Efficient Data Collection for Large-Scale Mobile Monitoring Applications

Haiying Shen*, *Senior Member, IEEE*, Ze Li, Lei Yu, *Member, IEEE,* and Chenxi Qiu .

Abstract—Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSNs) have been popular in the industrial field and both have undergone dramatic development. RFID and WSNs are well-known for their abilities in identity identification and data transmission, respectively, and hence widely used in applications for environmental and health monitoring. Though the integration of a sensor and an RFID tag was proposed to gather both RFID tag and sensed information, few previous research efforts explore the integration of data transmission modes in the RFID and WSN systems to enhance the performance of the applications. In this paper, we propose a Hybrid RFID and WSN system (HRW) that synergistically integrates the traditional RFID system and WSN system for efficient data collection. HRW has hybrid smart nodes that combine the function of RFID tags, the reduced function of RFID readers and wireless sensors. Therefore, nodes can read each other's sensed data in tags, and all data can be quickly transmitted to an RFID reader through the node that firstly reaches it. The RFID readers transmit the collected data to the back-end servers for data processing and management. We also propose methods to improve data transmission efficiency and to protect data privacy and avoid malicious data selective forwarding in data transmission. Comprehensive simulation and trace-driven experimental results show the high performance of HRW in terms of the cost of deployment, transmission delay and capability, and tag capacity requirement.

Index Terms—RFID, Wireless Sensor Networks (WSNs), Distributed hash tables (DHTs), Data routing.

1 Introduction

Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSNs) are two of the most important systems widely used in many monitoring applications such as environmental and health monitoring and enterprise supply chains. WSNs are mainly used for monitoring physical or environmental condition, collecting environmental data such as temperature, sound. RFID is a technology that uses radio waves to transfer data between RFID tags and RFID readers (readers in short). RFID can be implemented on the objects to be identified, improving the efficiency of individual object tracking and management. More than 104 Wal-Mart stores have installed RFID systems to monitor the stock levels and track merchandizes in the supply chain [1] so that the products will not be out of stock or lost.

RFID tag data usually is collected using *direct transmission* mode, in which an RFID reader communicates with a tag only when the tag moves into its transmission range. If many tags move to a reader at the same time, they will contend to access the channels for information transmission. Normally, the percentage of tags that can successfully transmit their data in one transmission is merely 34.6% to 36.8% [2]. Such a transmission architecture for RFID data collection is not sufficient to meet the requirements of low economic cost, high performance and real-time individual monitoring in large-scale mobile monitoring applications.

- *Economic cost*. An RFID reader cannot quickly receive information from tags due to its immobility and short transmission range. Thus, enormous RFID readers are required to increase their coverage for fast data collection. This would cause significant cost of the system deployment considering the high price of a high-quality RFID reader (at least $500) and the high cost of establishing connections between back-end servers and RFID readers. Therefore, it is important to constrain the number of RFID readers while still achieves efficient data collection.
- *High performance*. In traditional RFID monitoring applications, such as supply chain management and baggage checking in Delta Airlines, an RFID reader is required to quickly process several tags at different distances. An RFID reader can only read tags in its range. Limited communication bandwidth, background noise, multi-path fading and channel accessing contention between tags would severely deteriorate the performance of the data collection. These problems can be avoided by transmitting data in short distances via the multi-hop data transmission mode in WSNs.
- *Real-time individual monitoring*. In applications that require real-time monitoring on individual objects (e.g., endangered animals and patients), real-time information (e.g., body temperature, blood pressure, location) retrieval of individual objects is the most important. Though the integration of a sensor and an RFID tag helps gather both RFID tag and sensed information [3]–[6] from objects, quickly collecting the information still remains as a challenge.

In this paper, we propose a Hybrid RFID and WSN system (HRW) that synergistically integrates the RFID and WSN data transmission modes for efficient data collection in large-scale monitoring applications for moving objects (e.g., environmental and health monitoring). HRW novelly leverages the integration to reduce the number of required RFID readers hence economic cost and enhance data transmission efficiency. HRW has a new type of nodes called Hybrid Smart Nodes (smart nodes / nodes in short) that combine the function of RFID tags, and the reduced function of wireless sensors and RFID readers. The system mainly consists of three components: smart nodes, RFID readers and the back-end server infrastructure. The RFID readers collect data from

- * Corresponding Author. Email: shenh@clemson.edu; Phone: (864) 656 5931; Fax: (864) 656 5910.
- Haiying Shen, Ze Li, Lei Yu and Chenxi Qiu are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634. E-mail: {shenh, zel, leiy, chenxiq}@clemson.edu

Digital Object Indentifier 10.1109/TPDS.2013.122

smart nodes and transmit the data to the infrastructure. We summarize the contribution of this paper in below.

- *Proactive data transmission.* Inspired by the multi-hop transmission mode in WSNs, rather than passively waiting for RFID readers to read data, smart nodes actively transmit data to readers in a multi-hop manner. Smart nodes read tag data between each other. In this way, instead of reading every tag one by one when they move into the reading range, RFID reader can receive the information of a group of tags by reading only one first-encountered node. As a result, the channel contention and noise interference during the data transmission can be significantly reduced. In the traditional WSN, a node in the sleeping mode cannot receive and forward data. In HRW, a node can read data from the RFID tag of another node even if it is in sleep mode, which greatly increases transmission efficiency.
- *Algorithms to enhance efficiency.* We further improve the information collection efficiency by letting cluster nodes replicate their data to each other or to one specified cluster head that has high encountering frequency with cluster nodes and RFID readers. We also propose a tag clean-up algorithm to remove delivered data from tags to reduce transmission overhead.
- *Security strategies.* We propose solutions to handle two security threats in the data communication to readers to reduce privacy and security risks.

Our comprehensive simulation and trace-driven experimental results show that HRW can reduce the number of RFID readers, the transmission delay of each node, and the demand on the capacity of tags, compared to the traditional RFID monitoring system. The results also show the effectiveness of our proposed algorithms to enhance the efficiency and security. Compared to our previous conference version of this work [7], this paper enhances the performance of HRW with cluster-based data transmission and security mechanisms. It also presents theoretical analysis and additional experimental results.

The rest of this paper is organized as follows. Section 2 describes the HRW system for monitoring applications. Section 3 presents our security mechanisms. Section 4 presents the simulation results. Finally, Section 5 concludes this paper with remarks on future works. In the supplement material, we present a theoretical analysis of our proposed methods compared to the traditional RFID system, a bloom filter based data indexing to efficiently check redundant data in the local tag that does not need to be replicated or needs to be removed, additional experimental results and an overview of related works.

2 Hybrid RFID and WSN System (HRW)

2.1 Hybrid Smart Nodes

HRW has smart nodes that synergistically integrate RFID and WSN functions. A smart node has the following typical components.

- *Reduced-function sensor.* Unlike the normal sensors, this sensor does not have transmission function. It collects the environmental data and the sensed data (e.g., pressure, temperature) from hosts.
- *RFID tag.* As the normal RFID tags, it serves as traditional packet memory buffer for information storage. The RFID information such as identity and properties is configured into the RFID tag during the production stage.

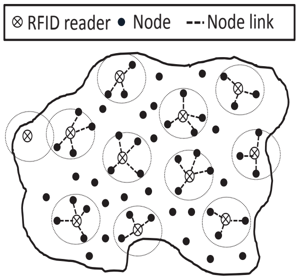


Fig. 1: Traditional RFID architecture.

