fight against the node mobility. Interference is a well-known issue that would degrade the network performance due to the broadcast characteristics of the wireless signals. Moreover, with infrastructure mobility, the interference becomes the key obstacle in pursuing the channel capacity. Legacy interference management mechanism through the channel access control in the MAC layer design of the 802.11 standard has some well-known drawbacks, such as exposed and hidden terminal problems, inefficient rate adaptation, and retransmission schemes, making the efficient interference management an everlasting research topic over the years. Recently, interference management through exploiting physical layer mechanisms has attracted much research interest and has been proven to be a promising way to improve the network throughput, especially under the infrastructure mobility scenarios which provides more indicators for node dynamics. In this paper, we introduce a series of representative physical layer techniques and analyze how they are exploited for interference management to improve the network performance. We also provide some discussions about the research challenges and give potential future research topics in this area.

1. Introduction

With the rapid development of mobile devices, the wireless local area networks (WLANs, often called Wi-Fi networks) have been widely deployed in various places, from home to office, restaurants, clubs, etc., due to its acceptable performance and easy-to-deploy characteristics [1]. Currently, the growth of unmanned aerial vehicles (UAVs) brings these networks into new applications with infrastructure mobility, where the access point (AP) is mobile to alleviate some unpredictable problems, such as natural disaster and poor coverage. However, the WLANs still face a continual challenge on the channel capacity owing to the huge amount of mobile data traffic. As forecast by the Cisco Visual Networking Index [2] (VNI), the global mobile data traffic will increase sevenfold from 2016 to 2021, and about 50% of the traffic will be offloaded to WLANs.

Actually, the IEEE task group has made great efforts on throughput improvement in WLANs through the evolution of the 802.11 standards in the past twenty years. From 1997 when the base version of IEEE 802.11 standard which supports 1/2 Mbps data rate was released, a series of 802.11 standards have been released to increase the physical layer data rate. 802.11b and 802.11a/g increase the data rate of single stream up to 11 Mbps and 54 Mbps, respectively, through exploiting higher-order modulations; 802.11n and 802.11ac increase the data transmission rate up to 600 Mbps and >6 Gbps, respectively, through further exploiting the Multiple Input and Multiple Output (MIMO) technique. Now an upcoming 802.11ax aims to achieve up to 10 Gbps data rate

and is expected to be released in 2019. However, there is a huge gap between the physical layer data rate and the network throughput [3], which is mainly due to the inefficient channel access mechanism for interference management in wireless mobile networks.

Interference is well known to degrade the network throughput due to the broadcast characteristics of the wireless signals. Theoretically, the interference occurs when the received signal's Signal to Interference and Noise Ratio (SINR) is below a required threshold. The problem will be even worse under the infrastructure mobility scenario as the mobile APs may lead to the dynamic change of the received signal power. The 802.11 family recommends the carrier sense multiple access (CSMA) mechanism in the MAC layer to manage interference. A transmitter can proceed its transmission only when it determines that the channel is idle; otherwise, it should keep silent until the channel becomes idle to avoid interference. This mechanism is simple but inefficient, as it has the following well-known drawbacks which may result in low network performance: (1) it may fail to avoid interference effectively due to the hidden terminal problem, where a node will interfere with an ongoing link as it cannot sense the data transmission from the transmitter but can interfere with the data reception at the receiver; (2) it may prohibit concurrent transmissions of noninterfering links due to the exposed terminal problem, where a node is prohibited to transmit signals as it can hear the data transmission from the transmitter, although it will not interfere with the data reception at the receiver; (3) it may be unable to transmit packets in the optimal data rate according to the actual channel situations, while data can certainly be transmitted at a higher rate when the wireless channel is better; (4) it has an inefficient retransmission scheme when a data packet is not detected correctly, as it makes the whole data packet retransmitted although a large partial of this packet can be detected correctly. The above listed drawbacks would degrade the network performance to a great extent in some situations and thus motivate the researchers to work on more effective interference management mechanisms.

Exploiting physical layer techniques is a promising way to manage interference from one or more of the above four aspects and has attracted much research interest in the recent decade. The benefits brought by the physical layer techniques, such as interference resistance and real-time channel estimation, inspire a new way for designing high efficient channel access mechanisms, especially under the infrastructure mobility situations. This paper will present a series of physical layer techniques and investigate how they are exploited for interference management, as shown in Figure 1. Cross correlation has the high interference-resistant characteristic and can tolerate the collision of control or data packets to some extent; it is always utilized to combat the exposed and hidden terminal problems or reduce the coordination overhead. SoftPHY provides each received bit's physical layer confidence to the upper layer [4], so as to determine which bits need to be retransmitted and what the optimal data rate would be. Successive interference cancellation (SIC) makes a strong interfered signal detected at first to recover an inferior strong signal, if the SINR is above the threshold after subtracting the strongest one [5]; it is mainly utilized to increase concurrent transmissions. Rateless coding aims to make the signal with fixed coding and modulation schemes decoded at any SNR environments, through making the signal transmitted multiple times and combining them at the physical layer of the receiver effectively, thus achieving an optimal data rate [6]. Some physical layer unique features are also exploited to design high efficient interference management mechanism; for example, the redundancy design for communication is commonly utilized to convey the coordination information, assisting in avoiding interference, increasing concurrent transmissions, or reducing coordination overhead.

Although with demonstrated high performance through hardware experiments or simulations, the adoption of these techniques represents some major challenges in current wireless networks, especially the high computational overhead in signal process, inflexible MAC layer design due to the scenario limitation of the techniques, etc. This survey will also investigate the challenges of each physical layer technique when applying to real networks for interference management.

The rest of this paper is organized as follows: Section 2 gives an overview of the IEEE 802.11 family, including its physical layer and MAC layer specifications. Section 3 discusses the problems in the legacy channel access mechanism recommended by the IEEE 802.11 MAC. Section 4 surveys a series of physical layer techniques and how they would be applied for interference management. Section 5 discusses advantages and limitations of each kind of interference management mechanisms and also puts forwards some future directions in this research area. Section 6 concludes this paper.

2. Review of IEEE 802.11 Standard

This section will briefly introduce the standards in the IEEE 802.11 family, including the physical layer (PHY) and MAC specifications.

2.1. The IEEE 802.11 PHY Specification. In the past twenty years, the IEEE 802.11 task group has made great efforts to increase the data rate of wireless devices through the physical layer evolutions.

As shown in Table 1, the 802.11 base version was released in 1997, which specifies two data rates of 1/2 Mbps working in the 2.4 GHz frequency band and uses the Direct-Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) as the modulation types. After that, they released a series of 802.11 standards to increase the physical layer data rate. In 1999, 802.11a, which operates in the 5 GHz frequency band and uses the Orthogonal Frequency Division Multiplexing (OFDM) as the modulation type, was introduced to enhance the data rate to 54 Mbps. In the same year, 802.11b, which operates in the same frequency band and uses the same modulation technique with the base version, was also introduced to support up to 11 Mbps data rate. Due to its dramatic throughput increase and the similar PHY technique compared with the base version, the 802.11b becomes the definitive WLAN technology. In 2003,

Table 1: IEEE 802.11 standards for throughput improvement.

Version	Release Time	Maximum Data Rate	Frequency Band (GHz)	Bandwidth (MHz)	Modulation
802.11-base version	Jun. 1997	2 Mbps	2.4	20	BPSK, QPSK DSSS, FHSS
802.11b	Sep. 1999	11 Mbps	2.4	20	BPSK, QPSK DSSS (CCK)
802.11a	Sep. 1999	54 Mbps	5	20	BPSK, QPSK, 16-QAM, 64-QAM OFDM
802.11g	Jun. 2003	54 Mbps	2.4	20	BPSK, QPSK, 16-QAM, 64-QAM OFDM, DSSS
802.11n	Oct. 2009	600 Mbps	2.4 & 5	20, 40	BPSK, QPSK, 16-QAM, 64-QAM OFDM, MIMO
802.11ac	Dec. 2013	6.933 Gbps	2.4 & 5	20, 40, 80, 160	BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM OFDM, MIMO, MU-MIMO
802.11ax	Approx. 2019	>10 Gbps	<6	----	BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM, 1024-QAM, OFDM, MIMO, MU-MIMO, OFDMA

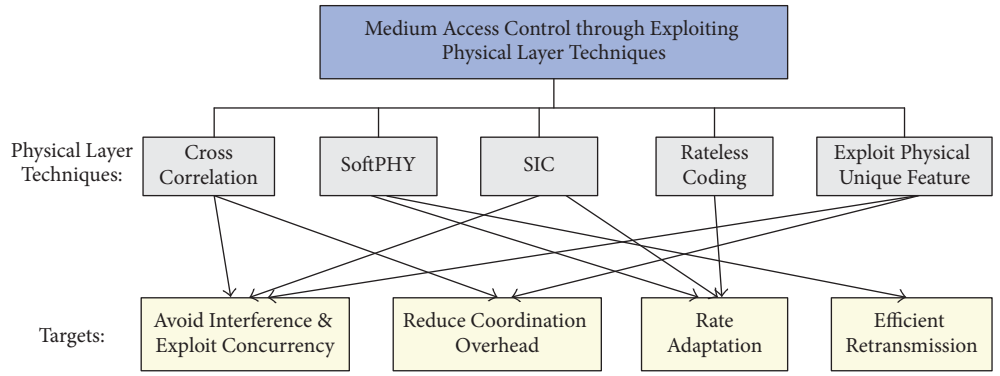


Figure 1: The classification of interference management mechanisms through exploiting physical layer techniques in WLANs.

the IEEE task group released 802.11g, which uses OFDM as its modulation technique (the same as 802.11a), and finally achieves up to 54 Mbps data rate in the 2.4 GHz band. It is always regarded as an extension of 802.11b.

The utilizing of Multiple Input and Multiple Output (MIMO) antennas substantially improves the spectral efficiency and finally improves the physical data rate of WLANs. In 2009, 802.11n, the first standard which supports MIMO, was ratified to support up to 600 Mbps data rate through four spatial streams and 40 MHz wider bandwidth. The latest released version is 802.11ac, which increases the data rate to >6 Gbps through 160 MHz wider bandwidth, eight spatial streams, higher-order 256-QAM modulation, and Multi-User MIMO (MU-MIMO). The upcoming 802.11ax, which is currently in an early stage of development, is predicted to achieve >10 Gbps data rate, through further introducing the Orthogonal Frequency Division Multiple Access (OFDMA) technique and the higher-order 1024-QAM modulation.

To illustrate the evolution of the physical layer process in WLANs more clearly, we show its basic diagram at the transmitter side, as shown in Figure 2. The receiver has the reverse processes. The components with dashed lines, including spreader, OFDM modulator, and MIMO encoder, are optional for specific 802.11 standard which supports the corresponding functions.

2.2. The IEEE 802.11 MAC Specification. Although the physical layer techniques have been updated to higher-order modulations, wider bandwidth and MIMO to increase the data rate, the interference management process which is

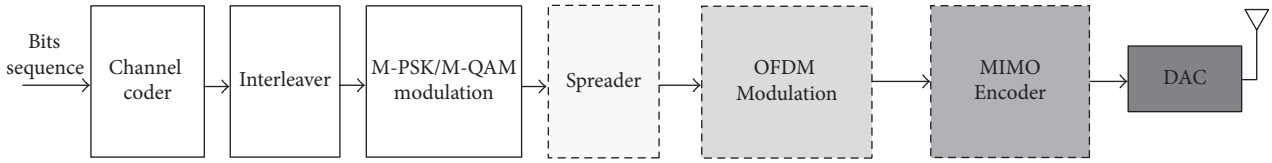


Figure 2: The basic diagram of the physical layer process at the transmitter in WLANs. The receiver has the reverse process.

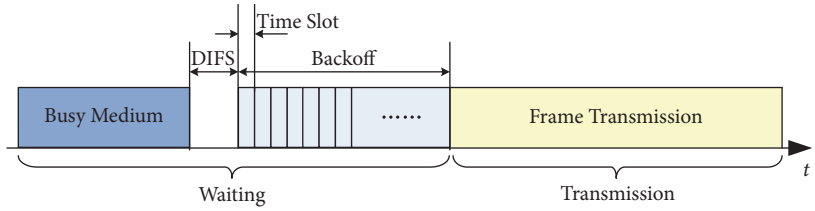


Figure 3: The basic access mechanism in DCF.

defined in the MAC layer of the 802.11 base version has only a little change along with the update of the standards. The 802.11 MAC recommends two kinds of coordination functions for interference management: a distributed mechanism called Distributed Coordination Function (DCF) and a centralized mechanism called Point Coordination Function (PCF). DCF is widely deployed in current WLANs, while PCF is not implemented in most devices as it is not part of the Wi-Fi Alliance's standard.

DCF further contains two mechanisms: a physical carrier sense mechanism called carrier sense multiple access (CSMA) and a virtual carrier sense mechanism called RTS/CTS. Both of them are designed to reduce the collision probability in a distributed way when multiple nodes access the same channel.

The CSMA and RTS/CTS follow the same basic channel access mechanism, as shown in Figure 3. Before transmitting, a sender first senses the medium to determine whether the channel is available; if the channel is determined to be busy when a nearby node is transmitting a signal, the sender should defer its transmission until the channel is available again to avoid interfering with the ongoing link. When the channel is idle for a DIFS (distributed interframe space) period, the node shall generate a random backoff period for an additional deferral time before transmitting. The backoff period is decremented along with the time when the channel stays idle and is suspended otherwise. Only when the backoff period expires can the node transmit its signal.

When the backoff expires, the two channel access mechanisms have different operations. CSMA makes the node transmit its data packet immediately, while RTS/CTS utilizes the exchange of the RTS and CTS frames to reserve medium for the following data transmission, through the NAV value carried in both frames; all the neighboring nodes that receive the RTS or CTS frame will keep silent during the NAV time to avoid interfering with this data transmission. For both mechanisms, the receiver should reply an acknowledgment (ACK) after receiving the data packet; the transmitter determines the data transmission to be successful only if it detects the ACK from the receiver; otherwise, it will retransmit this packet.

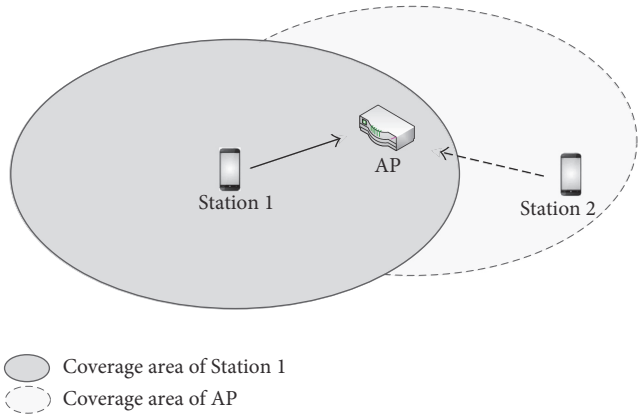


Figure 4: A scenario of the hidden terminal problem.

3. Problems in Legacy Channel Access Mechanism

In this section, we will give detailed description of four problems in current channel access mechanism.

3.1. The Hidden and Exposed Terminal Problems. CSMA has a serious hidden terminal problem which induces collisions, as it uses the channel situation at the transmitter side to determine that at the receiver side, which process is always improper in the real networks. As shown in Figure 4, when Station 1 is transmitting a data packet to the AP, Station 2 may determine that the channel is idle as it is far from Station 1 and out of its coverage area, it shall transmit signal simultaneously. This transmission will interfere with the AP's data reception from Station 1, making it not detect the data packet correctly. RTS/CTS is proposed to mainly solve this problem through the CTS frame transmitted by the receiver. According to the RTS/CTS mechanism, as Station 2 can receive the CTS frame transmitted by AP, it will keep silence during this transmission, thus avoiding inducing collisions.

However, both CSMA and RTS/CTS have the exposed terminal problem which prohibits concurrent transmissions.

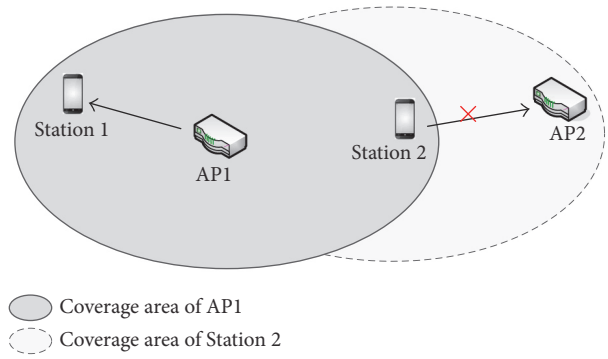


Figure 5: A scenario of the exposed terminal problem.

As shown in Figure 5, when AP1 is transmitting a data packet to Station 1, Station 2 which is near to AP1 is prohibited to transmit signals to AP2 simultaneously as it determines the channel is busy, although the data packet transmissions of these two links have no mutual interference.

3.2. The Coordination Overhead. There are two main kinds of coordination overhead in the 802.11 standard that may degrade the network throughput significantly: the backoff and the control frames' transmission overhead.

As shown in Figure 3, the backoff time will obviously induce a large amount of overhead as the nodes should wait for a long time before transmission, especially in the case when multiple nodes conduct the backoff simultaneously. In addition, a failure data transmission will lead to exponential increase in the backoff period and thus further increase the overhead. Researchers have pointed out that backoff leads to more than 30% throughput degradation in the 802.11b system [7], and the results would be even worse in the higher data rate situation [8].

The RTS, CTS, and ACK control frames' transmissions also induce too much overhead to the system, especially in the situation of the high data packet transmission rate, as the RTS and CTS are still transmitted at the lowest rate. For this reason, the current widely used mechanism is CSMA but not the RTS/CTS, although RTS/CTS can solve the hidden terminal problem efficiently.

3.3. Rate Adaptation Problem. The 802.11 standard recommends a set of data transmission rates which are related to different modulations, coding types, and number of spatial streams for MIMO. Table 2 lists the eight data rates supported by 802.11a as an example. Due to the different required SNR threshold for each data rate, it is easy to understand that optimizing the data transmission rate according to the channel quality can significantly improve the network throughput. The 802.11 standard does not give any recommendation for the rate adaptation process, leaving it an open research topic till now.

3.4. Low Efficient Retransmission. According to the DCF process, the transmitter determines the data transmission to be successful only if it detects the ACK from the receiver; otherwise, it will retransmit the whole data packet. However, as observed by researchers that the wireless channel often induces errors to only a few bits of a packet [4], retransmission of the entire packet is wasteful as the correct bits are transmitted for multiple times. How to organize the incorrect bits and make them retransmitted effectively also attracts much research interest.

Table 2: Data rates supported by 802.11A.

Data rate (Mbps)	SNR threshold (dB)	Modulation scheme	Coding rate
6	6.02	BPSK	1/2
9	7.78	BPSK	3/4
12	9.03	QPSK	1/2
18	10.79	QPSK	3/4
24	17.04	16-QAM	1/2
36	18.80	16-QAM	3/4
48	24.05	64-QAM	1/2
54	24.56	64-QAM	3/4

3.5. Discussion. There has already been a great amount of research on solving one or more of these problems to improve the network throughput through upper layer design, such as constructing an interference map to solve the exposed and hidden terminal problems [9], scheduling the data transmissions centrally to reduce the coordination overhead [10], adapting the transmission rate through observing either the packet loss rate or the SNR at the receiver side [11], etc. However, the mechanisms through exploiting physical layer techniques become more promising in recent years, which will be described in the following part.

4. Survey on Interference Management through Exploiting Physical Layer Techniques

In this section, we will investigate a series of physical layer techniques and how they are explored for interference management in wireless networks, as shown in Figure 1.

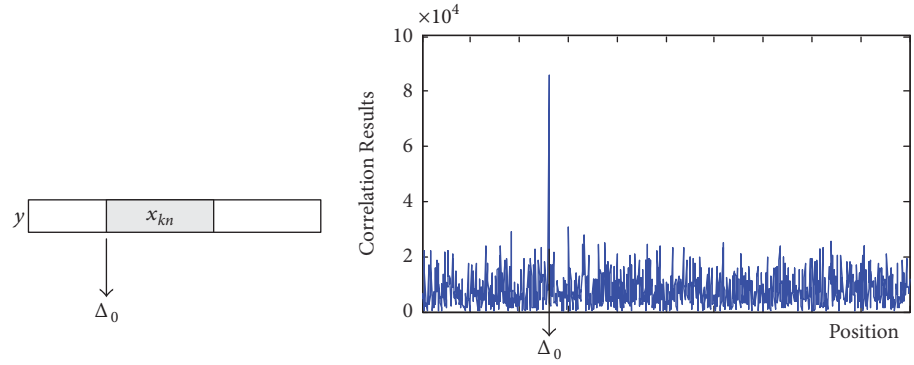


FIGURE 6: When a known sequence x_{kn} is in the received signal y at the position Δ_0, the correlation results between y and x_{kn} will have a peak value at Δ_0, while those at other positions will have small values.

4.1. Cross Correlation. The technology of cross correlation has been proposed for many years, and it has already been used in WLANs in the preamble synchronization and the DSSS modulation process when the 802.11 standard was established in 1997. However, it begins to be utilized for the purpose of interference management in WLANs only from about 2008.

4.1.1. Introduction of Cross Correlation. In the communication systems, cross correlation is commonly used to search for a known transmitted sequence in the received signal. According to the digital communication, a wireless signal is represented as a stream of complex samples, and a received signal $y[n]$ is different from the transmitted one $x[n]$ in amplitude, frequency, and phase, due to the channel attenuation, multiple path, the difference in the oscillator, and so on. Their relationship is always represented as $y[n] = H \cdot x[n]e^{j2\pi\delta_f T+\theta_0}$, where H is the amplitude attenuation factor, δ_f is the frequency offset, T is the sampling period, and θ_0 is the initial phase offset. When a node receives a signal $y[n]$ and intends to figure out whether a known sequence $x_{kn}[i]$ $(i = \{1, \ldots, L\})$ is received, it will calculate the correlation results between y and x_{kn} at each position of y, through the equation $R[\Delta] = \sum_{k=1}^{L} \overline{x_{kn}[k]} \cdot y[\Delta + k]$. As an example shown in Figure 6, the value of $R[\Delta]$ will be very small except that at the position Δ_0, where x_{kn} begins to appear.

4.1.2. Survey on Cross Correlation. This technology has been utilized to both combat the exposed and hidden terminals and reduce the coordination overhead.

