

Load Shedding Based Resource Management Techniques for RFID Data

Nova Ahmed
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0250
Email: nova@cc.gatech.edu

Umakishore Ramachandran
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0250
Email: rama@cc.gatech.edu

Abstract—RFID based systems are enjoying widespread adoption in a variety of application scenarios. Item tracking in a supply chain environment is one such application. From an application perspective, there are two challenges: (a) the data rates for large deployments are growing significantly; (b) the demands placed on the system for query processing in real time are also on the rise. Meeting these challenges in large-scale deployments is non trivial. The hardware base for RFID based systems compound these challenges due to the fact that RFID readers are error-prone and reliable reading of RFID tags is hampered by a number of physical limitations such as environmental conditions, and contents of the items carrying the tags. Naturally, the reliability of these systems become even more questionable when both the data rates increase and the need for real time processing of queries increases. We propose load shedding mechanisms that use the spatial and temporal properties of RFID deployments to combat the challenges due to increased demands for tag and query processing in real time. These mechanisms are piggy-backed on top of a middleware *Reliable Framework for Radio Frequency Identification (RF²ID)* that uses redundancy to improve the reliability of RFID deployment. The basic idea in RF²ID is to use the spatial notion of a *path* taken by items flowing from source to destination. By cumulatively aggregating the tags collected by entities called *Virtual Readers (VR)* that are placed along the path, the total reliability of the system is enhanced. The VRs cooperatively shed the load under heavy load conditions. The built-in redundancy in the RF²ID system allows the VRs to shed load with reasonable system performance thus enhancing the overall reliability of the deployment. Two different load shedding strategies are proposed in the literature: *space based approach* and *time based approach*. These strategies have been implemented in the RF²ID middleware and performance results show the efficacy of these mechanisms for dealing with increased data rates.

I. INTRODUCTION

In recent years, RFID based technology has become ubiquitous and has gained significant interest. The technology is used in a large variety of applications ranging from health care, industry, office and home applications for identification, location and tracking purpose. The current focus of our work is on item tracking applications where statically placed RFID readers track tagged objects in motion, e.g., tracking of tagged goods in a supply chain or tagged baggage in an airport by the reader. Large-scale deployment of RFID technology faces several challenges. There is an *overwhelming amount of data* generated in by the RFID readers in large applications [1], [2],

that can result in serious system overload. For example, UPS handles 13.6 million packages a day that produce 1.3 GB of raw data [3]. It seems reasonable to assume that the amount of data generated will only keep growing in the future [4]. A retail strategy consulting company estimates Wal-Mart could generate as much as 7.7 million terabytes of data per day if all its items are tagged [5], [6]. The problem is exacerbated by the *error prone nature* of the RFID generated data [7]–[10] and its *asynchronous flow* in the system [11]. Further, applications involving such data are typically *time sensitive* (for example, real-time forecasting and inventory control) [12]. A system designed to manage large scale RFID deployment must meet the above challenges.

There have been some recent studies addressing load management in RFID systems. The heuristic based solution [13] provides near optimal reader to tag assignment for statically placed reader and tags rather than considering tags in motion for item location applications. Load migration based approach is used in various research works for RFID data. Examples include a mobile agent [14] to re-adjust imbalance of load, and the connection pool based method [15] where load is re-adjusted from heavily loaded nodes to lightly loaded nodes. Load migration based techniques are not very suitable for streaming data to meet real time constraints [16], [17]. We review the relevant literature in more detail in the next section.

In our proposed work, we use the data flow in the system as an ally to make load distribution decisions. In item tracking applications, items follow a path from a source to destination. The key idea is to allow collaboration among resources along a particular data flow path. A system that has an understanding of the data flow at system level has been developed as RF²ID (Reliable Framework for Radio Frequency Identification) to combat the error prone nature of the RFID technology [18]. By using redundancy and exploiting the relative flow of RFID tagged objects with respect to the readers, RF²ID increases the reliability of an RFID deployment. RF²ID provides two system level abstractions for increasing the reliability: *Virtual Reader (VR)* and *Virtual Path (VPath)*. Each VR is assigned to a distinct geographical region and by collecting data from physical readers in its vicinity, it locally increases the reliability of the deployment. Further, the VRs in the direction of flow of the items form a logical path (VPath). By exchanging data

with one another along the VPath, the VRs further increase the global reliability of the deployment. The net result of this middleware is to increase the reliability of the entire deployment (by reducing the rate of *false negatives*, i.e., the number of tags that are missed by the system) in spite of the error prone nature of the physical readers.

The focus of our work is to deal with the volume of data and the time sensitive nature of both tag and query processing, by incorporating *cooperative load shedding* into the RF²ID middleware. The key insight is once again to leverage the notion of *path* for dynamic load distribution among the VRs under heavy load for item tracking applications. We propose intelligent load shedding mechanisms that avoid or minimizes data loss - when one VR sheds load, another VR along the same path most likely covers for the data loss by the first one. Two different load shedding approaches are proposed in this paper: *space based* load shedding mechanism and *time based* load shedding mechanism. In the space based approach, each VR dynamically decides to read a subset of the tags on the path that it is participating, based on their physical ordering in space. In time based load shedding, each VR dynamically decides to read tags for a chosen time interval. We show by experimentation that our load shedding schemes achieve the desired performance results without significantly sacrificing reliability. The evaluation testbed considers real world constraints such as tag mobility, presence of multiple RFID readers and multiple paths using real PRs as well as emulated PRs for item tracking applications. The workload is different for various application scenarios. The application requirements play a major role in the selection of a suitable load management strategy. The time based strategy achieves improved performance for synchronous data arrival, while minimizing the power consumption, as it allows readers to operate in dormant mode in different time intervals. The space based method is the simplest strategy that sheds load and achieves improved performance taking into account the built in redundancy in our system.

