A Robust Encounter Registration Protocol for Mapping the Social Networks of Wild Animals

Tong Shen*†, Yuexuan Wang*†, Yong Zhang‡, Sivan Toledo§, Heming Cui†, Francis C.M. Lau†
*Zhejiang University, Hangzhou, P.R. China
†The University of Hong Kong, Hong Kong, P.R. China
‡Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, P.R. China
§Tel-Aviv University, Tel Aviv, Israel
Email: shent0505@outlook.com, amywang@zju.edu.cn, zhangyong@siat.ac.cn, stoledo@tau.ac.il, heming@cs.hku.hk, fcmlau@cs.hku.hk

Abstract—Encounters between wild animals mark important events of the same or different species, such as mating, predation, and disease contraction, and they define the social structure of animal groups. Several real-world systems designed to record such encounters have been implemented in the wildlife-research community. In these systems, energy-restricted short-range radio devices are attached to wild animals, transmitting identification packets periodically, receiving and recording packets from others. However, non-trivial protocols for this problem have not been studied, and neither did the fundamental tradeoff between minimizing the power consumption and reducing the encounter latency. The key challenge is the uncertain mobility and herd characteristics of wild animals and our dominant strategy is to design a mechanism to identify the existence of an encounter in real-time.

In this paper, we propose a robust encounter registration protocol. Our protocol consists of two processes: detect other tags with low energy consumption; and make connections with the detected tags to record each other's ID. We present extensive simulations and experiments that verify the scalability of the protocol. Also, we present explicit animal models of mobility and evaluation results show the strengths of our protocol in mobile wildlife environment. To the best of our knowledge, this is the first detailed description of a encounter-registration protocol for real wireless mobile networks.

I. Introduction

Collecting detailed information about wild animals [1], [2] remains a significant technical challenge that limits the ability of ecologists to study the interactions between wildlife and their environment. One tool that emerged a little over a decade ago and has been gaining significance is encounter detection and logging [3]–[5].

Encounter-registration system [6]–[8] is a kind of wildlife tracing systems, consisting of radio devices called tags attached to wild animals (and sometimes also to fixed positions and livestock), as depicted in Fig. 1. The radios transmit identification packets periodically and listen to such packets from other tags, recording data about received packets in persistent memory in the tag. The radios are typically configured for short-range communication by using low transmitting power and by using high data-rates (both limit the signal-to-noise ratio at the receiver). Since the radios are configured for short-range communication, receiving a packet implies the transmitting tag is in close proximity to the receiving tag. Recent systems record each packet with received signal strength indication (RSSI) [9], helping to estimate the distance between the transmitter and receiver. This is the main goal of these systems: to log close-proximity events between two or more animals. The logs are downloaded either by physically retrieving the tags, or by remotely downloading to base-stations placed in locations that the animals pass by frequently.

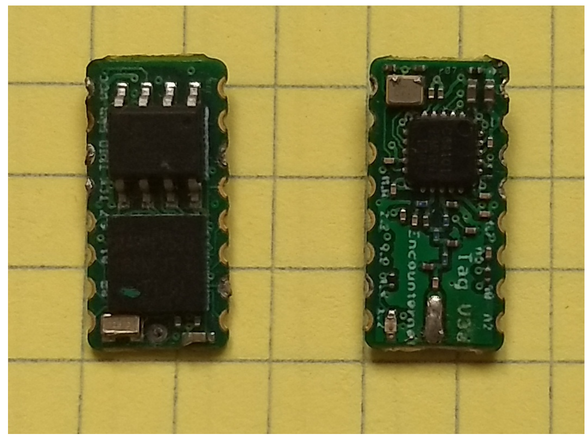


Fig. 1. Printed circuit boards for Encounternet tags.Lines are 5mm apart. Complete tags require the addition of miniature batteries, whip antenna, and coating. The small batteries make effective protocols important.

Such systems have gained popularity among ecologists because the tags are relatively inexpensive and can be very small. In addition, their deployment does not require much infrastructure in the field. More importantly, these systems help to study the close-proximity encounters of wildlife, which are key aspects of many significant events in the life of animals: mating, predation, disseminating diseases, etc.