(i) Avoid Interference and Exploit Concurrency. Many current strategies leverage the technology of cross correlation to recover the control information under interferences so as to coordinate between nodes [12–17]. In these protocols, authors carefully design some known symbol sequences to convey the control information and conduct the cross correlation between the received signal and the known sequences to determine which sequence is received, so as to obtain the control information under interferences, as shown in Figure 7.

CSMA/CN [13] attempts to implement the similar CSMA/CD (carrier sense multiple access/collision detection,

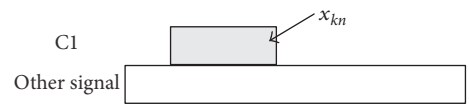


FIGURE 7: A scenario of permitting the control frame C1 collided by other signals. The known sequence x_{kn} here is C1 which can be detected correctly through cross correlation. Researchers exploit this method to reduce the control frames' transmission overhead, increase concurrent transmissions, and avoid interference more efficiently.

a well-known collision avoidance mechanism in the wired networks) to wireless networks. During a data packet reception, it makes the receiver transmit a collision notification information immediately when a collision occurs, through utilizing the correlated symbol sequences. The transmitter can then detect this sequence correctly under strong interference and abort the transmission instantly to avoid further collisions. 802.11ec [15] exploits the cross correlation to accomplish the control frames' transmissions. Comparing with the 802.11 standard, this protocol uses three kinds of known sequences to convey the RTS, CTS, and ACK information. As the known sequences can tolerate strong interferences, and the duration of these sequences is much less than that of the corresponding packets, this protocol can improve the network throughput through both avoiding collisions and reducing the transmission overhead of the control frames. RTS/S-CTS [14] presents a symbol-level detection mechanism to combat both the CTS collision problem and the remote hidden terminal problem, as the symbol sequences that carry useful information in the new S-CTS packet can be detected in very low SINR and SNR environments. IRMA [16] proposes to solve the exposed terminal problem through exploiting this technology. It designs a physical layer mechanism, called signature detection, to combat the CTS/ACK collisions at the transmitter side. It also designs new mechanisms in the MAC layer to differentiate the interfering and noninterfering links. Both mechanisms collaborate to exploit concurrent transmissions and avoid interference.

Some other researchers exploit this technology to recover the collided data signal. The first work is Zigzag [18], as shown in Figure 8. It focuses on the uplink transmissions from

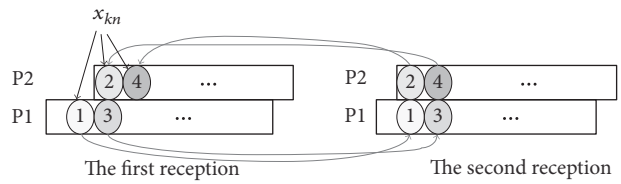


Figure 8: A scenario of detecting two collided data packets P1 and P2. This is a classical scenario of Zigzag, which lets the collided signals be transmitted again. The known sequence x_{kn} here is the collision-free or detected chunks. Zigzag first decodes the collision-free chunk 1 of P1 in the first reception, subtracts it in the second reception to decode chunk 2 of P2, then subtracts chunk 2 to decode chunk 3 in P1, etc. Through this way, it iteratively decodes both packets, thus solving the hidden terminal problems shown in Figure 4.

clients to AP and lets the collided signals be transmitted again for decoding. As these collisions have different interference-free blocks, it then exploits this collision diversity to bootstrap its decoding. Figure 8 can be extended to n-collided packets scenario. This method can solve the hidden terminal problem and achieve the same throughput as if the colliding packets were scheduled in a TDMA way. Symphony [19] extends the idea of Zigzag in the multiple-AP scenarios through utilizing the wired backbone among APs. It encourages collisions of packets transmitted from clients to APs and cooperatively decodes all the packets through the Zigzag-like process. This kind of idea has also been extended to some other scenarios, such as the wireless cooperative relay in DAC [20] and efficient broadcasting in Chorus [21].

(ii) Reduce Coordination Overhead. Researchers also exploit this technology to reduce the coordination overhead in backoff. The semidistributed backoff (SDB) algorithm in [22] is proposed to make nodes perform the receiver side backoff. Using SDB algorithm, they design a MAC protocol Semi-DCF, which exploits the collision detection capability of receivers for disseminating information on optimal backoffs to the contenders using signature vectors, so as to migrate backoff from random to deterministic and largely reduce the backoff time in 802.11 standard. CWM [23] exploits collision tolerance mechanism to reduce the backoff time and improve the channel utilization in the wireless networks. Upon detecting a collided signal from multiple senders, the receiver obtains the senders' IDs through exploiting the correlatable preamble in the physical layer and then allocates each sender a different timeslot so that the senders can transmit their data packets one after another in the following time, without mutual interference.

4.2. SoftPHY

4.2.1. Introduction of SoftPHY. The SoftPHY was first introduced in [24] to provide PHY-independent hints about the PHY's confidence in each bit to the upper layer, so as to determine which bits need to be retransmitted. As shown in Figure 9(a), the basic idea of SoftPHY is to calculate the distance φ between the received signal's constellation point y_i and the corresponding theoretical point a_i; that is, $\varphi_i = K_c \cdot \|y_i - a_i\|$, where K_c is a constant factor related to the modulation type.

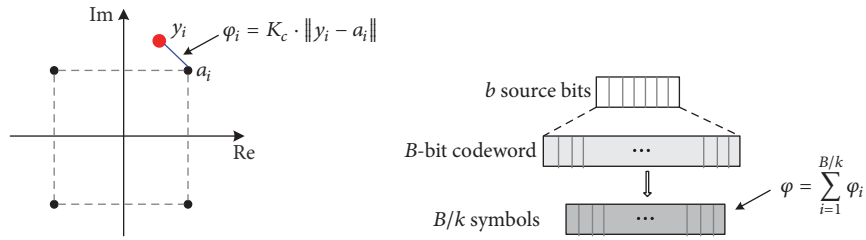
The SoftPHY is designed to be suitable with the real communication systems, where redundancy is added through coding to make the data transmission more robust to the interference existing in the wireless channel. That means the node maps each b-bit string to a B-bit codeword C_i ($b < B$), while the codeword is selected from a codebook $\{C_i, \ldots, C_M\}$. The B-bit codeword will be transformed to B/k symbols after modulation process, where k indicates the number of bits per symbol and is decided by the modulation type. Then the SoftPHY hint for any B-bit codeword is $\varphi = \sum_{i=1}^{B/k} \varphi_i$. The codeword with a larger φ value is determined to be detected correctly with higher probability and higher confidence.

4.2.2. Survey on SoftPHY. This technology has been mainly utilized either to increase the retransmission efficiency or for rate adaptation.

(i) Efficient Retransmission. Through observing that current wireless protocols make all the data packet retransmitted even when detecting a small number of bit errors, authors in [24] propose partial packet recovery (PPR) based on the SoftPHY interface to improve the retransmission efficiency. PPR contains a link-layer protocol design which permits a receiver to encode a retransmission request, so that the transmitter will only retransmit the bits with low confidence. It also contains a postamble scheme which can be utilized to recover the packet in a "roll back" way when the preamble is collided, so as to further improve the network throughput.

(ii) Rate Adaptation. Different from PPR [24] where SoftPHY is utilized to estimate the confidence of symbols for retransmission determination, SoftRate [25] begins to exploit the SoftPHY hints to estimate the BER of a received frame for rate adaptation. Nodes use this BER estimation to pick up optimal bit rates for the next frame transmission and thus achieve the rate adaptation on the frame level. This mechanism can rapidly respond to the varied channel conditions. It can also identify whether the changes in the BER estimation are induced by interference and only apply rate adaptation in the situation of channel errors. However, authors in [26] point out that it is very hard for one node to directly jump to the best rate according to the BER estimation. They propose AccuRate to "replay" the signal dispersion in the channel on all possible rates and select the data rate which achieves optimal throughput. This "replay" action is simulated at the receiver and there is no need for the transmitter to transmit signals each time. AccuRate can outperform SoftRate at the expense of implementation complexity.

Authors in [27] utilize Log-Likelihood Ratio (LLR) estimation to determine the confidence of each demodulated bits and make those with low confidence retransmitted. Meanwhile, they also leverage these low-confidence bits and combine them from multiple failed transmissions by adding up their LLR. All the results will be fed into the FEC decoder to increase the decoding efficiency. Due to the lower retransmission overhead in this mechanism, a higher data rate can be selected to improve the channel capacity.



(a) SoftPHY hint calculation for each symbol in uncoded channel

(b) SoftPHY hint calculation for each codeword in coded channel

Figure 9: An example of SoftPHY hint calculation under the QPSK modulation. For uncoded channel, the hint φ_i is calculated as the distance between the received signal y_i's constellation point and the decoded symbol a_i's constellation point. For coded channel where a group of b bits are coded to B codeword, then modulated to B/k symbols, the SoftPHY hint is calculated for this b-bit stream, which is the sum of φ_i for each symbol. The codeword with a larger φ value is determined to be detected correctly with a higher probability.

Recitation [28] exploits the SoftPHY-like idea at the transmitter side. The authors observe that the wireless physical layer is deterministic when the processes of signal scrambling, encoding, interleaving, and modulating are known. Thus, before actual transmission, the transmitter can "rehearse" every operation in the decoding process to predict the received bits' confidence, called error event probability (EVP), based on which information, it then takes the best action for the following data transmission through some possible schemes such as unequal error protection, partial packet recovery, and rate adaptation. The authors further propose UnPKT [29] to add more FEC redundancies for the bits with higher estimated EVP in the packet, so as to ensure its decoding at the receiver side.

4.3. SIC

4.3.1. Introduction of SIC. The technology of successive interference cancellation (SIC) is one version of the interference cancellation techniques which harness the data structure in the interference to mitigate its harmful effect, thus improving the channel utilization. SIC was first implemented in WLANs since about 2008.

Suppose a received signal $y[n]$ contains two signals $y_1[n]$ and $y_2[n]$ from different transmitters:

$$y[n] = y_1[n] + y_2[n] + w[n], \tag{1}$$

where $w[n]$ indicates the Gaussian background noise.

Assume the signal strength of y_1 is much stronger than that of y_2, and

$$\text{SINR}_1 = \frac{y_1}{y_2 + w} \geq \beta, \tag{2}$$

where β is the SINR threshold to demodulate the signal.

In this situation, the signal y_1 can be detected correctly through the normal demodulation process. The demodulated bit stream for y_1 can be utilized to reconstruct its received signal y_1'. At this time, the signal y_2, whose SINR in the received signal y is obviously below β, can also be detected through subtracting y_1' from y if the following condition is satisfied:

$$\text{SINR}_2\left(y - y_1'\right) = \frac{y_2}{y - y_1' + w} \geq \beta. \tag{3}$$

Figure 10 illustrates an example of the SIC process in a scenario when two signals are collided, and both signals can be detected correctly through this process. The SIC process can be extended to k ($k > 2$) collided packets scenarios theoretically.

4.3.2. Survey of SIC. This technology has been exploited to increase concurrent transmissions or adapt optimal transmission rate.

(i) Avoid Interference and Exploit Concurrency. SIC is introduced in WLANs to mainly increase the channel capacity through enabling more concurrent transmissions. Authors in [5] first implement this technique in WLANs. Figure 11 gives a simple example for its application, where three clients intend to transmit their data packets to the AP. Different from the 802.11 standard which utilizes CSMA to avoid their mutual interference, SIC permits their simultaneous transmissions through carefully adjusting the transmission power to satisfy the SIC decoding requirements. After receiving the collided signal, the AP first decodes the packet 1 with the strongest power and then reconstructs and cancels this packet from the received signal; it then repeats this process to decode packet 2 and packet 3.

With a different opinion on SIC, authors in [30] investigate the throughput gain of SIC from the MAC layer perspective. They point out that the network performance can be improved through SIC only under some limit conditions for the transmission data rate and the SNR requirements. These conditions are so restrictive and hard to be satisfied, making the gains of SIC not achievable in practical networks.

The idea of SIC has also been employed in [31–33] to improve the network performance through Aloha-based random access, where the transmissions are divided into time slots, and each client randomly selects one slot for its

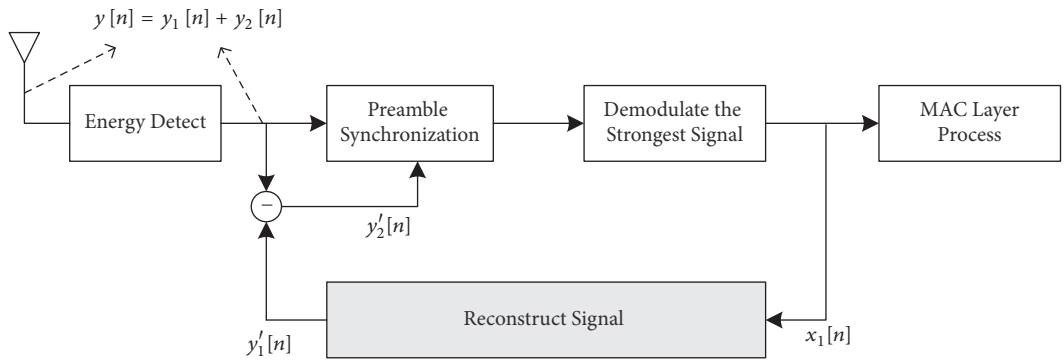
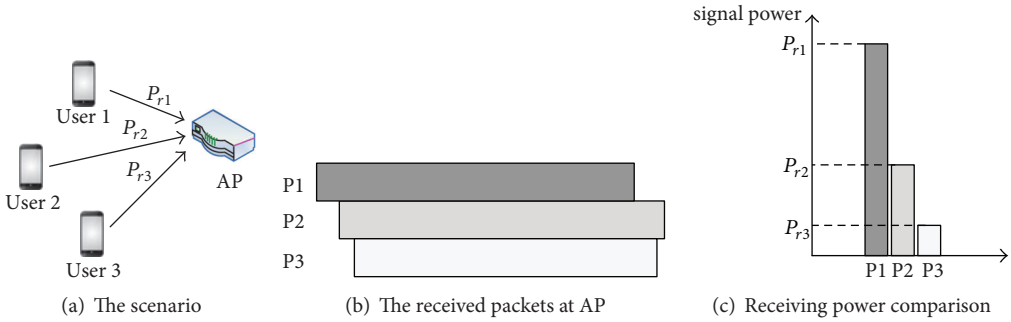


Figure 10: An example of SIC process when a node receives two signals $y_1[n]$ and $y_2[n]$ simultaneously, where the signal strength of $y_1[n]$ is much stronger than that of $y_2[n]$. The node first demodulates the strongest signal $y_1[n]$ and gets the corresponding transmitted bit stream $x_1[n]$; it then reconstructs the ideal received signal $y_1'[n]$ from $x_1[n]$ and subtracts $y_1'[n]$ from $y[n]$ to get the second signal $y_2[n]$; $y_2[n]$ is finally fed into the same preamble synchronization and demodulation process to get the corresponding bit stream $x_2[n]$. This scenario can be extended to k ($k > 2$) collided packets scenarios theoretically.



(a) The scenario (b) The received packets at AP (c) Receiving power comparison

Figure 11: A simple example when SIC is applied in WLAN. The three clients transmit their data packets to the AP simultaneously. Through adjusting the transmission power carefully, the receiving power of the three packets P_{r1}, P_{r2}, and P_{r3} can satisfy the SIC decoding requirement, making all the three packets decoded successfully.

data transmission. The SIC is applied to permit multiple transmissions in each slot, and the decoding constraints are satisfied through changing the clients' transmission power, data rate, etc.

ContraFlow [34] is a SIC MAC protocol designed for full-duplex wireless networks. Based on the integration of full-duplex and the CSMA/CA in the standard, it designs a dual-link which enables two links to be proceeded concurrently to both increase the spatial reuse and eliminate the hidden terminals. Contrabass [35] further exploits this technique in the MIMO system to enable the channel training when a receiver receives multiple packets from different transmitters. CSMA k-SIC [36] exploits the SIC in the distributed random access protocols and makes a receiver capable of cancelling up to k strongly interfering signals, through determining which links can be scheduled and which interferers a receiver must cancel.

BASIC [37] applies this technique in the enterprise WLANs. It utilizes the AP backbone to design an uplink transmission strategy, which permits simultaneous transmissions from multiple clients to APs and controls the data rates of the clients so as to collaboratively exploit the SIC mechanism at the APs to decode all the packets, through utilizing the diversity of received signal across multiple receivers.

(ii) Rate Adaptation. The basic concept of SIC has also been exploited with rateless coding to further improve the network performance, such as Strider [6] and AutoMAC [38], which will be discussed in the following part.

4.4. Rateless Coding

4.4.1. Introduction of Rateless Coding. Rateless coding has been introduced for a long time, starting from the automatic repeat request (ARQ) schemes. Its objective is to make the signals with fixed channel coding and modulation schemes decoded at any SNR environments. The first kind of practical rateless codes is LT codes [39], which is designed for the erasure channel where the transmission packets has some probability to be lost. Raptor codes [40] are later introduced to achieve a more computational efficient and capacity achieving performance. For the AWGN channel, a "layered" encoding and decoding manner [41] is proposed to combine the original codes with fixed rate to generate a rateless stream.

One of the main challenges of rateless coding is to reduce the processing complexity to make it more practical [42]. Some current works focus on designing practical rateless codes or the corresponding up-layer protocols and make

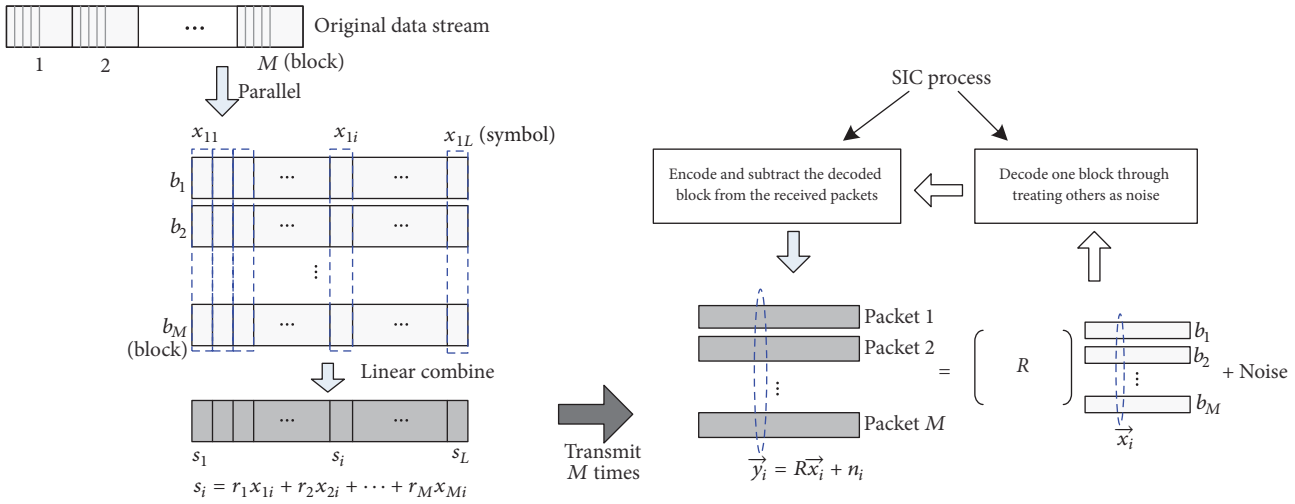


FIGURE 12: The rateless coding scheme in Strider. The original data stream is divided into M blocks, each with L complex symbols. The M blocks are linearly combined to form a new packet for transmission. The sender will create multiple packets using different linear combinations and transmit them until the receiver can decode all the M blocks. After receiving enough signals, the receiver first decodes the first block through treating the other blocks as noise, then reencodes it, and subtracts it from the received signal to proceed the next block. It will repeat this process until all the blocks are decoded.

implementations on the system testbeds to evaluate their performance.

4.4.2. Survey of Rateless Coding. The aim of rateless coding is to achieve optimal transmission rate under any wireless environments.

Strider [6] is the first work that exploits rateless coding in the WLAN-like scenarios and has been implemented on system testbed. It designs a rateless coding technique and combines it with SIC to permit concurrent transmissions of multiple coded streams, thus achieveing an optimal data rate in any SNR situation. As shown in Figure 12, the original data stream is divided into M blocks, each block with L complex symbols. The M blocks are linearly combined to form a new packet for transmission. The sender will create multiple packets using different linear combinations and transmit them until the receiver can decode all the M blocks. After receiving these packets, the receiver first decodes the first block through treating the other blocks as noise and then reencodes it and subtracts it from the received signal to proceed the next block. It will repeat this process until all the blocks are decoded. Through this way, the sender can achieve the optimal bit rate without knowing the channel states.

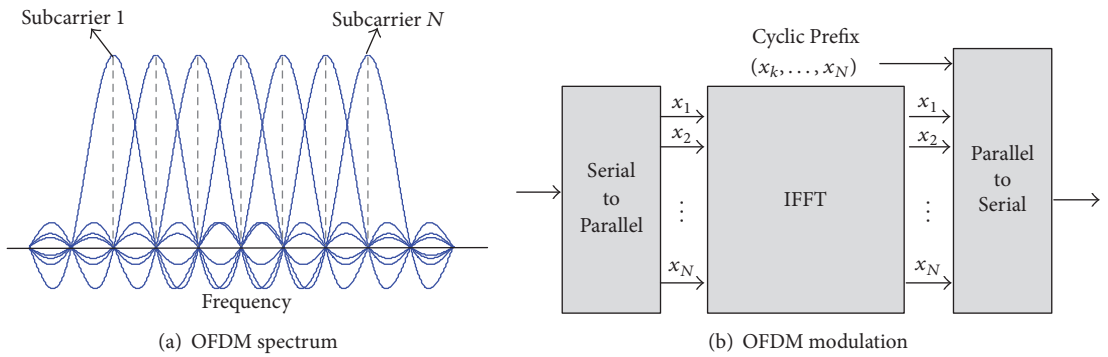
The authors further propose AutoMAC [38], a MAC protocol for Strider to improve its efficiency. In Strider, the number of rateless transmissions of each packet depends on the SNR environments at the receiver side, which information the transmitter does not know. AutoMAC makes the receiver ACK the corresponding transmitter once it determines that it has received enough number of signals to decode the packet. The ACKed transmitter then goes to the next packet transmission, without wasting the channel bandwidth.