Through this work, we make the following *contributions*:

- We present novel load shedding algorithms for dealing with both the large volume of data and its time sensitive nature in item tracking applications taking into account the data flow information using *space based* load shedding mechanism and *time based* load shedding mechanism.
- We present a detailed performance study of our RF²ID system that embodies these load shedding algorithms using a real experimental testbed deployment using real physical RFID readers as well as emulated physical RFID readers.

The paper is organized in the following manner: section 2 presents the system architecture of RF²ID followed by the motivation for load shedding in section 3, section 4 provides the problem description, section 5 details the path based load shedding strategies, the system evaluation results are presented in section 6, followed by the related work in section 7 and the

conclusion is presented in section 8.

II. SYSTEM ARCHITECTURE

Figure 1 shows a schematic of RF²ID system architecture. The system architecture has been presented in [18]. It is being summarized in this section for the sake of completeness and for setting the stage for presenting the load shedding algorithms.

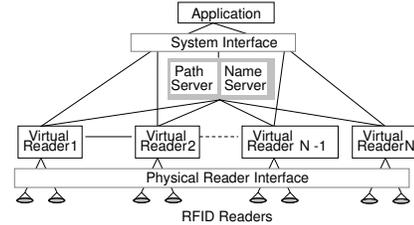


Fig. 1. System architecture of RF²ID

A. Virtual Readers (VRs) and Virtual Paths (VPaths)

An example deployment of VRs in a supply chain management system is shown in Figure 2. Every VR is in charge of a set of physical readers (PRs) in a particular geographic location as shown by the rectangular region. It has three physical routes: source A to destination D; source B to destination E; and source B to destination C. The VRs form a logical path that overlays these physical routes. As can be seen a given VR may participate in multiple Vpaths. The information accrued from different Vpaths and from the different physical routes supported by a given Vpath are kept distinct for the sake of inventory management by each VR. Each VR reports the tags it reads via inter-VR messaging to the VR that is downstream from it on the Vpath with the pertinent information (such as the physical route, number of tags, and the timestamp of the tags).

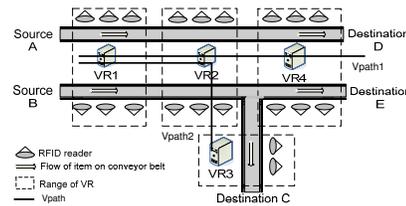


Fig. 2. A warehouse scenario using RF²ID deployment

Each VR is responsible for managing its connections, its load, and answering queries. As we said earlier, each Vpath is constituted by a set of VRs, and a given VR may be part of several Vpaths. The association of VR to its PRs happens at system initialization time (by the name server). However, the creation of Vpath is a bit more involved since it signifies commitment of system resources to manage the newly created path. Therefore, upon a Vpath creation request, each VR makes an assessment of its current network load and CPU load before it accepts to be part of a new Vpath. RF²ID uses

a distributed admission control mechanism orchestrated by the name server. A new path is created only if the current system load and the expected maximum system load are within a predefined threshold. It ensures that each VR can serve all the Vpaths that it is currently associated with, without any data loss. The system is well positioned to handle time specific queries since the VRs associate timestamps both with the information they gather as well as with the information they disseminate.

B. Name Server and Path Server

The name server illustrated in Figure 1 maps PRs to VRs based on the geographic location of the PRs. A path creation request to the name server identifies a source and destination. The name server identifies the possible physical route(s) to satisfy the request. It then contacts the VRs on the physical route to establish the Vpath, which then execute the distributed admission control algorithm in an attempt to establish the path (if such a path does not already exist). As the name suggests, the path server shown in Figure 1 maintains an inventory of active Vpaths in the system. Each path has a lifetime. On access to an existing path, it receives a new lease on its lifetime. When the lifetime expires, a given Vpath is garbage collected.

III. MOTIVATION

A VR is responsible for a set of physical RFID readers (PR) within its geographic vicinity. The volume of data produced in typical RFID deployments have been increasing significantly [16], [19]. Although the amount of data can be reduced by different filtering mechanisms, the system should be able to capture this amount of data in the first place for further processing and the problem gets complicated due to the bursty and often unpredictable nature of item arrival in many RFID deployments [11] that could lead to overloading the infrastructure and potential loss of data. To illustrate this problem, we have conducted some simple experiments using the RF²ID middleware. Each VR is a dual core Pentium-4 Xeon processor, which represents one node of a 56-node Linux cluster. The experiment studies the CPU load experienced by a VR as a function of the tag arrival rate. The CPU load indicates the load in a single VR. Figure 3 shows the impact on a VRs CPU load as the tag arrival rate is increased. As can be seen, the CPU load is close to 100% when the tag arrival rate is 50000 data per minute (y-axis of left hand side). Figure 3 also shows decrease in the accuracy in item detection as a result of increased tag arrival rate. The study on system accuracy uses 9 VRs, each attached to 5 PRs. It shows that we are able to achieve 100% accuracy for tag reading with a tag arrival rate of up to 20000 tags/minute (y-axis of right hand side). However, as we increase the tag arrival rate the accuracy of the system starts deteriorating. It can be seen that the system accuracy drops to 50% when the tag arrival rate is around 35000 tags/min. Basically, the VRs are overloaded and cannot cope with the arrival rate for tag processing. The interesting point to note is that due to the built-in redundancy of the

