FINDERS: A Featherlight Information Network With Delay-Endurable RFID Support

Zhipeng Yang, *Student Member, IEEE*, and Hongyi Wu, *Member, IEEE*

Abstract—**In this paper, we investigate the use of radio frequency identification (RFID) gear for wireless sensor network construction, aiming to find events of interest and gather aggregate information. In particular, we develop a *featherlight information network with delay-endurable RFID support* (FINDERS), composed of passive RFID tags that are ultralight, durable, and flexible, without power supply for long-lasting applications. FINDERS faces unprecedented challenges in communication and networking due to its sporadic wireless links, unique asymmetric communication paradigm, intermittent computation capability, and extremely small memory of tags. Several effective techniques are proposed to address these challenges, arriving at an efficient communication protocol for FINDERS. A prototype system is developed, and test-bed experiments are carried out with 38 participants and for 9 days, yielding interesting experimental results that offer valuable insights into RFID-based delay-tolerant networks and provide useful practical guidance for the setup of FINDERS systems.**

Index Terms—**Analysis, delay-tolerant network (DTN), experiments, passive radio frequency identification (RFID), wildlife tracking.**

I. INTRODUCTION

THIS paper aims to develop novel network technologies that support wildlife tracking under extreme weight constraints and harsh environments. As a rule of thumb to avoid unwanted effect on the wildlife, the weight of a portable tracking device (i.e., a sensor) must be under 5% of the weight of the animal. This requires very light sensors to be employed for small wildlife (such as nutrias, frogs, and various insects), which is in sharp contrast to other projects that deal with large animals including zebras [1], whales [2], deer [3], and more commonly human beings [4]–[8]. Earlier investigation has revealed that most of these small animals cannot carry any active devices (such as GPS receivers or the smallest Crossbow sensors) with battery and casing since overweight sensors often lead to high mortality rates of the animals being studied. Similar weight constraint also applies in many other fields of scientific studies and industrial applications. While efforts have been made to develop miniature sensors, the lowest weight of any active sensor that aims to achieve a given communication range and lifetime is bounded inevitably by its battery and casing, where the latter must be heavy-duty for the protection of power source and powered electronic circuits under harsh environments. This becomes a key hurdle that limits the applicability of active sensors in various wildlife tracking applications, not to mention extra hassle involved in ensuring adequate battery power.

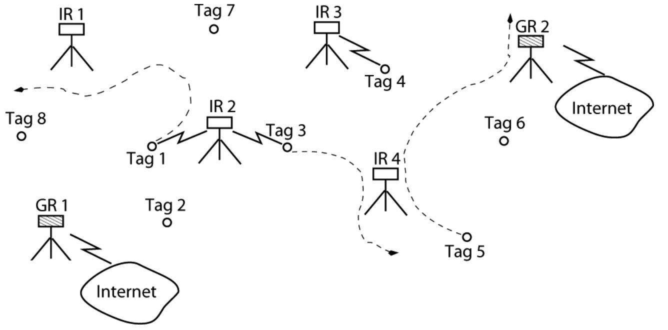


Fig. 1. Overview of FINDERS that consists of mobile tags, isolated readers (IRs) and gateway readers (GRs).

In this paper, we investigate the use of radio frequency identification (RFID) gear for information network construction, aiming to find events of interest and gather aggregate data. In particular, we develop a *featherlight information network with delay-endurable RFID support* (FINDERS), composed of passive RFID tags that are ultralight, durable, and flexible, without power supply for long-lasting applications.

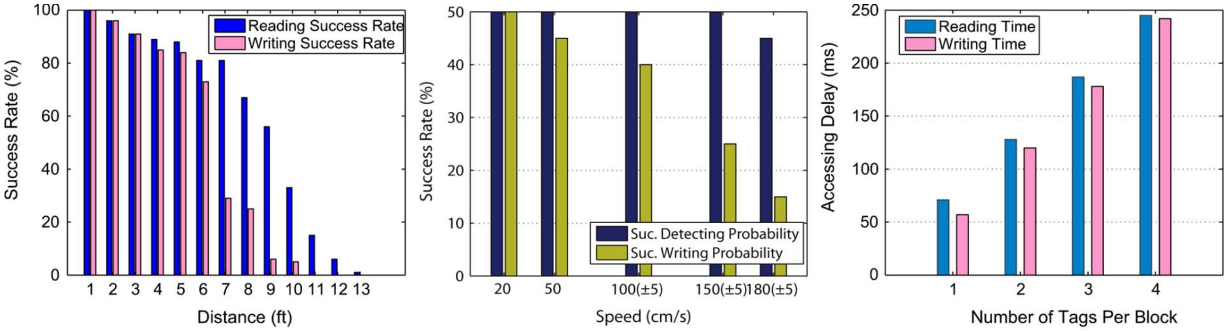
FINDERS consists of two types of nodes, readers and tags, as illustrated in Fig. 1. The readers are deployed at strategic locations according to specific applications. For example, in wildlife and biological studies, readers can be installed at hotspots (such as water sources) where the animals visit frequently or choke points (e.g., a tunnel entrance) that they have to move through because of significant movement barriers otherwise. In remote field deployment, infrastructure-based wireless networks (such as 2G/3G cellular, WiMAX, and telemetry systems) are not available to cover the readers. Moreover, since FINDERS is often deployed under harsh and complex wildlife environments with various obstacles, it is not practically viable for most readers to access satellites. Neither can they establish reliable connections to communicate with each other due to their sparse deployment where two readers are usually separated by a distance longer than the radio communication range of most portable wireless technologies (e.g., IEEE 802.11). As a result, they become isolated readers, or IRs (see IRs 1–4 in Fig. 1). Power supply is challenging for the IRs as well. Most of them rely on solar panels and high-capacity batteries, which can be recharged/replaced, but will post a limit on the daily duty circle of the readers. Only a few readers at convenient locations are equipped with reliable power source and network connections,

Manuscript received August 19, 2009; revised July 06, 2010 and October 08, 2010; accepted October 28, 2010; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Liu. Date of publication November 29, 2010; date of current version August 17, 2011. This work was supported in part by the National Science Foundation under Grant CNS-0831823. Part of the work was presented at the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Rome, Italy, June 22–26, 2009.

The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70503 USA (e-mail: zhipengyang@hotmail.com; wu@cacs.louisiana.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2010.2091425



(a) (b) (c)

Fig. 2. Experimental results of read/write operations-based ALR-9900 readers and ALN-9540 Squiggle (passive) tags: (a) Read/write range (average of 1000 tests); (b) Mobility impact (average of 10 tests); (c) Read/write delay (average of 1000 tests).

and they are dubbed gateway readers (GRs; see GRs 1 and 2 in Fig. 1). GRs serve as the gateways to deliver data from FINDERS to the destination (e.g., a data server). The readers have relatively large storage space compared to tags. While all readers are fixed, the tags (e.g., Tags 1–8 in Fig. 1) are attached to moving targets and thus become mobile.

FINDERS aims to gather both mobility information and sensor data. First, when a tag moves into the communication range of and is detected by a reader, the reader generates a data packet to reflect the meeting event. Second, sensors (such as light intensity sensors and body temperature sensors) can be integrated with a tag. When the tag is powered up, the sensors take measurement with corresponding data written into the tag's nonvolatile memory. A collection of meeting events and sensor readings intrinsically provide a discrete sampling of the interested habitat, valuable for a wide range of wildlife and biological researches such as the modeling of their population and movement. The goal of this research is to enable efficient delivery of data generated at tags and readers.

Since readers are static and most of them are isolated from each other, the communications in FINDERS rely on the mobile tags to establish time-varying opportunistic links with nearby readers, forming a delay-tolerant network (DTN) [9] for data delivery. For example, in the scenario shown in Fig. 1, Tags 1 and 3 are within the reading/writing range of IR 2. Thus, IR 2 may read the data from Tag 1 and write them into Tag 3. When Tag 3 passes by IR 4, it unloads its data to the latter, which in turn writes the data into Tag 5 when it comes into the writing range. Tag 5's trajectory passes through GR 2. The data can thus be delivered to the destination via this gateway.

A. Unique Characteristics of FINDERS

FINDERS is a unique wireless information network that distinguishes itself from conventional communication networks, sensor networks, and delay-tolerant networks by the following characteristics.

1) **Nodal heterogeneity**: FINDERS consists of two very different types of nodes. A reader is a static and powerful device, with large storage, high computing power, and long-lasting (but still limited) battery power. On the other hand, a passive tag can be mobile and has extremely limited resource.
2) **Asymmetric communication**: The communication in FINDERS can be established between a tag and a reader only, but not tags to tags or readers to readers. The communication must be initiated by a reader, in contrast to symmetric transmissions in most conventional networks.
3) **Intermittent connectivity**: The communication range between a reader and a tag is short. Our experiments based on Alien RFID gears show that the writing success rate drops sharply when the distance between a reader and a tag is beyond 6 ft [as depicted in Fig. 2(a)]. The reading range is longer, but its success rate also becomes low when the distance is more than 9 ft. When the tags are mobile, the success rate is further reduced. For example, we have carried out an experiment by using an iRobot Create Programmable robot. A RFID tag is attached to the robot. A RFID reader is set up at the door of an office room. The robot starts at a point 15 ft outside the door and moves into the room. Although the shortest distance between the tag and the reader is about 1 ft only during the movement (which should result in perfect read/write as shown in Fig. 2(a) if they were under static setup), the writing success rate degrades significantly with the increase of the moving speed. Due to short communication range and dynamic tag mobility, the connectivity of FINDERS is very low and intermittent, forming a sparse network where a tag is connected to a reader only occasionally. In fact, FINDERS is a DTN with unique communication and storage constraints.
4) **Intermittent computation**: Besides connectivity, the computation at the tag is also intermittent. It is available when the tag is powered up by a nearby reader, and once charged, the tag can hold its energy for up to about 5 s only [10]. Thus, such continuous functions necessary to many protocols as timers cannot be implemented here.
5) **Critical network resource**: The fundamental philosophy in DTN is to trade storage for sporadic connectivity. The DTN node is assumed to have sufficient storage space, which can hold a large volume of data to alleviate the needs of immediate transmission. This, however, is no longer valid in FINDERS because the buffer space of the tags (the main vehicle for data transportation) is so limited that it may become the critical network resource and communication bottleneck.
6) **Delay tolerability**: Data delivery delay in FINDERS is potentially high due to loose connectivity and extremely limited tag capacity. However, such delay, though not desirable, is usually tolerable by the applications that

aim at pervasive information gathering from a statistical perspective.