Different from linear combination, spinal code [43] exploits a pseudo-hash function to transform the original data bits into encoded symbols. due to the sequential application of the hash function, this encoding process ensures that two similar streams with even one-bit difference have different coded sequences, thus more resilient to noise and bit errors. The authors also exploit the sequential structure of the encoding for the decoding process through a tree-searching method. They demonstrate spinal code's higher throughput and its ability to be deployed practically through hardware prototype.

RateMore [44] is a link-layer protocol design for rateless codes to determine how much data needed to be transmitted, so as to both guarantee the successful data decoding and avoid wasting the transmission time. The sender learns the decoding CDF (cumulative probability distribution function), which represents the probability distribution of the number of symbols required to decode a packet, based on the acknowledgment from the receiver. With both the decoding CDF and the feedback delay, an optimal transmission schedule is derived to maximize the network performance.

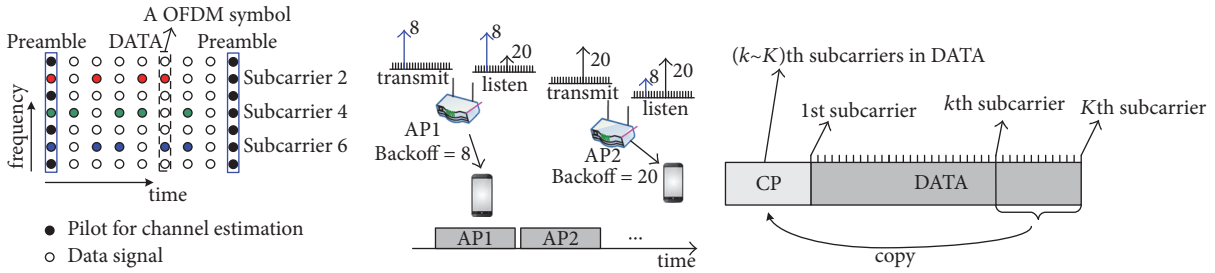
PRAC [45] exploits the feature of rateless linear coding to identify and recover the erroneous packet segments in the partial packet recovery context. The transmitter continually makes linear combination of k original packets to create new one for transmission. After receiving n ($n > k$) packets, the receiver begins to distinguish the erroneous symbols through utilizing a designed scheme called algebraic consistency rule check (ACR). For each symbol column, it iteratively conducts the ACR check, a searching algorithm to identify correct symbols, and conducts the CRC check until the ACR check is satisfied. This mechanism reduces the retransmitted segments, thus significantly improving the transmission efficiency.

4.5. Physical Unique Features. In this part, we will survey a series of features in the OFDM modulation process exploited



(a) OFDM spectrum (b) OFDM modulation

Figure 13: The OFDM modulation and spectrum.



(a) An example of attached information in hJam (b) An example of Back2F (c) The feature of cycle prefix exploited by CPRecycle

Figure 14: Three interference management schemes through exploiting the OFDM features.

for interference management; meanwhile, some features in other systems are also concluded.

4.5.1. Introduction of OFDM System. As shown in Figure 2, OFDM is part of the digital modulation and demodulation process in current 802.11 standard. An OFDM signal can be described as a group of closely spaced modulated subcarriers, which are orthogonal and will not interfere with each other theoretically, as demonstrated in Figure 13(a). The OFDM modulation process is shown in Figure 13(b); the modulated data are first passed through the serial-to-parallel (S/P) module to map the data in each subcarrier and then fed into IFFT (Inverse Fast Fourier Transform) to output the time-domain OFDM symbol. Cyclic Prefix is inserted in this stage to eliminate the intersymbol interference. The signal will finally be output as a serial stream for the following process.

4.5.2. Survey on Exploiting OFDM Features. To manage interference efficiently, some known OFDM features are exploited to avoid interference, increase concurrent transmissions, or reduce the coordination overhead.

(i) Avoid Interference and Exploit Concurrency. hJam [46] makes the control information attached to the data packet for transmission, so as to reduce its transmission overhead. It exploits a few of "clean" subcarriers which have no signal except noise in the packet preamble to carry the control information. It then proposes attachment coding to make the control information transmitted in these "clean" subcarriers of the data packet. As shown in Figure 14(a), the clean subcarriers are 2, 4, and 6, and the control information attached in subcarrier 2 is 0101100. The receiver can determine there is an attached signal on a subcarrier if it detects a relatively high level energy on that subcarrier. After detecting the attached signals, node can obtain the corresponding control messages. Meanwhile, the data signal can be recovered through detaching the attached control signal from the received signal. The authors further propose Attached-RTS [47] and FAST [48] to solve the hidden terminal and exposed terminal problems through exploiting the attached control information.

CPRecycle [49] exploits the redundancy in cycle prefix of the OFDM PHY to mitigate interference. As shown in Figure 14(c), the cycle prefix is a copy of a portion of the following symbol in the tail, and it is used to eliminate the intersymbol interference induced by the multipath propagation in the wireless channel. It is now well known to be overprovisioned. For example, the 802.11 recommends the CP duration to be 0.8 μs for 20 MHz bandwidth, but in most cases the multipath delay is only in the order of ns [50]. The authors get a key observation that when a receiver performs FFT from different positions in the redundancy cyclic prefix, the signal demodulation will not be affected but the interference from concurrent transmissions can be dramatically reduced. They then design an algorithm to find the optimal starting position to maximize the performance.

(ii) Reduce Coordination Overhead. T2F [51] and Back2F [52] propose to reduce the coordination overhead induced by the

backoff, through migrating the random backoff from time domain to frequency domain. Each node needs two antennas, one for data transmissions and the other for listening to all the subcarriers in the network. Before the transmission, each node selects a subcarrier and transmits signals in this subcarrier, while the subcarrier number indicates its backoff time. The node with the minimal subcarrier number wins the contention and transmits the data packet. As shown in Figure 14(b), when both AP1 and AP2 intend to transmit packets to their clients, AP1 wins the contention as its backoff value is 8. AP2 will transmit its packet immediately after AP1's transmission, as it has the subminimal backoff value. The authors also propose multiple contention rounds to combat some situations such as multiple contention domains. Different from T2F [51] and Back2F [52] which only use subcarriers to reduce the backoff time, REPICK [53] further exploits the OFDM subcarriers to reduce some other overhead induced by 802.11 MAC, including DIFS, and ACK; thus it can dramatically improve the network performance.

4.5.3. Survey on Other Systems. Some strategies exploit the known feature in other systems to reduce the coordination overhead and enable concurrent transmissions. Side Channel [54, 55] utilizes the Direct-Sequence Spread Spectrum (DSSS) system which has the ability to resist interferences to a certain extent [56]. The authors carefully design some "intended patterns" which carry the control information and make them transmitted simultaneously with the original data packet, so as to reduce the coordination overhead but without degrading the effective throughput of data transmissions. Coco [57] advocates simultaneous accesses from multiple senders to a shared channel, optimistically allowing collisions instead of simply avoiding them, through both utilizing the capture effect and exploiting the ability to tolerate collisions because of redundancy in the physical layer implementations. mZig [58] exploits the known shaping feature of the ZigBee physical layer design. Based on this shaping feature, it can resolve one m-packet collision by this collision itself and thus can achieve m-fold throughput improvement comparing with the legacy mechanisms.

Some strategies exploit the physical layer features to improve the retransmission efficiency. MISC [59] merges incorrect symbols from multiple transmissions to produce correct ones. It exploits constellation diversity by rearranging the constellation maps in retransmissions, so as to improve the combining and decoding efficiency at the receiver side. Epicenter [60] utilizes the backbone network in enterprise WLANs, where one packet transmitted by a client will be received by multiple APs and shared among each other. They exploit the multiple copies of the incorrectly received symbols to recover the transmitted signal and design a coarse representation of the symbols to reduce their overhead when transmitted between APs.

5. Discussion

The aim of the surveyed paper is to exploit the physical layer techniques to manage interference in wireless networks efficiently, so as to improve the network throughput. This part discusses the challenges of applying these techniques and gives some potential future directions for research in this area.

5.1. Discussions of the Physical Layer Techniques. The adoption of the physical layer techniques inspires new ways for designing high efficient channel access mechanisms. Here we want to discuss the advantages and challenges of these techniques as a summary.

5.1.1. Cross Correlation. Cross correlation can be exploited to detect both the control and data packets once a collision occurs. When this technology is utilized to recover the collided control packets, the coordination information can be exchanged among nodes more efficiently, thus avoiding interference, enabling concurrent transmissions, or reducing the coordination overhead. When this technology is utilized to recover the collided data packets, nodes have a new way to manage interference through embracing it but not avoiding it.

The main challenge of applying the cross correlation technology for interference management lies in the high computational overhead. According to Section 4.1.1., one cross correlation process needs L complex multiplication calculations. When leveraging this technology to recover the collided control packet and suppose the number of possible known sequences carried by the control packet is N, one node should conduct N cross correlation processes at each position of the received signal to determine which known sequence is detected [13, 22, 23]; this will lead to $N \times L$ complex multiplication calculations at each position. The computational overhead would be even higher when this technology is exploited to recover the collided data packets [18, 19], as the cross correlation should be conducted for multiple times when bootstrapping the packet decoding. None of the works has been implemented on off-the-shelf devices till now, and all of them are evaluated based on special platforms such as USRP [61] and WARP [62].

5.1.2. SoftPHY. SoftPHY provides PHY's confidence of each bit to the upper layer, so as to either increase the retransmission efficiency through only making the bits with low-confidence retransmitted or adapt the optimal transmission rate through calibrating the channel estimation.

The main problem of SoftPHY is its low performance when applied in poor wireless environments. SoftPHY relies on preamble to synchronize with the incoming signal and estimate the confidence of the following bits. However, under the situation of interference or comparatively high noise, the receiver would not be able to detect the preamble correctly, making the following bits decoding and confidence calculation not achievable. Authors in PPR [4] utilize a postamble at the end of each packet to make SoftPHY work when the preamble is corrupted. However, this design would not be practical in real networks as all parts of a packet may have a high probability to be corrupted when the wireless channel is poor.

5.1.3. SIC. SIC permits concurrent transmissions of multiple packets and detects them correctly when the receiving power

satisfies the constraints. In the SIC receiving process, one node first decodes the strongest signal and then subtracts it and decodes the inferior signal if the remaining SINR is above the threshold; the process can be repeated to decode more signals.

The practical hurdle of deploying SIC in wireless networks is how to satisfy the power constraints. Although some advances try to exploit SIC to improve the network throughput in a distributed way [34–36], it is intuitively very hard to enable multiple concurrent transmissions and make their receiving power satisfy the restrictive requirements through protocol design. Actually, some researchers have already investigated the throughput gain of SIC from the MAC layer perspective [30] and pointed out that the throughput gain can only be obtained under limit conditions for the transmission data rate and the SNR requirements, which are very hard to be satisfied in practical networks. Centralized coordination may be a better choice for the SIC application, as proposed by Basic [37] which utilizes the AP backbone to centrally control the data rate and transmission power of simultaneous transmissions.

5.1.4. Rateless Coding. Rateless coding makes the undecodable packet in a fixed data rate be recovered by combining multiple transmissions, resulting in different data transmission rate. Comparing with the traditional rate adaptation mechanisms which choose the best data rate through the estimated SNR of the wireless channel, this method can achieve the optimal bit rate without knowing the channel state in advance.

One of the main challenges of rateless coding is to reduce its processing complexity to make it more practical. Some current advances have focused on this challenge and designed mechanisms with affordable computational complexity for implementation, such as Strider [6] and spinal codes [43]. However, these kinds of mechanisms incur significant modification in the baseband process as they require the transmitters to transmit correlated symbols, making these mechanisms hard to be embedded to current wireless devices.

5.1.5. Unique Physical Layer Features. Several unique features in the physical layer are exploited for interference management, such as the OFDM subcarriers, redundancy design in both OFDM and DSSS systems.

The OFDM subcarriers are exploited to convey the coordination information, so as to effectively solve the hidden or exposed terminal problem [47, 48], or to reduce the coordination overhead induced by backoff [52]. It is a good idea to leverage the OFDM subcarriers to convey the coordination information. However, some of the current works need extra antenna to accomplish the transmission [48, 52]. Although multiple antennas are commonly deployed in current wireless devices due to the adoption of MIMO (multi-input and multi-output) technology in recent 802.11n and 802.11ac standards, it is still uneconomical to let one antenna only transmit the control information.

Coding redundancy in both OFDM and DSSS systems is originally designed for interference resistance, so that the signals can be decoded in comparatively lower SINR environments. When the control information is transmitted simultaneously with the data packets, it becomes an interference of the data packet and will reduce the data packet's received SINR, making the system have lower ability to resist other interference. Thus, these mechanisms will only work well in the high SNR environments.

5.1.6. Summary of Limitations. Based on the above discussions, we summarize the main limitations to deploy these physical layer techniques to real networks as follows:

(i) High computational overhead: the deploying of each physical layer technique will certainly induce extra computational overhead to the systems, which has become the main hurdle to make them applicable, such as cross correlation and rateless coding. Most of the works have just been evaluated based on special software defined radio platforms which have huge computation resources; few of them can be implemented on commercial devices.

(ii) Rigorous system constraints: the successful application of some physical layer techniques can only occur under rigorous constraints. For example, SoftPHY requires uncollided preamble for synchronization and SIC has strict power constraints for the multiple receiving packets. These requirements can hardly be fully satisfied in real networks, especially the mobility scenarios when the infrastructures (APs) and stations are mobile, making the interference and receiving power more uncertain in the network. This issue would significantly interrupt the performance improvement of the proposed mechanisms.

(iii) Hardware incompatibility: nearly all the physical layer techniques have added or revised some components in the baseband signal process, making them incompatible to current commercial devices.

5.2. Discussions of Future Directions

5.2.1. Overcome the Limitations of Physical Layer Techniques. As discussed above, the current mechanisms based on exploiting the physical layer techniques have some serious limitations which impede their applications, such as the high computational overhead, the rigorous system constraints, and hardware incompatibility. Overcoming these limitations would obviously enhance their deploy ability in real networks. For example, the hardware compatibility is now becoming an important specification to evaluate the proposed mechanisms, and this issue has been considered by many current works based on physical layer design, such as WeBee [63]. It is worth rethinking the mechanism design from this point of view.

5.2.2. Further Explore the Physical Layer Techniques. Current advances have already presented some examples on exploiting the physical layer techniques for interference management and demonstrated exciting throughput improvement through hardware experiments or simulations. However, we consider the potential of these techniques in these areas is far

from fully unleashed. Researchers may find some other effective methods to improve the network performance through utilizing these physical layer techniques. For example, SIC has been exploited only to enable concurrent transmissions of data packets, how to design the interference management mechanism when it is utilized for concurrent transmissions of data and control packets, and what is the expected performance?

5.2.3. Manage Interference Centrally. Researchers have observed that some of the problems in the legacy interference management, such as the exposed and hidden terminals and the large coordination overhead, are intrinsic properties existing in the distributed coordination methods [10]. That means optimizing CSMA or designing new distributed interference management mechanisms can only mitigate these problems but not eliminate them. Actually, the 802.11 standard already recommends a centralized coordination mechanism, called PCF (point coordination function), to make the AP schedule the uplink and downlink transmissions centrally to achieve a higher throughput. However, this mechanism is rarely used in current WLANs as it is not suitable to a common scenario when there are multiple APs in the network.

Nowadays, many researchers focus on centralized coordination mechanisms through exploiting the wired backbone network among APs in the enterprise WLANs. For example, Symphony [19] extends the idea of Zigzag [18] in the multiple-AP scenario through utilizing the AP backbone; Basic [37] also utilizes this backbone to control the data rate and simultaneous transmissions in the uplink direction and exploits SIC at APs to decode all the packets. Designing centralized interference management mechanisms seems like a more promising way to achieve high performance in wireless networks. This concept can also be extended for designing efficient rate adaptation and retransmission schemes. In addition, when considering the infrastructure mobility scenario with multiple mobile APs but without the backbone network, how to solve this kind of issues is a very meaningful topic.

5.2.4. Extend the Design to MIMO. Current researches on interference management through exploiting physical layer techniques mainly focus on the single stream systems; only very few of them are designed or tested based on the MIMO systems, such as Recitation [28]. Actually, there have already been some researches on theoretically analyzing the performance of exploiting these physical layer techniques to the MIMO communications, for example, exploiting rateless coding for MIMO Fading Channels [64] and proposing SIC for large-scale MIMO configuration [65] and 5G system [66]. However, these works are accomplished from the communication perspective and they do not consider the suitable design of channel access mechanisms in the network. It would be worthy of expectation to exploit these physical layer techniques in the MIMO systems.

6. Conclusion

In this paper, we investigate the interference management mechanisms which exploit the physical layer techniques in wireless networks. We first give some background information about the PHY and MAC characteristics in the 802.11 standards and then discuss the problems existing in the current interference management process. After that, we introduce five kinds of physical layer techniques and investigate how these techniques are exploited to manage interference and improve the network throughput. Based on this study, we finally present some discussions from both the physical layer and MAC layer perspective. We hope this survey would help the readers to summarize the current research progress and inspire their future work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by grants from China NSFC (61702343, 61472259, and 61601308), China Postdoctoral Funding (2017M610548), Joint Key Project of the National Natural Science Foundation of China (U1736207), Guangdong Natural Science Foundation (2017A030312008), Hong Kong RGC (PolyU-521312), Hong Kong PolyU (4-BCB6, M-N020, G-YBXY), Shenzhen Science and Technology Funding (JCYJ20170302140946299 and JCYJ20170412110753954), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161064), Guangdong Talent Project (2014TQ01X238 and 2015TX01X111), and GDUPS (2015). Kaishun Wu is the corresponding author.

References

[1] J. Xu, J. Yao, L. Wang, Z. Ming, K. Wu, and L. Chen, "Narrowband Internet of Things: Evolutions, Technologies and Open Issues," *IEEE Internet of Things Journal*, 2017.

[2] Cisco, *Cisco Visual Networking Index: global mobile data traffic forecast update, 2016 - 2021 white paper*, 2017.

[3] K. Tan, J. Fang, Y. Zhang et al., "Fine-grained channel access in wireless lan," in *Proceedings of the ACM SIGCOMM Conference (SIGCOMM '10)*, pp. 147–158, New Delhi, India, August 2010.

[4] K. Jamieson, *The SoftPHY abstraction: From packets to symbols in wireless network design [Ph.D. thesis]*, MIT, 2008.

[5] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: interference cancellation for wireless LANs," in *Proceedings of the 14th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '08)*, pp. 339–350, September 2008.

[6] A. Gudipati and S. Katti, "Automatic rate adaptation and collision handling," *Computer Communication Review*, vol. 41, no. 4, pp. 158–169, 2011.

[7] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding congestion in IEEE 802.11b wireless networks," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC '05)*, October 2005.

[8] Y. He, R. Yuan, J. Sun, and W. Gong, "Semi-random backoff: towards resource reservation for channel access in wireless LANs," in *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP '09)*, pp. 21–30, IEEE, Princeton, NJ, USA, October 2009.

[9] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing exposed terminals in wireless networks," in *Proceedings of the ACM NSDI*, 2008.

[10] Z. Yang, J. Zhang, K. Tan, Q. Zhang, and Y. Zhang, "Enabling TDMA for today's wireless LANs," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '15)*, pp. 1436–1444, Kowloon, Hong Kong, April 2015.

[11] L. Wang, X. Qi, J. Xiao, K. Wu, M. Hamdi, and Q. Zhang, "Exploring Smart Pilot for Wireless Rate Adaptation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4571–4582, 2016.

[12] S. Sen and N. Santhapuri, "Moving away from collision avoidance: towards collision detection in wireless networks," in *Proceedings of the ACM HotNets*, Moving away from collision avoidance, 2009.

[13] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: carrier sense multiple access with collision notification," in *Proceedings of the 16th Annual Conference on Mobile Computing and Networking (MobiCom '10)*, pp. 25–36, Chicago, Ill, USA, September 2010.

[14] T. Xiong, J. Zhang, J. Yao, and W. Lou, "Symbol-level detection: A new approach to silencing hidden terminals," in *Proceedings of the 20th IEEE International Conference on Network Protocols, ICNP '12*, USA, November 2012.

[15] E. Magistretti, O. Gurewitz, and E. W. Knightly, "802.11ec: collision avoidance without control messages," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 65–76, ACM, Istanbul, Turkey, August 2012.

[16] J. Yao, T. Xiong, J. Zhang, and W. Lou, "On Eliminating the Exposed Terminal Problem Using Signature Detection," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2034–2047, 2016.

[17] J. Yao, W. Lou, C. Yang, and K. Wu, "Efficient interference-aware power control in wireless ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '17)*, pp. 1–6, Paris, France, May 2017.

[18] S. Gollakota and D. Katabi, "Zigzag decoding: combating hidden terminals in wireless networks," in *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '08)*, pp. 159–170, August 2008.

[19] T. Bansal, B. Chen, P. Sinha, and K. Srinivasan, "Symphony: Cooperative packet recovery over the wired backbone in enterprise WLANs," in *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking, MobiCom '13*, pp. 351–362, USA, October 2013.

[20] X. Zhang and K. G. Shin, "DAC: Distributed asynchronous cooperation for wireless relay networks," in *Proceedings of the IEEE INFOCOM '10*, USA, March 2010.

[21] X. Zhang and K. G. Shin, "Chorus: collision resolution for efficient wireless broadcast," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, San Diego, Calif, USA, March 2010.

[22] S. Misra and M. Khatua, "Semi-distributed backoff: collision-aware migration from random to deterministic backoff," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1071–1084, 2015.

[23] J. Zhao, S. Fan, D.-A. Li, and B. Zhao, "Collision tolerance: Improving channel utilization with correlatable symbol sequences in wireless networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 678735, 2015.

[24] K. Jamieson and H. Balakrishnan, "PPR: partial packet recovery for wireless networks," in *Proceedings of the ACM SIGCOMM Conference on Computer Communications '07*, pp. 409–420, Japan, August 2007.

[25] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *Proceedings of the ACM SIGCOMM conference '09*, Barcelona, Spain, August 2009.

[26] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "AccuRate: constellation based rate estimation in wireless networks," in *Proceedings of the NSDI*, 2010.

[27] M. O. Khan, L. Qiu, A. Bhartia, and K. C. Lin, "Smart Retransmission and Rate Adaptation in WiFi," in *Proceedings of the 23rd IEEE International Conference on Network Protocols (ICNP '15)*, pp. 54–65, San Francisco, CA, USA, November 2015.

[28] Z. Li, Y. Xie, M. Li, and K. Jamieson, "Recitation: Rehearsing wireless packet reception in software," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom '15*, pp. 291–303, Paris, France, September 2015.