Unfortunately, there is no robust encounter registration protocol for tag radios to transmit and receive packets efficiently. Worse, existing protocols for relevant problems (e.g., neighbor discovery problem [10]) fail because high efficiency (when an encounter happens) and low-energy consumption (when there is no encounter) can be mutually exclusive. The key challenge is the uncertain mobility and herd characteristics of wild animals. There is variability among species in their movement and interaction behavior. This issue adds a great deal of difficulties to solve two challenging protocol-related problems. One problem is to minimize the power used for listening to other tags. All of the systems developed so far use low-power integrated UHF transceivers, and the power consumed by these while receiving is comparable to the power that they use to transmit. The other protocol-related problem is interference. If many tags transmit simultaneously, receiving tags may fail to reconstruct valid packets. Many species of animals, including many species of birds and bats, roost together, so encounters of tens or hundreds of individuals are not necessarily rare.

Our fundamental observation is that it is a waste of energy if an animal is not encountering with others while still keeps the tag working frequently. Thus it is reasonable for a tag to increase the working frequency of its radio when encounter happens; otherwise it keeps the radio in a low-power mode. To handle this issue while taking the uncertain mobility of animals into account, the dominant strategy in this paper is to design a mechanism to identify the existence of an encounter in real-time.

In this paper, we propose a robust protocol for encounter registration, addressing these problems mentioned above systematically and methodically. In our protocol, we design two stages for the tags, namely detecting stage and connecting stage. In the detecting stage, a tag works a fraction of time in order to save energy, and transmits a beacon periodically to detect whether an encounter is happening at the moment. When detected an acknowledgement, the tag switches to the connecting stage and increases the transmission frequency in order to establish links with other tags. To deal with interference happening in the connecting stage, a tag adaptively adjusts its transmitting probability: it increases the probability when the channel is detected idle and reduces the probability when interference is detected.

The contributions of the paper are summarized as follows:

1) We propose a robust encounter registration protocol with detailed theoretical analysis.
2) We conduct extensive simulations and experiments. Our evaluation results show that, compared to baseline methods, our protocol achieves better scalability.
3) We present explicit animal models of mobility and evaluation results show that, compared to baseline methods, our protocol has better performance, regarding three species animals.

Our protocol can be widely applied to various wireless mobile networks, e.g. mobile Ad hoc network management.

The remainder of the paper is organized as follows. The next section highlights the related works. Section III presents the system model and basic definitions. We present the concrete protocol design in Section IV. We carry out simulations for protocol validation in in Section VI and experiments for model validation in Section VII. The paper is concluded in Section VIII.

II. Related Work

A. Encounter-registration system

The first encounter-registration system was developed by a company called Sirtrack over a decade ago [1]. The tags are placed on collars attached to wild mammals. They are fairly heavy, weighing 45–450g, depending on the size of the battery. This system has been commercially used, for example, to study the possibility of disease transmission between cattle and wild badgers [4].

The next generalization is a system called Enounter-net [2], [7], which has been available commercially for a few years. The key innovation of Encounternet was tags' weight: tags used a tiny printed-circuit board and could be powered by miniature batteries, allowing tags weighing 1.3g and up to be manufactured. Clearly, the small batteries restrict the life-span of tags and increase the importance of effective protocols, especially with respect to low-power operation. The small size of Encounternet tags has enabled the study of interaction of small species, including small birds [6] and freshwater fish [3].

More recently, prototype Encounternet-like tags have been developed and tested on bats [5], [8]. These tags include a wake-up receiver, which enables nano-power listening without maintenance of inter-tag synchronized clocks. These tags do not appear to have been commercialized or widely-used.

B. Protocol for encounter registration

Unfortunately, none of the papers that describe these systems and ecological research give any details on the MAC-layer protocol that is used, nor on how the receiver is duty cycled. Existing methods to encounter registration problem are mainly based on fixed transmitting probability [2], [7] (i.e., agents transmit a beacon with a fixed probability p and listen with $1 - p$). In particular, it appears that many of these protocols are not particularly efficient. For example, switching an Encounternet tag with a particular battery from transmit-only mode to encounter-registration mode reduces the lifespan of the tag from 7.5 days to less than a day, indicating that the receiver is active a significant fraction of the time.

Although to the best of our knowledge, there has been no analyses of the protocols for encounter registration problem, several similar problems are well-studied in the wireless network literature, such as minimum dominating set problem [11], [12], neighbor discovery problem [10], [13], [14] and information exchange problem [15]–[17].

Minimum dominating set problem [11], [12] is studied to deal with interference challenges for the wireless multi-hop networks. This problem focus on the communication models, such as unit disk graph model [18] (UDG), graph-based model [19] and Signal to Interference plus Noise Ratio model [20] (SINR). Neighbor discovery problem is well studied in the wireless sensor networks. Protocols [10], [13], [14] are designed for two nodes to discover each other and are applied directly to the multi-node scenario. Information exchange problem [15]–[17] is studied on the information propagation in a single-hop network. In the network, there are active nodes with packets to transmit and inactive nodes waiting to receive.