7) **Fault tolerability**: Redundancy exists in FINDERS in data acquisition and delivery. Thus, a data packet may be dropped without degrading information gathering performance.

B. *Challenges in FINDERS*

The above characteristics make the development of FINDERS a unique, interesting, and challenging problem, calling for effective solutions to overhaul the data acquisition and delivery schemes in such a featherlight information network with extremely limited resources. While various sensor networks have been investigated and several prototypes and test-beds have been developed and deployed [1]–[8], data collection, storage, and communication solutions devised therein are not directly applicable to FINDERS. Few earlier studies have ever been conducted on information networks composed of RFID gears with sporadic communication between moving tags and nearby readers. In particular, FINDERS faces the following unprecedented challenges. First, the sporadically available wireless links render it impossible to form a well-connected mesh network for end-to-end communications, which are the basis of mainstream sensor network technologies. Moreover, the unique asymmetric communication paradigm, the intermittent computation capability, and the extremely small memory size of a passive tag overthrow the fundamental principle of DTN (where ample buffering is employed to alleviate the needs of immediate transmission), resulting in inefficiency if the existing DTN protocols [1], [2], [4], [11]–[20] are adopted in FINDERS. To this end, a series of interesting problems related to data delivery are investigated in this research.

1) **Routing metrics**: Routing metrics are usually adopted by routing protocols as an indicator of the available resource of a given path or link. For example, hop count and delay are popularly used in conventional networks. These metrics, however, do not reflect the unique network resource of FINDERS and may thus lead to poor network performance or even failure if used for routing. In this research, we investigate effective routing metrics based on meeting probabilities between tags and readers and devise a mechanism that consumes insignificant storage space for maintaining and updating such metrics.
2) **Duplication control**: In FINDERS and many other DTN networks, replication is necessary during data delivery for achieving a given success ratio. However, replication increases overhead, and worse yet, excessive replication may even degrade the delivery ratio due to frequent buffer overflow. Duplication control thus becomes a key issue to be tackled in this research. It arrives at a system-wide optimization problem. While the data delivery schemes in FINDERS are (and have to be) much simpler compared to the protocols for other wireless or wired networks, such optimization is nontrivial, and its performance largely determines the effectiveness and feasibility of FINDERS.
3) **Queue management**: Each reader maintains a data queue. Given the limited communication resource (due to sporadic connectivity and extremely limited storage of tags) and the relatively high loading factor (due to newly acquired and duplicated data), it is an overriding design issue in FINDERS to differentiate the packets in the queue by a simple and efficient parameter that signifies their importance and determines which packet to transmit if a communication opportunity becomes available or which packet to drop if the queue is full. In this research, we propose the prioritization parameter based on the probability that at least one copy of given information can be delivered to the GRs. Note that the goal of such queue management is to facilitate efficient communication in the network with redundancy, in contrast to the QoS-aware algorithms that aim to prioritize raw information.

Routing, duplication, and queue management are dependent. For instance, data duplication clearly affects queue management and, at the same time, itself is influenced by routing. Built on the three basic components, we carefully devise the data delivery protocol by synergizing the interaction among them in order to achieve efficient network resource utilization.

C. *Contribution of This Paper*

- As far as we know, FINDERS is the first work that investigates the RFID-based delay-tolerant mobile sensor networks.
- A set of technical challenges in such a unique network environment is identified and addressed, leading to an efficient communication protocol for FINDERS.
- A prototype system is developed, and test-bed experiments are carried out with 38 participants and for 9 days, yielding interesting experimental results that offer valuable insights into FINDERS, particularly the critical impact of nodal mobility on data generation and delivery.

The rest of this paper is organized as follows. Section II presents an analytic study that reveals the general capacity of FINDERS. Sections III and IV introduce our proposed protocol and its implementation. Section V presents test-bed experiments and results. Finally, Section VII concludes the paper.

II. GENERAL FEASIBILITY STUDY

While the featherlight weight of passive RFID tags provides unique opportunities to support various applications with strict weight constraints, the capacity of FINDERS is unsurprisingly low compared to many other data networks due to its extremely limited network resources. To understand its capacity and accordingly the applications that it can support, we first carry out a feasibility study of the FINDERS system before moving into the detailed protocol design and evaluation. This feasibility study aims to gain insights into the basics of FINDERS and unveil its general capacity, without consideration of specific protocols and network environments.

To this end, we investigate why a data delivery effort may fail and where the data packet may be dropped. First, since the GRs connect to conventional networks, they have sufficient communication bandwidth (for our target applications) and can safely deliver the received data to end-users. The tags keep data packets in their nonvolatile memory. If we ignore physical damage on tags, the data will not be lost. On the other hand, each IR maintains a data queue, which may overflow when packet arrival is

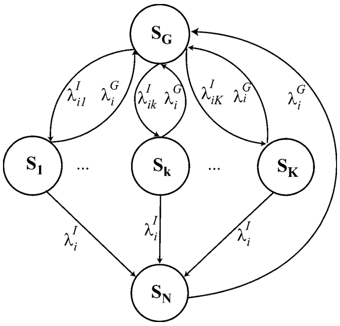


Fig. 3. Markovian analytic model for a tag (Tag i).

too high. Therefore, we focus on the queueing behavior of the IRs in our feasibility study.

1) Queueing Model: We consider a general queuing model (i.e., a $G/G/1$ queue) for each IR. While it is difficult to derive solutions for the G/G/1 queue without detailed knowledge of packet arrival and service (which heavily depend on the data transmission protocol), we intend to obtain an estimation of the network capacity (i.e., the amount of data that can be handled by the system) based on the condition of queue stability. More specifically, the arrival rate must be lower than the service rate to keep the queue stable. This study only requires us to discover the average data service rate, as to be discussed next.

2) Average Data Service Rate: A data packet is served (i.e., sent out from the IR's queue) when the IR meets a suitable tag. Thus, it is critical to understand how often such meeting occurs. Consider K IRs, J GRs, and T tags, where each tag can store up to m data packets. For simplicity, we assume that a tag receives data (i.e., becomes a suitable tag) only if it is empty (while more sophisticate schemes for data transmission will be discussed in Section III). For analytic tractability, we assume that a tag moves by following a random path, and the time to travel through such a path is a random variable under exponential distribution with a rate of λ_i for Tag i. This assumption, thought not always true under all network environments, greatly reduces the analytic complexity. Note that our feasibility study does not intend to provide an accurate analysis, but instead to have a general observation of the capacity of FINDERS. Besides, we assume Tag i has a long-term statistical probability of γ_{ik}^I to meet IR k and the probability of γ_{ij}^G to meet GR j. It is easy to verify that the time intervals to meet IR k and GR j are exponentially distributed with the rates of $\lambda_{ik}^I = \lambda_i\gamma_{ik}^I$ and $\lambda_{ij}^G = \lambda_i\gamma_{ij}^G$, respectively. In addition, let $\lambda_i^I = \sum_{k=1}^K \lambda_{ik}^I$, which is the total rate for Tag i meeting any IRs, and similarly $\lambda_i^G = \sum_{j=1}^J \lambda_{ij}^G$.

Based on above assumptions, a Markovian model (see Fig. 3) is established to analyze the meeting events between Tag i and the readers, and accordingly the data service rate of the IR's queue. The model consists of $K+2$ states. State S_k $(1 \le k \le K)$ is the state that IR k can transmit m packets to Tag i (i.e., the tag is empty). State S_N is for the state where Tag i meets any IRs, but it is not empty and thus cannot receive data packets. State S_G indicates that Tag i meets a GR and delivers the data packets it carries. Once the packets are delivered to the GR, they are removed from the tag, which thus becomes empty and ready for receiving new data packets.

Let us denote P_k, P_N, and P_G as the steady-state probabilities of states S_k, S_N, and S_G, respectively. Based on the Markovian model, we can derive the following state equations:

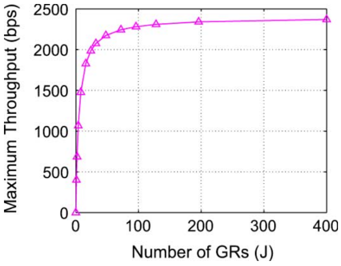


Fig. 4. Throughput increases nonlinearly with the number of GRs.

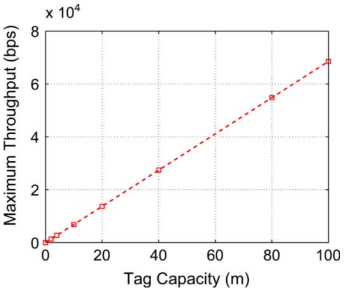


Fig. 5. Throughput increases linearly with tag capacity.

$$\begin{cases} P_k\left(\lambda_i^G + \lambda_i^I\right) = \lambda_{ik}^I P_G, \ \forall 1 \le k \le K \\ P_N\lambda_i^G = \sum\limits_{k=1}^K \lambda_i^I \\ P_G\lambda_i^I = P_N\lambda_i^G + \sum\limits_{k=1}^K \left(\lambda_{ik}^I P_k\right) \\ \sum\limits_{k=1}^K P_k + P_N + P_G = 1. \end{cases} \tag{1}$$

Solving them, we have $P_G = \frac{\lambda_i^G}{\lambda_i^I + \lambda_i^G}$, $P_k = \frac{\lambda_{ik}^I\lambda_i^G}{(\lambda_i^G + \lambda_i^I)^2}$, and $P_N = \frac{{\lambda_i^I}^2}{(\lambda_i^G + \lambda_i^I)^2}$. To facilitate the understanding of FINDERS' general capacity, we further simplify the above results by letting $\gamma_{ik}^I = \gamma_{ij}^G = \gamma$ (i.e., Tag i has equal probability to meet all readers), and accordingly $\lambda_{ik}^I = \lambda_{ij}^G = \lambda$. Then, we arrive at $P_k = \frac{J}{(K+J)^2}$, which is the steady-state probability that Tag i becomes empty and arrives at an IR. With an overall rate of $(K+J)\lambda$ for Tag i to visit any readers, it has a rate of $(K+J)\lambda P_k$ to meet IR k when it is empty. Assuming all tags are independent, the aggregated arrival rate of empty tags at an IR is $T(K+J)\lambda P_k$. Since each tag can carry m packets, we have the service rate of an IR's queue, $\mu = \frac{J}{J+K}\lambda mT$.