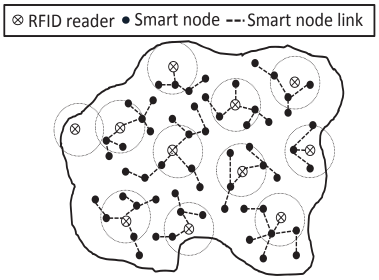


Fig. 2: The HRW architecture.

- *Reduced-function RFID reader (RFRR).* It is used for the data transmission between smart nodes. A smart node uses RFRR to read other smart nodes' tags and write the information into its own tag.

RFRR can just be a simple ultra-high frequency reader module from traditional RFID readers. An RFID reader module can be as low as $29, which costs much less than a high-quality RFID reader (at least $500). Using RFRR, nodes can exchange their tag data in a proactive manner. RFRR also helps to store the data sensed from monitored hosts and environment into the tags. The smart nodes are feasible to build as they consist of simpler and partial parts from nodes integrating RFID tag and sensor functions [3]–[6], [8] and RFRR. Compared to RFID tags, HRW achieves higher performance at the cost of additional components of reduced-function sensor and RFRR for each node. However, this additional cost is much less than the cost of its reduced many high-cost RFID readers. The nodes with integrated RFID tag and sensor functions can also use HRW for efficient data collection with RFRR modules.

Each smart node has two modes: sleep mode and active mode. In the active mode, the sensor unit in the smart node collects the physical information of the smart node host and asks RFRR to write the data into the node's tag. While in the sleep mode, smart nodes do nothing. The tag information in a node can be read by other active nodes; no matter it is in sleep mode or not [2]. Since there are many smart nodes in the system, and the transmission of the collected information to RFID readers is delay tolerant, it is not necessary to let all of the smart nodes remain active all the time, which otherwise consumes considerable battery power.

2.2 Proactive Data Transmission

Figure 1 shows the traditional RFID architecture, and Figure 2 shows the architecture of the HRW system. Both architectures are hierarchical. The upper layer is composed of RFID readers connected to the back-end infrastructure with high-speed backbone cables. The back-end infrastructure connects to the applications (e.g., database in a hospital). The lower layer is formed by a considerable number of object hosts that transmit data to RFID readers. The difference between these two architectures is the transmission mode. In Figure 1, only the nodes (hosts) in the transmission range of RFID readers can send their tag information to the RFID readers. As explained in Section 1, this direct transmission mode

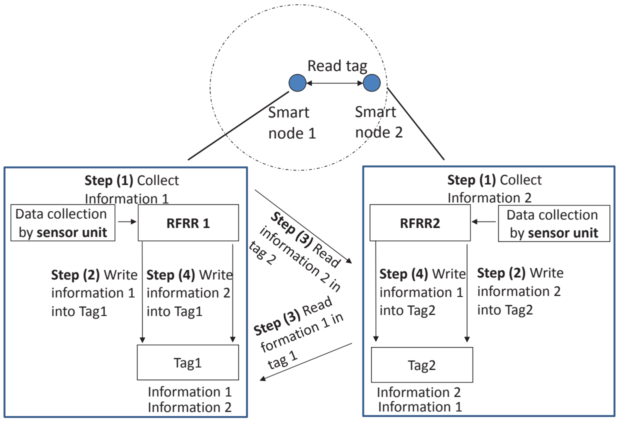

Read tag
Smart node 1
Smart node 2
Step (1) Collect Information 1
Data collection by sensor unit
RFRR 1
Step (2) Write information 1 into Tag1
Step (4) Write information 2 into Tag1
Tag1
Information 1
Information 2
Step (3) Read information 2 in tag 2
Step (3) Read formation 1 in tag 1
Step (1) Collect Information 2
RFRR2
Data collection by sensor unit
Step (4) Write information 1 into Tag2
Step (2) Write information 2 into Tag2
Tag2
Information 2
Information 1

Fig. 3: The replication process of two smart nodes

would lead to channel contention and hence low successful transmission rate and slow data collection. In Figure 2, the nodes are smart nodes that can exchange and replicate tag information with each other using wireless RF channels. Each RFID reader reads tags within its transmission range. Since the data can be transmitted to the RFID reader using a multi-hop transmission mode, each RFID reader can also receive the information in tags outside of its transmission range. In this way, HRW can quickly collect data and expedite the data collection.

After smart node A collects the sensed data, it appends the sensed data with a timestamp and stores the data in its tag through RFRR. Figure 3 shows an example of data collection process of two smart nodes. After the sensor unit in a smart node collects the information about its tag host (Step 1), it asks RFRR to store the information into its tag (Step 2). Once two nodes move into the transmission range of each other, the RFRR in a node reads the information stored in another node's tag (Step 3). Based on the host ID and timestamp, the node checks if it has stored the information previously. If not, the RFRR then stores the acquired information into the local tag (Step 4).

When node i replicates node j's data, node i also records the timestamp of the replication time denoted by t_{ij}. Next time when node i meets node j, node i will not replicate node j's data with timestamps prior to t_{ij}. Suppose the timestamp of smart node 3 for node 4 is 11230337, which represents the time $03 : 37$ am, Nov. 23th. When node 3 meets node 4 next time, node 3 ignores the information with timestamp less than 11230337 in the information replication. In this way, smart nodes avoid recording duplicated information, and hence avoid the unnecessary overhead in the transmission. Algorithm 1 shows the pseudo-code of the information collection process.

By the data exchange, the data of a node can be replicated to a number of other nodes in the system and any one of these nodes that meets the RFID reader can transmit the information to the reader. In this way, the likelihood that the information is delivered to the RFID reader is greatly increased and the number of RFID readers needed for fast information delivery is reduced.

When a node enters the reading range of an RFID reader, the RFID reader reads the information in the node's tag. If several nodes enter the range of RFID reader at the same time, the RFID reader gives the first meeting tag the highest priority to access the channel, reducing channel contention and long distance transmission interference. The RFID reader can erase the information in the tag once after obtaining it. Algorithm 2 shows the pseudocode of the reading process of an RFID reader.

Algorithm 1 Pseudo-code of the process of information replication executed by smart node i.

```
if this.state=active then
    Collect the sensed data of its host $D_i$
    //Store$D_i$ into its $tag_i$
    Store($D_i$, $tag_i$)
    for every node $j$ in its transmission range do
        if this.linkAvailable(j) then
            Read data $D_j$ with timestamp $> t_{ij}$ from $tag_j$
            //Store data $D_j$ in its $tag_i$
            Store($D_j$, $tag_i$)
            Update timestamp $t_{ij}$ with current time
        end if
    end for
end if
```

Algorithm 2 Pseudo-code of the process of information reading executed by RFID reader i.

```
for every node $j$ in its transmission range do
    if this.linkAvailable(j) then
        Read data $D_j$ from $tag_j$ in node $j$
        //Store data $D_j$ in storage $S_i$ in the RFID reader
        Store($D_j$, $S_i$)
        Erase $D_j$ from $tag_j$ in node $j$
    end if
end for
```

With this data transmission algorithm, after an RFID reader receives the information of a node, many nodes still hold the replicas of the information. Exchanging such delivered and redundant information incurs high transmission overhead but does not contribute to information collection. In order to reduce the unnecessary message transmission, we use a tag clean-up algorithm to delete the delivered messages in the system. Specifically, after an RFID reader reads the information from a node, the reader sends the node a directory containing the tag IDs and timestamps of recently received data items. This directory has a TTL (Time to live) with it. It is then broadcasted among nodes and will be deleted when TTL expires. After receiving the directory, the nodes delete the delivered information in their own tags. Considering that the timestamp and ID of each information item have much less size than a complete data item, the overhead of the directory broadcasting is small.