However, protocols for these problems are not well applicable to encounter-registration systems. Whether an encounter is happening at the moment is unknown, making the problem much more complicated, and thus high efficiency (when an encounter happens) and low-energy consumption (when it does not) can hardly be mutually exclusive.

III. System Model

In this paper, we study the encounter registration problem in a wildlife tracing system. We call individual animals as agents, and peers are referred to as other agents that distinguish from a specific one. The definition of the encounter process is formulated as follows.

Definition 1: Encounter is defined as the process that an agent detects and records other peer(s) if they keep a period of close proximity $\Delta \leq D$ in the wildlife tracking system.

In the following, we describe the system model in this paper.

A. Radio Communication Model

In the wildlife tracing system Encounternet, the encounter behavior is a common biological phenomenon and happens when more than one agents gather closely, constituting a single clique of size k ($k \geq 2$). Note that, k is not known to each agent and the whole clique composes a sing-hop network for communication due to the proximity.

Each agent is equipped with a radio tag. An agent that has its radio on can choose to be in the *transmit* state or the *listen* state:

- Transmit state: an agent transmits (broadcasts) a message containing its ID on the channel;
- Listen state: an agent listens on the channel to receive messages from peers.

We also call an agent keeps in the listen state for a period of consecutive slots as quiet state.

Suppose time is divided into synchronized slots of equal length $2\hat{t_0}$ [21], [22] , where $\hat{t_0}$ is assumed to be sufficiently large to finish a complete communication process (one agent transmits a message including its ID and a peer receives the message).

An agent transmits successfully in a time slot if and only if it is the only one transmitting and all the other peer(s) will receive its message and record its ID in this single-hop network. Otherwise the channel is detected as idle if there is no transmission and busy if there are simultaneous messages incurring collisions on the channel.

In the wildlife tracing system Encounternet, on the one hand, each agent is equipped with an energy-restricted tag; on the other hand, encounter process happens occasionally, and thus it is a waste of battery energy if an agent turns on the radio while it does not encounter with any peer(s) at the moment. Therefore, in order to keep a balance between the energy consumption and the efficiency of the encounter process, we introduce the duty cycle mechanism [23].

Duty cycle mechanism. An agent has the capability to turn off the radio to save energy for most of the time, and only be active (transmit or listen) during a fraction θ of the time.

Incorporating the duty cycle mechanism into the Mac layer of the radio tag, in each time slot an agent u_i is able to adopt an action as:

$$s_i^t = \begin{cases} Sleep & sleep\ with\ probability\ (1-\theta_i) \\ Transmit & transmit\ with\ probability\ \theta_i p \\ Listen & listen\ with\ probability\ \theta_i(1-p) \end{cases}$$

Duty cycle is defined as the fraction of time an agent turns its radio on, which is formulated as:

$$\theta_i = \frac{|\{t : 0 \leq t < t_0, s_i(t) \in \{Transimit, Listen\}\}|}{t_0}.$$

Next, we introduce another efficient technique called collision detection mechanism. This technique is carried out by the physical carrier sensing [24], which is part of 802.11 standard and provided by a Clear Channel Assessment (CCS) circuit.

Collision detection mechanism. A listening agent can distinguish whether the channel is idle or busy, apart from successfully receiving a message.

B. Problem formulation

We formulate the problem in this paper as follows.

Problem 1: Consider $\hat{T}$ slots which is a small enough period in reality. We define an encounter registration problem as to design a protocol to guarantee all the agents in the clique can receive message from each other at least once if they encounter for at least $\hat{T}$ time slots and record the encounter process.

We look into the problem and find the key challenge is the uncertainty of dynamic movements of agents. Despite the dynamicity in this real system, when $\hat{T}$ is short enough relative to the time required for an agent to move a short distance in reality (e.g., less than 1 second), we can make a reasonable assumption that the communication connectivity of the agents is stable during each $\hat{T}$ time slots.

IV. Encounter registration protocol

In this section, we present our encounter registration protocol. The pseudo-code of the protocol is given in Algorithm 2 and Algorithm 3.

The protocol consists of two stages: detecting stage and connecting stage.

- Stage 1: detecting stage. In this stage, an agent attempts to detect whether there are nearby peers, regardless of who they are.
- Stage 2: connecting stage In this stage, an agent attempts to identify nearby peer(s) and record their IDs to its log.

Initially, each agent starts from the detecting stage. In the detecting stage, an agent turns its radio to the *sleep* state most of the time, and switches to *transmit* state or *listen* state at intervals. In the connecting stage, agents only switch between *transmit* state and *listen* state.