[29] Y. Xie, Z. Li, M. Li, and K. Jamieson, "Augmenting wideband 802.11 transmissions via unequal packet bit protection," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM '16*, USA, April 2016.

[30] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Successive interference cancellation," in *Proceedings of the the Ninth ACM SIGCOMM Workshop*, pp. 1–6, Monterey, California, October 2010.

[31] S. P. Weber, J. G. Andrews, X. Yang, and G. de Veciana, "Transmission capacity of wireless ad hoc networks with successive interference cancellation," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 53, no. 8, pp. 2799–2814, 2007.

[32] J. Blomer and N. Jindal, "Transmission capacity of wireless ad hoc networks: Successive interference cancellation vs. joint detection," in *Proceedings of the 2009 IEEE International Conference on Communications, ICC '09*, Dresden, Germany, June 2009.

[33] M. Mollanoori and M. Ghaderi, "On the performance of successive interference cancellation in random access networks," in *Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON '12*, pp. 461–469, Seoul, South Korea, June 2012.

[34] N. Singh, D. Gunawardena, A. Proutiere, B. Radunović, H. V. Balan, and P. Key, "Efficient and fair MAC for wireless networks with self-interference cancellation," in *Proceedings of the International Symposium of on Modeling and Optimization of Mobile, Ad Hoc, and Wireless Networks (WiOpt '11)*, pp. 94–101, IEEE, Princeton, NJ, USA, May 2011.

[35] S. Yoon, I. Rhee, B. C. Jung, B. Daneshrad, and J. H. Kim, "Contrabass: Concurrent transmissions without coordination for ad hoc networks," in *Proceedings of the IEEE INFOCOM*, pp. 1134–1142, China, April 2011.

[36] A. Sankararaman and F. Baccelli, "CSMA k-SIC - A class of distributed MAC protocols and their performance evaluation," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM '15*, pp. 2002–2010, Kowloon, Hong Kong, May 2015.

[37] W. Zhou, T. Das, L. Chen, K. Srinivasan, and P. Sinha, "BASIC: backbone-assisted successive interference cancellation," in *Proceedings of the the 22nd Annual International Conference, ACM MOBICOM '16*, pp. 149–161, New York, NY, USA, October 2016.

[38] A. Gudipati, S. Pereira, and S. Katti, "AutoMAC: rateless wireless concurrent medium access," in *Proceedings of the 18th annual international conference on Mobile computing and networking, ACM MOBICOM '12*, Istanbul, Turkey, 2012.

[39] M. Luby, "LT codes," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, IEEE, Vancouver, Canada, November 2002.

[40] A. Shokrollahi, "Raptor codes," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[41] U. Erez, M. D. Trott, and G. W. Wornell, "Rateless coding for Gaussian channels," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 58, no. 2, pp. 530–547, 2012.

[42] F. Mehran, K. Nikitopoulos, P. Xiao, and Q. Chen, "Rateless wireless systems: Gains, approaches, and challenges," in *Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing, ChinaSIP '15*, pp. 751–755, China, July 2015.

[43] J. Perry, P. A. Iannucci, K. E. Fleming, H. Balakrishnan, and D. Shah, "Spinal codes," in *Proceedings of the ACM SIGCOMM Conference Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM '12*, pp. 49–60, Helsinki, Finland, August 2012.

[44] P. Iannucci, J. Perry, H. Balakrishnan, and D. Shah, "No symbol left behind: A link-layer protocol for rateless codes," in *Proceedings of the International Conference on Mobile Computing and Networking*, pp. 17–27, 2012.

[45] G. Angelopoulos, M. Medard, and A. P. Chandrakasan, "Harnessing Partial Packets in Wireless Networks: Throughput and Energy Benefits," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 694–704, 2017.

[46] K. Wu, H. Li, . Lu Wang et al., "HJam: Attachment transmission in WLANs," in *Proceedings of the IEEE INFOCOM - IEEE Conference on Computer Communications*, pp. 1449–1457, Orlando, FL, USA, March 2012.

[47] L. Wang, K. Wu, and M. Hamdi, "Attached-RTS: Eliminating an exposed terminal problem in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1289–1299, 2013.

[48] L. Wang, K. Wu, and M. Hamdi, "Combating hidden and exposed terminal problems in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 4204–4213, 2012.

[49] S. Rathinakumar, B. Radunovic, and M. K. Marina, "CPRecycle: Recycling cyclic prefix for versatile interference mitigation in OFDM based wireless systems," in *Proceedings of the 12th ACM Conference on Emerging Networking Experiments and Technologies, ACM CoNEXT '16*, pp. 67–81, USA, December 2016.

[50] C.-P. Lim, J. L. Volakis, K. Sertel, R. W. Kindt, and A. Anastasopoulos, "Indoor propagation models based on rigorous methods for site-specific multipath environments," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 6, pp. 1718–1725, 2006.

[51] S. Sen, R. R. Choudhury, and S. Nelakuditi, "Listen (on the frequency domain) before you talk," in *Proceedings of the the Ninth ACM SIGCOMM Workshop*, pp. 1–6, Monterey, California, October 2010.

[52] S. Sen, R. Roy Choudhury, and S. Nelakuditi, "No time to countdown: Migrating backoff to the frequency domain," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, MobiCom'11 and Co-Located Workshops*, pp. 241–252, USA, September 2011.

[53] X. Feng, J. Zhang, Q. Zhang, and B. Li, "Use your frequency wisely: Explore frequency domain for channel contention and ACK," in *Proceedings of the IEEE Conference on Computer Communications, INFOCOM '12*, pp. 549–557, USA, March 2012.

[54] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. Ni, "Side channel: Bits over interference," in *Proceedings of the Sixteenth Annual International Conference, ACM MOBICOM '10*, pp. 13–24, Chicago, Illinois, USA, September 2010.

[55] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. M. Ni, "Side channel: Bits over interference," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1317–1330, 2012.

[56] K. Wu, H. Tan, H.-L. Ngan, Y. Liu, and L. M. Ni, "Chip error pattern analysis in IEEE 802.15.4," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 543–552, 2012.

[57] X. Ji, Y. He, J. Wang et al., "On Improving Wireless Channel Utilization: A Collision Tolerance-Based Approach," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 787–800, 2017.

[58] L. Kong and X. Liu, "MZig: enabling multi-packet reception in ZigBee," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom '15)*, pp. 552–565, Paris, France, September 2015.

[59] J. Ou, Y. Zheng, and M. Li, "MISC: Merging incorrect symbols using constellation diversity for 802.11 retransmission," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM '14*, pp. 2472–2480, Canada, May 2014.

[60] M. Gowda, S. Sen, R. R. Choudhury, and S.-J. Lee, "Cooperative packet recovery in enterprise WLANs," in *Proceedings of the 32nd IEEE Conference on Computer Communications, IEEE INFOCOM '13*, pp. 1348–1356, Italy, April 2013.

[61] Ettus Inc, Universal software radio peripheral.

[62] "WARP: Wireless Open Access Research Platform," https://warpproject.org/trac.

[63] Z. Li and T. He, "WEBee," in *Proceedings of the the 23rd Annual International Conference*, pp. 2–14, Snowbird, Utah, USA, October 2017.

[64] F. Yijia, L. Lifeng, E. Erkip, and H. V. Poor, "Rateless coding for MIMO fading channels: performance limits and code construction," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1288–1292, 2010.

[65] A. Elghariani and M. Zoltowski, "Successive interference cancellation for large-scale MIMO OFDM," in *Proceedings of the IEEE International Conference on Electro/Information Technology, EIT '15*, pp. 657–661, USA, May 2015.

[66] R. Ruby, S. Zhong, H. Yang, and K. Wu, "Enhanced Uplink Resource Allocation in Non-Orthogonal Multiple Access Systems," *IEEE Transactions on Wireless Communications*, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9567302, 11 pages
https://doi.org/10.1155/2018/9567302

Research Article

Leveraging Mobile Nodes for Preserving Node Privacy in Mobile Crowd Sensing

Qinghua Chen,[1,2] **Shengbao Zheng**,[3] **and Zhengqiu Weng**[2]

[1]*College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou, China*
[2]*Department of Information Technology, Wenzhou Vocational & Technical College, Wenzhou, China*
[3]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*

Correspondence should be addressed to Qinghua Chen; kegully@qq.com

Received 29 December 2017; Accepted 25 March 2018; Published 30 April 2018

Academic Editor: Lu Wang

Mobile crowd sensing has been a very important paradigm for collecting sensing data from a large number of mobile nodes dispersed over a wide area. Although it provides a powerful means for sensing data collection, mobile nodes are subject to privacy leakage risks since the sensing data from a mobile node may contain sensitive information about the sensor node such as physical locations. Therefore, it is essential for mobile crowd sensing to have a privacy preserving scheme to protect the privacy of mobile nodes. A number of approaches have been proposed for preserving node privacy in mobile crowd sensing. Many of the existing approaches manipulate the sensing data so that attackers could not obtain the privacy-sensitive data. The main drawback of these approaches is that the manipulated data have a lower utility in real-world applications. In this paper, we propose an approach called P^3 to preserve the privacy of the mobile nodes in a mobile crowd sensing system, leveraging node mobility. In essence, a mobile node determines a routing path that consists of a sequence of intermediate mobile nodes and then forwards the sensing data along the routing path. By using asymmetric encryptions, it is ensured that a malicious node is not able to determine the source nodes by tracing back along the path. With our approach, upper-layer applications are able to access the original sensing data from mobile nodes, while the privacy of the mobile node is not compromised. Our theoretical analysis shows that the proposed approach achieves a high level of privacy preserving capability. The simulation results also show that the proposed approach incurs only modest overhead.

1. Introduction

Mobile crowd sensing has been a very important paradigm for collecting sensing data from a large number of mobile nodes dispersed over a wide area. A mobile crowd sensing system [1] has the following basic framework of data collection. First, smartphone nodes collect environmental data with the embedded sensors, such as noise level, air pollution level, GPS trajectories [2, 3], and radio signal strength [4–6]. Next, these nodes send the data directly to the central server (note that the sever is able to get the ID of the data sender). Finally, the server uses all the collected data to do some computations and analysis and finally draw some conclusions for further use. A wide spectrum of applications of mobile crowd sensing is envisioned, such as road traffic sensing [7], traffic light sensing [8], urban monitoring [9], and indoor localization [10, 11].

Although it provides a powerful means for sensing data collection, mobile nodes are subject to privacy leakage risks since the sensing data from a mobile node may contain sensitive information about the sensor node such as physical locations [12]. For instance, one intends to collect the noise level of a city and proposes a task. The sensing data contain not only the noise information, but also the location information acquired by GPS. Therefore, the data server easily matches the participant ID and the user's location. If the sensing is continuous, the time-series data may even reveal the trajectories of participants. This can be used to locate the sensitive locations (e.g., home, workplace) of participants. A smartphone user may be willing to participate in sensing but may not want to disclose his location. Thus, it is of great importance to preserve privacy throughout the data collection process in mobile crowd sensing.

Therefore, it is essential for mobile crowd sensing to have a privacy preserving scheme to protect the privacy of mobile nodes. It is challenging for mobile crowd sensing to preserve node privacy while maintaining a high level of the utility of the collected sensing data. A number of approaches [13, 14] have been proposed for preserving node privacy in mobile crowd sensing. Many of the existing approaches manipulate the sensing data so that attackers could not obtain the privacy-sensitive data. The main drawback of these approaches is that the manipulated data have a lower utility in real-world applications.

In this paper, we propose an approach called P^3 to preserve the privacy of the mobile nodes in a mobile crowd sensing system. In essence, a mobile node determines a routing path that consists of a sequence of intermediate mobile nodes and then forwards the sensing data along the routing path. By using asymmetric encryptions, it is ensured that a malicious node is not able to determine the source nodes by tracing back along the path. Note that our approach leverages mobile nodes to protect the privacy of a node generating sensing data. In addition, mobile nodes are dynamic and may come and leave the system at any time. Routing paths consist of different sets of mobile nodes. With our approach, upper-layer applications are able to access the original sensing data from mobile nodes, while the privacy of the mobile node is not compromised. Our theoretical analysis shows that the proposed approach achieves a high level of privacy preserving capability. The simulation results also show that the proposed approach incurs only modest overhead.

The rest of the paper is structured as follows. Section 2 reviews the related work. Section 3 presents the problem model and formally defines the problem. Section 4 presents the proposed approach of P^3. Section 5 provides the theoretical analysis of our approach. The performance evaluation is presented in Section 6. Finally, we conclude our paper in Section 7.

2. System Model and Problem Formulation

2.1. System Model. In this paper, our system is composed of three parts—smartphones, a central server, and a public key handler. Each smartphone is regarded as a node in our system, and each one knows its own data. Each node also has a public key and a private key for encryption and decryption. Just as its name, the public key can be seen by every other node, and all the public keys are stored in the public key handler. However, on the contrary, the private key is only kept by the node itself. No one can know it except itself. The central server intends to collect all the sensing data provided by nodes and utilizes them to form knowledge.

Now assume that there are N online nodes in our system. The nodes are denoted as u_i ($i = 1, 2, 3, \ldots, N$). Besides, assume each node has M sensing data; therefore, the sensing data are denoted as

$$D_i = \left\{d_{i1}, d_{i2}, d_{i3}, \ldots, d_{ij}\right\} \tag{1}$$

($i = 1, 2, 3, \ldots, N$; $j = 1, 2, 3, \ldots, M$), where d_{ij} means the jth data of node u_i. In particular, when each node has only one data, the data set of node u_i has only one parameter d_{i1}, and we simplify it to D_i.

Furthermore, the data collection function of the server S is denoted as f. Obviously,

$$f = \bigcup_{i=1}^{N}\bigcup_{j=1}^{M} d_{ij}. \tag{2}$$

In particular, when each node has only one sensing data, it is simplified to

$$f = \bigcup_{i=1}^{N} D_i. \tag{3}$$

We define a collection set of sensing data

$$C = \{D_1, D_2, D_3, \ldots, D_i\}. \tag{4}$$

Besides, for each node, it has a public key and a private key. Take node u_i as an example; its public key is denoted as P_{K_i} and its private key is denoted as S_{K_i}. All the public keys are stored at the public key handler, H, and it has a set of public keys

$$S = \left\{P_{K_1}, P_{K_2}, P_{K_3}, \ldots, P_{K_i}\right\}. \tag{5}$$

There also exists a global clock in our system, and every node synchronizes their clocks to it to get the current time t. The key handler also calculates a forwarding threshold T for each node. Each time a node attempts to forward a data, it needs to check whether the current time t is before T ($t < T$). If not, drop the data.

At last, we use a package to represent the data transferred on the forwarding routes, which is denoted as P, such as $P_1, P_2, \ldots, P_m$.

The definitions of all notations adopted are described in "Notations."

2.2. Design Requirements. Obviously, the most significant aspect of our privacy preserving data collection approach is protecting the privacy. But there also exist other aspects such as the efficiency of data forwarding that we need to consider. The following two aspects are the main requirements of our privacy preserving data collection approach.

(1) Preserving Node Privacy. We treat each smartphone as a node in our system, either honest or malicious. The attacker in our system can either be the server or any other node in the mobile crowd sensing network. The target of our aim is to realize anonymity of sensing data and make sure the server cannot match the sensing data with their originators. In other words, our goal is that the server could only get the sensing data while not knowing whom this data belongs to.

(2) Maintaining High Efficiency. The mobile crowd sensing system is usually a real-time system, in which the real-time attribute of sensing data is significant. In addition, due to the limitation of the smartphones, the instability of battery duration, and the availability of smartphones, the efficiency

of our data collection approach must be taken into account. The processing ability was also a limitation in the past, but as the development of smartphones, it becomes not as important as before. In this paper, we use the overhead to evaluate the efficiency, including the time cost of encryption, decryption, data forwarding, and storage utilization.

2.3. The Attack Model. In our system, we have three different types of entities: a server, a public key handler, and N online smartphone nodes. First, we assume the public key handler is fully trusted. Second, we assume that there are malicious nodes that collude with the server and the number is up to γN $(0 < \gamma < 1)$. We also assume honest nodes can keep their private keys securely. Each node sees the package addressed to it but cannot see packages between other nodes. Honest nodes will forward the new package at the time point as instructed in the old package. The goal of the adversary is to identify the originator of the package from the given piece of data. We suppose that the attacker is successful if he/she determines the node that generated the data, out of all the nodes in the network.

3. Design of P^3

In this section, we present two parts: the first part is an overview of our privacy preserving data collection approach, and we describe our approach in detail in the second part. The approach consists of four phases: setup, route selection, data encryption, and data forwarding.

Figure 1 shows the system architecture of our approach. The proposed system consists of a central server, a public key handler, and some smartphone devices. The central server aims to collect all the sensing data. And the public key handler is responsible for managing all the public keys.

In our system, the public key handler has an additional function—checking the state of a node. It says *Hello* to each node periodically. If a node is online, then its name will be on the online list. However, on the contrary, if a node is offline, its name will be removed from the list. This guarantees that there will not be an offline node on the forwarding route which leads to package missing. Besides, the selection of forwarding routes is also conducted by the public key handler.

Each node obtains sensing data through the sensors embedded in their smartphones. It then sends a sending query to the public key handler and encrypts the original data into a package with the public keys provided by the public key handler and then sends this package to the next node.

Once the next node receives this data, it decrypts the package with its own private key. Then it will get three parts from it—the address of the next node, the time threshold, and the package to be sent. After checking the time requirement, the next step is to pass the package to the next node and iterate this until the central server gets the data. What the server needs to do is to add this data into the data collection set.

The first step of our approach is the system setup. In a crowd sensing system, nodes may come and go. In this step, the central server first needs to confirm the nodes and then inform them the sensing task. Besides, each node should get prepared for sensing and initialize their sensors. In order to encrypt the data, each node should build a pair of key—a public key used for encrypting the data and a private key for decrypting the package. This pairs of keys are constant and will not change. The private key is stored by the node locally, while the public key is stored by a fully trusted third party—the public key handler.

The public key handler acquires the name list of nodes from the central server and contact with them. It triggers a query by a *Hello* message to each node to make sure whether they are online or not. The first time a node receives the query, it passes its public key to the public key handler to answer *Hello*. After all the available nodes pass their public keys to the public key handler, it then forms a public key set of all the nodes. Meanwhile, an available nodes list is built as well. Besides, due to the limitation of smartphones, some nodes may be unavailable at some time. So we assume the public key handler is also in charge of checking the state of the nodes. It may send a *Hello* message to all the nodes. As soon as a node receives the message, it will answer with a signal. If the public key handler does not receive this signal, this node will be considered to be unavailable and cannot be utilized for forwarding package. Then it will be removed from the available nodes list. By this way, the public key handler checks whether a smartphone node is online.

This process is carried out periodically. The period will be neither too long nor too short. If it is set too long, some nodes may have been offline but not removed from the available nodes list, which will lead to some forwarding routes' breaking. On the other hand, if set too short, the communication overhead may be too large. Dynamics does not create a problem to our protocol. The data transmission finishes in a very short time. In case a transmission is stopped by a node going offline, the data is just ignored. In a crowd sensing system, data are typically redundant and small loss of data is tolerable.

3.1. Determining a Data Forwarding Path. The second step is to select a suitable route to forward the sensing data. All the mobile phones are addressed by IP addresses. As we have assumed before, there exists an available nodes list in the key handler, which is fully trusted. So it can generate a forwarding route for each node and send the routes to them. In this way, each data originator can easily know who can be depended on to upload its data. During the system setup step, we need to set a hop number n, which means how many nodes participate in this forwarding process. However, we can also rule a range for n, which has an upper bound and a lower bound, to avoid attackers using the constant number of hops to guess the data originator. To simplify the analysis, we only discuss the constant n. On the forwarding route, except the beginning node, there is also the need for $n - 1$ extra nodes. As we have set in the model part, there are N online nodes in the system, so this route selection problem is equivalent to a combination problem, that is, to select $n - 1$ nodes from $N - 1$ nodes, so the number of combination methods is

$$C_{N-1}^{n-1} = \frac{(N-1)!}{(n-1)!\,(N-n)!}. \tag{6}$$

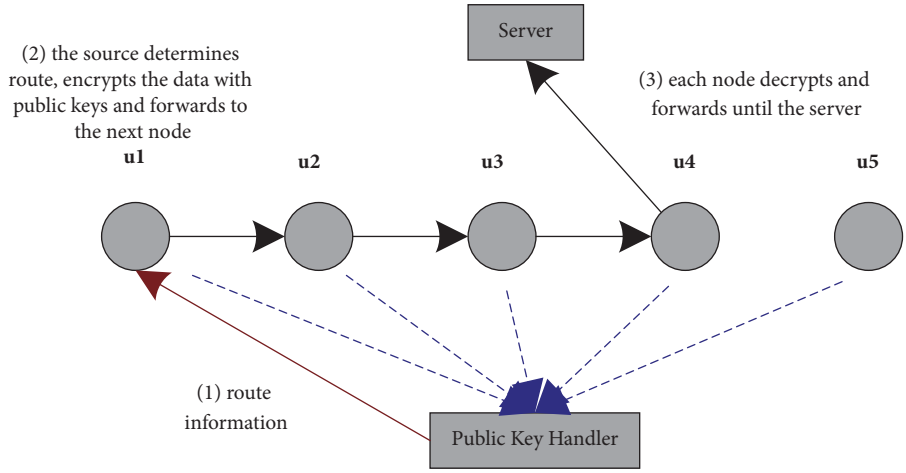


Figure 1: The main framework of the proposed approach.

It also means the number of routes can be selected when a sensing data originator intends to pass a data to the server. What to do next is just to select one route from them. The route information that public key handler sends to a node consists of the intermediate nodes' public key and their address.

Note that we assume that the data communications between any pair of nodes are always possible through the Internet. It is practical in the real world since any mobile node has the cellular Internet access.

3.2. Encrypting along the Forwarding Path. We use an example to illustrate how to encrypt the data. Figure 2 shows the process of data encryption. Node A intends to send its sensing data d_A to the central server and the intermediate nodes are node B and node C. P_1 represents the package that node A sends to B and P_2 represents the package that node B sends to node C. P_3 is what C sends to the central server.