2.3 Cluster-based Data Transmission

Replicating data between any two encountered smart nodes generates a high cost. Concurrent data transmission from many nodes to an RFID reader causes channel access congestion. Also, it is not easy to erase duplicate data that is already reported to the RFID readers from replica nodes. We propose enhanced data transmission algorithms to mitigate these problems.

A simple algorithm to reduce the cost is to enable a source node to replicate its data to a limited number of nodes. Here, we describe two enhanced algorithms called cluster-member based and cluster-head algorithms, in which smart nodes are clustered to different virtual clusters and each cluster has a cluster head. In the cluster-member based algorithm, cluster members replicate their tag data between each other. When a cluster member of a virtual cluster enters the reading range of an RFID reader, by reading the aggregated tag information from the cluster member, the RFID reader receives all information of nodes in this virtual cluster. In the cluster-head based algorithm, cluster members replicate their tag data to the cluster head. When a cluster head of a virtual cluster reaches an RFID reader, the RFID reader receives all information of nodes in this virtual cluster. This enhanced method greatly reduces channel access congestion, reduces the information exchanges between nodes and makes it easy to erase duplicate information in a cluster. The method is suitable to the applications where monitored objects (e.g., zebras, birds, people) tend to move in clusters.

Algorithm 3 Pseudo-code of cluster head determination and data transmission conducted by smart node i.

```
Receive cluster head candidates from an RFID reader
for each cluster head candidate $j$ do
    Calculate $(f_{n_{ij}} * f_{r_j})$
end for
Choose the cluster head with $\max(f_{n_{ij}} * f_{r_j})$
if it is a cluster head and meet its cluster member then
    Read data from the cluster member
end if
if it is a cluster head and meet an RFID reader then
    Send its data to the RFID reader
end if
```

To form the clusters in the cluster-member based algorithm, nodes report their encountering frequency to the server through the RFID readers. The server forms nodes with high encountering frequency into a cluster using the method in [9] and notifies the cluster nodes through the RFID readers. The cluster head for a cluster can be selected in a number of ways depending on the application requirement. For example, in a health monitoring application where real-time data collection is required, the nodes with the most contact frequency with cluster members and RFID readers should be the cluster heads. In the supply chain where nodes are always close to each other, the nodes with the highest energy should be the cluster heads. We use the former example to show how to choose cluster heads. Algorithm 3 shows the pseudocode of cluster head determination and data transmission conducted by each smart node in the second algorithm. RFID readers record the meeting frequency with each node and report the data to the back-end server. The server calculates the sum of the frequencies from different readers for each node j, denoted by f_{r_j}, and selects N nodes with the highest f_{r_j} as the cluster heads. The information of the selected cluster heads along with their f_r is transmitted back to the RFID readers, which will forward the information to the nodes. We use $f_{n_{ij}}$ to denote the meeting frequency between node i and a cluster head j. A node measures its $f_{n_{ij}} * f_{r_j}$ for each cluster head candidate, and selects the one with the highest metric as its cluster head. The metric of $f_{n_{ij}} * f_{r_j}$ indicates how fast cluster head j can forward node i's data to an RFID reader. Through RFID readers, each node reports its selected cluster head to the server and the server then notifies all heads about their cluster members. The head determination can also be solely conducted at the server to reduce the communication. As a result, each cluster head is associated with a group of nodes, and it can most quickly forward the data to RFID readers for its cluster members.

In the HRW system, since the data is stored in tags, active nodes can retrieve the information at any time from a sleeping node. In traditional WSNs, however, nodes in sleeping mode cannot conduct data transmission. Therefore, the HRW system can greatly improve packet transmission efficiency with the RFID technology.

3 Communication Security Mechanisms

The multi-hop message transmission mode in HRW improves the communication efficiency. However, such method introduces privacy and security risks. Low-cost RFID nodes are not tamper-resistant and deployed in open environment, thus the attackers can easily physically access and take control of these nodes. The attacker can obtain all the information in the compromised nodes and use the compromised nodes to obtain sensitive information and disrupt system functions. Thus, in this section, we consider two security threats arising from node compromise attacks: data manipulation and data selective forwarding.

3.1 Data Privacy and Data Manipulation

In the system, each smart node replicates its information to other nodes. Once a node is compromised, all the information of other nodes is exposed to the adversaries, which is dangerous especially in privacy sensitive applications such as health monitoring. A malicious node can also manipulate the gathered information and provide false information to the readers. Therefore, it is important to protect the confidentiality and authenticity of tag information in data transmission.

Public key operations are too expensive for the smart nodes due to their limited computing, storage and bandwidth resources. We then develop a symmetric key based security scheme in our system. In this paper, we focus on the threats due to the compromised smart nodes and assume the readers are secure. In our security scheme, each smart node N is initially assigned with an individual key K_N. The pairs (N, K_N) of all smart nodes are stored in a central server, which can be securely accessed by the readers. To achieve data confidentiality, each smart node N generates a temporary key $K'_N = H(Nonce|K_N)$, where $Nonce$ is a nonce number which can be the timestamp of RFID data, $H(*)$ is a system-wide secure hash function known by every node, and "|" represents the concatenation of two strings. Node N uses this symmetric key to encrypt its data D_N and sends the encrypted data, denoted by $En(K'_N, D_N)$, to other nodes. The use of temporary keys for every data transmission further enhance the security against the ciphertext-only attacks by interpreting historical transmissions. To protect data authenticity, node N also computes the message authentication code with the temporary key K'_N, denoted by $MAC(K'_N, N|D_N)$. The message from a smart node is in the format of

$$(N, Nonce, En(K'_N, D_N), MAC(K'_N, N|D_N)).$$

Figure 4 shows the procedure of data reading with encryption and authentication. When a reader receives the data, it first sends to the central server the tag ID N and $Nonce$. The server finds K_N and computes the temporary key K'_N, and then securely sends K'_N to the reader. After receiving K'_N, the reader is able to decrypt the data D_N from $En(K'_N, D_N)$ and then verifies whether MAC is correct. If the recomputed MAC is consistent with the MAC received from the smart node, the reader considers the MAC is correct and the data set is authentic. Otherwise, the $En(K'_N, D_N)$ is changed by an adversary node.

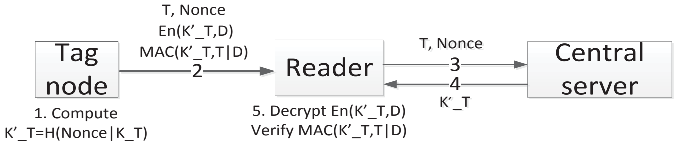


Fig. 4: The procedure for secure data reading and verification.

To avoid being detected for changing data, an adversary may launch old message replay attack by replacing a new message from a node with an old message from the node. When a reader forwards the N and $Nonce$ to the central server, the central server can easily detect outdated nonce values which were reported previously. As a result, the old message replay attack can be detected.