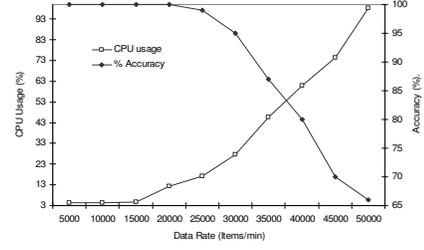


Fig. 3. CPU usage and data accuracy with respect to variable tag arrival

RF²ID middleware, the VRs may be spending time processing the same set of tags. We have observed the overlap in the set of tags read by distinct VRs to be between 40% and 80% in this experimental set up. Clearly, the VR CPU bandwidth used in processing this overlapping set of tags represents wasted resources. If we could harness this wasted CPU bandwidth for processing disjoint sets of tags then obviously we could achieve higher accuracy even at higher tag arrival rates. This is the intuition behind the cooperative load shedding approach presented in this work.

IV. PROBLEM DESCRIPTION

The RF²ID system comprises of multiple VRs denoted as $\{VR_1, VR_2, \dots, VR_n\}$. Each VR_i is associated with number of PRs as $\{PR_1, PR_2, \dots, PR_m\}$ within its geographic vicinity. The set of data items D arriving in the system is defined as $\{D_1, D_2, \dots, D_l\}$. The data items are directed to various paths in the system and the total data along a path is defined as $D_{(pathID, ptotal)}$ where ptotal is the total number of items intended along that path. For a total of p paths in the system, the total amount of data in the system is defined as: $D_{total} = \sum_{i=0}^p D_{(p, ptotal)}$. The PR accuracy level $A_{(PR, i)}$ is defined as the proportion of items correctly detected by PR_i ($D_{PR, i}$) and the total data for path p ($D_{(p, ptotal)}$): $A_{(PR, i)} = \frac{D_{(PR, i)}}{D_{(p, ptotal)}}$. The VR level accuracy A_{VR} is determined as a proportion of processed data of VR_j ($D_{(VR, j)}$) and the total data along path p ($D_{(p, ptotal)}$): $A_{(VR, j)} = \frac{D_{(VR, j)}}{D_{total}} = \frac{(\sum_{j=0}^m D_{PR, j})}{D_{(p, ptotal)}}$. The path level accuracy $A_{p, k}$ for a $path_k$ is determined by the proportion of processed data along a path p ($D_{(p, k)}$) and the total data along path p ($D_{(p, ptotal)}$): $A_{(p, k)} = \frac{D_{(p, k)}}{D_{(p, ptotal)}} = \frac{(\sum_{k=0}^n D_{VR, k})}{D_{(p, ptotal)}}$. The system level accuracy A_{sys} for p paths is determined by the following relation: $A_{sys} = \frac{D_{(sys)}}{D_{total}} = \frac{(\sum_{l=0}^p D_{path, l})}{D_{total}}$.

The goal is to achieve desirable accuracy level defined as threshold accuracy (A_{th}) even when the system is under heavy load. The basic idea is to let each VR process a subset of total data ($D_{proc} \subset D_{total}$) along a path under heavy load. Each VR_i makes decision on shedding a certain amount of load (D_{shed}) where $D_{proc} + D_{shed} = D_{total}$ making sure $A_{sys} \geq A_{th}$ holds.

V. LOAD SHEDDING STRATEGIES

Operationally, the load shedding strategy can be split into two phases. The first phase, namely, *load monitoring and overload detection* is common to all the load shedding strategies.

A. Load Monitoring and Overload Detection Phase

The computational load on an individual VR has two components:

- A *communication* component that comprises receiving messages from the PRs physically associated with it and for sending/receiving messages to/from the peer VRs logically associated by the paths with this VR.
- A *computation* component that comprises a complex mixture of processing time for reading and classifying tags from PRs into the physical routes and logical paths associated with this VR; processing time for filtering the tags (elimination of false positives, etc.) based on the messages received from peer VRs; executing admission control algorithm in response to path creation requests; and processing queries for inventory control.

Each VR has to ensure that the cumulative load does not exceed its computational capacity (C). We express the above computational load on a VR in terms of its physical and logical connectivity. Let each VR be connected to m PRs and p paths. Let PR_i be the computational load due to an individual physical connection i ; let $Path_j$ be the computational load due to an individual path j . The instantaneous load L on a VR is expressed as: $L = \sum_{i=1}^m PR_i + \sum_{j=1}^p Path_j$. The system as a whole is not overloaded so long as $L \leq C$ holds for all VRs. Since items flow along paths, if one VR is overloaded then it is likely that other VRs on the same path are overloaded as well. Fortunately, the path abstraction gives a common basis for each VR to make a *local* determination on the load for taking load shedding decisions. To ensure that there is some performance cushion for dealing with overloads and for avoiding sudden drop in the quality of service, we preset an upper limit for the instantaneous load on a VR as L_t ($L_t < C$). Each VR monitors its instantaneous load (L) and when it exceeds the preset threshold L_t , it enters a load shedding phase.