First, node A issues a query to the public key handler for a route to upload data and the public key handler answers the query with a forwarding route information, which contains the public keys of nodes B, C, the central server, and their address, as well as some time thresholds. When the node A receives the route, it then encrypts the sensing data d_A with the server's public key P_{K_S} and gets P_3. After that it needs to combine three parts: P_3, the route information, which is the server's address, and the time threshold used to judge whether to send the package out or drop it. After it gets the combination, encrypt this with node C's public key. This is P_2. Next, utilize the node B's public key to encrypt the combination of P_2, the route information, node C's address, and the time threshold and generate P_1. This is what node A sends to node B. In addition, node A can also get node B's address and a time threshold.

Besides, we also utilize methods to ensure all the packages have the same size, so that attackers cannot judge the number of nodes on the route by means of analyzing the size changes of the packages.

3.3. Forwarding Sensing Data along the Path. The final step of our privacy preserving data collection approach is the data forwarding, which describes how to move the encrypted packages. With the Ps we have got, what we need to do is to transfer them along the route.

First, node A sends P_1 to node B at the time threshold T_1. Once node B gets P_1, decrypt it with its private key S_{K_B} and get P_2. Besides, it is also aware of the destination, node C's address. However, it needs to get the current time t and compare it with the time threshold T_2. If t is bigger than T_2, it means that this package is out of date and should be dropped. If not dropped, the package will arrive at the server very late and the server can easily identify its originator. If the time requirement is satisfied, send P_2 to node C at the time threshold T_2. As soon as P_2 arrives, node C decrypts it with S_{K_C} and verifies the time requirement and, if satisfied, issues it to the next hop, the central server, at the time threshold T_3. The central server can easily acquire the exact sensing data d_A by decrypting the P_3 it received but without acknowledging its originator.

All the nodes use this method to hand their sensing data to the central server. In a short time, the server will receive all the data and build a data collection set.

4. Theoretical Analysis of P^3

The most essential aspect to evaluate a privacy preserving problem is its security level. In this section, we provide a theoretical security analysis of our privacy preserving data collection approach.

In the paper, we assume the server is unauthentic and has the authority to get all the sensing data but should not know their originators, though it is eager to. So privacy is defined as the relationship or the ownership between sensing data and their originators, which are nodes in our system. Our target is to realize the anonymity of the sensing data. For this kind of privacy, the server is the attacker. It cannot distinguish to

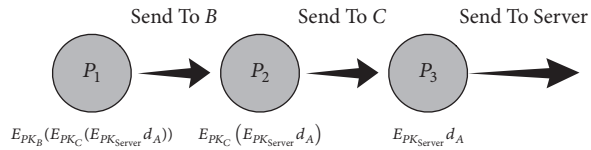


Figure 2: Data encryption.

whom each data belongs in our system. Besides, there may be other attackers who are nodes. They collude with the central server to help it realize the attack target.

The malicious nodes on the forwarding route can infer the originator of the data by means of colluding with each other. The server can query its former node that it received data from. A malicious node will answer the query, but an honest node would not answer this query. The basic idea is that the server asks its former node who sent the package to it. If this node answers the query, it would ask its former node the same question, until it gets no answer. And we can assume that the node which does not reply to the query is the beginning node (the originator of the sensing data). When all the nodes on the forwarding route except the data originator are malicious, the attack target is achieved.

However, this is just the simplest way to finish an attack, and there exist some clever means as well. For example, if there exists an honest node between two malicious nodes, as Figure 3 shows, where nodes in dark color represent malicious nodes and those in light color are honest (Figures 4 and 5 use the same way to indicate a node). When node D queried, it will say that node C sent data to it. And when node C queried, it will not reply to the query. So the attack appears to have failed. However, node B may tell the server that it has sent data to node C and it receives it from node A. So the server may infer that there may be a route $A \to B \to C \to D$.

The security level of our approach depends on the distribution of malicious nodes in the route. We can classify all possible distributions into three cases:

(i) Case 1: the next hop of the originator is honest.

(ii) Case 2.1: the next hop of the originator is malicious, and there are two consecutive nodes on the route which are honest.

(iii) Case 2.2: the next hop of the originator is malicious, and there are not two consecutive nodes on the route which are honest.

4.1. Case 1. The next hop of the originator is honest.

In this case, as is showed in Figure 4, we know that if the next hop of the data originator on the forwarding route is an honest node, it will not answer any query from the server. So the originator's identity will not be exposed. Therefore, the server will never figure out the originator of the sensing data transferred by this route. The best circumstance is that the server can infer that the second node on the route is the owner of the data. Actually, it is just an intermediate node that is honest, but not the data originator.

The probability of this case is $1 - \gamma$, which equals the probability of the next node of the data originator on the forwarding route which is honest.

Theorem 1. *In Case 1, the success probability of the attacker is* $1/(1-\gamma)N$.

Proof. When an attacker receives a piece of data, it will attempt to link the data to the originator by tracing the data back. For each node, it knows only who are the senders or receivers of the packages it receives or sends. In Case 1, we know that if the next hop of the originator is honest; thus, it will not disclose who is the sender of a package to the attackers. Therefore, even all other nodes in the route are malicious; the server cannot do better than a random guess. Since the data originator can be any honest node, the success probability of attackers is $1/(1-\gamma)N$. □

When analyzing this case, we know that it has no relationship with the number of hops n. The total probability of attackers guessing correctly under Case 1 is

$$p = \frac{p_{\text{case 1}}}{(1-\gamma)N} = \frac{1}{N}. \tag{7}$$

Table 1 shows the concrete values of p.

4.2. Case 2. If the next hop of the originator is a malicious node. This case is divided into two cases: there exist two consecutive honest nodes on the forwarding route and two consecutive honest nodes do not exist. The latter case we will discuss is Case 2.2.

4.3. Case 2.1. The next hop of the data originator is a malicious node and there exist two consecutive honest nodes on the forwarding route.

If there exist two consecutive honest nodes on the forwarding route, the server cannot finish an attack, because the server cannot restore the route. For example, Figure 5 shows this circumstance. Node B tells the server that node A sent a package to it and it has forwarded it to node C. However, nodes C and D are honest nodes and will not answer any question of the server and malicious nodes. So although the next hop of node D, which is E, is malicious, it can only provide the information that it received a package from D. With only this information, the server cannot restore the whole route because of the absence of relation between C and D. The server may consider that these nodes belong to two routes. So the privacy is preserved.

However, the malicious nodes may add a large delay before they forward package. As a result, an honest node's

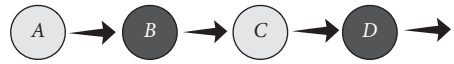


Figure 3: Clever guess.

Figure 4: Case 1.

Figure 5: Case 2.1.

sensing data may arrive at the server very late. The first malicious node can tell the server who sent this late-arrived data, and the privacy we wish to preserve is exposed. We know that the trusted public key handler also assists in selecting the forwarding routes. Besides, the encryption time of data originator can be estimated; meanwhile, the forwarding time and the decryption time of each node can also be estimated. So the public key handler can calculate the package's arrive time for each node. To avoid the estimating deviation resulting in dropping package by mistake, a small value is added to release the strict limitation of time threshold. So when the key handler selects a route, it also calculates the time threshold for each node. And with it, the malicious delay cannot help the attack.

The probability of Case 2.1 is

$$p_{\text{case 2.1}} = \gamma - \sum_{i=0}^{[(n-1)/2]} \frac{P_{\gamma N}^{(n-1-i)} P_{(n-1-i)}^{i} C_{(1-\gamma)N-1}^{i}}{P_{N-1}^{n-1}} \tag{8}$$

Theorem 2. *In Case 2.1, the success probability of attackers is* $1/(1-\gamma)N$.

Proof. In this case, the next hop of the data originator is malicious; thus, it knows who is the originator. However, the data package is encrypted at this hop and the cipher-text cannot be matched to the data received by the server. Therefore, to link a piece of data to its originator, the attackers still need to trace back. If there are two consecutive honest nodes in the route, tracing back will fail at this point. Then similar to Case 1, the success probability of attackers is $1/(1-\gamma)N$. □

Therefore, the total probability p is

$$p = \frac{p_{\text{case 2.1}}}{(1-\gamma)N}. \tag{9}$$

Tables 2, 3, and 4 show the exact values of this circumstance. From Tables 2, 3, and 4, we draw the conclusion that when the number of hops n is larger than 10, n has little impact on p. What influence it are γ and N.

4.4. Case 2.2. The next hop of the data originator is a malicious node and two consecutive honest nodes on the forwarding route do not exist.

In this case, because the second hop node is malicious, the server can acknowledge that this node sent the data. So what it needs to do is to identify the data transferred by this route. However, if no consecutive two honest nodes on the route exist, it is not hard to know this data. The simplest circumstance is that all the nodes after the second node are malicious. The server just needs to ask the former node until a node does not answer the query. While the most complex circumstance is that there is an honest node between every two malicious nodes, as Figure 3 shows. But the server and malicious nodes can still complete the attack. With the clever guess means we talked above, we know that a Malicious-Honest-Malicious combination can also be restored, though an honest node exists. Therefore, this case can be completely attacked.

The probability of Case 2.2 is

$$p_{\text{case 2.2}} = \sum_{i=0}^{[(n-1)/2]} \frac{P_{\gamma N}^{(n-1-i)} P_{(n-1-i)}^{i} C_{(1-\gamma)N-1}^{i}}{P_{N-1}^{n-1}}. \tag{10}$$

Theorem 3. *In Case 2.2, the success probability of attackers is* 1.

Proof. In this case, the simplest circumstance is that all the nodes after the second node are malicious. While the most complex circumstance is that there is an honest node between every two malicious nodes. With the basic and the clever guess means we talked above, the package forwarding route can also be restored, which means a successful attack. So the success probability of attackers is 1. □

Tables 5, 6, and 7 show that the probability of Case 2.2 happens when γ is 0.03, 0.05, and 0.1.

5. Performance Evaluation

Our simulation experiment simulates the whole process of conducting a sensing task, including the central server allocating sensing task, the public key handler collecting public keys, and selecting random forwarding route for each node, all the nodes sending out their sensing data and the server colluding with the malicious nodes to realize attack. By

Table 1: Probability of attackers guessing correctly under Case 1.

	$\gamma = 0.03$	$\gamma = 0.05$	$\gamma = 0.10$
$N = 4000$	$2^{-11.97}$	$2^{-11.97}$	$2^{-11.97}$
$N = 6000$	$2^{-12.55}$	$2^{-12.55}$	$2^{-12.55}$
$N = 8000$	$2^{-12.97}$	$2^{-12.97}$	$2^{-12.97}$
$N = 10000$	$2^{-13.29}$	$2^{-13.29}$	$2^{-13.29}$

Table 2: Probability of attackers guessing correctly under Case 2.1 when $\gamma = 0.03$.

$N = 4000$	n	11	12	13	14	15
$\gamma N = 120$	p	$2^{-16.98}$	$2^{-16.98}$	$2^{-16.98}$	$2^{-16.98}$	$2^{-16.98}$
$N = 6000$	n	11	12	13	14	15
$\gamma N = 180$	p	$2^{-17.57}$	$2^{-17.57}$	$2^{-17.57}$	$2^{-17.57}$	$2^{-17.57}$
$N = 8000$	n	11	12	13	14	15
$\gamma N = 240$	p	$2^{-17.98}$	$2^{-17.98}$	$2^{-17.98}$	$2^{-17.98}$	$2^{-17.98}$
$N = 10000$	n	11	12	13	14	15
$\gamma N = 300$	p	$2^{-18.30}$	$2^{-18.30}$	$2^{-18.30}$	$2^{-18.30}$	$2^{-18.30}$

Table 3: Probability of attackers guessing correctly under Case 2.1 when $\gamma = 0.05$.

$N = 4000$	n	11	12	13	14	15
$\gamma N = 120$	p	$2^{-16.21}$	$2^{-16.21}$	$2^{-16.21}$	$2^{-16.21}$	$2^{-16.21}$
$N = 6000$	n	11	12	13	14	15
$\gamma N = 180$	p	$2^{-16.80}$	$2^{-16.80}$	$2^{-16.80}$	$2^{-16.80}$	$2^{-16.80}$
$N = 8000$	n	11	12	13	14	15
$\gamma N = 240$	p	$2^{-17.21}$	$2^{-17.21}$	$2^{-17.21}$	$2^{-17.21}$	$2^{-17.21}$
$N = 10000$	n	11	12	13	14	15
$\gamma N = 300$	p	$2^{-17.54}$	$2^{-17.54}$	$2^{-17.54}$	$2^{-17.54}$	$2^{-17.54}$

Table 4: Probability of attackers guessing correctly under Case 2.1 when $\gamma = 0.1$.

$N = 4000$	n	11	12	13	14	15
$\gamma N = 120$	p	$2^{-15.14}$	$2^{-15.14}$	$2^{-15.14}$	$2^{-15.14}$	$2^{-15.14}$
$N = 6000$	n	11	12	13	14	15
$\gamma N = 180$	p	$2^{-15.72}$	$2^{-15.72}$	$2^{-15.72}$	$2^{-15.72}$	$2^{-15.72}$
$N = 8000$	n	11	12	13	14	15
$\gamma N = 240$	p	$2^{-16.14}$	$2^{-16.14}$	$2^{-16.14}$	$2^{-16.14}$	$2^{-16.14}$
$N = 10000$	n	11	12	13	14	15
$\gamma N = 300$	p	$2^{-16.46}$	$2^{-16.46}$	$2^{-16.46}$	$2^{-16.46}$	$2^{-16.46}$

Table 5: Probability of attackers guessing correctly under Case 2.2 when $\gamma = 0.03$.

$N = 4000$	n	11	12	13	14	15
$\gamma N = 120$	p	$2^{-25.06}$	$2^{-27.92}$	$2^{-30.03}$	$2^{-32.78}$	$2^{-35.00}$
$N = 6000$	n	11	12	13	14	15
$\gamma N = 180$	p	$2^{-25.02}$	$2^{-27.85}$	$2^{-29.97}$	$2^{-32.70}$	$2^{-34.91}$
$N = 8000$	n	11	12	13	14	15
$\gamma N = 240$	p	$2^{-24.99}$	$2^{-27.83}$	$2^{-29.93}$	$2^{-32.65}$	$2^{-34.86}$
$N = 10000$	n	11	12	13	14	15
$\gamma N = 300$	p	$2^{-24.98}$	$2^{-27.81}$	$2^{-29.91}$	$2^{-32.63}$	$2^{-34.83}$

Table 6: Probability of attackers guessing correctly under Case 2.2 when $\gamma = 0.05$.

$N = 4000$	n	11	12	13	14	15
$\gamma N = 120$	p	$2^{-21.14}$	$2^{-23.40}$	$2^{-25.28}$	$2^{-27.48}$	$2^{-29.42}$
$N = 6000$	n	11	12	13	14	15
$\gamma N = 180$	p	$2^{-21.11}$	$2^{-23.37}$	$2^{-25.44}$	$2^{-27.43}$	$2^{-29.37}$
$N = 8000$	n	11	12	13	14	15
$\gamma N = 240$	p	$2^{-21.10}$	$2^{-23.35}$	$2^{-25.22}$	$2^{-27.40}$	$2^{-29.34}$
$N = 10000$	n	11	12	13	14	15
$\gamma N = 300$	p	$2^{-21.09}$	$2^{-23.34}$	$2^{-25.21}$	$2^{-27.39}$	$2^{-29.32}$

Table 7: Probability of attackers guessing correctly under Case 2.2 when $\gamma = 0.1$.

$N = 4000$	n	11	12	13	14	15
$\gamma N = 120$	p	$2^{-15.76}$	$2^{-17.33}$	$2^{-18.79}$	$2^{-20.33}$	$2^{-21.82}$
$N = 6000$	n	11	12	13	14	15
$\gamma N = 180$	p	$2^{-15.74}$	$2^{-17.31}$	$2^{-18.77}$	$2^{-20.31}$	$2^{-21.79}$
$N = 8000$	n	11	12	13	14	15
$\gamma N = 240$	p	$2^{-15.74}$	$2^{-17.30}$	$2^{-18.76}$	$2^{-20.30}$	$2^{-21.78}$
$N = 10000$	n	11	12	13	14	15
$\gamma N = 300$	p	$2^{-15.73}$	$2^{-17.30}$	$2^{-18.75}$	$2^{-20.29}$	$2^{-21.77}$

changing different parameters, we have acquired different results. Through analyzing them, we can draw the conclusion that our approach has a high security level.

Figure 6 shows the experimental ratio of exposed data originators of different γ. The number of all the online nodes N is 10000 and we care about the circumstance that γ equals 0.05 and 0.10. The range of hops is from 9 to 14. The simulation results of the experiment are similar to our theoretical analysis. And when then proportion of malicious nodes γ is 0.10 and the system select $n = 9$ as the number of hops, the ratio of exposed data originators is nearly 1.50×10^{-4}, which means that the server can only identify 1.50 sensing data's originator. This is clearly a beautiful result.

Figure 7 shows the different ratios of exposed data originators changing tendency as the number of hops changing in theoretical analysis. The number of online nodes N in this figure is still 10000. From the figure, we can see that as the number of hops increases, the ratio of exposed data originators declines and finally converges towards zero. All the ratios are in a magnitude of 10^{-4}. Obviously, when the number of hops is constant, the results show that the higher γ the system has, the less data originators are exposed.

The comparison of experimental and theoretical results is shown in Figure 8. The parameters are set as follows: $\gamma = 0.1$ and $N = 10000$. We can easily find that two lines which represent experimental and theoretical results nearly overlap. This is a strong evidence to prove our security analysis discussed before is right.

6. Related Work

Privacy preservation is a promising area in wireless networks [15], where there have been many applications [16, 17]. Many works have addressed varieties of privacy and security issues in mobile crowd sensing networks, such as [18–23]; however, the goal of data collection is omitted. From the existing works, it is not hard to find some excellent works related to data aggregation, which have some aspects similar to our work and can be referred to, though having some differences with our work. Unfortunately, some of those existing works that solve the privacy preservation in data aggregation usually assume a trusted aggregator or server. They cannot deal with the circumstance that the aggregator or the server is unauthentic, for instance, [24–27].

Generally, data aggregation usually cares about the statistic results of the data, not the data themselves. Sometimes the real values of data are less useful than the statistic results. For example, we commonly prefer to acknowledge the maximum or the minimum or the average of a set of data. The real values of data are not necessary. The most common data aggregation is *Sum* and Max/Min: Yang et al. [13] proposed an encryption approach that helps an aggregator obtain the *Sum* of data without knowing each participants' specific data. Shi et al. [28, 29] proposed a construction for *Sum* aggregation which does not need the extra round of interaction. Li and Cao [14] proposed an efficient protocol to obtain the *Sum* aggregation, which employs an additive homomorphic encryption and a novel key management technique to support large plaintext space. And they also extend *Sum* aggregation to *Min* aggregation. Although these approaches solve the privacy preservation in data aggregation, they only provide *Sum* and *Min* function.

While, on the other hand, our work and data collection, which aims to collect all the data, must know all the exact values. Merely obtaining statistic results cannot satisfy our request, but these existing works also have significance to us. Boutsis and Kalogeraki [30] proposed an efficient way to collect location information based on exchanging data with other participants. But there is no central server in its model.

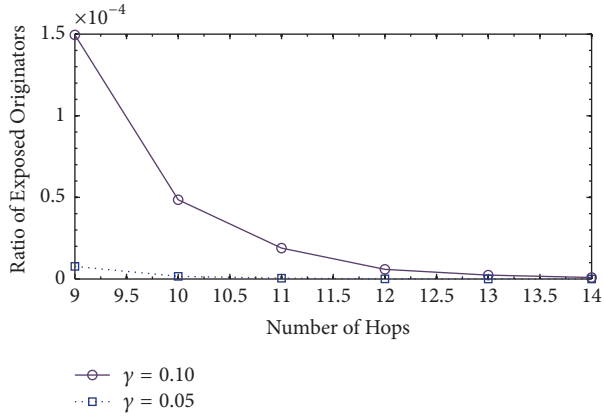


Figure 6: Experimental ratio of exposed data originators.

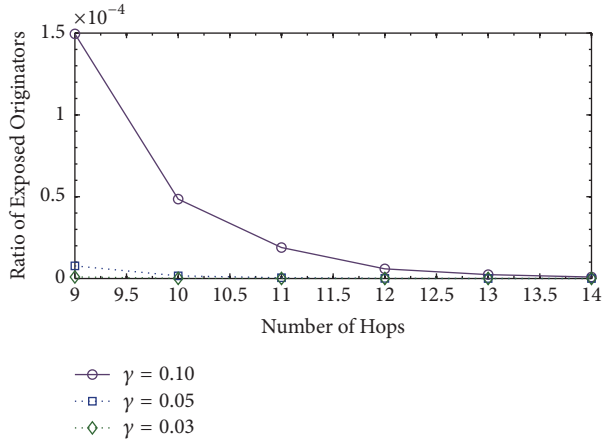


Figure 7: Theoretical ratio of exposed data originators.

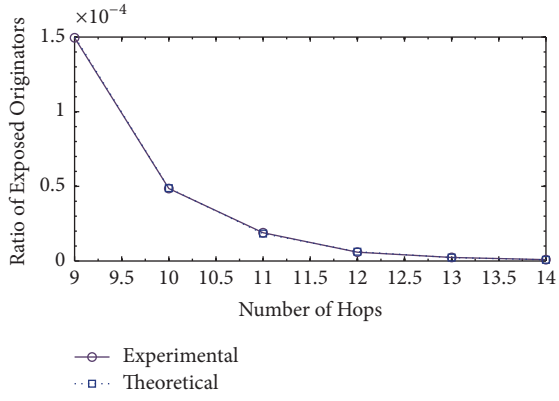


Figure 8: Comparison of experimental and theoretical results.

Besides, the participants may be malicious, and it will lead to an unguaranteed security level.

The most frequently adopted method is slicing the data first and then mixing it. Shi et al. [31] utilized this method and proposed an aggregation approach that realized the privacy preservation. Destroying original data has little impact on data aggregation, while for data collection, restoring the data is troubling. In order to mix the data slice, each slice coming from the same original data needs to be signed with the same ID, which is harmful to realizing the anonymity of data.