3) Estimated Capacity: To ensure a stable queue at the IR, the service rate must be greater than the arrival rate. Therefore, the entire FINDERS with K IRs can accept no more than $\frac{JK}{J+K}\lambda mT$ packets, which can serve as an estimation of the network capacity. It shows that, under typical scenarios, FINDERS can easily achieve a data rate of multiple kbps or higher (see Figs. 4 and 5, where by default a tag can hold 96 bits, $K = 5$, $m = 1$, $T = 100$, $\lambda = 0.05$, and $J = 2$), which is sufficient to support a wide range of applications, such as wildlife tracking and lightweight monitoring. The capacity increases linearly with m, λ, and T and nonlinearly with J and K.

In addition to the overall network capacity, we also have several interesting findings on the data generation in individual events. In general, new data are fed into the IR's queue only when it meets a tag. Without loss of generality, let us assume α packets arrive at the IR when such a meeting event occurs. Thus, we have $\alpha\lambda T \leq \mu$, or $\alpha \leq \frac{Jm}{J+K}$. In other words, when a tag meets an IR, no more than $\frac{Jm}{J+K}$ packets can be generated and fed into the IR's queue. This result provides a guideline for traffic control in FINDERS and leads to two interesting observations. First, α is not related to the number of tags. This is because the tags contribute to both data arrival and service of an individual IR. Second, increasing the number of IRs, K, results in a lower α. Though this is anti-intuitive (as IRs help data relaying), it can be explained. The IRs directly contribute to the network traffic load (when they meet tags), but do not directly consume the data (which can be successfully delivered only when they arrive at the GRs but not IRs). Thus, more IRs decrease the maximum allowed α. However, a larger K does help improve the overall network capacity because they relay and buffer data packets. Additionally, a sufficiently large K is usually required by the applications in order to achieve the needed coverage and granularity in data acquisition.

III. PROPOSED DATA DELIVERY PROTOCOL

With extremely limited resources in buffer space, communication links, and battery power, FINDERS faces unprecedented challenges in communication and networking. As discussed in Section I, the sporadically available wireless links render it impossible to form a well-connected mesh network for end-to-end communications. Moreover, the basic principle of DTN is to employ ample buffering for alleviating the needs of immediate transmission. This, however, becomes challenging in FINDERS due to extremely small memory and intermittent computing capability of its constituent tags (which are responsible for data delivery). In this paper, we explore novel adaptive approaches for efficient data transmission in FINDERS by dynamically creating redundancy, managing data queues, and routing data packets. Next, we first introduce two basic parameters that are key to our proposed scheme, and then elaborate the communication protocol for FINDERS.

A. *Effective Delivery Capability (EDC)*

FINDERS is an opportunistic network, where the communication links exist with certain probabilities. It is essentially important to define an appropriate routing metrics to represent the key resource and to indicate the nodal delivery capability in such networks. To this end, several approaches have been investigated, and they are largely based on the nodal meeting probabilities (or more sophisticated parameters derived from nodal meeting probabilities) in contrast to end-to-end delay or path length adopted in conventional networks. For example, in [1], a node estimates its capability of transmitting data to a destination by direct contact probability. While simple, it has been shown inefficient [4], especially when the source is far away from the destination. Another approach is to let each node maintain the pairwise contact probability of any two nodes in the network. Then, the shortest path algorithm is applied to identify the path with the highest probability between a given source and destination pair [20]. Its drawback is the high overhead in storage and communications, which is unaffordable in FINDERS where the tags have extremely limited storage space. To lower overhead, [21] employs a transitive equation to update the routing metrics to each possible destination. In addition, several systems [2], [17] do not maintain any routing metrics by treating all nodes equally in routing. This is inefficient in many practical applications where nodes have nonequal probabilities to reach a destination.

In this research, we investigate effective routing metrics based on *cascaded* meeting probabilities between tags and readers and devise a mechanism that consumes insignificant storage space for maintaining and updating such metrics. Specifically, we have explored simple and efficient online estimation approaches to obtain the effective delivery capability (EDC) of tags and readers. Such online estimation usually involves tradeoffs among complexity, stability, agility, and required memory space. While many online estimators are available, the exponentially weighted moving average (EWMA) is one of the most effective schemes. It is simple, reacts effectively to small shifts, and is memory-efficient, requiring only constant space. Note that although EWMA is a well-defined method, it remains challenging to design an appropriate online updating mechanism to obtain EDC in FINDERS. In particular, since the tag cannot perform continuous computation, effective schemes must be devised for calculating EDC based on the necessary state information saved in its nonvolatile memory. More specifically, let ξ_i denote the EDC of Node i, which may be a reader or a tag. The GRs always have their $\xi_i = 1$. For other nodes, ξ_i is initialized as zero and updated upon an event of meeting with another node. Let t_o denote the time when the last event occurred and ξ_i was updated. If Node i meets Node j at time t_1, ξ_i is updated as follows:

$$\begin{cases} \xi_i = (1-\eta)^{\left\lfloor \frac{t_1 - t_o}{\Delta} \right\rfloor}[\xi_i] + \eta\xi_j, & \xi_j > \xi_i \\ \xi_i = (1-\eta)^{\left\lfloor \frac{t_1 - t_o}{\Delta} \right\rfloor}[\xi_i], & \xi_j < \xi_i \end{cases} \tag{2}$$

where $[\xi_i]$ is the parameter of Node i before it is updated, Δ is a constant that regulates the decrease of ξ_i, and $0 \leq \eta \leq 1$ is a constant weight to keep partial memory of historic status. Since the meeting event occurs between a tag and a reader only, the above computation can be performed by the reader, and then the updated ξ_i and t_o are written into the tag's nonvolatile memory. Clearly, this scheme requires time synchronization among readers, which can be achieved by various means with different degrees of accuracy. The impact of synchronization errors will be discussed in Section V-C.

Our studies show that EWMA accurately captures the special characteristics of FINDERS, with two interesting findings outlined below that make EDC an effective parameter to reflect the communication resource of individual nodes. First, the expectation of ξ_i converges to constant under statically distributed mobility. The proof is given in the Appendix. This result shows the stability of EDC, which is essentially important for it to serve as the routing parameter in FINDERS, ensuring efficient and stable routing. Second, EDC is an effective estimation of end-to-end data delivery rate. It exhibits excellent consistency and closely follows the actual delivery rate. Thus, when EDC serves as the routing metrics, data packets always flow from the nodes with lower delivery rate to the nodes with higher delivery rate, achieving efficient transmission in FINDERS.

B. *Fault Tolerance Degree (FTD)*

In a typical store-and-forward network, the packets are deleted from the buffer after they are transmitted to the next hop successfully. In FINDERS (like many other DTN networks), however, the reader may still keep a copy of the data after its transmission in order to maintain necessary redundancy for achieving a given success ratio. Therefore, multiple copies of the data could be created and stored by different readers and transported by different tags in the network. The fault tolerance is introduced to represent the amount of redundancy and to indicate the importance of a given data packet. We assume that each packet carries a field that keeps its fault tolerance degree (FTD). Let $\mathcal{F}_i^M$ denote the FTD of Packet M in the queue of Reader i. It is defined to be the estimated probability that at least one copy of Packet M is delivered to a GR by the tags in the network. When a packet is generated, its FTD is initialized to be zero and updated during transmissions. Let us consider a general scenario where Reader i transmits Packet M to Tag j. This transmission essentially creates two copies of Packet M, each of which shares part of the responsibility to deliver the data to GRs. An appropriate FTD needs to be assigned to them, respectively. More specifically, the packet transmitted to Tag j is associated with an FTD of $\mathcal{F}_j^M$

$$\mathcal{F}_j^M = 1 - \left(1 - \left[\mathcal{F}_i^M\right]\right)(1 - \xi_i) \tag{3}$$

and the FTD of the packet at Reader i is updated as

$$\mathcal{F}_i^M = 1 - \left(1 - \left[\mathcal{F}_i^M\right]\right)(1 - \xi_j) \tag{4}$$

where $\left[\mathcal{F}_i^M\right]$ is the FTD of Packet M at Reader i before the transmission. Equations (3) and (4) are derived based on the definition of FTD, previously introduced, to reflect the probability that the corresponding data is delivered. The update of FTD repeats each time upon a data transmission. In general, the more times the data packet is forwarded, the more copies are created, thus increasing its delivery probability. Accordingly, it is associated with a larger FTD. As will be discussed next, FTD is important in redundancy control and queue management.

C. Overview of Proposed Data Delivery Protocol

The data transmission protocol needs to be carefully devised based on EDC and FTD in order to achieve efficient network resource utilization. To this end, we propose a simple and effective scheme with six steps.