Once a smart node N is compromised, its individual key K_N is exposed and the adversary can derive all previous temporary keys to decrypt data in the old messages. Thus, it is important to achieve the backward security by updating the individual key periodically. However, periodically distributing new keys from a central server to all smart nodes incurs expensive communication cost. Therefore, we propose a key hash chain method to avoid the key distribution cost. Initially, each node is loaded with a key hash chain computed with a secure one-way hash function $H(*)$ as follows

$$K_0 \overset{\text{H}}{\Longrightarrow} K_1 \overset{\text{H}}{\Longrightarrow} K_2 \cdots \overset{\text{H}}{\Longrightarrow} K_L, \tag{1}$$

where $K_i = H(K_{i-1})$. The smart node uses a key as its individual key on the chain in the order from K_0 to K_L. It periodically updates its individual key from K_i to K_{i+1} and erases K_i from its storage. Because H is one-way hash function, even the attacker obtains K_i, it cannot derive any keys K_j with $j < i$, and thus cannot decrypt previous transmissions encrypted by K_j.

In a large-scale system with a large amount of nodes, it could be an expensive and time-consuming operation to find the individual key of a specific smart node among all nodes' keys. The searching time is linear to the total number of nodes. We propose two methods to resolve this problem. First, we propose to compute individual keys in run time rather than storing all keys in advance and searching keys on-demand. To this end, the central server maintains a secret key K_c. For each node with the tag ID N, its individual key K_N is computed by the cryptographical secure hash function H with K_c, i.e., $K_N = H(N|K_c)$. In this way, the server does not need to store any individual keys. When receiving the tag ID N, the server directly recomputes $H(N|K_c)$ and obtains the individual key K_N, which avoids the searching. Since the computation time of the hash function is independent of the number of nodes, the time for finding individual keys can be significantly reduced in large-scale systems compared to linear searching. Second, we propose distributed key storage in the back-end servers. We form the back-end servers into a distributed hash table (DHT). The DHT overlay supplies *Insert(key, data)* and *Lookup(key)* functions. *Insert(key, data)* stores the data into its assigned server. *Lookup(key)* retrieves the data with the indicated key. Regarding key here as the consistent hash [10] value of a node's tag ID and data here as the node's individual key, the back-end servers can efficiently store and retrieve a node's individual key using $O(\log n)$ path length, where n is the number of nodes.

3.2 Data Selective Forwarding

In the cluster-head based transmission algorithm, the cluster head in each cluster is responsible for forwarding the tag data of all cluster members to the reader. A malicious cluster head can drop part of the data and selectively forward the gathered information to the reader. Since an RFID reader may not know all the smart nodes in a head's cluster in advance, it cannot detect such attacks. To prevent the selective forwarding attack, we can exploit the cluster-member based data transmission algorithm, in which all cluster members hold the data of all other nodes in the cluster. A reader can compare cluster members' reported data with the cluster head's reported data to verify the correctness of the latter.

We use D_{all} to denote the set of all encrypted tag data $(N, Nonce, En(K'_N, D_N), MAC(K'_N, N|D_N))$ in a cluster. After node N collects encrypted data from all other nodes in its cluster, it creates its MAC on D_{all_N} and sends its $(N, Nounce, MAC(K'_N, N|D_{all_N}))$ to the reader.

After receiving D_{all_c} from a cluster head and the MACs of D_{all} from cluster members, the reader can verify the authenticity of D_{all_c}. Based on a cluster member's N and K'_N, the reader creates $MAC(K'_N, N|D_{all_c})$ and compares it with the received $MAC(K'_N, N|D_{all_N})$ from node N. If two MAC values are different, it means that the data from the cluster head or from node N is not valid. After conducting many comparisons for many cluster nodes, if the majority comparisons are valid, then the data from the cluster head should be valid, otherwise, it is not valid.

Obviously, it causes excessive communication cost if the reader needs every cluster member to send its MAC for D_{all}. A simple solution is to let the reader only collect MACs from T ($T \geq 1$) number of cluster members. Once T number of MACs are collected, the reader verifies the authenticity of the data set and considers it valid if all the MACs are correct. However, this method cannot prevent the collusion attack of multiple compromised nodes. Suppose that a node sent a pruned data set to the reader, other T compromised nodes can compute valid MACs for the pruned data set and send them to the reader.

To prevent the collusion attack, we propose a secure randomized solution, in which each smart node randomly decides whether to send its MAC to the reader. Suppose F is a cryptographically secure pseudo-random function which uniformly maps the input values into the range of $[0, 1)$. Each node N checks the inequality

$$F(N|K'_N) < \rho \ (0 < \rho < 1), \tag{2}$$

where ρ is a threshold which decides the expected number of MACs the reader will receive. If the inequality holds, the node sends its MAC to the reader. Otherwise, it does not. As a result, each smart node in the cluster has a probability of ρ to send its MAC to the reader. When the reader receives the MAC from a smart node N, it recomputes F and accepts the MAC only when the inequality holds. Once the reader finds that all received MACs are correct, it considers the data set valid and complete. In this way, the collusion attack is prevented through verifying the legitimacy of nodes for providing their MACs (i.e., checking whether Inequality (2) holds), while the communication cost between the nodes and the reader is reduced. The threshold ρ is a system parameter loaded into the tag nodes and servers when the system is initialized. Larger threshold means stronger security strength at the expense of higher communication cost. The threshold ρ is initially decided by the users according to their security strength demand.

4 Performance Evaluation

4.1 Evaluation on Data Transmission

Since the transmission links between hosts are usually intermittently connected, we created a delay tolerant network environment for the performance evaluation.

The simulation was built on a custom discrete event-driven simulator [11].

We use the random way-point model [12] to simulate the situation where there is no movement pattern and use the real mobile traces [13], [14] to simulate the situation where there is movement pattern for the applications where monitored objects (e.g., zebras, birds, people) tend to move in clusters. In the random way-point model, each node waits for a pause time randomly chosen from $(1-5)$s, then moves to another random position with a speed chosen between 1 to 10m/s. In the simulation, 50 nodes and 5 RFID readers were independent identically distributed (i.i.d.) over a $600m \times 700m$ area.

We used two transmission modes in HRW: epidemic [15] and source-replication. In the epidemic transmission, the packets of nodes are replicated to other nodes within TTL hops, which was set to 6 by default. In the source-replication transmission, a source node allows a certain number (10 by default) of nodes to read its packets. We compared these methods with the “direct” transmission method in the traditional RFID systems, in which a node keeps its collected information in its tag until it reaches the range of an RFID reader. If one of the copies of a packet arrives at an RFID reader, we consider this packet successfully delivered. We only considered the first delivered replica of a packet in the measurement.

The entire simulation time was set to 4000s. The warmup time was set to 100s during which nodes randomly move around. Then, during the following 1000s, at every second, we randomly selected a node to generate a packet in its own tag. By default, the active time of each node was randomly chosen from 10-15s and the sleeping time was randomly chosen from 0-10s. Unless otherwise specified, the tag capacity of each node was set to 30 packets, and the reading range of the RFID and smart node readers was set to 30m. There was no tag capacity limit for RFID readers. In the simulation, a node dropped packets if its tag was full and these dropped packets would not be retransmitted. The packets dropped by sleeping nodes would be retransmitted. We run each simulation test for 10 times and report the average value.