Data perturbation is another common technique. Many existing works protect individuals' privacy by adding noise to

the original data at the client side and allows the server to reconstruct the statistics of the original data [32–34]. Kargupta et al. [35] utilized the independent random noise, which can be separated from the perturbed data by studying the spectral properties of the perturbed data. Zhang et al. [36] proposed a data perturbation algorithm that can be used to protect participants' private information. But the disadvantage of these works is that the server can only acquire statistics result. Privacy preservation problem also exists in other networks, such as wireless sensor networks. Although our work focuses on the mobile crowd sensing network, there are many works in wireless sensor networks that can be referred to. The privacy preserving problem in wireless sensor networks usually has the same attack model as our work. The difference between these two networks is that wireless sensor networks usually have a constant topology, which seldom changes. However, in mobile crowd sensing networks, almost every participant moves all the time. Besides, some smartphones may be unavailable at some time. But if we only care about the method they preserve privacy, it is similar to our work.

Horney et al. [37] proposed a set of protocols that enable anonymous data collection in a sensor network. Sensor nodes, instead of transmitting their actual data, transmit a sample of the data complement to a base station, which then uses the negative samples to reconstruct a histogram of the original sensor readings. There also exist some approaches that can only do an approximate query. Besides, the topology graphs must be known to each node in most approaches in wireless sensor networks, which demands that each node owns enough storage and calculating ability. Therefore, these approaches are unpractical in mobile crowd sensing networks.

However, we propose a privacy preserving data collection approach. Onion routing approach [38] which operates by dynamically building anonymous connections within a network is referred. Specially, our approach has the following advantages over the previous approach. First, we can collect all the exact data values which can be further used to analyze, not just some aggregation results. Second, an anonymous function is realized to preserve data privacy in a high security level. Last, our approach is efficient and suitable for use.

Given the existing methods for preserving privacy of nodes in a crowdsensing system, our approach contributes to expanding the tool set for privacy preserving in that with our approach the server is able to access the original data, not just aggregated data.

A preliminary version of the work is reported in [39].

7. Conclusions

In this paper, we focus on the potential privacy risk of the mobile node in a mobile crowd sensing system. A mobile node's privacy may be compromised if its sensing data is associated with the node by a malicious attacker. Existing approaches try to protect node privacy by manipulating the sensing data sent to other nodes and the data center. Such approaches have a main drawback that the collected sensing data have a lower utility. We have proposed an approach called P^3 for preserving node privacy. This approach determines a data forwarding path by which a source node sends its sensing data along the forwarding path. By using asymmetric encryptions, the source node is protected against malicious attackers which may try to trace back along the forwarding path. Through the security analysis based on the attack model, the possibility of malicious nodes compromising the privacy of data originators is low. Simulation results demonstrate that our approach has a low overhead of encryption computations and storage.

Notations

u_i: The th node
d_{ij}: The jth data of node u_i
D_i: The data set of node u_i
P_{K_i}: The public key of node u_i
S_{K_i}: The private key of node u_i
S: The central server
C: Data collection set in the server
H: The public key handler
P_i: The th package be transferred by nodes
t: The current time
T: The forwarding time threshold
n: The number of hops.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The research is supported in part by Wenzhou Vocational & Technical College Research Project (no. WZY2017002), Wenzhou Science and Technology Bureau Program (no. G2017037), and Education Department of Zhejiang Province (2016 Educational Technology Research Project no. JB084).

References

[1] T. Liu and Y. Zhu, "Social welfare maximization in participatory smartphone sensing," *Computer Networks*, vol. 73, pp. 195–209, 2014.

[2] Y. Zhu, Y. Wang, G. Forman, and H. Wei, "Mining large-scale GPS streams for connectivity refinement of road maps," *The Computer Journal*, vol. 58, no. 9, pp. 2109–2119, 2014.

[3] R. Jiang, Y. Zhu, X. Wang, and L. M. Ni, "TMC: Exploiting Trajectories for Multicast in Sparse Vehicular Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 262–271, 2015.

[4] K. Wu, H. Li, L. Wang et al., "HJam: attachment transmission in WLANs," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2334–2345, 2013.

[5] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. M. Ni, "Side channel: Bits over interference," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1317–1330, 2012.

[6] K. Wu, H. Tan, H.-L. Ngan, Y. Liu, and L. M. Ni, "Chip error pattern analysis in IEEE 802.15.4," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 543–552, 2012.

[7] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2289–2302, 2013.

[8] Y. Zhu, X. Liu, M. Li, and Q. Zhang, "POVA: traffic light sensing with probe vehicles," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1390–1400, 2013.

[9] Y. Zhu, Y. Bao, and B. Li, "On maximizing delay-constrained coverage of urban vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 804–817, 2012.

[10] Y. Zhu, R. Jiang, J. Zhao, and L. M. Ni, "Correlating mobility with social encounters: distributed localization in sparse mobile networks," *Wireless Networks*, vol. 21, no. 1, pp. 201–215, 2015.

[11] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "CSI-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2013.

[12] J. A. Burke, D. Estrin, M. Hansen et al., "Participatory sensing," *Center for Embedded Network Sensing*, 2006.

[13] Z. Yang, S. Zhong, and R. N. Wright, "Privacy-preserving classification of customer data without loss of accuracy," in *Proceedings of the SDM*, pp. 92–102, 2005.

[14] Q. Li and G. Cao, "Efficient and privacy-preserving data aggregation in mobile sensing," in *Proceedings of the 2012 20th IEEE International Conference on Network Protocols, ICNP 2012*, USA, November 2012.

[15] T. Abdelzaher, Y. Anokwa, P. Boda et al., "Mobiscopes for human spaces," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 20–29, 2007.

[16] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing," in *Proceedings of the 2nd Annual International Workshop on Wireless Internet (WICON '06)*, pp. 1–14, ACM, August 2006.

[17] J. Corburn, "Confronting the Challenges in Reconnecting Urban Planning and Public Health," *American Journal of Public Health*, vol. 94, no. 4, pp. 541–546, 2004.

[18] Z. Zhu and G. Cao, "APPLAUS: a privacy-preserving location proof updating system for location-based services," in *Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011)*, pp. 1889–1897, Shanghai, China, April 2011.

[19] Q. Li and G. Cao, "Mitigating routing misbehavior in disruption tolerant networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 664–675, 2012.

[20] Q. H. Li, S. C. Zhu, and G. H. Cao, "Routing in socially selfish delay tolerant networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, March 2010.

[21] Q. Li and G. Cao, "Providing privacy-aware incentives for mobile sensing," in *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications, PerCom 2013*, pp. 76–84, USA, March 2013.

[22] E. De Cristofaro and C. Soriente, "Short paper: PEPSI: Privacy-Enhanced Participatory Sensing Infrastructure," in *Proceedings of the 4th ACM Conference on Wireless Network Security, WiSec'11*, pp. 23–28, Germany, June 2011.

[23] Q. Li, W. Gao, S. Zhu, and G. Cao, "A routing protocol for socially selfish delay tolerant networks," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1619–1632, 2012.

[24] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC '09*, pp. 169–178, usa, June 2009.

[25] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 1–36, 2009.

[26] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of Cryptography*, vol. 3378 of *Lecture Notes in Computer Science*, pp. 325–341, Springer, Berlin, Germany, 2005.

[27] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: a secure hop-by-hop data aggregation protocol for sensor networks," *ACM Transactions on Information and System Security*, vol. 11, no. 4, article 18, 2008.

[28] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proceedings of the NDSS*, p. 4, 2011.

[29] T.-H. H. Chan, E. Shi, and D. Song, "Privacy-preserving stream aggregation with fault tolerance," in *Financial Cryptography and Data Security*, pp. 200–214, Springer, Berlin, Germany, 2012.

[30] I. Boutsis and V. Kalogeraki, "Privacy preservation for participatory sensing data," in *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications (PerCom '13)*, pp. 103–113, IEEE, March 2013.

[31] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: privacy-preserving data aggregation in people-centric urban sensing systems," in *Proceedings of the IEEE INFOCOM*, San Diego, Calif, USA, March 2010.

[32] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the ACM PODS*, pp. 247–255, 2001.

[33] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting Privacy Breaches in Privacy Preserving Data Mining," in *Proceedings of the ACM PODS*, pp. 211–222, June 2003.

[34] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, "PoolView: Stream privacy for grassroots participatory sensing," in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems, SenSys 2008*, pp. 281–294, USA, November 2008.

[35] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Proceedings of the IEEE ICDM*, pp. 99–106.

[36] F. Zhang, L. He, W. He, and X. Liu, "Data perturbation with state-dependent noise for participatory sensing," in *Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2012*, pp. 2246–2254, USA, March 2012.

[37] J. Horey, M. M. Groat, S. Forrest, and F. Esponda, "Anonymous data collection in sensor networks," in *Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '07)*, pp. 1–8, August 2007.

[38] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing," *Communications of the ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[39] Q. Chen, S. Zheng, and Z. Weng, "Data collection with privacy preserving in participatory sensing," in *Proceedings of the IEEE ICPADS*, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 8130152, 13 pages
https://doi.org/10.1155/2018/8130152

Research Article

Mathematical Performance Evaluation Model for Mobile Network Firewall Based on Queuing

Shichang Xuan,[1] **Dapeng Man**,[1] **Jiangchuan Zhang**,[1] **Wu Yang**,[1] **and Miao Yu**[2]

[1]*Information Security Research Center, Harbin Engineering University, No. 145 Nantong Street, Harbin, Heilongjiang 150001, China*
[2]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*

Correspondence should be addressed to Shichang Xuan; xuanshichang@hrbeu.edu.cn and Wu Yang; yangwu@hrbeu.edu.cn

Received 21 January 2018; Accepted 21 March 2018; Published 30 April 2018

Academic Editor: Wei Wang

While mobile networks provide many opportunities for people, they face security problems huge enough that a firewall is essential. The firewall in mobile networks offers a secure intranet through which all traffic is handled and processed. Furthermore, due to the limited resources in mobile networks, the firewall execution can impact the quality of communication between the intranet and the Internet. In this paper, a performance evaluation mathematical model for firewall system of mobile networks is developed using queuing theory for a multihierarchy firewall with multiple concurrent services. In addition, the throughput and the package loss rate are employed as performance evaluation indicators, and discrete-event simulated experiments are conducted for further verification. Lastly, experimental results are compared to theoretically obtained values to identify a resource allocation scheme that provides optimal firewall performance and can offer a better quality of service (QoS) in mobile networks.

1. Introduction

Due to the rapid development of mobile Internet technologies and the propagation of networks, the mobile Internet is progressively becoming significant part of everyone's personal and professional lives. Consequently, the security issues of mobile networks tend to attract more attention. Packet-based filtering in traditional firewalls fails to satisfy the security requirements of the end users. Being an essential component of the entire trusted intranet, a traditional firewall suffers from the disadvantages of simple packet filtering and lower-layer processing. Compared to traditional firewalls, new firewalls are capable of comprehensive data analysis at the application layer, which defines higher-level processing and more comprehensive protection.

Although new firewalls are able to provide a trusted intranet with better security, they consume large amounts of time for analyzing and processing the network packets because they require higher-level assessments and analysis. Furthermore, heavier traffic may reduce the processing efficiency. It makes the firewall prone to becoming a performance bottleneck for the communication between the intranet and the Internet, consequently reducing the quality of service (QoS) of mobile networks. To overcome this problem, limited system resources (such as number of threads and CPU) must be rationally assigned when developing a firewall. Modifying the assignment of resources may directly affect the overall system performance. In general, when developing systems such as firewall, the corresponding firewall performance test needs to be performed. The throughput of the current system needs to be tested and performance optimization be conducted by software engineers. Further tests should be carried out until the optimal resource allocation scheme is obtained. This method for testing and performance tuning is very time-consuming and complex. However, if a mathematical model is constructed to test the common performance indicators of the firewall, such as throughput and packet loss rate, considerable savings in developer labor costs may be achieved, and the optimal resource allocation system may be derived directly.

In this paper, a performance evaluation model for the firewall system of mobile networks is proposed using the queuing theory. Generally, this model can be divided into two phases. Phase one involves handling the traffic at the lower

layers, including examinations at the network layer and the transmission layer. Phase two involves handling the upper-layer applications, consisting of HTTP, IMAP, and DNS. Each phase involves concurrent processing with multiservice stations. A discrete-event simulated experiment is conducted to validate the model.

The rest of the paper is organized as follows: Section 2 reviews the current state of related research and designates the innovations proposed in this paper. Section 3 discusses the mathematical modeling of a firewall system for mobile networks. Section 4 describes the simulated experiments performed using the developed model and analyzes the results of conducted experiments. Finally, Section 5 summarizes the results and conclusions.

2. Related Work

With the increasing complexity of mobile network security issues, many researchers have been studying and analyzing firewall equipment. As the methods and means of network attacks continue to increase [1], the functions of firewall systems are being enhanced and extended gradually, for example, intrusion detection, DDoS attack detection, and user-behavior detection [2, 3]. Moreover, a number of researchers have conducted network security analysis. Peng et al. mainly performed analysis with respect to user behaviors [4–6]. Chen et al. provided a survey of security problems in mobile networks [7]. A resource optimization approach in mobile networks was proposed in [8] under imperfect prediction.

The queuing theory, which has significant potential in terms of its application to the mathematical modeling of network equipment, may be useful for analyzing the performance of such equipment [9]. Some researchers have conducted modeling and analyzing of systems such as firewalls based on this theory, and they have made certain relatively advanced achievements in the study of network equipment performance. Herrmann constructed a vacation model for network service analysis, where batch arrival is used rather than the common Poisson flow [10]. Furthermore, in some studies, large-scale cloud services [11] have been modeled into a complex processing system with finite queues [12–14]. Salah et al. made considerable achievements by proposing a service system with a finite queue and single window, where packets arrive in a Poisson flow pattern and the service time follows a negative exponential distribution [15, 16]. Salah [17] found that the differences between the applicability of the Erlang distribution modeling and that of a Markov chain modeling have been compared. Zapechnikov et al. conducted an in-depth study based on Salah's work. They proposed a two-phase service system, where Phase 1 was based on a Markov chain model and Phase 2 employed a hyper-Erlang distribution model [18]. Some researchers have also investigated cloud-service-based firewall systems [19, 20].

Several recent studies have only compared modeling methods, while some have merely discussed the handling process of a single-service window or conducted simple hierarchical modeling [21]. However, neither can describe the firewall workflow nor depict its handling process in detail. With the advances in science and technology, the widespread use of multicore processors and the increased availability of resources in terms of equipment, it is becoming difficult for the above-mentioned methods to fully exploit hardware performance. In this paper, an in-depth examination is conducted with respect to these issues. Further, based on firewall handling and processing characteristics, a two-phase multiservice station [22] and multiprotocol firewall model with multiple concurrent applications is proposed, which is further employed to analyze the overall system performance [23].

3. Model Analysis

Using an Erlang queuing model, this paper proposes a two-phase multiservice station and multiprotocol firewall model with multiple concurrent applications. This model may be hierarchically divided into two phases. When a packet enters the system, it will first be processed on the basis of the rules at the network layer and the transmission layer and then handled by the DNS, FTP, and HTTP protocols. The process flow varies slightly with the characteristics of various protocols. The hierarchy of the firewall model is shown in Figure 1.

The related parameters with respect to the model are defined as follows.

λ denotes the arrival rate of packets that arrive at the firewall. In Phase 1, K_L represents the phase 1 buffer, N_L denotes the number of service windows for Phase 1, r_L is the number of rules for Phase 1, and μ_L is the processing rate of each rule. Phase 2 is divided into three main parts, namely, DNS, FTP, and HTTP. Here, q_D, q_F, and q_H denote the proportions of DNS, FTP, and HTTP traffic, respectively. In the DNS handling part, K_D denotes the buffer, N_D represents the number of service windows, r_D represents the number of rules, and μ_D is the handling rate of each rule. The FTP handling component consists of FTP command handling operations and FTP data transmission handling operations. In terms of the FTP command handling part, K_{F1} denotes its buffer, N_{F1} represents its number of service windows, r_{F1} is the number of rules, and μ_{F1} represents the handling rate of each rule. The FTP data transmission handling part may be divided into multiple transmissions. For example, n concurrent FTP data transmission handling operations are shown in Figure 2, where the traffic proportion is $q_{F\text{-}i}$. Further, $K_{F2\text{-}i}$ represents the buffer for data transmission handling, $N_{F2\text{-}i}$ denotes the number of service windows for data transmission handling, $r_{F2\text{-}i}$ is the number of rules for data transmission handling, and $\mu_{F2\text{-}1}$ denotes the handling rate of each rule. The HTTP part consists of HTTP-request line handling, HTTP-header handling, and HTTP-body handling. Here, $q_{H\text{-}i}$ denotes the traffic proportion of each HTTP application; K_{H1}, K_{H2}, and $K_{H3\text{-}i}$ denote the buffers for HTTP line handling, header handling, and body handling for each application, respectively; N_{H1}, N_{H2}, and $N_{H3\text{-}i}$ represent the number of service windows for HTTP-request line handling, header handling, and body handling for each application, respectively; r_{H1}, r_{H2}, and $r_{H3\text{-}i}$ denote the number of rules for HTTP line handling, header handling, and body handling for each application, respectively; and μ_{H1}, μ_{H2}, and $\mu_{H3\text{-}i}$

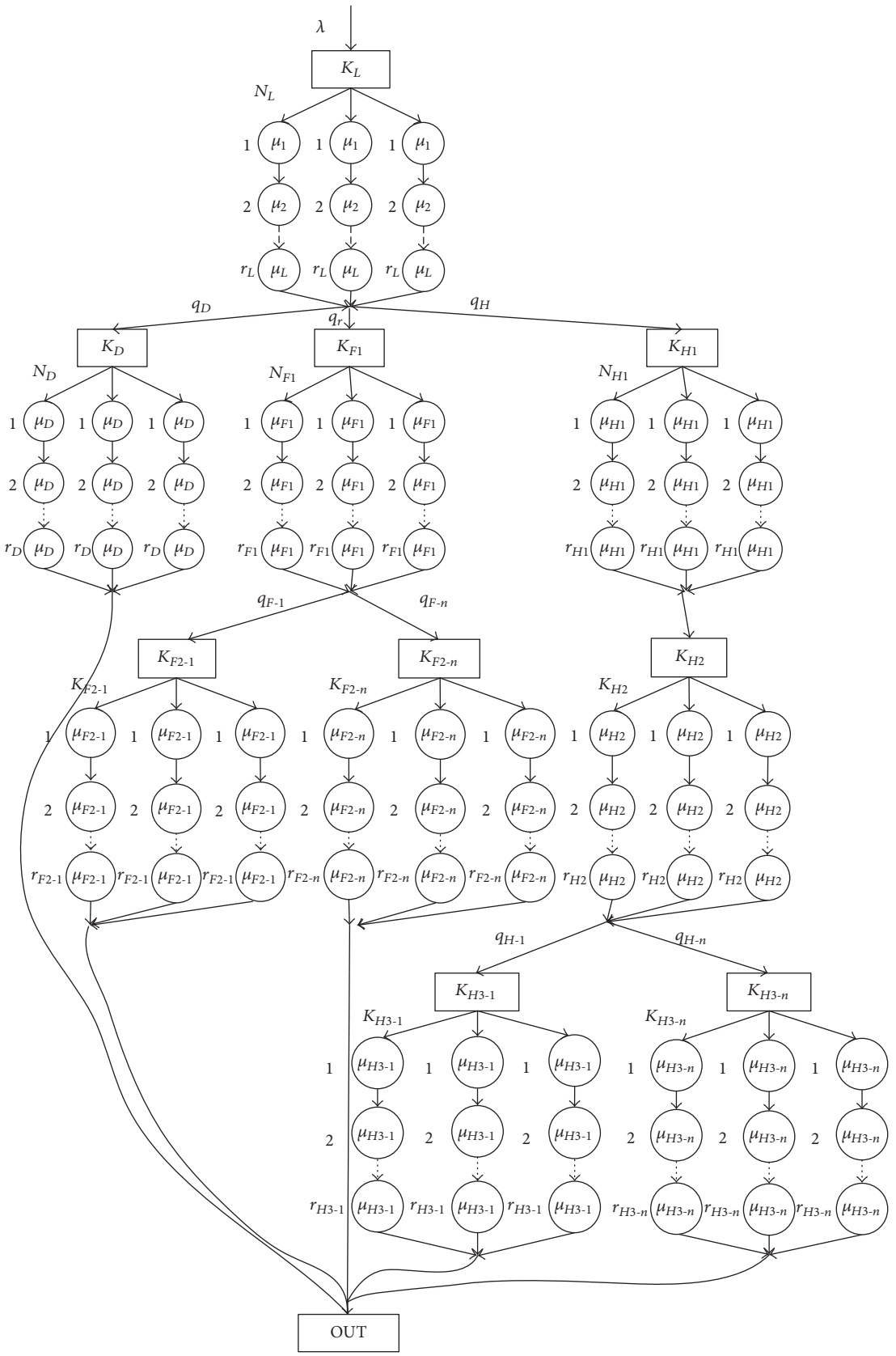


Figure 1: Hierarchy of the firewall model.

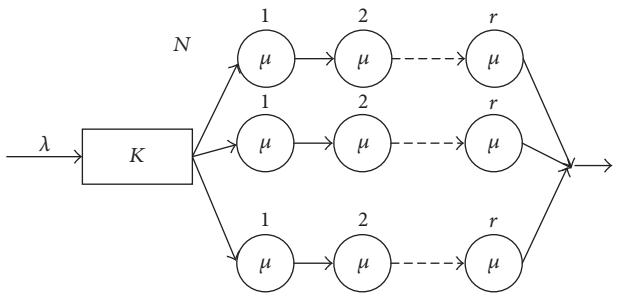


Figure 2: Submodule structure.

denote the handling rate of each rule for HTTP line handling, header handling, and body handling for each application, respectively.

The firewall model proposed in this paper refines a number of essential upper-layer handling processes in which the traffic flow proportion or number of applications may be easily modified. In addition, this model comprises the commonly used handling processes for DNS, FTP, and HTTP, each of which may be extended as an individual template for other upper-layer application protocols.

3.1. Low-Level Analysis. As shown in Figure 1, the entire model consists of multiple single-buffer and multiconcurrent Erlang processes. Hence, the computation of the entire model may be based on cumulative computations of individual submodules. Packets enter the system sequentially, following a Poisson flow pattern; then, they wait in the buffer to be further handled by the concurrent service windows. Each service window comprises multiple subservice stations that correspond to each rule. The handling efficiency of the service windows follows an Erlang distribution. The structure of the submodules is shown in Figure 2.

The following key formula may be derived on the basis of the results of the previous work.