- Step I: *Data Acquisition*. Each reader periodically broadcasts query signals. When a nearby tag is detected, the reader first acquires the information on the tag, and then generates a new data packet to reflect the meeting event.
- Step II: *Data Queuing*. The reader inserts data packets into its queue, which is sorted according to FTD.
- Step III: *Update of EDC*. EDCs are calculated for the readers and tags according to (2).
- Step IV: *Receiver Identification*. One or multiple nearby tags will be identified for data transportation.
- Step V: *Data Duplication*. The data packets at the top of the queue are selected for transmission. For each such packets, two or more copies are created and processed (including the update of their FTDs).
- Step VI: *Data Transmission*. For each chosen tag, a package that includes the updated EDC of the tag and the selected data packets with updated FTDs is created and transmitted via a writing operation.

Next, we elaborate each of the six steps and several problems, tradeoffs, and alternatives involved therein. To facilitate our discussion, we start with Step II (Data Queuing), followed by Steps IV and V, and finally Steps I and VI. Step III is done straightforwardly according to (2).

1) Queue Management: Queue management is critical to the efficiency of FINDERS. It is meant to appropriately sort the data packets in the queue and determine which packet is to be sent and dropped. Our proposed queue management scheme is based on FTD, which signifies the importance of data packets. The packet with a smaller FTD is more important and should be transmitted with a higher priority. This is done by sorting packets in the queue in an increasing order of their FTDs. A packet is dropped on the following two occasions. First, when the queue is full, the packet with the highest FTD is dropped. Second, if the FTD of a packet is larger than a threshold, it is dropped, even if the queue is not full, aiming to reduce transmission overhead, since the data contained in the packet will be delivered to GRs with a high probability by other nodes that carry duplicate copies of the packet.

2) Receiver Identification: In Steps IV and V, the reader needs to identify one or multiple tags for data transportation and to determine the amount of duplication to be created. Without loss of generality, let us consider Reader i, which has a set of tags in its reading/writing range. Reader i first learns their EDCs and sorts them in a descending order. Let Ξ designate the set of sorted tags whose EDCs are greater than ξ_i. Next, Reader i simply picks Packet M at the top of its queue and writes M into the package to be transmitted to the first tag in Ξ. As we have discussed earlier, this transmission will create two copies of the data. Each of them is associated with an FTD calculated according to (3) and (4). Then, the copy at Reader i is put back into its queue at the appropriate position based on its newly updated FTD (i.e., $\mathcal{F}_i^M$), or dropped if $\mathcal{F}_i^M$ is larger than a threshold. This process repeats until all tags in Ξ have been occupied or the queue of Reader i becomes empty.

3) Reading and Writing Strategies: Data acquisition and transmission (i.e., Steps I and VI) are realized via reading and writing operations, respectively. Our studies have shown that the reading and writing strategies may significantly affect the communication efficiency of FINDERS. In general, three strategies are available for reading and writing, respectively, as outlined.

1) Limited-Read (LR): Only GRs (but not IRs) read data from tags.
2) Conservative-Read (CR): Any reader with a higher EDC than that of the tag can read data from the tag.
3) Aggressive-Read (AR): The reader reads data from any neighboring tags.
4) Limited-Write (LW): The reader writes data into a tag only if the tag is empty.
5) Conservative-Write (CW): The reader only writes its data to the tag which has higher EDC.
6) Aggressive-Write (AW): The reader writes data to any neighboring tags.

These reading and writing strategies yield a total of nine combinations as shown in Table I. Among them, “LR/LW,” “CR/LW,” and “AR/LW” are obviously inefficient because the reader is allowed to write data into a tag only when the tag is empty. Under this strategy, only the tags that commute directly

TABLE I
COMBINATIONS OF READING AND WRITING STRATEGIES

	Limited-Read (LR)	Conservative-Read (CR)	Aggressive-Read (AR)
Limited-Write (LW)	LR/LW	CR/LW	AR/LW
Conservative-Write (CW)	LR/CW	CR/CW	AR/CW
Aggressive-Write (AW)	LR/AW	CR/AW	AR/AW

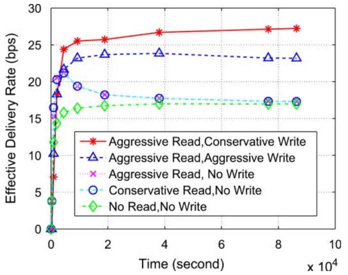
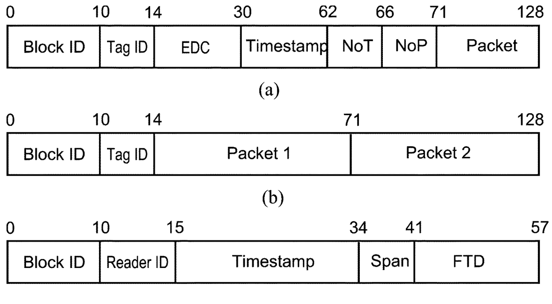


Fig. 6. Reading/writing strategies: AR/CW performs best.

TABLE II
MEMORY MAPPING MODES OF ALIEN C1G2 TAGS

Mapping-Mode	EPC (bits)	USER (bits)	Access-Pwd	Kill-Pwd
EPC96	96	0	32	32
EPC96USER64	96	64	0	0
EPC128	128	0	0	32



(a)

(b)

(c)

Fig. 7. Format of blocks and data packets: (a) Head tag; (b) Storage tag; (c) Data packet format.

between the GRs and the IRs are used for data delivery. This seriously limits the utilization of the tags, which are the critical transportation vehicles in FINDERS. We further observe that "LR/CW," "LR/AW," "CR/CW," and "CR/AW" are inefficient as well, as the reader may overwrite a tag without reading out its data in some or all scenarios, resulting in high data loss rate. Therefore, we finally arrive at two feasible approaches: "AR/CW" and "AR/AW." The difference between them lies in their writing strategies. Intuitively, "CW" is a better approach, as it generates less redundancy than that of "AW" and avoids unnecessary data duplications by controlling the data packets to flow from nodes with lower EDC to nodes with higher EDC. This has been verified by our simulations. As shown in Fig. 6 (with 1 GR, 4 IRs, and 500 tags and detailed setup deferred to Section VI), "AR/CW" achieves the highest data delivery rate, and thus is adopted for FINDERS by default.

IV. PROTOTYPE OF FINDERS

To demonstrate the feasibility and efficiency of the proposed data transmission protocol and to gain useful empirical insights into FINDERS, we have developed a prototype built upon the off-the-shelf RFID gears supplied by Alien Technologies. We have acquired five sets of Alien passive Class1Gen2 (C1G2) RFID systems that consist of five ALR-9900 readers with 10 circular polarized antennas and 1000 ALN-9540-WR Squiggle tags for our implementation and experiments. The reader is programmed by using the vender's Applications Development Kit (ADK). No sensors are integrated with the ALN-9540-WR tags. However, note that the implementation of our data transmission protocol remains the same with or without sensors. It affects traffic load only. In this section, we first discuss off-the-shelf passive tags and the expansion of their capacity, and then introduce the implementation of FINDERS.

A. Adaptive Expansion of Tag's Capacity

Most low-cost passive RFID tags have extremely limited capacity. For example, the Alien ALN-9540-WR "Squiggle" tag adopted in our research has total 160 bits memory only. Such limited capacity becomes the bottleneck for communication in FINDERS. It is thus highly desirable to support expansion of tag memory, adaptive to the communication needs in specific applications. To this end, we devise an adaptive expansion scheme based on off-the-shelf Alien ALN-9540-WR tags. Since the Alien tag is C1G2 standard compliant, our scheme can be applied to other standard passive tags as well.

A C1G2 tag comprises four memory banks: Reserved memory, Electronic Product Code (EPC) memory, Tag Identifier (TID) memory, and User memory. The data generated in FINDERS can be stored in EPC memory and User memory. The sizes of the memory banks vary depending on the configurations. For instance, the Alien C1G2 tags can be configured as one of three memory maps: EPC96, EPC128, and EPC96USER64, as summarized in Table II. Each memory map provides a different space allocation for EPC and User memory. At first glance, EPC96USER64 seems the best choice since we can use the entire 160 bits. However, in order to read the EPC96USER64-mapped tags, two APIs must be called, which will dramatically increase the delay to identify a moving object. Therefore, EPC128 mode is chosen instead in our implementation, where each tag can hold 128 bits of data. Note that here the EPC memory is used to store data instead of EPC code in typical RFID applications.

We introduce "blocks" for the expansion of tag capacity. A block is an integral unit attached to an object being tracked. Each block consists of a head tag and one or multiple storage tags. The tags within a block share the same Block ID [i.e., the first field of the tags shown in Fig. 7(a) and (b)]. Since the Block ID is essentially used to identify individual moving objects, its length can be chosen according to the estimated number of objects in specific applications. For example, we allocate 10 bits for Block ID in our implementation. The next field after Block ID is the Tag ID, which uniquely identifies each tag within a block. Its length is again dependent on the application (or more specifically, the maximum storage space needed for a block). Four bits are used for Tag ID in our implementation. The Tag ID field for the head tag is predefined to 1111, and 1110 for the first storage tag, 1101 for the second storage tag, so on and so forth. This design can effectively shorten communication delay. For example, to read nearby tags, the reader first sets its mask to 1111 for head tags only to reduce unnecessary collisions between head tags and storage tags. Once a head tag is identified, the reader can always use the Block ID and the predefined Tag ID to generate a unique mask for the next tag in the block, thus eliminating collision among storage tags in the same block.

Large blocks are desired to offer high network capacity, but note that it is not always beneficial to build large blocks. A block that consists of more tags usually requires longer delay for read/write operations. Such delay may lead to communication failures when the object being tracked moves quickly. Fig. 2(c) illustrates the read/write delay under different block sizes. As we can see, the communication between a reader and a block of tags can be completed within a very short delay, which grows linearly with block size. Writing results in slightly lower delay than reading because the reading operation needs to perform tag arbitration, while the IDs are already known in writing.