The key idea of the encounter protocol is that, any single agent keeps in detecting stage to reduce ineffective energy consumption. When encounter happens, it detects the existence of nearby peers and turns to the connecting stage to identify those peers (or a peer) as fast as possible, and record the encounter process to its log. when the encounter process is determined to be finished in the connecting stage, the agent turns back to the detecting stage.

Remark 1: In the encounter protocol, there is no need to synchronize the stage between agents and the encounter protocol still works when encounter peers are in different stage, e.g., an agent in detecting stage comes into a stable clique in connecting stage. The proof of correctness will be presented in section IV-C.

In the following, we describe the operations of these two stages in detail.

A. Detecting stage

In the detecting stage, energy efficiency is achieved by the duty cycle mechanism, e.g., denote the predefined duty cycle for all the agents is θ, the tag radio of each agent will work θT_0 slots in every period of T_0 slots.

However, it is ineffective when two agents encounter and one is in *sleep* state while the other is transmitting or listening. To technically achieve synchronizing the time that agents turn on the radio without extra cost, we introduce the technique of Relax Difference Set (RDS) [25]. We use the RDS technique to guarantee every encounter pair of agents turn on the radio in the same slot at least once in each round T_0.

RDS is an efficient tool to construct cyclic quorum systems [26]. The definition is:

Definition 2: A set $R = \{a_1, a_2, ..., a_k\} \subseteq Z_T$ (the set of all non-negative integers less than T) is called a RDS if for every $d \neq 0 \pmod{T}$, there exists at least one ordered pair (a_i, a_j) such that $a_i - a_j \equiv d \pmod{T}$, where $a_i, a_j \in D$.

We now give an example to explain how RDS works to help synchronization. Suppose the duty cycle is set as 0.4,

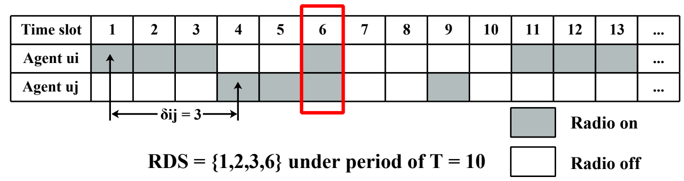


Fig. 2. A example of how RDS works to help synchronization. Consider a period of *ten* slots and the time drift between two agents u_i and u_j is 3. There exists an ordered pair $(6, 3)$ in the constructed RDS such that $6 - 3 \equiv 3 \pmod{10}$. Thus they will determinately turn on the radio at the same slot in every period T, which is the 6^{th} slot in a period of u_i and the 3^{th} slot in that of u_j respectively.

i.e., there are 4 active slots in every 10 slots. It is easy to show that $R = \{1, 2, 3, 6\}$ is a RDS under Z_{10}:

$$\begin{aligned} &2-1=1, \quad 3-1=2, \quad 6-3=3, \quad 6-2=4, \\ &6-1=5, \quad 2-6=6 \ (mod\ 10), \quad \ldots, \quad \ldots, \end{aligned}$$

In every period of ten slots, for any $i = \{0, 1, \ldots, 9\}$, if $i \in R$, then the agent turns on its radio in the i^{th} slot in this period; otherwise it turns off the radio to the *sleep* state. An example is depicted in Figure 2.

Algorithm 1 RDS Construction Algorithm

```
1: R := ∅; λ := ⌈√N⌉, μ := ⌈⌈√N⌉/2⌉;
2: for i = 1 : λ do
3:     R := R ∪ i;
4: end for
5: for j = 1 : μ do
6:     R := R ∪ (1 + j * λ);
7: end for
```

It has been proved that any RDS must have cardinality $|R| \geq \sqrt{N}$ [25]. We present a linear algorithm to construct a RDS with cardinality $\lceil \frac{3\sqrt{T_0}}{2} \rceil$ under Z_{T_0} in Alg. 8.

We show the correctness of the construction formally.

Lemma 1: Set $R = \{r_0, r_1, ..., r_{\lambda+\mu-1}\}$ constructed in Alg. 8 is a RDS, where $|R| = \lambda + \mu = \lceil \sqrt{T_0} \rceil + \lceil \frac{\lceil \sqrt{T_0} \rceil}{2} \rceil \approx \lceil \frac{3\sqrt{T_0}}{2} \rceil$.