4.1.1 Comparison of Data Delivery Delay

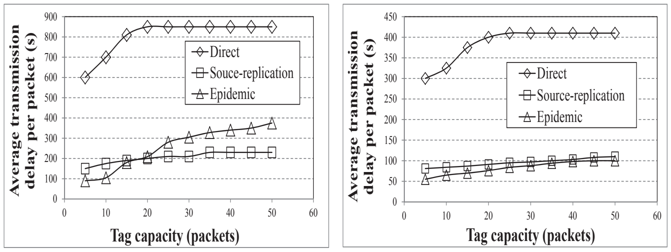
We use the delivery latency of the copy of a packet that first arrives at an RFID reader as the packet’s transmission delay. We calculated the average transmission delay per packet for successfully delivered packets. Figure 5 shows the average transmission delay versus the tag capacity with two different reading ranges of all nodes denoted by $R = 20m$ and $R = 40m$. Comparing the two figures, we find that as the reading range increases, the average transmission delay of all three protocols decreases. This is because a larger reading range makes it easier for a node to find other neighbor nodes, which may either be the RFID readers or promising relay nodes moving to RFID readers. Moreover, the movement speed of the electromagnetic waves is much faster than the nodes, thus message delivery delay is reduced with a larger transmission range. In each figure, we find that as the tag capacity grows, the average transmission delay also increases. With a smaller tag capacity, more packets are dropped without retransmission, leading to lower average transmission delay. With the increase of the tag capacity, more packets are able to reside in the tags long enough to be delivered to readers, leading to longer transmission delay.

The comparison of the three protocols indicates that the direct transmission always produces higher average transmission delay than epidemic and source-replication. Epidemic and source-replication proactively transmit data to RFID readers using multi-hop routing, while the direct transmission lets nodes wait until meeting readers to transmit data. The result indicates the higher efficiency of the HRW transmission mode than the traditional RFID system.

We see that when R=20m and the tag capacity is less than 15, source-replication has higher delay than epidemic, and when the tag capacity is larger than 15, source-replication has lower delay than epidemic. In the former case, many packets are dropped in epidemic. Such packet dropping reduces the delivery delay in epidemic. Source-replication does not have so many drops, thus it has longer delay. As the tag size increases, more packets can be buffered in the tag. As some of the previously dropped packets have long transmission delay, the average delay in epidemic increases. However, as source-replication is not greatly affected by the tag size, the delay performance does not increase significantly. When R=40 and tag capacity is small, source-replication leads to higher delay than epidemic due to the same reason as when R=20m.

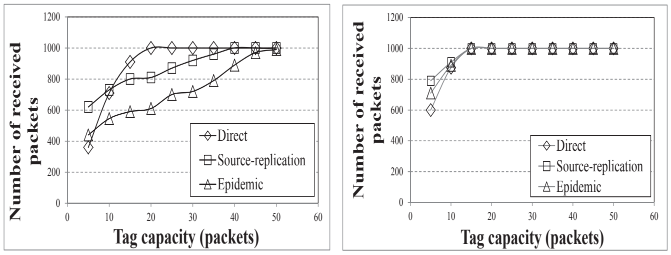
4.1.2 Comparison of Data Delivery Capacity

Figure 6 shows the number of successfully delivered packets (i.e., received packets) versus the tag capacity. Both figures indicate that as the tag capacity increases, the number of received packets increases due to fewer packet drops as explained previously. Comparing the two figures, we see that as the transmission range of the nodes increases, the number of received packets also increases. The reason is that with the increasing reading range, it is more likely for a certain node to



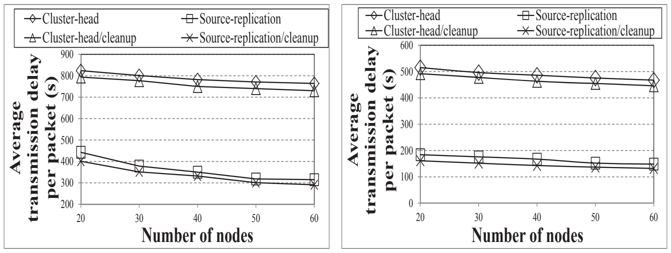
(a) Range =20m (b) Range = 40m

Fig. 5: Transmission delay versus tag capacity.



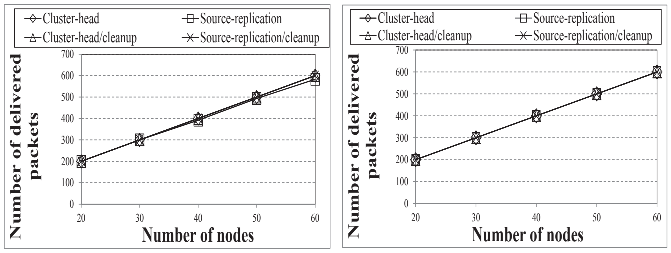
(a) Range =20m (b) Range = 40m

Fig. 6: Delivery capability versus tag capacity.



(a) Range =20m (b) Range = 40m

Fig. 7: Comparison of transmission delay versus network size.



(a) Range =20m (b) Range = 40m

Fig. 8: Comparison of the delivery capacity versus network size.

meet an RFID reader or to find more neighbor nodes for the information replication. With R=40m, when the tag capacity is larger than 10, all packets in three protocols can be successfully delivered. This is because packets can be quickly transmitted to RFID readers without a long-time buffering, leading to fewer packet drops.

Figure 6 also shows that the direct transmission suffers from more congestion than the other two protocols when a tag has small capacity. This is because the transmission delay of direct transmission is much longer than the other two protocols, so that nodes have much less free tag buffer than the nodes in the other two protocols. Therefore, when the tag capacity of nodes is limited, the traditional RFID system is not a wise choice for environmental and health monitoring applications. We can also see that as the tag size increases, the number of delivered packets in the direct transmission increases much faster than those in source-replication and epidemic and finally exceeds them. This is because the nodes in the source-replication and epidemic need to buffer multiple copies of a packet, thus resulting in more packet drops due to tag capacity limitation. Source-replication produces more successfully delivered packets than epidemic because it buffers fewer duplicated copies of a packet, leading to fewer drops of different packets.

4.2 Evaluation on Cluster-based Data Transmission

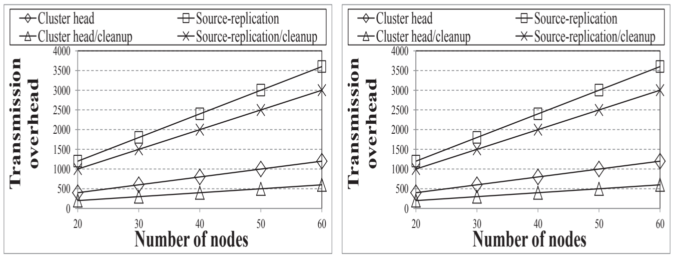
We then compare the cluster-head based and source-replication (i.e., cluster-member) based data transmission. In the simulation, every 5 nodes form a cluster and move together to random places based on the random way-point model to simulate the group moving features of the monitored objects. During the 1000s packet initiation time, every node generates one packet every 100s. According to the specification of MICA2 motes [16], we assume that when reading range $R = 20m$ and $R = 40m$, the energy consumed to receive a byte is 0.0057 microjoules (mJ) and 0.0228 mJ, and to transmit a byte takes 0.0144 mJ and 0.0576 mJ, respectively. The size of a tag data is 30 bytes. We also evaluated the tag clean-up algorithm in this section.