The format of data packet is illustrated in Fig. 7(c). In our application of object tracking, a data packet is created when a reader detects an object. The data packet records information of when and where the meeting event happens. As shown in Fig. 7(c), Block ID and Reader ID indicate which object appears around which reader; Timestamp and Span give the timing and the period during which the object stays with the reader. We allocate 19 bits for timestamp. With a granularity of minutes, the timestamp with 19 bits is long enough to accommodate continuous tracking for almost a year before it wraps around. The use of Span field can avoid generating many separate data packets when the object stays with the reader for a long time. Finally FTD is included for duplication control and queue management. All of these fields sum up to total 57 bits for a packet. If sensors are incorporated with tags or readers, Bits 1–41 can be used to accommodate sensor data.

With the capacity of 128 bits, each storage tag can hold two data packets (besides the Block ID and Tag ID fields) as shown in Fig. 7(b). The head tag can accommodate one data packet only, as it needs to include EDC, Timestamp, NoT, and NoP fields. The Timestamp field is to keep t_o that is needed for the update of EDC [see (2)]. NoT and NoP are the number of tags enclosed in the block and the total number of packets carried by the block, respectively. Since the Tag ID field has 4 bits, a block can include up to 16 tags that store up to 31 data packets. This offers great flexibility to configure the storage capacity of individual objects. How to determine appropriate block size for each object is application-dependent and will be discussed in Section V.

Note that a block of tags is an integral unit attached to an object. Many early discussions based on individual tags are applied to blocks hereafter.

B. Protocol Implementation

Here, we discuss the implementation details and related issues. To present a clear view of our implementation, we focus on the case where a block consists of a single tag only, while the general configuration with multiple tags in a block is a direct extension of the discussion that follows. We will point out how to make such an extension by the end of this section. Our prototype, test-bed and experimental results are all based on the general configuration.

To realize FINDERS, we have devised an implementation scheme based on the low-level operations defined in EPCglobal Class1Gen2 [10], the latest industrial standard adopted by the majority of current RFID systems. The implementation schemes discussed here are applicable to not only our test-bed, but also other standard compliant RFID gears. *Our implementation does not require any modification on the off-the-shelf tags or standard reader commands*. Only a small amount of code for calculation and decision making are added to the reader's program to support our proposed protocol.

The reader manipulates a set of commands (such as Select, Query, Req_RN, Read, Write, and ACK) to communicate with tags. A tag can be in one of the seven states depending on its interaction with the reader: Ready, Arbitrate, Reply, Acknowledged, Open, Secured, and Killed, which determine the reaction of the tag upon receiving a command from the reader [10]. Whenever a reader enters its active mode, it sends out a *Select* command with the parameter of *all*, which informs all nearby tags to respond to its following commands. Then, the reader initiates *Inventory* by transmitting a *Query* command, which contains a parameter of Q. Upon receiving *Query*, each tag picks a random number between $(0, 2^Q - 1)$, loads it into its slot counter, and enters the *Arbitrate* state. The reader then issues a series of *QueryRep* commands. Upon receiving each *QueryRep*, the tag decreases its counter by one. When the counter becomes 0, the tag backscatters its RN16 and moves to the *Reply* state. RN16 is another random number generated by the tag. It is used as a "handle" for the reader to access the tag during later communications. The reader then sends an *ACK* command with the received RN16 as its parameter. Once the tag receives the *ACK*, it enters the *Acknowledged* state and backscatters its EPC (along with other information not interested in FINDERS). Till now, the reader has successfully read the data on the first tag. The data packet(s) are inserted into the reader's queue according to their FTDs. The reader may continue sending *QueryRep* to acquire the replies of other tags. Note that collision may occur if more than one tag chooses the same random number. The tags involved in collisions are simply excluded from this round of communication. However, the collision probability can be controlled to a very low level as shown in [10]. This discussion is based on the tag arbitration scheme adopted in the standard. There are other enhanced or newly proposed approaches. We do not intend to investigate these alternatives in this paper. Any efficient arbitration scheme can be employed in FINDERS. In addition, since the readers are isolated, collisions due to simultaneous transmission of multiple readers are rare.

Next, the reader transmits its data by writing the packets into a set of suitable tags. As described in Section III-C, the reader calculates EDC, identifies the receiver, creates duplicate data copies, and then transmit them. More specifically, the reader sends a *Req_RN* command to the first identified tag, with the corresponding RN16 received earlier from this tag as the parameter. Upon receiving the *Req_RN*, the tag generates a new RN16, backscatters it, and enters *Open* state (where further write operation is allowed). The reader then issues the *Write* command, which includes the data and the just received RN16. If the tag receives the data correctly, it backscatters *ACK* with the RN16 as parameter (or an error code if the received packet is incorrect). Such a procedure repeats for the transmission to each tag.

The above discussions are based on the case where a single tag is attached to an object. When a block of tags is employed, the procedure is similar, with only minor modifications needed

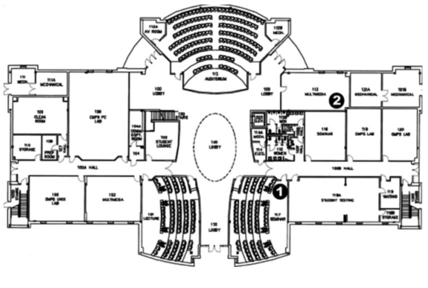


Fig. 8. Experiment setup (1st/2nd floor).

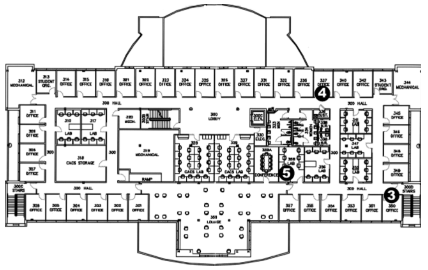


Fig. 9. Experiment setup (3rd floor).

to be made. Specifically, the *Select* command sent by the reader should include a mask of "1111" to allow only head tags to reply, instead of using the parameter of *all* that opens to all tags. Once a head tag is identified, the remaining storage tags in the block can be individually queried by using their Block IDs and Tag IDs as masks, which are thus collision-free. Similar changes are needed in writing, where the head tag is processed first, followed by other storage tags.

V. Experiments and Results

Based on our prototype, we have carried out test-bed experiments on campus. In this section, we first introduce our test-bed setup and then present the experimental results.

A. *Test-Bed Setting*

Our test-bed consists of five readers deployed in the building of the Computer Science Department, University of Louisiana at Lafayette. A reader is located at the entrance of a major classroom (Room 117) on the first floor (see Reader 1 in Fig. 8). Another reader (Reader 2 in Fig. 8) is installed at a large research lab (Room 228) on the second floor, which has a similar floor plan as the first floor and thus is not shown separately. Three readers are on the third floor, close to the doors of a small lab and two faculty offices (see Readers 3–5 in Fig. 9). Each reader is equipped with two side-by-side 6-dBi circular polarized antennas. Readers 1–4 serve as IRs, while Reader 5 is the GR. All readers issue queries at a frequency of one per second and have a maximum queue size of 500 packets.

Thirty-eight volunteers had participated in our experiments, including faculty, senior Ph.D. students (who do not have classes), M.S.-level graduate students (who go to classrooms regularly), and undergraduate students. Each participant carries a badge (a typical plastic badge used in conferences), which is embedded with a block of ALN-9540-WR Squiggle tag(s). For simplicity, we employ two types of blocks only: a block with a single tag and a block with four tags. Based on the estimated traffic load, we distributed 30 single-tag badges and 8 four-tag badges. Our experiment lasted 9 days, starting from Wednesday 6:00 p.m. to Thursday of the next week.

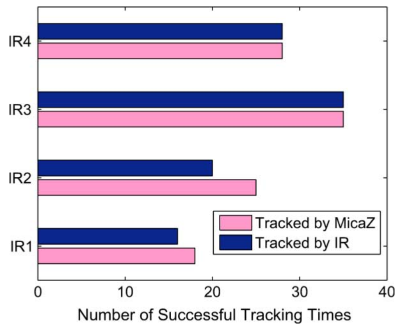


Fig. 10. Tracking efficiency.

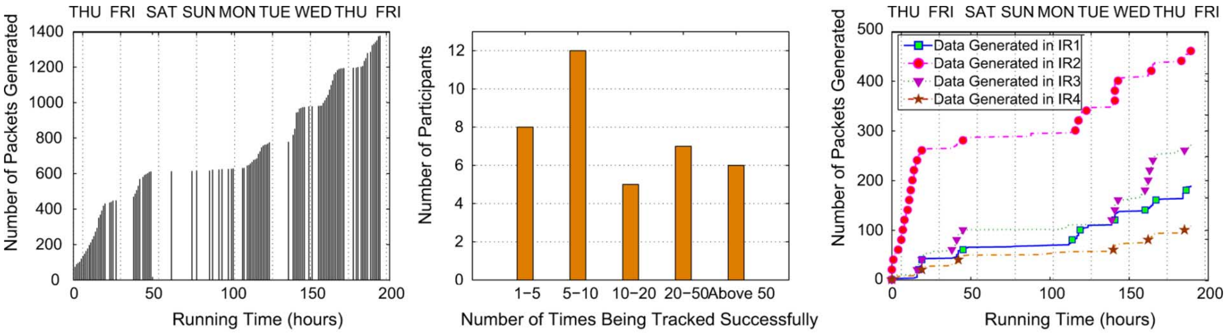
B. *Experimental Results*

We evaluate FINDERS on two aspects: tracking efficiency and data delivery efficiency.

1) Tracking Efficiency and Data Generation: FINDERS is designed for wildlife tracking. Therefore, tracking efficiency is a key performance metrics to be evaluated. Generally, tracking failures can be classified into two categories: false positive and false negative. The former is rare in our experiments because irrelevant tags do not exist in our test-bed. Therefore, we focus on the latter, which is defined to be the ratio of actual events being successfully tracked, or the tracking success rate. While our experimental trace naturally provides the number of successfully tracked events, it is difficult to find the "actual" number of events occurred in the experiment without active devices. To this end, we carried out a supplemental experiment for 7 days with six participants, where each participant carried not only a block of RFID tags, but also an active Crossbow MICAz mote. We installed a beacon mote at each IR, which kept broadcasting its ID every 1 s. The received beacon signals were used as ground truth to evaluate the tracking efficiency of FINDERS. As shown in Fig. 10 (based on the supplemental experiment), the readers exhibit a high tracking success rate in general.