B. Load Shedding Strategies

Two load shedding approaches - *space based load shedding* and *time based load shedding* mechanisms are discussed here.

1) *Space Based Approach*: In this approach, each VR decides to read a *subset of the tags* flowing through a given path in a given *time quantum* T_{window} . Each VR has a finite *per path queue* of size Q for processing incoming tags. Each VR knows *a priori* the number of VRs (VR_{num}) on a particular path. Let MAX_{tags} be the expected maximum number of tags per time quantum flowing through the path. This quantity can be readily computed knowing the item arrival rate into the system. A particular VR_i reads a number of items D_{read} in a time window which is defined by the queue length as: $D_{read} = Q = MAX_{tag} / VR_{num}$. In an ideal situation (i.e., assuming PRs with 100% accuracy), it is sufficient if each VR reads MAX_{tags} / VR_{num} disjoint set of tags to ensure 100% accuracy of the deployment. However, there is the complication that the PRs do not have 100% accuracy. The term Δ is used in the evaluation of space based load shedding where Δ is a sensitivity parameter for keeping the

error rate within acceptable bounds. The performance impact of Δ is studied in the evaluation section. The evaluation study considering the sensitivity parameter is given by: $D_{read} = Q = MAX_{tag} / VR_{num} + \Delta$. We propose two variations of space based load shedding which differ in the method of choosing the amount of data to read and process (D_{read}) based on its queue length Q .

- The *random space based load shedding strategy* provides read-autonomy to individual VRs, each VR reads a random subset of items bounded by queue size Q .
- The *cooperative space based load shedding strategy* allows individual VRs in a path to cooperatively select a disjoint portion of the RFID data stream bounded by the queue size Q . The disjoint set is determined by choosing a data set in the same sequence as the sequence of VRs along the path in a particular window of operation. Ideally, each VR would read separate sets of tags. However, RFID readers use their internal protocol to reduce collision [20], [21] and read tags in the reading range in a random order.

2) *Time Based Approach*: This approach can be considered as the time analog to the space-based approach. First, we introduce some terminologies:

- *Cycle Time* (T_{cycle}) is the time a particular item takes to traverse from source to destination along a particular physical route.
- *Read Time* (T_{read}) is the time a VR accepts data read by PRs, where, $T_{read} \ll T_{cycle}$.
- *Interval Time* ($T_{interval}$) is the time interval between the read times of two consecutive VRs, where, $T_{interval} \ll T_{cycle}$.

If we consider the conveyor belt speed as S m/sec, conveyor width of W and number of items in a palette as N items, then we can define the data rate (r) as: $r = (S \times N) / W$ (items/sec). Each VR reads (D_{read}) items within a predefined read time: $D_{read} = r \times T_{read}$. We use $\tau \leq T_{interval}$ that serves as the sensitivity parameter to increase the reliability of the deployment. Thus the read time is defined as: $D_{read} = r(T_{read} + \tau)$. Note that neither of the load shedding strategies requires any additional intelligence in the PRs. The choice of a specific strategy depends on the application requirement. In the next section, we evaluate these strategies.

VI. EVALUATION

This section discusses the experimental setup and the performance results using the proposed load shedding strategies.

A. Experimental Setup

RF²ID is currently implemented in Java using sockets for inter-VR communication. The implementation supports both physical RFID readers (Alien 9800 model) as well as emulated physical readers that are indistinguishable so far as the VRs are concerned. The emulated physical readers are a faithful reproduction of the Alien 9800 readers, and allow controlled experiments to be carried out varying different sensitivity

TABLE I
COMPARISON OF A REAL SUPPLY CHAIN DEPLOYMENT WITH
EXPERIMENTAL TESTBED

Parameters	Real Supply Chain Deployment	Experimental Testbed
Area	22 meter x12 meter	2 meter x 2 meter
Read Points	9	8
PR/Read Point	order of 10	1
Number of tags	order of thousands	500
Conveyor Speed	≥ 18 meter/min	15meter/min

parameters of the physical readers. Each VR executes on a 3 GHz dual core Intel Xeon processor with 1 GB RAM, 512 KB L2 cache and 1 MB L3 cache. The processors are interconnected by Gigabit Ethernet and form part of a 56-node cluster. We have extended the implementation of the base RF²ID system to include the load shedding strategies discussed in the previous section. At system initialization time any particular load shedding strategy may be chosen providing the system with the parameters specific to that strategy (for e.g., queue size for space-based approach; read and interval times for time-based approach). The purpose of the experimental study is to understand the effectiveness of the load shedding strategies under heavy load to increase the reliability of an RFID deployment without data loss. In our lab, we have constructed a mock up of a supply chain using toy trains (the trains move carrying the tags on stationary tracks instead of stationary tagged items moving on a conveyor belt) illustrated in Figure 4-(c). The various parameters considered for the experimental testbed is presented as a comparison toward an actual testbed in Table I.