The distribution density function of the service time may be denoted by

$$f(t) = \frac{N\mu (N\mu t)^{r-1}}{(r-1)!} e^{-N\mu t} \quad (t \geq 0). \tag{1}$$

Further, ∂_k denotes the probability that k packets arrive at the system during the service time of a packet:

$$\partial_k = \int_0^\infty \frac{(\lambda x)^k}{k!} e^{-\lambda x} f(x)\, dx. \tag{2}$$

Hence, the following formula can be obtained:

$$\begin{aligned} \partial_k &= \frac{\lambda^k}{k!} \frac{(N\mu)^r}{(r-1)!} \cdot \frac{\Gamma(k+r)}{(\lambda + N\mu)^{k+r}} \\ &= \frac{\lambda^k}{k!} \frac{(N\mu)^r}{(r-1)!} \cdot \frac{(k+r-1)!}{(\lambda + N\mu)^{k+r}}. \end{aligned} \tag{3}$$

In addition, P_{jk} represents the state transition probability, that is, the probability that the number of network packets in the system changes from j to k at any arbitrary moment. The transition of states is correlated with the number of packet arrivals within the service time; hence, the relationship between P_{jk} and α_k may be obtained as follows:

$$p_{0k} = \begin{cases} \partial_k & 0 \leq k \leq K+N-2 \\ \sum_{m=K+N-1}^{\infty} \partial_m & k = K+N-1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \leq k \leq K+N-2 \\ \sum_{m=K+N-1}^{\infty} \partial_m & k = K+N-1 \end{cases} \tag{4}$$

$$1 \leq j \leq K+N-1.$$

In terms of the steady-state probability of the system following the instantaneous exit of the packets π_k ($0 \leq k \leq K+N-1$), the following relationship exists amongst various states:

$$\pi_k = \sum_{j=0}^{K-1} \pi_j p_{jk} \quad 0 \leq k \leq K+N-1. \tag{5}$$

The following may be derived:

$$\pi_k = \pi_0 \partial_k + \sum_{j=1}^{k+1} \pi_j \partial_{k-j+1} \quad 0 \leq k \leq K+N-2. \tag{6}$$

According to the regularity, we can derived

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K-1} (\pi_k / \pi_0)}. \tag{7}$$

The load offered at the network layer is given by

$$\rho = \lambda \cdot \frac{r}{N\mu}. \tag{8}$$

The packet loss rate is denoted by

$$p_{\text{loss}} = p_K = 1 - \frac{1-p_0}{\rho} = \frac{p_0 + \rho - 1}{\rho}. \tag{9}$$

And the throughput is expressed by

$$\gamma = \lambda (1 - p_{\text{loss}}). \tag{10}$$

3.2. Phase 1 Analysis. As shown in Figure 1, it can be learned that Phase 1 hierarchy is consistent with the subhierarchy described above, where the arrival rate of the packets is λ, the buffer size is denoted by K_L, the number of service windows is represented by N_L, the number of rules is r_L, and the service rate of the rules is indicated by μ_L.

By applying the conclusion drawn in the previous section, it can be learned that the probability that k packets arrive at a queue within the service time of a packet may be expressed by

$$\partial_k = \frac{\lambda^k}{k!} \frac{(N_L \mu_L)^{r_L}}{(r_L - 1)!} \cdot \frac{(k + r_L - 1)!}{(\lambda + N_L \mu_L)^{k+r_L}}. \tag{11}$$

The state transition probability for Phase 1 is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \le k \le K_L + N_L - 2 \\ \sum_{m=K_L+N_L-1}^{\infty} \partial_m & k = K_L + N_L - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \le k \le K_L + N_L - 2 \\ \sum_{m=K_L+N_L-1}^{\infty} \partial_m & k = K_L + N_L - 1 \end{cases} \quad 1 \le j \le K_L + N_L - 1. \tag{12}$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_L+N_L-1} \pi_j p_{jk} \quad 0 \le k \le K_L + N_L - 1. \tag{13}$$

By performing ratio computation with π_0, we obtain

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_L+N_L-1} (\pi_k/\pi_0)}. \tag{14}$$

Phase 1 packet loss rate is given by

$$p_{\text{loss-}L} = 1 - \frac{1}{\pi_0 + \lambda \cdot r_L / N_L \mu_L}. \tag{15}$$

Phase 1 throughput is given by

$$\gamma_L = \lambda (1 - p_{\text{loss-}L}). \tag{16}$$

3.3. Phase 2 Analysis. Phase 2 analysis is comparatively complex. It involves handling operations implemented by DNS, FTP, and HTTP. Phase 2 throughputs, that is, the overall throughput of the system, are jointly comprised of the throughputs from DNS, FTP, and HTTP.

3.3.1. DNS Handling Analysis. As shown in Figure 1, the DNS handling process is based on a single-phase concurrent Erlang model. If applications with similar handling processes exist in the system, the DNS handling process may be used as a template. During the DNS handling operations, the arrival rate of packets equals Phase 1 throughput $\gamma_L * q_D$, the buffer size is denoted by K_D, the number of service windows is N_D, the number of rules is represented by r_D, and the service rate of the rules is indicated by μ_D.

As with Phase 1 handling process, the probability that k packets arrive at the queue during the service time of a packet may be expressed by

$$\partial_k = \frac{\lambda^k}{k!} \frac{(N_D \mu_D)^{r_D}}{(r_D - 1)!} \cdot \frac{(k + r_D - 1)!}{(\lambda + N_D \mu_D)^{k+r_D}}. \tag{17}$$

The state transition probability for DNS handling operations is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \le k \le K_D + N_D - 2 \\ \sum_{m=K_D+N_D-1}^{\infty} \partial_m & k = K_D + N_D - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \le k \le K_D + N_D - 2 \\ \sum_{m=K_D+N_D-1}^{\infty} \partial_m & k = K_D + N_D - 1 \end{cases} \quad 1 \le j \le K_D + N_D - 1. \tag{18}$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_D+N_D-1} \pi_j p_{jk} \quad 0 \le k \le K_D + N_D - 1. \tag{19}$$

By performing ratio computation with π_0, the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_D+N_D-1} (\pi_k/\pi_0)}. \tag{20}$$

The packet loss rate for DNS handling operations is given by

$$p_{\text{loss-}D} = 1 - \frac{1}{\pi_0 + \lambda \cdot q_D \cdot r_D / N_D \mu_D}. \tag{21}$$

The throughput for DNS handling operations is expressed by

$$\gamma_D = \lambda \cdot q_D (1 - p_{\text{loss-}D}). \tag{22}$$

3.3.2. FTP Handling Analysis. As shown in Figure 1, the FTP handling process is based on a two-phase concurrent Erlang model. If applications with similar handling processes exist in the system, the FTP handling process may be used as a template. The FTP handling operations may be divided into two parts. When a packet arrives at the FTP handling process, it will first be processed by the FTP command handling module, followed by the filtering and analysis of the transmitted data.

(1) FTP Command Handling Operations. During the FTP command handling operations, the arrival rate of the packets equals Phase 1 throughput $\gamma_L * q_F$, the buffer size is denoted by K_{F1}, the number of service windows is N_{F1}, the number of rules is represented by r_{F1}, and the service rate of the rules is μ_{F1}.

The probability that k packets arrive at a queue within the service time of a packet may be expressed by

$$\partial_k = \frac{\lambda^k}{k!} \frac{(N_{F1} \mu_{F1})^{r_{F1}}}{(r_{F1} - 1)!} \cdot \frac{(k + r_{F1} - 1)!}{(\lambda + N_{F1} \mu_{F1})^{k+r_{F1}}}. \tag{23}$$

The state transition probability for FTP command handling operations may be expressed by

$$p_{0k} = \begin{cases} \partial_k & 0 \le k \le K_{F1} + N_{F1} - 2 \\ \sum_{m=K_{F1}+N_{F1}-1}^{\infty} \partial_m & k = K_{F1} + N_{F1} - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \le k \le K_{F1} + N_{F1} - 2 \\ \sum_{m=K_{F1}+N_{F1}-1}^{\infty} \partial_m & k = K_{F1} + N_{F1} - 1 \end{cases} \quad 1 \le j \le K_{F1} + N_{F1} - 1. \tag{24}$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{F1}+N_{F1}-1} \pi_j p_{jk} \quad 0 \le k \le K_{F1} + N_{F1} - 1. \tag{25}$$

By performing ratio computation with $\pi 0$, the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{F1}+N_{F1}-1} (\pi_k/\pi_0)}. \tag{26}$$

The packet loss rate for FTP command handling operations is given by

$$p_{\text{loss-}F1} = 1 - \frac{1}{\pi_0 + \lambda \cdot q_F \cdot r_{F1}/N_{F1}\mu_{F1}}. \tag{27}$$

The throughput for FTP command handling operations is denoted by

$$\gamma_{F1} = \lambda \cdot q_F (1 - p_{\text{loss-}F1}). \tag{28}$$

(2) FTP Data Transmission Handling Operations. In terms of FTP data transmission handling operations, assume that there are n' concurrent data transmission handling modules, and let i denote the ith data transmission handling module. The packet flow proportion for each data transmission handling module is denoted by $q_{F\text{-}i}$. The arrival rate of packets equals the throughput for FTP command handling operations, $\gamma_{F1} * q_{F\text{-}i}$, the buffer size is $K_{F2\text{-}i}$, the number of service windows is expressed by $N_{F2\text{-}i}$, the number of rules is $r_{F2\text{-}i}$, and the service rate of the rules is represented by $\mu_{F2\text{-}i}$.

The probability that k packets arrive at a queue within the service time of a packet may be denoted by

$$\partial_k = \frac{\lambda^k}{k!} \frac{(N_{F2\text{-}i}\mu_{F2\text{-}i})^{r_{F2}}}{(r_{F2\text{-}i}-1)!} \cdot \frac{(k + r_{F2\text{-}i} - 1)!}{(\lambda + N_{F2\text{-}i}\mu_{F2\text{-}i})^{k+r_{F2\text{-}i}}}. \tag{29}$$

The state transition probability for FTP data transmission handling operations is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \le k \le K_{F2\text{-}i} + N_{F2\text{-}i} - 2 \\ \sum_{m=K_{F2\text{-}i}+N_{F2\text{-}i}-1}^{\infty} \partial_m & k = K_{F2\text{-}i} + N_{F2\text{-}i} - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \le k \le K_{F2\text{-}i} + N_{F2\text{-}i} - 2 \\ \sum_{m=K_{F2\text{-}i}+N_{F2\text{-}i}-1}^{\infty} \partial_m & k = K_{F2\text{-}i} + N_{F2\text{-}i} - 1 \end{cases} \quad 1 \le j \le K_{F2\text{-}i} + N_{F2\text{-}i} - 1. \tag{30}$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{F2\text{-}i}+N_{F2\text{-}i}-1} \pi_j p_{jk} \quad 0 \le k \le K_{F2\text{-}i} + N_{F2\text{-}i} - 1. \tag{31}$$

By performing ratio computation with $\pi 0$, the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{F2\text{-}i}+N_{F2\text{-}i}-1} (\pi_k/\pi_0)}. \tag{32}$$

The packet loss rate for FTP data transmission handling operations is given by

$$p_{\text{loss-}F2\text{-}i} = 1 - \frac{1}{\pi_0 + \gamma_{F1} \cdot q_{F\text{-}i} \cdot r_{F2\text{-}i}/N_{F2\text{-}i}\mu_{F2\text{-}i}}. \tag{33}$$

The throughput for FTP data transmission handling operations is denoted by

$$\gamma_{F2\text{-}i} = \gamma_{F1} \cdot q_{F\text{-}i} (1 - p_{\text{loss-}F2\text{-}i}). \tag{34}$$

The total throughput for FTP handling operations is expressed by

$$\gamma_F = \sum_{i=1}^{n} \gamma_{F2\text{-}i}. \tag{35}$$

3.3.3. HTTP Handling Analysis. As shown in Figure 1, the HTTP handling process is based on a three-phase concurrent Erlang model. If applications with similar handling processes exist in the system, the HTTP handling process may be used as a template. The HTTP handling operations may be divided into three parts. When a packet arrives at the HTTP handling process, it will first be handled by HTTP-request line handling module; then, it will be processed by HTTP-header handling module, and finally, it will be handled by the HTTP-body handling module and delivered to upper-layer applications.

(1) HTTP-Request Line Handling Operations. In terms of HTTP-request line handling operations, the arrival rate of

packets equals the throughput for Phase 1, $\gamma_L * q_H$, the buffer size is K_{H1}, the number of service windows is denoted by N_{H1}, the number of rules is r_{H1}, and the service rate of the rules is expressed by μ_{H1}.

The probability that k packets arrive at a queue within the service time of a packet may be denoted by

$$\partial_k = \frac{\lambda^k}{k!} \frac{(N_{H1}\mu_{H1})^{r_{H1}}}{(r_{H1}-1)!} \cdot \frac{(k+r_{H1}-1)!}{(\lambda + N_{H1}\mu_{H1})^{k+r_{H1}}}. \quad (36)$$

The state transition probability for HTTP-request line handling operations is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \le k \le K_{H1} + N_{H1} - 2 \\ \sum_{m=K_{H1}+N_{H1}-1}^{\infty} \partial_m & k = K_{H1} + N_{H1} - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \le k \le K_{H1} + N_{H1} - 2 \\ \sum_{m=K_{H1}+N_{H1}-1}^{\infty} \partial_m & k = K_{H1} + N_{H1} - 1 \end{cases} \quad 1 \le j \le K_{H1} + N_{H1} - 1. \quad (37)$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{H1}+N_{H1}-1} \pi_j p_{jk} \quad 0 \le k \le K_{H1} + N_{H1} - 1. \quad (38)$$

By performing ratio computation with $\pi 0$, the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{H1}+N_{H1}-1} (\pi_k/\pi_0)}. \quad (39)$$

The packet loss rate for HTTP-request line handling operations is given by

$$p_{\text{loss-}H1} = 1 - \frac{1}{\pi_0 + \lambda \cdot q_H \cdot r_{H1}/N_{H1}\mu_{H1}}. \quad (40)$$

The throughput for HTTP-request line handling operations is denoted by

$$\gamma_{H1} = \lambda \cdot q_H (1 - p_{\text{loss-}H1}). \quad (41)$$

(2) HTTP-Header Handling Operations. With respect to HTTP-header handling operations, the arrival rate of packets equals the throughput for the HTTP-request line handling operations, γ_{H2}, the buffer size is K_{H2}, the number of service windows is represented by N_{H2}, the number of rules is r_{H2}, and the service rate of the rules is μ_{H2}.

The probability that k packets arrive at a queue within the service time of a packet may be denoted by

$$\partial_k = \frac{\lambda^k}{k!} \frac{(N_{H2}\mu_{H2})^{r_{H2}}}{(r_{H2}-1)!} \cdot \frac{(k+r_{H2}-1)!}{(\lambda + N_{H2}\mu_{H2})^{k+r_{H2}}}. \quad (42)$$

The state transition probability for HTTP-header handling operations may be expressed by

$$p_{0k} = \begin{cases} \partial_k & 0 \le k \le K_{H2} + N_{H2} - 2 \\ \sum_{m=K_{H2}+N_{H2}-1}^{\infty} \partial_m & k = K_{H2} + N_{H2} - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \le k \le K_{H2} + N_{H2} - 2 \\ \sum_{m=K_{H2}+N_{H2}-1}^{\infty} \partial_m & k = K_{H2} + N_{H2} - 1 \end{cases} \quad 1 \le j \le K_{H2} + N_{H2} - 1. \quad (43)$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{H2}+N_{H2}-1} \pi_j p_{jk} \quad 0 \le k \le K_{H2} + N_{H2} - 1. \quad (44)$$

By performing ratio computation with $\pi 0$, the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{H2}+N_{H2}-1} (\pi_k/\pi_0)}. \quad (45)$$

The packet loss rate for HTTP-header handling operations may be expressed by

$$p_{\text{loss-}H2} = 1 - \frac{1}{\pi_0 + \gamma_{H1} \cdot r_{H2}/N_{H2}\mu_{H2}}. \quad (46)$$

The throughput for the HTTP-header handling operations may be expressed by

$$\gamma_{H2} = \gamma_{H1} (1 - p_{\text{loss-}H2}). \quad (47)$$

(3) HTTP-Body Handling Operations. In terms of HTTP-body handling operations, assume that there are n' concurrent HTTP-body handling modules and let i denote the ith HTTP-body handling module. The packet flow proportion for each HTTP-body handling module is denoted by $q_{H\text{-}i}$. The arrival rate of packets equals the throughput for the HTTP-header handling operations, $\gamma_{H2} * q_{H\text{-}i}$, the buffer size is $K_{H3\text{-}i}$, the number of service windows is expressed by $N_{H3\text{-}i}$, the number of rules is $r_{H3\text{-}i}$, and the service rate of the rules is represented by $\mu_{H3\text{-}i}$.

The probability that k packets arrive at a queue within the service time of a packet may be denoted by

$$\partial_k = \frac{\lambda^k}{k!} \frac{(N_{H3\text{-}i}\mu_{H3\text{-}i})^{r_{H3}}}{(r_{H3\text{-}i}-1)!} \cdot \frac{(k+r_{H3\text{-}i}-1)!}{(\lambda + N_{H3\text{-}i}\mu_{H3\text{-}i})^{k+r_{H3\text{-}i}}}. \quad (48)$$

The state transition probability for HTTP-body handling operations is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \le k \le K_{H3\text{-}i} + N_{H3\text{-}i} - 2 \\ \sum_{m=K_{H3\text{-}i}+N_{H3\text{-}i}-1}^{\infty} \partial_m & k = K_{H3\text{-}i} + N_{H3\text{-}i} - 1 \end{cases}$$

$$j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j - 1 \le k \le K_{H3\text{-}i} + N_{H3\text{-}i} - 2 \\ \sum_{m=K_{H3\text{-}i}+N_{H3\text{-}i}-1}^{\infty} \partial_m & k = K_{H3\text{-}i} + N_{H3\text{-}i} - 1 \end{cases} \quad (49)$$

$$1 \le j \le K_{H3\text{-}i} + N_{H3\text{-}i} - 1.$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{H3\text{-}i}+N_{H3\text{-}i}-1} \pi_j p_{jk} \quad 0 \le k \le K_{H3\text{-}i} + N_{H3\text{-}i} - 1. \quad (50)$$

By performing ratio computation with $\pi 0$, the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{H3\text{-}i}+N_{H3\text{-}i}-1} (\pi_k/\pi_0)}. \quad (51)$$

The packet loss rate for HTTP-body handling operations may be denoted by

$$p_{\text{loss-}H3\text{-}i} = 1 - \frac{1}{\pi_0 + \gamma_{H2} \cdot q_{H\text{-}i} \cdot r_{H3\text{-}i}/N_{H3\text{-}i}\mu_{H3\text{-}i}}. \quad (52)$$

The throughput for HTTP-body handling operations may be expressed by

$$\gamma_{H3\text{-}i} = \gamma_{H2} \cdot q_{H\text{-}i} (1 - p_{\text{loss-}H3\text{-}i}). \quad (53)$$

The total throughput for HTTP-body handling operations may be expressed by

$$\gamma_H = \sum_{i=1}^{n'} \gamma_{H3\text{-}i}. \quad (54)$$

The overall traffic handled by the firewall should be the sum of the traffic handled by DNS, FTP, and HTTP. The overall packet loss rate of the firewall is given by

$$p_{\text{loss}} = p_{\text{loss-}L} + q_D \cdot (1 - p_{\text{loss-}L})\, p_{\text{loss-}D} + q_F \cdot (1 - p_{\text{loss-}L})\, p_{\text{loss-}F1} + \sum_{i=1}^{n} q_{Fi} \cdot [1 - q_F \cdot (1 - p_{\text{loss-}L})\, p_{\text{loss-}F1}] \cdot p_{\text{loss-}F2\text{-}i} + q_H \cdot (1 - p_{\text{loss-}L})\, p_{\text{loss-}H1} + [1 - q_H \cdot (1 - p_{\text{loss-}L})\, p_{\text{loss-}H1}] \cdot p_{\text{loss-}H2} + \sum_{i=1}^{n'} q_{Hi} \cdot \{1 - [1 - q_H \cdot (1 - p_{\text{loss-}L})\, p_{\text{loss-}H1}] \cdot p_{\text{loss-}H2}\} \cdot p_{\text{loss-}H3\text{-}i}. \quad (55)$$

4. Experiment Evaluation

This section describes a discrete-event simulated experiment conducted with respect to the firewall model described above. The arrival time of packets and processing time of service windows are stochastically generated, where the arrival process is a Poisson process and the service process follows an Erlang distribution.

Let the number of total resources be 12, enumerate the combinations of the allocated resources, compute the values of the throughput and packet loss rate under each combination using the theoretical formulae and simulation programs, respectively, and calculate the error between the theoretical and simulated values.

The total number of service stations is 12, and the arrival rate of data packets λ = 80 kpps (kilo packets per second). In Phase 1, the buffer size K_L = 100, the number of rules r_L = 5, and the handling rate μ_L = 250 kpps. In Phase 2, the DNS traffic accounts for 20%; FTP traffic, 20%; and HTTP traffic, 60%. In terms of the DNS handling part, the buffer size K_D = 100, the number of rules r_D = 5, and the handling rate μ_D = 200 kpps. In the FTP handling part, the FTP command handling buffer size K_{F1} = 100, the number of rules r_{F1} = 5, and the handling rate μ_{F1} = 350 kpps. The FTP data transmission handling part may be divided into two data transmission processes, whose traffic proportions are 40% and 60%, respectively. For the first data transmission process, the buffer size $K_{F2\text{-}1}$ = 80, the number of rules $r_{F2\text{-}1}$ = 5, and the handling rate $\mu_{F2\text{-}1}$ = 150 kpps. For the second transmission process, the buffer size $K_{F2\text{-}2}$ = 90, the number of rules $r_{F2\text{-}1}$ = 5, and the handling rate $\mu_{F2\text{-}1}$ = 180 kpps. In the HTTP handling part, the buffer size for HTTP-request line handling K_{H1} = 100, the number of rules r_{H1} = 5, and the handling rate μ_{H1} = 500 kpps. The buffer size for HTTP-header handling K_{H2} = 150, the number of rules r_{H2} = 5, and the handling rate μ_{H2} = 200 kpps. The HTTP-body handling operations consist of two HTTP applications, whose traffic proportions are 40% and 60%, respectively. The buffer sizes are $K_{H3\text{-}1}$ = 100 and $K_{H3\text{-}2}$ = 100, respectively, the number of rules is $r_{H3\text{-}1}$ = 5 and $r_{H3\text{-}2}$ = 5, respectively, and the handling rates are $\mu_{H3\text{-}1}$ = 80 kpps and $\mu_{H3\text{-}2}$ = 100 kpps, respectively. The results are presented in Table 1.