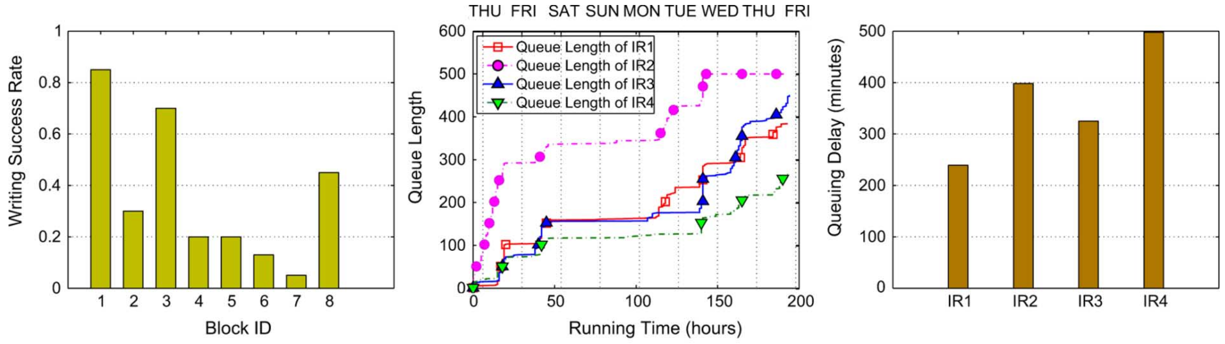
Fig. 11(a) illustrates data generation, which varies from day to night and from weekdays to weekends. The x-axis indicates the running time of our experiments, with the hours shown at the bottom and days at the top. The dotted lines divide the time span into days. Generally, data generation reflects the regular moving activity characteristics of the objects. More data are generated during daytime than night (see the denser bar-lines and sharper increase at the middle part of each day interval, roughly from 9 a.m. to 11 p.m.). More data are generated during weekdays than weekends (see the sparse bar-lines and flat interval on Saturday and Sunday). An unexpected high data generation rate occurred during the first Wednesday night (see the beginning of the graph). Our investigation shows that a student frequently visited Room 228 (where IR 2 was installed) to look for his teaching assistant (TA) during that night, yielding a large number of packets.

The density distribution of the tracked events is illustrated in Fig. 11(b). The results match the activities of the tracked objects very well. For example, the graduate students whose laboratories locate on the third floor usually have more events tracked because they spent more time in the building and frequently



(a) (b) (c)

Fig. 11. Experimental results on data generation: (a) Time distribution; (b) Density distribution; (c) Location distribution.



(a) (b) (c)

Fig. 12. Statistics of experimental results at individual readers: (a) Success write rate on detected tags; (b) Run-time queue length; (c) Average queuing delay.

walked around from the first floor to the third floor, while the undergraduate students only used classrooms on the first floor and usually left the building after their classes.

Fig. 11(c) shows the location dependency of data generation, i.e., the number of packets generated by each reader. This can also serve as an active index for the IRs because more objects visit it, so more data packets will be generated by the IR. For example, IR 2 exhibits the highest data generation rate because it is located at a large research lab that houses multiple active graduate students and a set of equipment frequently used by a group of students. Moreover, the TA of a large class holds office hours in this lab and regularly receives visits from his students for question answering. Overall, we observe that data generation is highly location-dependent.

2) Data Delivery Efficiency: The data delivery performance is mainly determined by three factors: data generation rate, mobility pattern of the objects, and the number of packets each object can carry (i.e., the block size). In the following discussions, we first investigate the performance of individual readers and then the end-to-end data delivery.

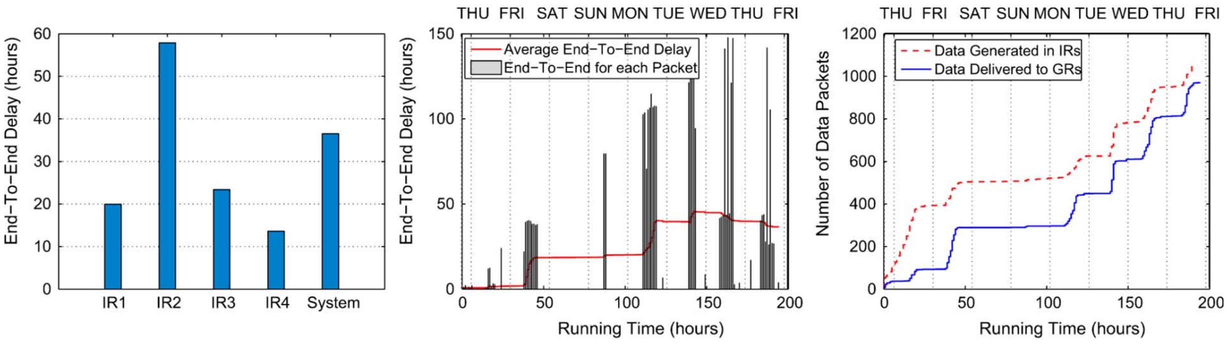
Performance at Individual IRs: Data delivery in FINDERS relies on the transportation of tags. It is thus essentially important to successfully write into the tags the data to be delivered. To this end, we have focused on the eight badges, each of which embeds a block of four tags. We study the ratio of successful writing once a block is read by an IR. As shown in Fig. 12(a), the difference among those badges is dramatic. The success rate ranges from 0.85 to 0.05. After individual interviews with the participants, we discover that their moving habits have strong impact on the successful write probability. The faster one moves, the lower probability it is written successfully. As we have discussed for Fig. 2(a) and (b), there is a sharp decrease of write success rate when the range exceeds 6 ft, and the problem is exacerbated when the object is mobile. If the object moves too quickly, it might be detected (i.e., read), but cannot be used to transport any data packets because of the low chance to be written. This experiment suggests that the objects that have lower moving speed are more suitable for carrying large blocks.

Fig. 12(b) shows the run-time queue length of each IR. For all IRs, the queue length increases with the time until the queue is full (i.e., reaches 500). The patterns of the curves are similar to but higher than those in Fig. 11(c) because the queue length indicates not only data generation, but also interactions between readers. In our experiment, IRs 2 and 3 are key readers that bridge the participants from different groups and with different social activities. They naturally hold more packets in their queues.

Another important metrics is queuing delay. Generally, the queuing delay is affected by both packet arrival and service at the queue. The former depends on how many new packets are generated and how many existing packets are received, while the latter is determined by the number of writable tags the reader can access. As shown in Fig. 12(c), IR 4 exhibits the highest queuing delay (over 8 h). The reason is that IR 4 is less active than other readers. It has a lower chance to meet tags, and thus its data packets reside in the queue for a longer time before being transmitted.

Network-Wide End-to-End Performance: From the system's perspective, the most important performance metrics are the end-to-end delay and end-to-end delivery rate.

Fig. 13(a) illustrates the average end-to-end delay for packets generated at different IRs. First of all, we observe that the packets generated at IR 4 have the lowest end-to-end delay.



(a) (b) (c)

Fig. 13. Experimental results of end-to-end data delivery delay: (a) Location dependency; (b) Time dependency; (c) Data generation versus delivery.

This appears to conflict with the results in Fig. 12(c) at first glance. However, it is in fact reasonable. IR 4's location is close to the GR (i.e., Reader 5 shown in Fig. 9). Thus, most of its packets were delivered via a single-hop transmission, i.e., by tags moving from IR 4 to the GR directly. This is evident by the results that show only slightly longer end-to-end delay than the individual queuing delay at IR 4. Other IRs usually need multihop transmissions, and consequently the data packets reside in multiple queues before they arrive at the GR. Thus, the end-to-end delay becomes longer. IR 2 exhibits the longest delay because of the unexpected high volume of data generated at the first Wednesday night (as discussed in Section V-B.1), and many of those data cannot be delivered during the following Saturday and Sunday. The network-wide average delay is between 1 to 2 days, which is acceptable in such a resource-constrained delay-tolerant network.

In addition to its location dependency, we have also investigated the density and distribution of delay for individual data packets. As we can see from Fig. 13(b), end-to-end delay keeps increasing till Wednesday of the second week. This is mainly due to the unusual high data generation at IR 2 during the night of the first Wednesday. Many date packets were piled up at IR 2's queue first and other IRs' queues later with the propagation of the packets. Only a limited number of data packets can be delivered to the GR in each day. Moreover, very few packets were delivered on Saturday and Sunday due to low activity of students and faculty. As a result, many packets had to stay in the readers' queues during the weekend, resulting in further increased delay when they were transmitted on Monday and Tuesday. Most "old" packets had been delivered by Tuesday evening. Therefore, packets have significantly lower delay on the following Wednesday and Thursday, except a few leftovers. The network-wide average delay reaches about 1.5 days by the end of Thursday.

Fig. 13(c) depicts the number of packets generated and delivered at each time instance. The two curves show similar pattern because data generation and transmission are highly correlated (i.e., a meeting even between a block of tags and a reader results in both data generation and transmissions). The shift between the two curves indicates data delivery delay.

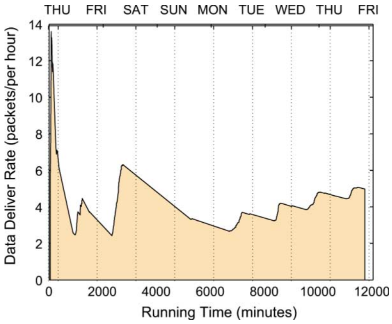


Fig. 14. Overall end-to-end delivery rate.