Figure 7 shows the comparison results of the average transmission delay versus the network size excluding readers when R=20m and R=40m, respectively. We see that as the network size increases, the packet transmission delay of both algorithms decreases slightly. The reason is that given the same number of packets, increasing the number of nodes in the same area increases the node density. Therefore, source nodes gain higher probability of meeting other nodes or cluster heads to forward their packets, which reduces the transmission delay. We see that source-replication decreases slightly faster than the cluster-head as the network size increases. This is because the probability of the head of a cluster to meet readers is not increased as much as that of any node in a cluster as the network size increases.

We also see that cluster-head has longer delay than source-replication. In the cluster-head method, the cluster head holds the replicas of the packets in the cluster and sends the replicas to an RFID reader when it meets an RFID reader. In the source-replication method, every node in a cluster holds a copy of packets from other nodes in the cluster. All information can be transmitted to an RFID reader whenever one cluster member meets an RFID reader, which greatly reduces the packet transmission delay.

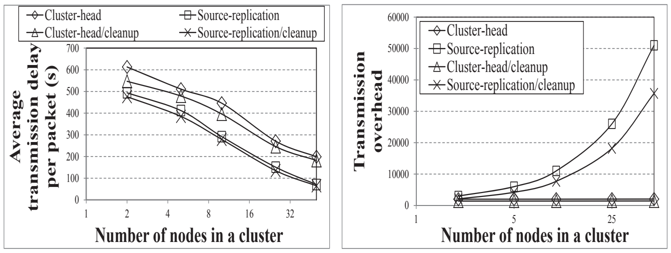
Comparing Figure 7(a) and Figure 7(b), we find that as the transmission range of the nodes increases, the packet transmission delay decreases. This is because a larger transmission range enables the nodes to communicate with RFID readers at longer distances, which decreases the packet transmission delay. Both figures show that the tag clean-up algorithm helps to reduce transmission delay. This algorithm reduces the number of redundant packets in the tag, saving more space for the transmission of other undelivered packets. More replicas of a packet increases the probability that one replica is delivered, thus reducing its transmission delay.

Figure 8 shows the comparison results of the number of delivered packets versus different network sizes. We see that as the network size increases, the number of



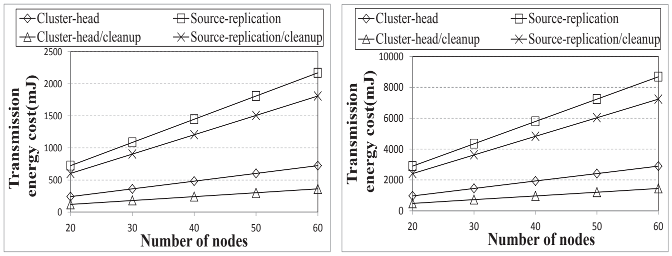
(a) Range =20m (b) Range = 40m

Fig. 9: Comparison of transmission overhead versus network size.



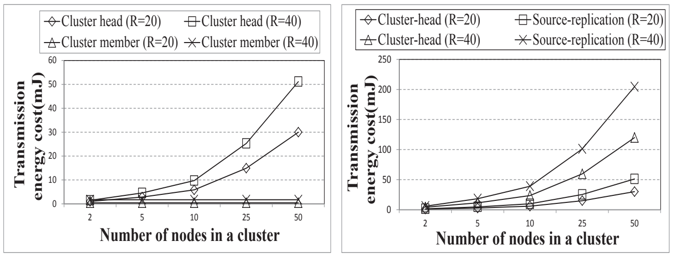
(a) Transmission delay (b) Transmission overhead

Fig. 10: Transmission delay and overhead versus cluster size.



(a) Range =20m (b) Range = 40m

Fig. 11: Comparison of total transmission energy cost versus network size.



(a) Cluster head vs cluster member (b) Cluster head energy cost

Fig. 12: Transmission energy cost of a cluster head versus cluster size.

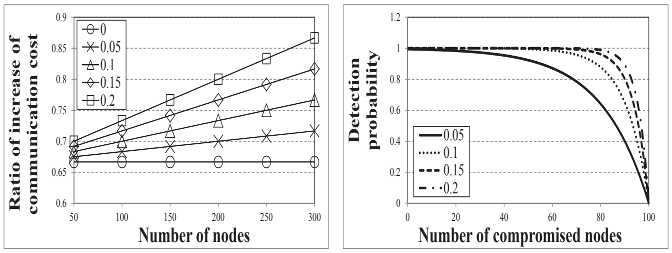
delivered packets in the system increases. The reason is that more nodes in the system generate more packets and also increase the network density, thus increasing the probability of successful packet delivery. We also see that when the transmission range of nodes is 40m, all methods can deliver all packets due to the same reason as for Figure 6. When R=20, no less than 97% packets were successfully delivered. Also, source-replication has very slightly less number of delivered packets than other methods. As source-replication does not use the clean-up algorithm and the probability for a cluster member to meet an RFID reader is smaller with a small reading range, packets are more likely to be dropped because of the congestion.

We define the transmission overhead as the number of transmission operations between nodes, and between nodes and RFID readers for all packets. Figure 9 shows the comparison results of transmission overhead versus the network size. We see that cluster-head generates much less overhead than source-replication. This is because in cluster-head, only the cluster head holds the replicas of the packets in all nodes in the cluster while in source-replication, every node in the cluster holds a replica of the packets of all other cluster members. We also see that as the network size increases, the overhead grows since more packets are initiated. We also see that the transmission overhead increases as the number of nodes increases, but it is not greatly affected by the reading range. The transmission overhead increases in proportion to the number of nodes. After a cluster head gathers all information in its cluster in cluster-head or after cluster members exchange the information in source-replication, no more additional replicas are generated. As nearly all packets were finally delivered to the readers in the simulation time, the transmission range does not affect the transmission overhead greatly. The figures also show that the tag clean-up algorithm reduces the transmission overhead. Since this algorithm helps nodes reduce redundant packets, fewer outdated information is exchanged among nodes and between nodes and readers, leading to less overhead.

Figure 10(a) shows the comparison results of the average transmission delay versus different cluster sizes. As the cluster size increases, the transmission delay of both transmission algorithms decreases. In cluster-head, as the cluster size increases, the cluster head holds more information of the nodes in the system. Therefore, once the cluster head meets an RFID reader, all information of the cluster is delivered to it at one time. This greatly reduces the packet transmission delay when the packets are sent to the RFID readers separately by multiple nodes. Similarly, in source-replication, with more nodes in a cluster, more information of the nodes is gathered for transmission. Once a cluster member meets an RFID reader, all information of the cluster members is delivered. Since the probability that any cluster member of a cluster meets an RFID reader is higher than the probability that a cluster head meets an RFID reader, the transmission delay of source-replication is much lower than cluster-head. We also see that the tag clean-up algorithm reduces the packet transmission delay due to the same reason as for Figure 7.

Figure 10(b) shows the transmission overhead versus the number of nodes in a cluster. In source-replication, every node in a cluster needs to exchange packets with each other. In cluster-head, only the cluster head needs to collect packets from its cluster members. Therefore, the transmission overhead of source-replication is much higher than that of cluster-head. We see that the tag clean-up algorithm reduces the transmission overhead due to the same reason as for Figure 9.