1) *System Parameters*: In this study, we consider a distribution center of moderate size [19], [22]. The distribution center uses 9 disjoint read points which we have considered to be individual VR operating points. Each VR is attached to 5 PRs. A moderate sized center expects 1 million data items every day, which amounts to 7000 items per minute. A uniform distribution of this much amount of data may not cause the system to get overloaded. However, the arrival of RFID tagged data is bursty in nature that could cause the system to overload [16]. The various system parameters used for the experiment are discussed here. The *system performance* (A_{sys}) is the measured quantity for evaluation which is a proportion of detected items compared to actual items. *Incoming Data Rate* is the data rate defined from the conveyor speed, palette width and items per palette (defined as *bulk size*) - we have fixed the belt speed and palette width and vary the bulk size to vary the data rate. A time window (t_{window}) for n tags is considered within which a VR performs the following tasks: processing of the tags ($n \times t_{tag}$), processing of incoming VR messages t_{in} , processing of outgoing VR messages (t_{out}) and processing of queries (t_{query}). The VR needs to make sure $t_{process} < t_{window}$ holds before it can send its processed data to the next VR defined as $t_{process} = (n(t_{tag}) + t_{in} + t_{out} + t_{query})$. We have considered a *time window* (t_{window}) of 1.5 minutes based on real time application requirements. We have considered a

threshold system accuracy (A_{th}) of 80% that indicates the system's tolerance to error.

B. PR Level Evaluation

1) *PR Performance*: The PR performance depends on various environmental factors such as tag to reader distance, orientation, and angle has been reported elsewhere [18]. In this study, we look at two other limiting factors, namely, belt speed and bulk size. We use item speed and belt speed synonymously in this paper, since in our laboratory testbed the items move on the train instead of a moving belt in a real warehouse. Figure 4 shows the sensitivity of processed items to the item speed and bulk size. We use Alien ALR 9800 RFID reader and the experiment uses 7 tagged items with our experimental set up for the speed evaluation (y-axis of left hand side). The number of detected items decreases as the item speed is increased. For example, the number of detected item decreases from 62% to 42% when the item speed increases from 9 cm/sec to 36 cm/sec. Figure 4-(a) also shows the sensitivity of the processed items to bulk size (y-axis of right hand side). We fix the item speed at 18 cm/sec. The number of detected items deteriorates as the bulk size increases. As can be seen, 98% of the items are detected when the bulk size is 5, which drops to 16% when the bulk size reaches 98. These results clearly show that as the data rate is increased (either due to belt speed or bulk size or both), a single PR is incapable of meeting the reliability requirements of an RFID deployment.

2) *Validation of Emulated PRs*: The scalability study to be discussed in the next subsection uses emulated PRs that have been designed to faithfully reproduce the behavior of the real PRs. We have designed experiments to validate the emulated PRs using the RF²ID middleware and the laboratory testbed. In this experiment, we fix the tag arrival speed at 15 meter per minute and use 7 RFID tags. Each VR is associated with a single PR, thus making it a one-to-one comparison of the performance of real PRs and emulated PRs. We vary the number of VRs from 1 to 4 and observe the accuracy (i.e., the number tags read compared to actual number of tags) as shown in Figure 4-(b). As can be seen, the observed results validate the accuracy of the emulated PRs in faithfully reproducing the results of the real PRs. For one PR, the readings are almost identical. However, for higher number of PRs, there is a small difference between the real and emulated results. This is due to the variance in the accuracy levels of the individual real PRs, which is not present for the emulated ones. Still the discrepancy between real and emulated PRs is less than 5%.

C. Performance of RF²ID

We study the performance of our proposed system under heavy load with and without exploiting the load shedding mechanisms.

1) *Comparison of Baseline with the Two Load Shedding Mechanisms*: The study uses 9 VRs, each associated with 5 PRs. The baseline performance deteriorates when the data rate increases beyond 25000. As we discussed earlier, within the time window of 1.5 min, each VR has to do all the

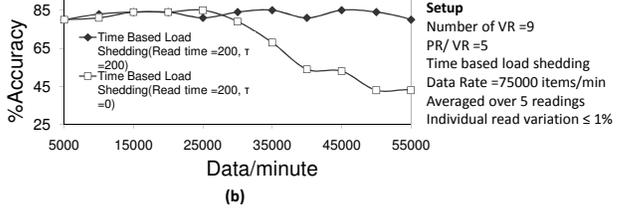
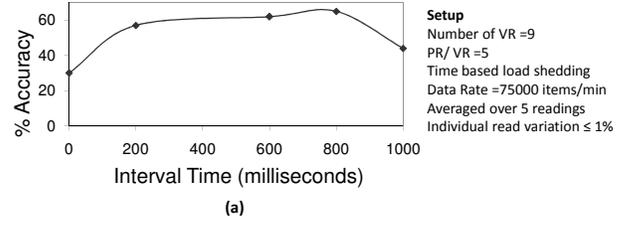
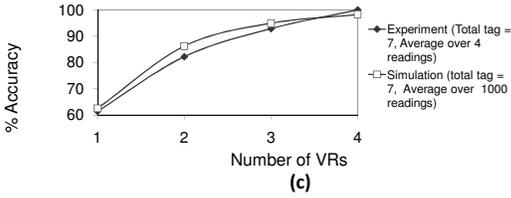
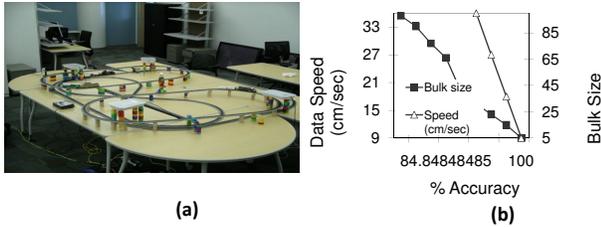