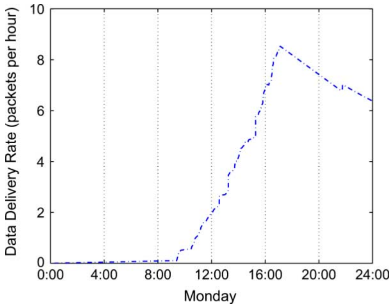


Fig. 15. End-to-end delivery rate on Monday.

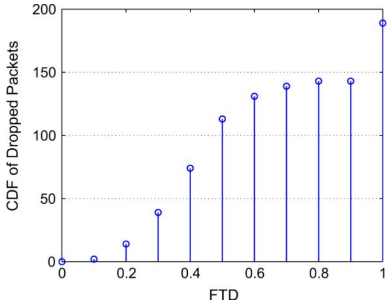


Fig. 16. Cumulative distribution function (CDF) of the FTD of dropped packets.

Figs. 14 and 15 show the overall data delivery rate and its blowup on Monday, respectively. The data delivery has a surge at the beginning (see Fig. 14) because data queues are largely empty, thus resulting in a high delivery rate. Thereafter, the average delivery rate increases during daytime and decreases at night, especially after midnight. Fig. 15 illustrates details on a particular day. Most packets were delivered between 9 a.m. and 5 p.m., which closely follows the working hours and activity patterns of the participants.

Finally, Fig. 16 illustrates the distribution of FTD of the dropped packets. The bigger the FTD, the higher probability it is dropped. The results largely match our estimation.

TABLE III
POWER CONSUMPTION MEASUREMENT

Scanning Frequency (Hz)	Theoretical Estimation (W)	Measured Data (W)	Supply Time (Hours)
1/60	12.3	13.0	35.2
1/10	13.6	15.0	31.7
1/5	15.0	17.0	28.8
1/3	16.5	18.0	26.2
1	21.0	22.0	20.6

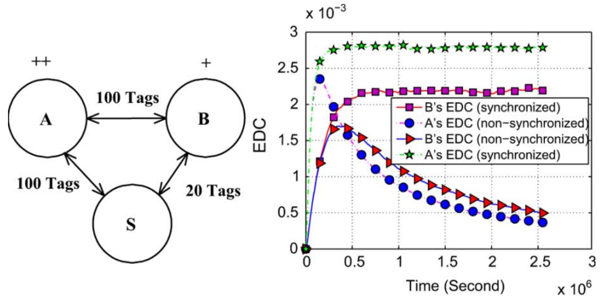
C. Further Discussions

In addition to the results presented, we would like to discuss two other issues not addressed in our experiments, but which could become potential problems in field implementation.

1) Power Supply: While the passive tags do not need power supply, the readers (especially IRs) usually operate on batteries charged by solar panels in field deployment. It is critical to ensure the battery's capacity is sufficient to support continuous function of the reader during the period when its solar charger is off (e.g., at night). To this end, we have investigated the power consumption of readers under various duty cycles. Our results show that a typical car battery with a suitable solar charger is capable to provide power supply for FINDERS.

When the reader is in its working mode, it can either be idle or initiate a scanning to communicate with nearby tags. The latter consumes more energy than the former does. Apparently, the scanning frequency affects both energy consumption and network performance. First of all, the scanning frequency is a key factor to determine the communication opportunities between readers and tags (and accordingly the effective network capacity). The optimal scanning frequency is highly application-dependent. For example, it depends on how fast a target animal passes by a reader. The slower the animal, the lower the frequency (as required to avoid missing any contacts). It also depends on the habitat of animals that may be more active during night than daytime, or vice versa. A higher scanning frequency yields higher delivery rate and lower delivery delay. However, excessive increase of scanning frequency (e.g., to higher than 1 Hz in our experiments) does not further improve the network performance in most practical applications. On the other hand, the scanning frequency also dictates power consumption, which can be estimated by the product of voltage, current, and the power factor, whose values under idle and communication states are usually given in the specification. Alternatively, the energy consumed by readers can be measured by the power meter. Table III shows the results for our Alien RFID reader, based on both theoretic estimation and actual test-bed measurement. A typical car battery (used in our experiments) with a capacity of 12 V × 60 Ah can supply from 35 to 20 h when the scanning frequency increases from 1/60 to 1 Hz, which is sufficient in a wide range of applications. The battery can be supplied by a solar charger. A typical solar cell generates 15 mW/cm^2 under direct sunlight. With a minimum of 6 h of daytime, the total energy harvesting is 90 mWh/cm^2 per day. Therefore, a solar panel of less than 1 m^2 is sufficient for each IR.

2) Synchronization Among Readers: As we have discussed in Section III-A, time synchronization among the readers is required to update EDC according to (2). Any inaccuracy in time synchronization will lead to errors in its calculation. For example, if an IR has a clock faster than the clocks of other readers, its EDC will be less than its true value because t_1 in (2) becomes bigger due to the fast clock. Similarly, a slow clock leads to a higher EDC. A small error of EDC does not hurt the network performance as long as the order of the EDCs remains the same (i.e., the node with higher true EDC still maintains a higher EDC although with errors). When the errors are high enough to change the orders, however, performance degradation can be observed. An example is given in Fig. 17(a) with two IRs (A and B) and one GR (S). There are 100 tags moving between A and B and A and S, respectively, and 20 tags moving between B and S. If all clocks are accurate, A maintains a higher EDC than that of B. Therefore, data packets flow from B to A. However, if S's clock is accurate, but B's clock is faster than S, and A's clock is faster than B's, the EDC of A will eventually become smaller than that of B [see Fig. 17(b)]. Accordingly, packets will incorrectly flow from A to B, leading to lower data delivery rate.



(a) (b)

Fig. 17. Illustration of synchronization errors: (a) Model; (b) EDC.

In practice, all readers are synchronized to the standard time before they are deployed. Such synchronization may be realized via the network time protocol (NTP), which can achieve an accuracy of 10 ms or lower [22]. Once a reader (especially an IR) is deployed, however, it is difficult to keep the synchronization because it is isolated from the time servers. Note that the tag, though it communicates with readers (IRs and GRs) from time to time, does not help synchronization because its intermittent computation capability does not allow it to maintain a clock. Hereafter, the clock skew will lead to bigger and bigger errors in synchronization and eventually result in the problem discussed. However, fortunately, this process usually takes a very long time. Considering the typical precision of a CPU clock, the skew may range from 10^{-6} to 10^{-20} s per second. Therefore the system can maintain its highly efficient data delivery for years without the need of resynchronization, which is sufficient in most applications.

VI. SIMULATION RESULTS

Besides the experiments discussed, extensive simulations are indispensable for a comprehensive evaluation of FINDERS with a large number of readers and tags, which are not practical to build in labs.

Our simulated network consists of 25 readers uniformly deployed in a bounded area. A number of tags are distributed in the network and move according to power-law distribution, which is deemed as one of the most realistic mobility models for delay-tolerant mobile networks [23]. More specifically, the network area is partitioned into cells. Each tag is randomly assigned a home cell. Tag i makes its decision to move into one of the neighboring cells or stay at the current cell in every time

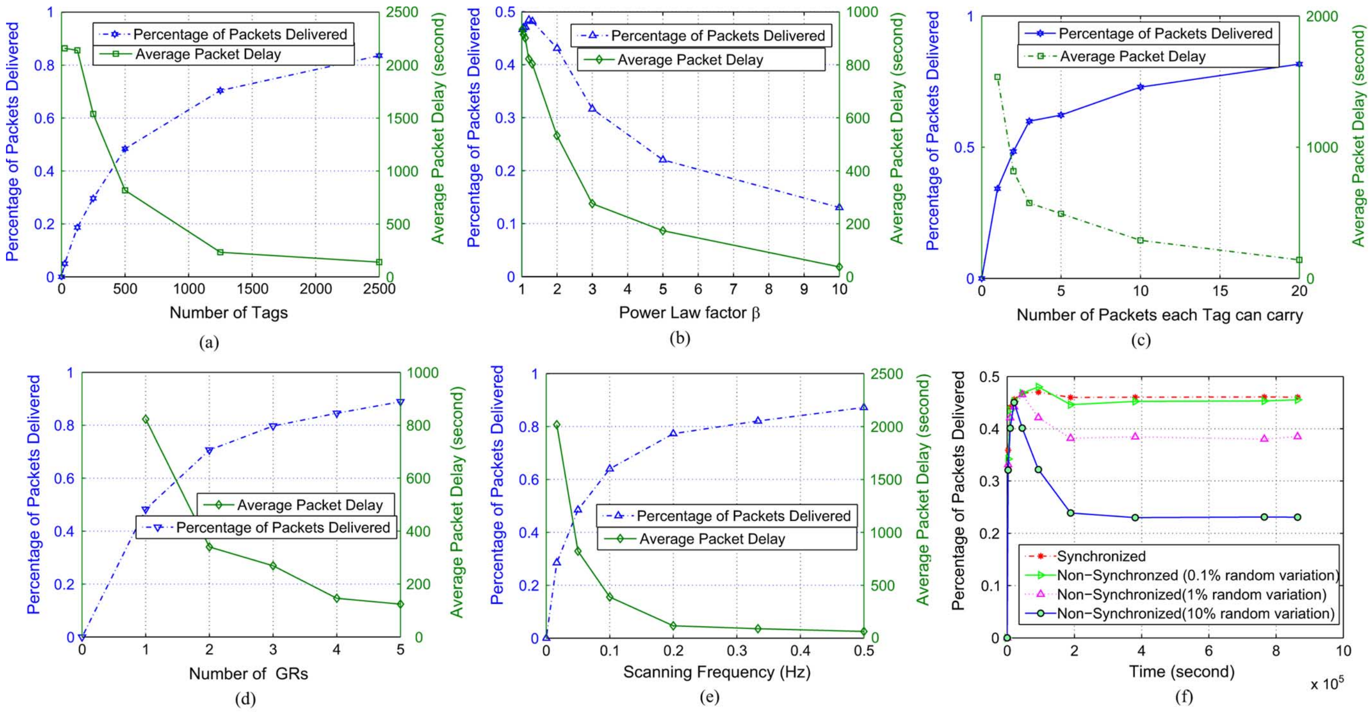


Fig. 18. Impact of several key parameters on the performance of FINDERS: (a) Number of tags; (b) Tag mobility; (c) Tag capacity (m); (d) Number of GRs; (e) Scanning frequency; (f) Synchronization errors.

slot. For instance, if it is currently in Cell 0, it may move into one of four adjacent cells (i.e., Cells 1–4) or stay in the current cell (Cell 0) in the next time slot. Its probability to be in Cell x in the next time slot is $P_i(x|0) = P_i(x)/\sum_{z=0}^{4} P_i(z)$, where $x = 0, 1, 2, 3, 4$, $P_i(x) = k_i(\frac{1}{d_i(x)})^{\beta}$. k_i is a constant, and β is the exponent of the power-law distribution, respectively. $d_i(x)$ denotes the distance from Cell x to the home cell of Tag i. The summary of default parameters used in our simulations follows. One GR and 500 tags are simulated. $\beta = 1.2$. The tag capacity is $m = 2$ packets. The data arrival rate at a reader is one packet per 10 time slots. We are mainly interested in data delivery rate and average packet delay.