Figure 11 shows the comparison results of total transmission energy cost versus the network size. We see that the results of transmission energy cost of different methods coincide with the corresponding transmission overhead in Figure 9, since the transmission overhead represents the number of packet transmissions. However, unlike Figure 9, the total transmission energy cost for each method with $R = 40$ is significantly larger than the corresponding result with $R = 20$, since larger



(a) Overhead

(b) Detection probability

Fig. 13: Overhead and detection probability on security mechanism.

reading range requires higher transmission power. Due to the consistent relationship between the transmission energy cost and the transmission overhead, we focus on the transmission overhead in the following evaluations, which directly reflects the transmission energy cost.

Figure 12(a) compares the energy cost of a cluster head with that of a cluster member in a cluster-head based transmission for collecting the data from all nodes in the cluster. As we see, in both cases of $R = 20$ and $R = 40$, the transmission energy cost of the cluster head is significantly larger than that of each cluster member. The energy cost gap between a cluster head and a cluster member becomes larger when the number of nodes increases, because the energy cost of the cluster head increases with the cluster size, and the energy cost of each cluster member remains the same. We see that the energy cost of a cluster head is not extremely high and can be afforded by a node when a cluster size is limited. Figure 12(b) compares the energy cost of the cluster head in the cluster-head based transmission with that of the cluster member node that forwards the aggregated data in the source-replication algorithm. The figure shows that the cluster head has much less energy consumption than the forwarding cluster-member node. This is because the latter consumes much energy to replicate its data to all other cluster members. The experimental results show that though a cluster head consumes high energy as it forwards all data of its cluster members to a reader, the energy cost of the cluster-head method is much lower than that of the source-replication method.

4.3 Analysis of the Security Mechanisms

In this section, we evaluate the performance of our security mechanisms. We use cluster-head and cluster-member replication method for data transmission.

Cost analysis. In the transmitted packet with the security mechanisms $(N, Nonce, En(K'_N, D_N), MAC(K'_N, N|D_N))$, the size of the tag ID N, Nonce and the encrypted data $En(K'_N, D_N)$ was set to 32 bits respectively, and the size of the encrypted data $En(K'_N, D_N)$ was set to 64 bits. The size of the plain data without the security mechanism was set to 64 bits. We used the total transmitted bits to measure the communication cost. We evaluated the ratio of the increase of total communication cost, including the data replication cost of nodes and the communication cost between nodes and the reader. The ratio was computed by $(C_s - C)/C$, where C and C_s are the total communication cost without and with the security mechanisms respectively.

Figure 13(a) shows the ratio of the cost increase with regard to different threshold ρ value and the number of nodes in a cluster. In the figure, $\rho = 0$ means that only message encryption and authentication are used and there is no protection against data selective forwarding. Because of additional nonce and message authentication code, the size of each data message increases by more than half, about 67%. We see that a larger threshold leads to higher ratio of cost increase as it causes more nodes to send their MACs to the RFID reader. Given a threshold, the ratio of cost increase grows as the network size increases because larger network size makes more nodes to send their MACs to the RFID readers.

Security analysis. Compromised tags can collude together to provide enough MACs to authenticate a false data. Thus, a critical problem for our secure randomized solution is how much resiliency it has against compromised nodes. To answer this question, we analyze the detection probability of false data when a number of compromised nodes exist in the system.

Theorem 4.1: Suppose in a group of n tags, n_c number of compromised tags collude with a compromised node which sends pruned data set to the RFID reader. Given the threshold ρ for the probability of a node sending its MAC to the reader in the cluster, then the probability of the reader successfully detecting data selective drop attack, denoted by Pr_d, is the probability that at least one non-compromised tag choose to send its MAC for all data set to the sender. Then, we have

$$Pr_d = 1 - (1 - \rho)^{(n-n_c)}. \tag{3}$$

Proof: According to Formula (2), each node is selected with probability ρ to send MAC to the reader. Iff only the compromised nodes send their MACs to the reader, the false data cannot be detected. The probability of such case is $(1 - \rho)^{(n-n_c)}$. Then, the detection probability can be derived as $1 - (1 - \rho)^{(n-n_c)}$. □

Given $n = 100$, we show the relationship between Pr_d and n_c with regard to different ρ values in Figure 13(b). The figure shows that our security mechanism is resilient to the node compromising attack, since the detection probability is greater than 90% even when half of nodes in a cluster are compromised. A larger ρ value further increases the resiliency of our security mechanism. When ρ is greater than 0.05, the detection probability is almost one even when 50% nodes are compromised.

5 Conclusion

In this paper, we propose Hybrid RFID and WSN System (HRW) that integrates the multi-hop transmission mode of WSNs and direction transmission mode of RFID systems to improve the efficiency of data collection, hence to meet the requirements of low economic cost, high performance and real-time monitoring in mobile monitoring applications. HRW is composed of RFID readers and hybrid smart nodes. Instead of waiting for RFID readers to read data, smart nodes replicate packets with neighbor nodes using special reduced functional RFID readers. The collected packets are sent to a RFID reader when one of the replica nodes moves into the range of the RFID reader. We further propose enhanced data transmission algorithms and security mechanisms. Extensive simulation and trace-driven experimental results show that HRW outperforms traditional RFID in terms of the cost of deployment, transmission capacity and delay and tag capacity requirement. In the future, we plan to evaluate HRW in a real-world testbed.

ACKNOWLEDGEMENTS

This research was supported in part by U.S. NSF grants CNS-1249603, CNS-1049947, CNS-1156875, CNS-0917056 and CNS-1057530, CNS-1025652, CNS-0938189, CSR-2008826, CSR-2008827, Microsoft Research Faculty Fellowship 8300751, and U.S. Department of Energy's Oak Ridge National Laboratory including the Extreme Scale Systems Center located at ORNL and DoD 4000111689.

REFERENCES

[1] RFID progress at Wal-Mart in IDTechEx website. http://www.idtechex.com/research/articles/rfid_progress_at_wal_mart_00000161.asp.

[2] R. Clauberg. RFID and sensor networks. In *RFID Workshop*, 2004.

[3] L. Zhang and Z. Wang. Integration of rfid into wireless sensor networks: Architectures, opportunities and challenging problems. In *Proc. of the Grid and Cooperative Computing Workshops*, 2006.

[4] H. Liu, M. Bolic, A. Nayak, and I. Stojmenovic. Taxonomy and challenges of the integration of RFID and wireless sensor networks. *Network, IEEE*, 22(6):26 –35, 2008.

[5] J. Y. Daniel, J. H. Holleman, R. Prasad, J. R. Smith, and B. P. Otis. NeuralWISP: A Wirelessly Powered Neural Interface With 1-m Range . *IEEE Transactions on Biomedical Circuits and Systems*, 2009.

[6] A. P. Sample, D. J. Yeager, and J. R. Smith. A capacitive touch interface for passive rfid tags. In *Proc. of the 2009 IEEE International Conference on RFID*, 2009.

[7] Z. Li, H. Shen, and B. Alsaify. Integrating rfid with wireless sensor networks for inhabitant, environment and health monitoring. In *Proc. of ICPADS*, 2008.

[8] T. Lez and D. Kim. Wireless sensor networks and rfid integration for context aware services. Technical report, The Auto-ID Labs, 2007.

[9] Feng Li and Jie Wu. MOPS: Providing Content-Based Service in Disruption-Tolerant Networks. In *Proc. of ICDCS*, 2009.

[10] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and Panigrahy R. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In *Proc. of STOC*, pages 654–663, 1997.

[11] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in termittently connected mobile networks: The single-copy case. *ACM/IEEE Transactions on Networking*, 2007.