Fig. 4. (a)PR Performance with variable Speed and Bulk Size (b)Comparison of emulated and real PR (c)Laboratory setup for experiment

Fig. 6. Time Based Load Shedding with (a) various interval time (b) various read time

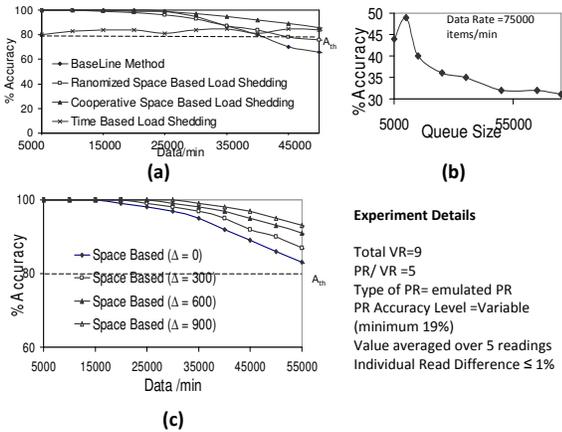


Fig. 5. (a)Baseline System Performance Compared to Space Based and Time Based Load Shedding (b)Space Based Load Shedding -Varying value of Δ (c)Space Based Load Shedding using various Queue Size

chores associated with processing the items and communicate the result to the VR downstream from it. As the data rate increases, the baseline has a hard time keeping up with the demands of processing all the items (see Figure 5-(a)). On the other hand, it can be seen that the space-based approaches is able to sustain good reliability with higher data rates. The cooperative space based approach performs the best among all the evaluation strategies. The randomized space based load shedding performs better than the baseline method for higher data rates. The time-based approach as seen in Figure 5-(a), although has a lower reliability compared to baseline for low data rates, does better at high data rates.

2) *Space Based Load Shedding*: Figure 5-(c) shows the sensitivity of the space-based approach to the Δ parameter (the

effective queue length is equal to $5000 + \Delta$). With increasing Δ values, the system reliability increases. However, arbitrarily increasing the queue length would eventually degrade the performance. This is illustrated in Figure 5-(b). For this experiment, the data rate is fixed at 75000 items/min, and Δ is 0 using cooperative space based load shedding. The number of processed tags deteriorates for larger queue lengths as there is more data to process at each reader. This coupled with the incoming data from neighbor VRs results in degradation of performance. The accuracy level increases with increasing queue length up to 15000 and then deteriorates for larger queue size.

3) *Time Based Load Shedding*: The impact of variable interval time is shown in Figure 6-(a). The overlapping reading (Interval time > 0) shows improved performance over non overlapping reads (Interval time = 0) in the study. However, the number of processed items decreases as the interval time increases over 800ms. For example, the percentage of processed items are 30%, 62% and 44% for interval time of 0, 600 and 1000 milliseconds, respectively. Basically, increasing the overlapping times effectively increases the load on the associated VRs, and this detrimental effect eventually dominates leading to deterioration of performance. The sensitivity of the time-based strategy to its parameters is shown in Figure 6-(b). This study compares the effect of overlapping read times among the PRs. In the non overlapping case, the interval time ($\tau = 0$) is considered to be 0. For the overlapping case, each reader reads for 200 ms more ($\tau = 200$) time along with its distinct read time. It is shown that the larger read time increases the system performance in terms of processed items. The load shedding based approach is an effective way to handle temporary overload in the VRs where the technique itself is not computation intensive. The choice of a specific load shedding strategy depends on the application scenario and the available resources. Time based load shedding mechanism

requires synchronization among PRs as well as VRs in real time which is desirable in a smaller setup. It allows set of PRs to be in dormant mode while other PRs are in operation. This feature can be very useful considering resources that have power constraints. On the other hand, the space based method operates independently among VRs and is able show improved performance over the time based approach. This approach is preferable in large distributed system setup.

VII. RELATED WORK

It is interesting to note that related work falls across fairly diverse domains. We briefly review each of these related areas in this section.

A. Load balancing and Streaming Application

There is a significant amount of work on load balancing in distributed systems where the aim is to distribute the load evenly using techniques such as load migration [13], [23], [24]. However, the streaming nature and the unpredictable burstiness of the data being handled in RFID based systems calls for a different treatment of the problem. Recently, there has been research work focusing on load management aspects of RFID based systems. A heuristic based solution for near optimal reader to tag assignment to deal with the load in the system is presented by Dong et al. [25]. However, the solution only addresses statically placed readers and tags. Producing a system that accounts for tags in motion adds another layer of complexity. The mobile agent based load management system [14] uses a mobile agent to re-adjust imbalance of load in the system using load migration. The connection pool based load balancing [15] also readjusts load among different nodes from heavily loaded one with a lighter load. Load migration is not very suitable for streaming data that needs to meet real time constraints [16], [17]. There is a need for load management that deals with a large amount of error prone data while simultaneously meeting the challenge of real time query processing requests, which is the focus of the proposed research. Traditional resource migration techniques may not be applicable for RFID based systems since such techniques require additional processing to make such decisions when the system is already under stress to meet the real time query processing requests and data overload. *Load shedding* based load management has been incorporated as an effective stream handling mechanism systems such as Aurora [26]–[28], that are designed to meet real time application constraints. It uses the notion of load shedding that allows dynamic insertion and deletion of drop operations of tuples into the query which can be done in a random fashion or based on priority to allow graceful degradation in the presence of load. Our ideas are also based on load shedding but we use the built-in redundancy of the RF²ID middleware to cope with the load with minimal data loss. TelegraphCQ [29] is aimed to support streaming data but differs from our approach in its system structure which uses stream partitioning for load balancing as opposed to load shedding techniques. The pipelined framework [10] processes correlated data locally which has similarity to the proposed VR

abstraction; however, the power of our proposed architecture is in having the VRs operate in concert via the Vpath abstraction.