We first study the performance of FINDERS by varying several parameters related to tags. As shown in Fig. 18(a), higher performance is achieved when the number of tags increases from 25 to 2500 since more tags result in more communication opportunities (i.e., higher service rate). However, the gain becomes smaller when the tag density is already high because additional tags will carry unnecessarily duplicated data and thus do not further improve network performance. The power-law factor β determines the mobility pattern of tags. With the increase of β, each tag has a high chance to stay at its home location, which means less mobility. Therefore, the delivery rate decreases. Meanwhile, although it is anti-intuitive to observe the delay decreases as well, it is in fact reasonable because the average delay is calculated based on the delivered data packets only. When β is large, most packets are delivered by the tags whose home locations are close to the GRs and thus experience short delay. In addition, as shown in Fig. 18(c), increasing the tag capacity can improve the performance of FINDERS by allowing more packets be read from and written into the tag. Fig. 18(d) illustrates the results with the number of GRs increasing from 1 to 5. Clearly, more GRs improve the chance for the tags to meet them, thus achieving higher data delivery rate and lower packet delay. We have also studied the impact of readers' scanning frequency [see Fig. 18(e)]. As the scanning frequency increases from 1/60 to 1/2 Hz, the readers can grasp more opportunities to communicate with tags for data delivery, leading to higher throughput and lower delivery delay. Finally, the impact of synchronization errors is shown in Fig. 18(f), which verifies our earlier discussions, i.e., no obvious performance degradation for over a year when the clock screw is less than 0.1%.

VII. CONCLUSION

We have developed a *featherlight information network with delay-endurable RFID support* (FINDERS), composed of passive RFID tags that are ultralight, durable, and flexible, without power supply for long-lasting applications. It expands the use of RFID gear for wireless sensor network construction, aiming to find events of interest and gather aggregate information. We have proposed an efficient data transmission protocol for FINDERS with effective techniques to address such challenges as sporadic wireless links, unique asymmetric communication, intermittent computation, and extremely small memory of tags. We have developed a prototype and established a test-bed for empirical studies. Our experiments have involved 38 participants and lasted for 9 days, yielding interesting results that offer valuable insights into RFID-based delay-tolerant mobile sensor networks and provide useful practical guidance for the setup of FINDERS systems.

APPENDIX

Lemma: The expectation of ξ_i converges to constant under statically distributed mobility.

Proof: Let $\tilde{\xi}_i$ denote the limit of the expectation of ξ_i, i.e., $\tilde{\xi}_i = \lim_{t\to\infty} E(\xi_i(t))$. The result is clearly true for GRs, with $\tilde{\xi}_{gr} = 1$. We now focus on tags and IRs only. Let p_{is} denote the probability that Tag i meets the GRs, and p_{ij} the contact probability between Tag i and IR j. Note that the contact probability between two tags or two IRs is zero.

We first show that if Node i has the highest $\tilde{\xi}_i$ among all non-GR nodes in the network, $\tilde{\xi}_i = p_{is}$ in the steady state. The node with the highest $\tilde{\xi}_i$ must be a tag because an IR has no direct contact with GRs. Based on the EDC estimation mechanism given in (2), the node with the highest $\tilde{\xi}_i$ increases its ξ_i only when it meets the GRs. Considering a time slot t, Node i has probability of p_{is} to meet the GRs and $(1-p_{is})$ to decrease its EDC. Let $\xi_i(t)$ be the EDC value in the time slot t. According to (2), we have

$$E(\xi_i(t)) = (1-\eta)^t \xi_i(0) + (1-\eta)^{t-1}\eta p_{is} + \cdots + \eta p_{is}.$$

Letting $t \longrightarrow \infty$, we can obtain the limit of $E(\xi_i(t))$

$$\tilde{\xi}_i = \lim_{t\to\infty} E(\xi_i(t)) = \eta p_{is}\frac{1}{\eta} = p_{is}$$

which is constant under statically distributed mobility.

Now we consider an arbitrary node. Since Node i increases its $\tilde{\xi}_i$ only when it meets the GRs or another node with higher $\tilde{\xi}_j$, we can derive $E(\xi_i(t))$ as follows, in a way similar to the discussions above

$$E(\xi_i(t)) = \sum_{j}^{\tilde{\xi}_j>\tilde{\xi}_i} p_{ij}\eta(1+\cdots+(1-\eta)^{n-1})E(\xi_j(t)).$$

Taking limit on both sides of the above equation, we arrive at

$$\tilde{\xi}_i = \lim_{t\to\infty} E(\xi_i(t)) = \sum_{j}^{\tilde{\xi}_j>\tilde{\xi}_i} p_{ij}\tilde{\xi}_i.$$

Therefore, $\tilde{\xi}_i$ can be safely obtained for every node through a recursive process, and all of them are constants. ■

REFERENCES

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Proc. 10th Annu. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2002, pp. 96–107.

[2] T. Small and Z. J. Haas, "The shared wireless infostation model—A new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proc. ACM MobiHoc*, 2003, pp. 233–244.

[3] "DeerNet," [Online]. Available: http://www.wu.ece.ufl.edu/projects/DeerNet/DeerNet.html

[4] Y. Wang and H. Wu, "DFT-MSN: The delay fault tolerant mobile sensor network for pervasive information gathering," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.

[5] 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. SenSys*, 2007, pp. 87–101.

[6] E. Miluzzo, N. D. Lane, K. Fodor, R. A. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application," in *Proc. SenSys*, 2008, pp. 337–350.

[7] P. Klasnja, B. L. Harrison, L. LeGrand, A. LaMarca, J. Froehlich, and S. E. Hudson, "Using wearable sensors and real time inference to understand human recall of routine activities," in *Proc. 10th Ubicomp*, 2008, pp. 154–163.

[8] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. SenSys*, 2008, pp. 323–336.

[9] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay tolerant network architecture," draft-irtf-dtnrg-arch-02.txt, 2004.

[10] *UHF Class 1 Gen 2 Standard*, v. 1.1.0, 2005.

[11] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. SIGCOMM*, 2003, pp. 27–34.

[12] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," in *Proc. Workshop Sensor Netw. Protocols Appl.*, 2003, pp. 30–41.

[13] D. J. Goodman, J. Borras, N. Mandayam, and R. Yates, "Infostations: A new system for data and messaging services," in *Proc. IEEE Veh. Technol. Conf.*, 1997, pp. 969–973.

[14] B. Burns, O. B. Brian, and N. Levine, "MV routing and capacity building in disruption tolerant networks," in *Proc. IEEE INFOCOM*, 2005, pp. 398–408.

[15] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, base stations, and meshes: Enhancing mobile networks with infrastructure," in *Proc. ACM MobiCom*, 2008, pp. 81–91.

[16] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proc. ACM SIGCOMM*, 2007, pp. 373–384.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[18] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proc. ACM SIGCOMM WDTN*, 2005, pp. 244–251.

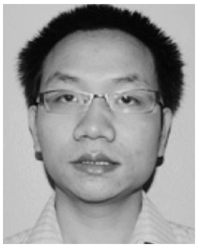
[19] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. ACM MobiHoc*, 2004, pp. 187–198.

[20] J. Ghosh, J. Philip, and C. Qiao, "Sociological orbit aware location approximation and routing (SOLAR) in MANET," *Ad Hoc Netw.*, vol. 5, no. 2, pp. 189–209, 2007.

[21] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Proc. Workshop Service Assurance Partial Intermittent Resources*, 2004, pp. 19–20.

[22] D. Mills, "Network Time Protocol version 4 reference and implementation guide," Dept. Elect. Comput. Eng., Univ. Delaware, Newark, Tech. Rep. 06-06-1, 2006, .

[23] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *Proc. ACM SIGCOMM05 WDTN*, 2005, pp. 276–283.



Zhipeng Yang (S'08) received the B.S. and M.S. degrees in computer science from Tianjin University, Tianjin, China, in 2001 and 2004, respectively, and has been pursuing the Ph.D. degree in computer science at the Center for Advanced Computer Studies, University of Louisiana at Lafayette, since 2007.

From 2004 to 2006, he was a Software Engineer with Nortel and Lucent China. His current research interests include delay-tolerant networks, radio frequency identification (RFID) systems, and wireless sensor networks.

Hongyi Wu (M'02) received the B.S. degree in scientific instruments from Zhejiang University, Hangzhou, China, in 1996, and the M.S. degree in electrical engineering and Ph.D. degree in computer science and from the State University of New York (SUNY) at Buffalo in 2000 and 2002, respectively.

He is currently an Associate Professor with the Center for Advanced Computer Studies, University of Louisiana at Lafayette. His research spans delay-tolerant networks, radio frequency identification (RFID) systems, wireless sensor networks, and integrated heterogeneous wireless systems.

Dr. Wu received the National Science Foundation (NSF) CAREER Award in 2004.