[12] Y. Wu, T. Ho, W. Liao, and C. Tsao. Epoch, length of the random waypoint model in mobile ad hoc networks. *Communications Letters, IEEE*, 2005.

[13] N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. *PNAS*, 106(36), 2009.

[14] A. Chaintreau, P. Hui, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Impact of human mobility on opportunistic forwarding algorithms. *IEEE TMC*, 6(6):606–620, 2007.

[15] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, 2000.

[16] Crossbow Inc. Mpr-mote processor radio board user's manual.

[17] S. M. Ross. *Introduction to Probability Models, 8th Edition*. Amsterdam: Academic Press, 2003.

[18] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 1970.

[19] D. Simplot-Ryl, I. Stojmenovic, A. Micic, and A. Nayak. A hybrid randomized protocol for RFID tag identification. *Sensor Review*, 2006.

[20] B. Firner, P. Jadhav, Y. Zhang, R. Howard, W. Trappe, and E. Fenson. Towards continuous asset tracking: Low-power communication and fail-safe presence assurance. In *Proc. of SECON*, 2009.

[21] Y. Gu, G. Yu, X. Li, and Y. Wang. RFID data interpolation algorithm based on dynamic probabilistic path-event model. *Journal of Software*, 2010.

[22] C. Lee and C. Chung. RFID data processing in supply chain management using a path encoding scheme. *TKDE*, 2011.

[23] C. Tan, B. Sheng, and Q. Li. How to monitor for missing RFID tags. In *Proc. of ICDCS*, 2008.

[24] B. Sheng, Q. Li, and W. Mao. Efficient continuous scanning in RFID systems. In *Proc. of INFOCOM*, 2010.

[25] W. Luo, S. Chen, T. Li, and Y. Qiao. Probabilistic missing-tag detection and energy-time tradeoff in large-scale RFID systems. In *Proc. of Mobihoc*, 2012.

[26] W. Luo, S. Chen, T. Li, and S. Chen. Efficient missing tag detection in rfid systems. In *Proc. of INFOCOM, mini-conference*, 2011.

[27] Y. Bai, F. Wang, P. Liu, Zaniolo, and S. Liu. RFID data processing with a data stream query language. In *Proc. of ICDE*, 2007.

[28] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In *Proc. of SenSys*, 2004.

[29] A. Maffei, K. Fall, and D. Chayes. Ocean instrument internet. In *Proc. of AGU*, 2006.

[30] E. Jovanov, A. Milenkovic, C. Otto, and P. C. Groen. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. *J. Neuroeng*, 2005.

[31] M. Guizani H. Nait-Charif T. Taleb, D. Bottazzi. Angelah: a framework for assisting elders at home. *JSAC*, 2009.

[32] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proc. of SenSys*, 2008.

[33] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proc. of SenSys*, 2009.

[34] M. Li, Y. Liu, J. Wang, and Z. Yang. Sensor network navigation without locations. In *Proc. of Infocom*, 2009.

[35] X. Wu G. Chen S. Li, A. Zhan. Ern: emergence rescue navigation with wireless sensor networks. In *Proc. of IPDPS*, 2009.

[36] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In *Proc. of SenSys*, 2010.

[37] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: Mobility-centric data dissemination algorithm for vehicular networks. In *Proc. of ACM on VANET*, 2004.

[38] G. Zhou, J. Lu, C. Y. Wan, M. Yarvis, and J. Stankovic. BodyQoS: adaptive and radio-agnostic QoS for body sensor networks. In *Proc. of INFOCOM*, 2008.

[39] D. Vlasic, R. Adelsberge, G. Vannucc, J. Barnwell, M. Gross, W. Matusik, and J. Popovi. Practical motion capture in everyday surroundings. *ACM Trans. Graph*, 2007.

[40] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G. S. Ahn, and A. T. Campbell. The bikenet mobile sensing system for cyclist experience mapping. In *Proc. of SenSys*, 2007.

[41] X. Chen, K. Makki, Kang Yen, and N. Pissinou. Sensor network security: a survey. *IEEE Communications Surveys Tutorials*, 2009.

[42] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. A. Arbaugh. Toward resilient security in wireless sensor networks. In *Proc. of ACM MobiHoc*, pages 34–45, 2005.

[43] Z. Yu and Y. Guan. A dynamic en-route scheme for filtering false data injection in wireless sensor networks. In *Proc. of IEEE INFOCOM*, April 2006.

[44] K. Ren, W. Lou, and Y. Zhang. LEDS: Providing location-aware end-to-end data security in wireless sensor networks. In *Proc. of IEEE INFOCOM*, April 2006.

[45] L. Yu and J. Li. Grouping-based resilient statistical en-route filtering for sensor networks. In *Proc. of INFOCOM*, 2009.

[46] Z. Yang and H. Wu. FINDERS: A Featherlight Information Network With Delay-Endurable RFID Support. *IEEE/ACM ToN*, 19(4):961 –974, 2011.

[47] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *Proc. of IFIP*, 2006.

[48] Y. Wang and et al. Erasure-coding based routing for opportunistic networks. In *Proc. of SIGCOMM*, 2005.

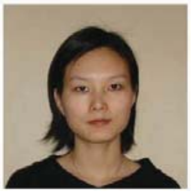
[49] Y. Wang and H. Wu. Delay/fault-tolerant mobile sensor network (DFT-MSN): A new paradigm for pervasive information gathering. *IEEE Transactions on mobile computing*, 2006.

[50] J. Burgess, B. Gallagher, D. Jensen, and B. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proc. of IEEE INFOCOM*, 2006.

[51] F. Li and J. Wu. MOPS: Providing Content-based Service in Disruption-tolerant Networks. In *Proc. of ICDCS*, 2009.

[52] J. Ghosh, S. J. Philip, and C. Qiao. Sociological orbit aware location approximation and routing (SOLAR) in MANET. *Ad Hoc Networks*, 2007.

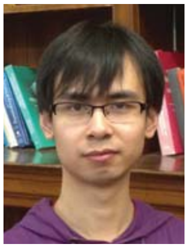
[53] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proc. of Mobihoc*, 2007.



Haiying Shen Haiying Shen received the BS degree in Computer Science and Engineering from Tongji University, China in 2000, and the MS and Ph.D. degrees in Computer Engineering from Wayne State University in 2004 and 2006, respectively. She is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Clemson University. Her research interests include distributed computer systems and computer networks, with an emphasis on P2P and content delivery networks, mobile computing, wireless sensor networks, and cloud computing. She is a Microsoft Faculty Fellow of 2010 and a member of the IEEE and ACM.

Ze Li Ze Li received the BS degree in Electronics and Information Engineering from Huazhong University of Science and Technology, China in 2007, and the Ph.D. degree in Computer Engineering from Clemson University. His research interests include distributed networks, with an emphasis on peer-to-peer and content delivery networks. He is currently a data scientist in the MicroStrategy Incorporation.

Lei Yu Lei Yu received the PhD degree in computer science from Harbin Institute of Technology, China, in 2011. He currently is a postdoctoral research fellow in the Department of Electrical and Computer Engineering at Clemson University, SC, United States. His research interests include sensor networks, wireless networks, cloud computing and network security.



Chenxi Qiu Chenxi Qiu received the BS degree in Telecommunication Engineering from Xidian University, China, in 2009. He currently is a Ph.D student in the Department of Electrical and Computer Engineering at Clemson University, SC, United States. His research interests include sensor networks and wireless networks.