As determined from the experimental results, when the resources are allocated as N_L = 2, N_D = 1, N_{F1} = 1, $N_{F2\text{-}1}$ = 1, $N_{F2\text{-}2}$ = 1, N_{H1} = 1, N_{H2} = 2, $N_{H3\text{-}1}$ = 1, and $N_{H3\text{-}2}$ = 2, the maximum throughput reaches up to 76.69 kpps with the lowest packet loss rate. Further, the error between the theoretical computation results and the simulated experiment results remains within 3.3%; the mean error is 0.848%.

5. Conclusion

A two-phase multiservice station and multiprotocol firewall model with multiple concurrent applications was proposed

Table 1: Relationship of resource allocation versus throughput and packet loss rate.

N_L	N_D	N_{F1}	$N_{F2\text{-}1}$	$N_{F2\text{-}2}$	N_{H1}	N_{H2}	$N_{H3\text{-}1}$	$N_{H3\text{-}2}$	Theoretical throughput	Theoretical packet loss rate	Simulated throughput	Simulated packet loss rate	Err
1	1	1	1	1	1	1	1	4	50.252	0.372	49.525	0.381	0.015
1	1	1	1	1	1	1	2	3	50.777	0.365	49.545	0.381	0.025
1	1	1	1	1	1	1	3	2	50.322	0.371	49.504	0.381	0.017
1	1	1	1	1	1	1	4	1	50.166	0.373	49.297	0.384	0.018
1	1	1	1	1	1	2	1	3	50.273	0.372	49.545	0.381	0.015
1	1	1	1	1	1	2	2	2	50.181	0.373	49.647	0.379	0.011
1	1	1	1	1	1	2	3	1	51.012	0.362	49.421	0.382	0.032
1	1	1	1	1	1	3	1	2	50.456	0.369	49.507	0.381	0.019
1	1	1	1	1	1	3	2	1	50.257	0.372	49.380	0.383	0.018
1	1	1	1	1	1	4	1	1	49.927	0.376	49.496	0.381	0.009
1	1	1	1	1	2	1	1	3	50.483	0.369	49.405	0.382	0.022
1	1	1	1	1	2	1	2	2	50.720	0.366	49.518	0.381	0.024
1	1	1	1	1	2	1	3	1	50.525	0.368	49.289	0.384	0.025
1	1	1	1	1	2	2	1	2	50.519	0.369	49.496	0.381	0.021
1	1	1	1	1	2	2	2	1	50.690	0.366	49.453	0.382	0.025
1	1	1	1	1	2	3	1	1	50.775	0.365	49.425	0.382	0.027
1	1	1	1	1	3	1	1	2	50.291	0.371	49.455	0.382	0.017
1	1	1	1	1	3	1	2	1	50.226	0.372	49.471	0.382	0.015
1	1	1	1	1	3	2	1	1	50.228	0.372	49.112	0.386	0.023
1	1	1	1	1	4	1	1	1	50.665	0.367	49.351	0.383	0.027
1	1	1	1	2	1	1	1	3	50.308	0.371	49.676	0.379	0.013
1	1	1	1	2	1	1	2	2	50.315	0.371	49.469	0.382	0.017
1	1	1	1	2	1	1	3	1	50.211	0.372	49.351	0.383	0.017
1	1	1	1	2	1	2	1	2	50.310	0.371	49.505	0.381	0.016
1	1	1	1	2	1	2	2	1	50.627	0.367	49.433	0.382	0.024
1	1	1	1	2	1	3	1	1	50.212	0.372	49.396	0.383	0.017
1	1	1	1	2	2	1	1	2	50.889	0.364	49.567	0.380	0.027
1	1	1	1	2	2	1	2	1	50.514	0.369	49.484	0.381	0.021
1	1	1	1	2	2	2	1	1	50.446	0.369	49.150	0.386	0.026
1	1	1	1	2	3	1	1	1	50.177	0.373	49.393	0.383	0.016
1	1	1	1	3	1	1	1	2	50.569	0.368	49.578	0.380	0.020
1	1	1	1	3	1	1	2	1	50.724	0.366	49.376	0.383	0.027
1	1	1	1	3	1	2	1	1	50.240	0.372	49.408	0.382	0.017
1	1	1	1	3	2	1	1	1	50.818	0.365	49.380	0.383	0.029
1	1	1	1	4	1	1	1	1	50.548	0.368	49.206	0.385	0.027
1	1	1	2	1	1	1	1	3	50.689	0.366	49.436	0.382	0.025
1	1	1	2	1	1	1	2	2	50.800	0.365	49.524	0.381	0.026
1	1	1	2	1	1	1	3	1	50.616	0.367	49.140	0.386	0.030
1	1	1	2	1	1	2	1	2	50.589	0.368	49.485	0.381	0.022
1	1	1	2	1	1	2	2	1	50.552	0.368	49.057	0.387	0.030
1	1	1	2	1	1	3	1	1	50.585	0.368	49.365	0.383	0.025
1	1	1	2	1	2	1	1	2	50.558	0.368	49.503	0.381	0.021
1	1	1	2	1	2	1	2	1	50.264	0.372	49.157	0.386	0.023
1	1	1	2	1	2	2	1	1	50.307	0.371	49.194	0.385	0.023
1	1	1	2	1	3	1	1	1	49.989	0.375	49.261	0.384	0.015
1	1	1	2	2	1	1	1	2	50.404	0.370	49.533	0.381	0.018
1	1	1	2	2	1	1	2	1	50.710	0.366	49.593	0.380	0.023

Table 1: Continued.

N_L	N_D	N_{F1}	$N_{F2\text{-}1}$	$N_{F2\text{-}2}$	N_{H1}	N_{H2}	$N_{H3\text{-}1}$	$N_{H3\text{-}2}$	Theoretical throughput	Theoretical packet loss rate	Simulated throughput	Simulated packet loss rate	Err
1	1	1	2	2	1	2	1	1	50.271	0.372	49.426	0.382	0.017
1	1	1	2	2	2	1	1	1	50.344	0.371	49.333	0.383	0.020
1	1	1	2	3	1	1	1	1	50.468	0.369	49.524	0.381	0.019
1	1	1	3	1	1	1	1	2	50.982	0.363	49.477	0.382	0.030
1	1	1	3	1	1	1	2	1	50.504	0.369	49.180	0.385	0.027
1	1	1	3	1	1	2	1	1	50.608	0.367	49.106	0.386	0.031
1	1	1	3	1	2	1	1	1	50.326	0.371	49.294	0.384	0.021
1	1	1	3	2	1	1	1	1	50.096	0.374	49.408	0.382	0.014
1	1	1	4	1	1	1	1	1	50.104	0.374	49.302	0.384	0.016
1	1	2	1	1	1	1	1	3	50.105	0.374	49.656	0.379	0.009
1	1	2	1	1	1	1	2	2	50.371	0.370	49.459	0.382	0.018
1	1	2	1	1	1	1	3	1	50.209	0.372	49.377	0.383	0.017
1	1	2	1	1	1	2	1	2	50.027	0.375	49.517	0.381	0.010
1	1	2	1	1	1	2	2	1	50.769	0.365	49.471	0.382	0.026
1	1	2	1	1	1	3	1	1	50.644	0.367	49.533	0.381	0.022
1	1	2	1	1	2	1	1	2	50.417	0.370	49.507	0.381	0.018
1	1	2	1	1	2	1	2	1	49.918	0.376	49.445	0.382	0.010
1	1	2	1	1	2	2	1	1	50.770	0.365	49.469	0.382	0.026
1	1	2	1	1	3	1	1	1	50.103	0.361	49.309	0.384	0.016
1	1	2	1	2	1	1	1	2	50.238	0.372	49.503	0.381	0.015
1	1	2	1	2	1	1	2	1	50.409	0.370	49.476	0.382	0.019
1	1	2	1	2	1	2	1	1	50.696	0.366	49.397	0.383	0.026
1	1	2	1	2	2	1	1	1	50.456	0.369	49.504	0.381	0.019
1	1	2	1	3	1	1	1	1	50.461	0.369	49.256	0.384	0.024
1	1	2	2	1	1	1	1	2	50.134	0.373	49.695	0.379	0.009
1	1	2	2	1	1	1	2	1	50.997	0.363	49.517	0.381	0.030
1	1	2	2	1	1	2	1	1	50.661	0.367	49.401	0.382	0.026
1	1	2	2	1	2	1	1	1	50.414	0.370	49.497	0.381	0.019
1	1	2	2	2	1	1	1	1	50.497	0.369	49.395	0.383	0.022
1	1	2	3	1	1	1	1	1	50.559	0.368	49.366	0.383	0.024
1	1	3	1	1	1	1	1	2	50.271	0.372	49.617	0.380	0.013
1	1	3	1	1	1	1	2	1	50.593	0.368	49.325	0.383	0.026
1	1	3	1	1	1	2	1	1	50.775	0.365	49.488	0.381	0.026
1	1	3	1	1	2	1	1	1	50.413	0.370	49.475	0.382	0.019
1	1	3	1	2	1	1	1	1	50.368	0.370	49.496	0.381	0.018
1	1	3	2	1	1	1	1	1	50.521	0.368	49.376	0.383	0.023
1	1	4	1	1	1	1	1	1	50.204	0.372	49.476	0.382	0.015
1	2	1	1	1	1	1	1	3	50.709	0.366	49.569	0.380	0.023
1	2	1	1	1	1	1	2	2	50.261	0.372	49.497	0.381	0.015
1	2	1	1	1	1	1	3	1	50.264	0.372	49.469	0.382	0.016
1	2	1	1	1	1	2	1	2	50.617	0.367	49.596	0.380	0.021
1	2	1	1	1	1	2	2	1	51.092	0.361	49.449	0.382	0.033
1	2	1	1	1	1	3	1	1	50.560	0.368	49.199	0.385	0.028
1	2	1	1	1	2	1	1	2	50.335	0.371	49.425	0.382	0.018
1	2	1	1	1	2	1	2	1	50.678	0.367	49.385	0.383	0.026
1	2	1	1	1	2	2	1	1	50.582	0.368	49.533	0.381	0.021
1	2	1	1	1	3	1	1	1	50.046	0.374	49.250	0.384	0.016
1	2	1	1	2	1	1	1	2	50.790	0.365	49.413	0.382	0.028

Table 1: Continued.

N_L	N_D	N_{F1}	$N_{F2\text{-}1}$	$N_{F2\text{-}2}$	N_{H1}	N_{H2}	$N_{H3\text{-}1}$	$N_{H3\text{-}2}$	Theoretical throughput	Theoretical packet loss rate	Simulated throughput	Simulated packet loss rate	Err
1	2	1	1	2	1	1	2	1	50.258	0.372	49.320	0.384	0.019
1	2	1	1	2	1	2	1	1	50.477	0.369	49.399	0.383	0.022
1	2	1	1	2	2	1	1	1	50.587	0.368	49.272	0.384	0.027
1	2	1	1	3	1	1	1	1	50.573	0.368	49.371	0.383	0.024
1	2	1	2	1	1	1	1	2	50.867	0.364	49.499	0.381	0.028
1	2	1	2	1	1	1	2	1	50.508	0.369	49.524	0.381	0.020
1	2	1	2	1	1	2	1	1	50.313	0.371	49.481	0.381	0.017
1	2	1	2	1	2	1	1	1	50.419	0.370	49.468	0.382	0.019
1	2	1	2	2	1	1	1	1	50.919	0.364	49.391	0.383	0.031
1	2	1	3	1	1	1	1	1	50.633	0.367	49.496	0.381	0.023
1	2	2	1	1	1	1	1	2	50.756	0.366	49.582	0.380	0.024
1	2	2	1	1	1	1	2	1	50.920	0.363	49.349	0.383	0.032
1	2	2	1	1	1	2	1	1	50.259	0.372	49.465	0.382	0.016
1	2	2	1	1	2	1	1	1	50.410	0.370	49.364	0.383	0.021
1	2	2	1	2	1	1	1	1	50.116	0.374	49.503	0.381	0.012
1	2	2	2	1	1	1	1	1	50.377	0.370	49.360	0.383	0.021
1	2	3	1	1	1	1	1	1	50.517	0.369	49.444	0.382	0.022
1	3	1	1	1	1	1	1	2	50.352	0.371	49.692	0.379	0.013
1	3	1	1	1	1	1	2	1	50.652	0.367	49.458	0.382	0.024
1	3	1	1	1	1	2	1	1	50.375	0.370	49.480	0.381	0.018
1	3	1	1	1	2	1	1	1	50.294	0.371	49.354	0.383	0.019
1	3	1	1	2	1	1	1	1	50.999	0.363	49.476	0.382	0.031
1	3	1	2	1	1	1	1	1	51.162	0.360	49.525	0.381	0.033
1	3	2	1	1	1	1	1	1	50.366	0.370	49.316	0.384	0.021
1	4	1	1	1	1	1	1	1	50.761	0.365	49.410	0.382	0.027
2	1	1	1	1	1	1	1	3	72.170	0.098	71.534	0.106	0.009
2	1	1	1	1	1	1	2	2	72.857	0.089	71.463	0.107	0.019
2	1	1	1	1	1	1	3	1	68.360	0.146	67.650	0.154	0.010
2	1	1	1	1	1	2	1	2	77.768	0.028	76.693	0.041	0.014
2	1	1	1	1	1	2	2	1	72.132	0.098	70.706	0.116	0.020
2	1	1	1	1	1	3	1	1	68.340	0.146	67.672	0.154	0.010
2	1	1	1	1	2	1	1	2	72.507	0.094	71.572	0.105	0.013
2	1	1	1	1	2	1	2	1	68.342	0.146	67.676	0.154	0.010
2	1	1	1	1	2	2	1	1	68.404	0.145	67.688	0.154	0.011
2	1	1	1	1	3	1	1	1	68.321	0.146	67.742	0.153	0.009
2	1	1	1	2	1	1	1	2	72.072	0.099	71.473	0.107	0.008
2	1	1	1	2	1	1	2	1	68.403	0.145	67.604	0.155	0.012
2	1	1	1	2	1	2	1	1	68.357	0.146	67.684	0.154	0.010
2	1	1	1	2	2	1	1	1	68.236	0.147	67.687	0.154	0.008
2	1	1	1	3	1	1	1	1	68.010	0.150	67.678	0.154	0.005
2	1	1	2	1	1	1	1	2	72.688	0.091	71.439	0.107	0.017
2	1	1	2	1	1	1	2	1	68.301	0.146	67.607	0.155	0.010
2	1	1	2	1	1	2	1	1	68.221	0.147	67.736	0.153	0.007
2	1	1	2	1	2	1	1	1	68.200	0.148	67.688	0.154	0.008
2	1	1	2	2	1	1	1	1	67.916	0.151	67.650	0.154	0.004
2	1	1	3	1	1	1	1	1	68.191	0.148	67.682	0.154	0.008
2	1	2	1	1	1	1	1	2	72.500	0.094	71.590	0.105	0.013
2	1	2	1	1	1	1	2	1	68.189	0.148	67.531	0.156	0.010
2	1	2	1	1	1	2	1	1	68.121	0.148	67.753	0.153	0.005

Table 1: Continued.

N_L	N_D	N_{F1}	$N_{F2\text{-}1}$	$N_{F2\text{-}2}$	N_{H1}	N_{H2}	$N_{H3\text{-}1}$	$N_{H3\text{-}2}$	Theoretical throughput	Theoretical packet loss rate	Simulated throughput	Simulated packet loss rate	Err
2	1	2	1	1	2	1	1	1	68.334	0.146	67.689	0.154	0.010
2	1	2	1	2	1	1	1	1	68.261	0.147	67.690	0.154	0.008
2	1	2	2	1	1	1	1	1	67.930	0.151	67.670	0.154	0.004
2	1	3	1	1	1	1	1	1	68.234	0.147	67.619	0.155	0.009
2	2	1	1	1	1	1	1	2	72.719	0.091	71.508	0.106	0.017
2	2	1	1	1	1	1	2	1	68.576	0.143	67.641	0.154	0.014
2	2	1	1	1	1	2	1	1	68.479	0.144	67.734	0.153	0.011
2	2	1	1	1	2	1	1	1	68.157	0.148	67.639	0.155	0.008
2	2	1	1	2	1	1	1	1	68.602	0.142	67.681	0.154	0.014
2	2	1	2	1	1	1	1	1	68.034	0.150	67.622	0.155	0.006
2	2	2	1	1	1	1	1	1	68.375	0.145	67.679	0.154	0.010
2	3	1	1	1	1	1	1	1	68.229	0.147	67.611	0.155	0.009
3	1	1	1	1	1	1	1	2	72.655	0.092	71.542	0.106	0.016
3	1	1	1	1	1	1	2	1	68.046	0.149	67.601	0.155	0.007
3	1	1	1	1	1	2	1	1	68.268	0.147	67.725	0.153	0.008
3	1	1	1	1	2	1	1	1	68.143	0.148	67.674	0.154	0.007
3	1	1	1	2	1	1	1	1	68.494	0.144	67.685	0.154	0.012
3	1	1	2	1	1	1	1	1	68.394	0.145	67.707	0.154	0.010
3	1	2	1	1	1	1	1	1	68.241	0.147	67.700	0.154	0.008
3	2	1	1	1	1	1	1	1	68.154	0.148	67.638	0.155	0.008
4	1	1	1	1	1	1	1	1	67.796	0.153	67.642	0.154	0.002

in this paper. Based on different phases, protocols, and applications, the values of performance indicators such as system throughput and packet loss rate were obtained. An optimal solution was derived after the results of simulated experiments and theoretical computation were compared, and the combinations of resource allocations were enumerated using a total of 12 resources. Furthermore, by comparing the error between the simulated experiment values and the theoretical computation values, it was found that this model may precisely represent the handling process of the firewall. Therefore, it may save a considerable amount of time in development and testing from the viewpoint of utilizing mobile networks, thus exhibiting significant potential for practical application. In the future, we plan to continue with our in-depth research by emphasizing the discussion of DDoS detection and user-behavior analysis.

Disclosure

The funding agency had no role in the study design, the collection, analysis, or interpretation of data, the writing of the report, or the decision to submit the article for publication.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program (2016YFB0801503).

References

[1] T. Wang, Z. Su, Y. Xia, B. Qin, and M. Hamdi, "Towards cost-effective and low latency data center network architecture," *Computer Communications*, vol. 82, pp. 1–12, 2016.

[2] K.-K. R. Choo, "The cyber threat landscape: challenges and future research directions," *Computers and Security*, vol. 30, no. 8, pp. 719–731, 2011.

[3] V. Prokhorenko, K.-K. R. Choo, and H. Ashman, "Web application protection techniques: a taxonomy," *Journal of Network and Computer Applications*, vol. 60, pp. 95–112, 2016.

[4] J. Peng, K.-K. R. Choo, and H. Ashman, "Bit-level n-gram based forensic authorship analysis on social media: identifying individuals from linguistic profiles," *Journal of Network and Computer Applications*, vol. 70, pp. 171–182, 2016.

[5] J. Peng, K. K. R. Choo, and H. Ashman, "User profiling in intrusion detection: a review," *Journal of Network and Computer Applications*, vol. 72, pp. 14–27, 2016.

[6] J. Peng, R. K.-K. Choo, and H. Ashman, "Astroturfing detection in social media: using binary n-gram analysis for authorship attribution," in *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*, 2016.

[7] M. Chen, Y. Qian, S. Mao, W. Tang, and X. Yang, "Software-defined mobile networks security," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 729–743, 2016.

[8] N. Bui and J. Widmer, "Mobile network resource optimization under imperfect prediction," in *Proceedings of the 16th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM '15*, 9, 1 pages, USA, 2015.

[9] A. Nikravesh, H. Yao, S. Xu, D. Choffnes, and Z. M. Mao, "Mobilyzer: an open platform for controllable mobile network measurements," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '15*, pp. 389–404, 2015.

[10] C. Herrmann, "Complete analysis of the discrete time finite DBMAP/G/1/N queue," *Performance Evaluation*, vol. 43, no. 2-3, pp. 95–121, 2001.

[11] T. Wang and M. Hamdi, "Presto: towards efficient online virtual network embedding in virtualized cloud data centers," *Computer Networks*, vol. 106, pp. 196–208, 2016.

[12] H. Khazaei, J. Misic, and V. B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 936–943, 2012.

[13] H. Khazaei, J. Mišić, V. B. Mišić, and S. Rashwand, "Analysis of a pool management scheme for cloud computing centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 849–861, 2013.

[14] H. Khazaei, J. Misic, and V. B. Misic, "A fine-grained performance model of cloud computing centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2138–2147, 2013.

[15] K. Salah, "Queuing analysis of network firewalls," in *Proceedings of the 53rd IEEE Global Communications Conference, GLOBECOM '10*, 2010.

[16] K. Salah, K. Elbadawi, and R. Boutaba, "Performance modeling and analysis of network firewalls," *IEEE Transactions on Network and Service Management*, vol. 9, no. 1, pp. 12–21, 2012.

[17] K. Salah, "Analysis of Erlangian network services," *AEÜ - International Journal of Electronics and Communications*, vol. 68, no. 7, pp. 623–630, 2014.

[18] S. Zapechnikov, N. Miloslavskaya, and A. Tolstoy, "Modeling of next-generation firewalls as queueing services," in *Proceedings of the 8th International Conference on Security of Information and Networks, SIN '15*, ACM, 2015.

[19] R. Ghosh, F. Longo, V. K. Naik, and K. S. Trivedi, "Modeling and performance analysis of large scale IaaS clouds," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1216–1234, 2013.

[20] K. Salah, J. M. A. Calero, S. Zeadally, S. Al-Mulla, and M. Alzaabi, "Using cloud computing to implement a security overlay network," *IEEE Security and Privacy*, vol. 11, no. 1, pp. 44–53, 2013.

[21] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile Micro-Task Allocation in spatial crowdsourcing," in *Proceedings of the 32nd IEEE International Conference on Data Engineering, ICDE '16*, pp. 49–60, IEEE, Helsinki, Finland, 2016.

[22] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "SLADE: a smart large-scale task decomposer in crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1-1, 2018.

[23] S. Xuan, W. Yang, H. Dong, and J. Zhang, "Performance evaluation model for application layer firewalls," *PLoS ONE*, vol. 11, no. 11, Article ID e0167280, 2016.