B. RFID Middleware and Path Abstraction

A number of middleware architectures have been proposed for managing the collection, and processing of the queries related to RFID generated data. These include the Savant architecture [30], the High Fan-in system [31], and WinRFID [32]. Savant is a component-based architecture and there have been specific instances of middleware built using that architecture (e.g., RFIDStack [33]). RFID middleware for object tracking include the work by Kumar et al. [12] and De et al. [34]. MAX [35] is a middleware for object location using RFID. The focus of these projects is different and complementary to the goals of RF²ID and the proposed research. Specifically, we are seeking to use the inherent dataflow in RFID-tagged asset movement in cyber physical systems to aid high level inferencing and close the sense-process-actuate control loop in such applications. Jiang [36] defines a Virtual Route Tracking algorithm to form a virtual route among RFID readers that are in close proximity to one another. The idea of virtual route is similar in spirit to Vpath of RF²ID, with the exception that Vpath spans geographically distributed elements.

The concept of path is used in many different contexts including fault tolerance [37], compiler optimization [38], profiling distributed systems [39], [40], and resource allocation [41], [42]. Our proposed architecture is inspired by the use of paths in these various contexts.

VIII. CONCLUSION

RFID based deployments for large-scale supply chain environments are becoming widespread. These deployments face the fundamental challenge in RFID based systems that the devices are error prone. The demands placed on the deployments from the applications in the form of increased data rates and requests for real time query processing are compounding these challenges even more. We have designed cooperative load shedding mechanisms for dealing with the increased data rates and query processing requests in large-scale RFID deployments. The mechanisms are piggy-backed on a middleware called RF²ID that deals with the error-prone nature of the physical RFID readers. The middleware uses the spatial notion of path along which the items flow in a supply chain to cumulatively aggregate the tags collected by the virtual readers to increase the overall reliability. The load shedding algorithms leverage the built-in redundancy in this middleware to cooperatively shed the load among the virtual readers with minimal data loss thus increasing the overall reliability of the deployment. Our implementation and performance results confirm the utility of these load shedding mechanisms.

REFERENCES

- [1] H. Gonzalez, J. Han, X. Li, and D. Klabjan, "Warehousing and analyzing massive rfid data sets," in *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering*. IEEE Computer Society, 2006, p. 83.

- [2] Y. Hu, S. Sundara, T. Chorma, and J. Srinivasan, "Supporting rfid-based item tracking applications in oracle dbms using a bitmap datatype," in *VLDB '05: Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 1140–1151.
- [3] M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, Eds., (*e*)*Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*. Morgan Kaufmann, 2004.
- [4] R. Kimball, "Rfid tags and smart dust," http://www.intelligententerprise.com/030718/612warehouse1_.1.jhtml.
- [5] B. Violino, "Rfid opportunities and challenges," <http://www.rfidjournal.com/article/articleview/537>.
- [6] I. Corp., "Tomorrow's high performance network," <http://developer.intel.com/technology/comms/perfnet/index.htm>.
- [7] T. Hassan and S. Chatterjee, "A taxonomy for RFID," in *HICSS '06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences*. IEEE Computer Society, 2006, p. 184.2.
- [8] "The basics of RFID technology," <http://www.rfidjournal.com/article/articleview/1337/1/129/>.
- [9] C. Martins, "RFID data management: It's not the mass, it's the momentums," AIM Global, 2005.
- [10] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, "A pipelined framework for online cleaning of sensor data streams," in *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering*. IEEE Computer Society, 2006.
- [11] S. Sarma, "Integrating RFID," *Queue*, vol. 2, no. 7, pp. 50–57, 2004.
- [12] S. Kumar, C. Alaettinoglu, and D. Estrin, "Scalable object-tracking through unattended techniques (SCOUT)," in *Proceedings of the 8th International Conference on Network Protocols (ICNP)*, 2000, pp. 253–262.
- [13] D. L. Eager, E. D. Lazowska, and J. Zahorjan, "Adaptive load sharing in homogeneous distributed systems," *IEEE Trans. Softw. Eng.*, vol. 12, no. 5, pp. 662–675, 1986.
- [14] J. F. Cui and H. S. Chae, "Load balancing for RFID middlewares based on agent technology," in *SE'07: Proceedings of the 25th conference on IASTED International Multi-Conference*. Anaheim, CA, USA: ACTA Press, 2007, pp. 136–141.
- [15] S.-M. Park, J.-H. Song, C.-S. Kim, and J.-J. Kim, "Load balancing method using connection pool in RFID middleware," in *SERA '07: Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management & Applications*. IEEE Computer Society, 2007, pp. 132–137.
- [16] M. Teskey, "Turning RFID Data into Information," <http://www.devx.com/enterprise/Article/32251/0/page/1>.
- [17] R. Kanters, "Data Management Risks of Radio Frequency Identification (RFID)," Master's thesis, Tilburg University, 2007.
- [18] N. Ahmed, R. Kumar, R. S. French, and U. Ramachandran, "RF²ID: A reliable middleware framework for RFID deployment," in *International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE Computer Society, 2007.
- [19] B. C. Hardgrave and R. Miller, "The myths and realities of RFID," SAM. M. Walton College of Business, University of Arkansas, Tech. Rep. ITRI-WP067-0306, 2006.
- [20] K. Romer, "Smart playing cards: A ubiquitous computing game," *Personal and Ubiquitous Computing*, vol. 6, pp. 371–377, 2002.
- [21] N. Vaidya and S. R. Das, "Rfid-based networks: exploiting diversity and redundancy," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 12, no. 1, pp. 2–14, 2008.
- [22] B. C. Hardgrave, R. Miller, and M. Waller, "Does RFID reduce out of stocks? a preliminary analysis," SAM. M. Walton College of Business, University of Arkansas, Tech. Rep. ITRI-WP058-1105, 2005.
- [23] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *J. Parallel Distrib. Comput.*, vol. 7, no. 2, pp. 279–301, 1989.
- [24] A. Hac and T. Johnson, "A study of dynamic load balancing in a distributed system," *SIGCOMM Comput. Commun. Rev.*, vol. 16, no. 3, pp. 348–356, 1986.
- [25] Q. Dong, A. Shukla, V. Shrivastava, D. Agrawal, S. Banerjee, and K. Kar, "Load balancing in large-scale RFID systems," *Comput. Netw.*, vol. 52, no. 9, pp. 1782–1796, 2008.
- [26] M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, Y. Xing, and S. Zdonik, "Scalable Distributed Stream Processing," in *CIDR 2003 - First Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, January 2003.
- [27] B. Babcock, M. Datar, and R. Motwani, "Load shedding techniques for data stream systems," in *Proc. of the 2003 Workshop on Management and Processing of Data Streams (MPDS)*, 2003.
- [28] N. Tatbul, U. Çetintemel, S. Zdonik, M. Cherniack, and M. Stonebraker, "Load shedding in a data stream manager," in *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*. VLDB Endowment, 2003, pp. 309–320.
- [29] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, and M. A. Shah, "TelegraphCQ: Continuous dataflow processing for an uncertain world," in *CIDR*, 2003.
- [30] "Savant technical manual," OAT SYSTEMS & MITAUTO-ID CENTER, Tech. Rep. Version 0.1 (Alpha), 2002.
- [31] M. J. Franklin, S. R. Jeffery, S. Krishnamurthy, and F. Reiss, "Design considerations for high fan-in systems: The HiFi approach," in *CIDR*, 2005, pp. 290–304.
- [32] B. S. Prabhu, X. Su, C. Qiu, H. Ramamurthy, P. Chu, and R. Gadh, "WinRFID middleware for distributed RFID infrastructure," in *International Workshop on Radio Frequency Identification (RFID) and Wireless Sensors*, Indian Institute of Technology, Kanpur, India, 2005.
- [33] C. Floerkemeier and M. Lampe, "RFID middleware design: addressing application requirements and RFID constraints," in *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*. ACM, 2005, pp. 219–224.
- [34] P. De, K. Basu, and S. K. Das, "An ubiquitous architectural framework and protocol for object tracking using RFID tags," in *International Conference on Mobile and Ubiquitous Systems*. IEEE Computer Society, 2004, pp. 174–182.
- [35] K.-K. Yap, V. Srinivasan, and M. Motani, "Max: human-centric search of the physical world," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005, pp. 166–179.
- [36] W. Jiang, D. Yu, and Y. Ma, "A tracking algorithm in RFID reader network," in *FCST '06: Proceedings of the Japan-China Joint Workshop on Frontier of Computer Science and Technology*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 164–171.
- [37] M. Y. Chen, A. Accardi, E. Kiciman, J. Lloyd, D. Patterson, A. Fox, and E. Brewer, "Path-based failure and evolution management," in *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*. USENIX Association, 2004, pp. 23–23.
- [38] G. Ammons and J. R. Larus, "Improving data-flow analysis with path profiles," *ACM SIGPLAN Notices*, vol. 39, no. 4, pp. 568–582, 2004, best of PLDI 1979-1999.
- [39] P. Barham, A. Donnelly, R. Isaacs, and R. Mortier, "Using magpie for request extraction and workload modelling," in *OSDI'04: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation*. USENIX Association, 2004, pp. 18–18.
- [40] T. Gschwind, K. Eshghi, P. K. Garg, and K. Wurster, "Webmon: A performance profiler for web transactions," in *WECWIS '02: Proceedings of the Fourth IEEE International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS'02)*. IEEE Computer Society, 2002, p. 171.
- [41] M. Welsh, D. E. Culler, and E. A. Brewer, "SEDA: An Architecture for Well-Conditioned, Scalable Internet Services," in *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, Oct. 2001.
- [42] J. Reumann, A. Mehra, K. G. Shin, and D. Kandlur, "Virtual services: a new abstraction for server consolidation," in *ATEC '00: Proceedings of the USENIX Annual Technical Conference*. USENIX Association, 2000.