Proof: Obviously, if there exists one ordered pair (a_i, a_j) satisfying $a_i - a_j \equiv d \pmod{T_0}$, an opposing pair (a_j, a_i) exists such that $a_j - a_i \equiv (T_0 - d) \pmod{T_0}$. Thus we only need to find at least one ordered pair (a_i, a_j) for each $d \in [1, \lfloor T_0/2 \rfloor]$.

In the construction, λ in Line 1 is the smallest integer satisfying $\lambda^2 \geq T_0$. Every d in range $[1, \lfloor T_0/2 \rfloor]$ can be represented as: $d = 1 + j \times \lambda - i$, where $1 \leq j \leq \mu, 1 \leq i \leq \lambda$. Thus, there exists $a_j = 1 + j \times \lambda$ from Line. 3 and $a_i = i$ from Line. 6 satisfying $a_j - a_i \equiv d$. Then, the lemma holds. ■

Based on the RDS, we present the operations in the detecting stage, as depicted in Alg. 2. Agents turn on and off the radio according to the RDS sequence. Consider a slot the radio is on, and then the slot is divided into two sub-slots. In the first sub-slot an agent transmits a beacon

Algorithm 2 Detecting Algorithm

$T_0 := \lceil \frac{9}{4\theta^2} \rceil$; $\omega_0 := \frac{1}{2}$; $t := 0$;
Invoke Alg. 8 to construct $R = \{r_0, r_1, ..., r_{\lceil \frac{3\sqrt{T_0}}{2} \rceil}\}$ under Z_{T_0};
while *True* **do**
  if $(t+1) \in R$ **then**
    In the first sub-slot:
    Transmit a beacon with probability ω_0 and listen with probability $1 - \omega_0$;
    In the second sub-slot:
    if the agent is in *listen* state in the first sub-slot **then**
      if detects energy (a beacon or a collision by multiple beacons) in the first sub-slot **then**
        Transmit a beacon and turn to the connecting stage;
      end if
    else
      if detects energy (a beacon or a collision by multiple beacons) in this sub-slot **then**
        Turn to the connecting stage;
      end if
    end if
  else
    Sleep in the whole slot;
  end if
  $t := (t+1)\%T_0$;
end while

Algorithm 3 Connecting Algorithm

$t := 0$; $\omega_t := \zeta$;
while *True* **do**
  In the first sub-slot:
  Transmit a message containing ID with probability ω_t and listen with probability $1 - \omega_t$;
  In the second sub-slot:
  if the agent is in *listen* state in the first sub-slot **then**
    if reveive a message successfully **then**
      Record the source ID and transmit a beacon;
      Set $\omega_{t+1} := \frac{\omega_t}{(1+\epsilon)}$;
    else if channel is idle **then**
      Set $\omega_{t+1} := min\{(1+\epsilon) \cdot \omega_t, \zeta\}$;
    else
      Set $\omega_{t+1} := \frac{\omega_t}{(1+\epsilon)}$;
    end if
  else
    if detect beacons in this sub-slot **then**
      $\omega_{t+1} := 0$;
    else
      Set $\omega_{t+1} := \frac{\omega_t}{(1+\epsilon)}$;
    end if
  end if
  $t := (t+1)$;
  if $t == \hat{T}$ **then**
    if no peer is found in this round **then**
      Turn to the detecting stage;
    else
      $t := 0$; $\omega_t := \zeta$;
    end if
  end if
end while

with probability ω_0 and listens with probability $(1 - \omega_0)$. In the second sub-slot:

1) The agent is in *listen* state in the first sub-slot:
   - if the agent detects a beacon (or beacons) in the first sub-slot, it transmits a beacon (a bit is OK) as an acknowledgement on the channel in the second sub-slot and turn to the connecting stage; otherwise it does nothing.
2) The agent is in *transmit* state in the first sub-slot:
   - if the agent detects a beacon (or beacons) in this sub-slot, it turns to the connecting stage; otherwise it does nothing.

As discussed before, the aim of this stage is to detect nearby peer(s) as fast as possible (if exists), and either successful transmission or detecting busy on the channel activates the agent to switch to the connecting stage.

B. Connecting stage

In the connecting stage, agents attempt to identify the nearby peers and record the encounter processto its local log.

In this stage, pre-defined $\hat{T}$ slots consists of a round. The transmitting probability is initially set as ζ, which is a pre-defined constant. Each agent repeats the operations in Alg. 3 round by round and it turns to the detecting stage when it cannot find any peer in a complete round.

Each slot is divided into two sub-slots. In the first sub-slot of each slot t, an agent transmits a message containing ID with probability ω_t and listen with probability $(1-\omega_t)$. In the second sub-slot:

1) The agent is in *listen* state in the first sub-slot:
   - if the agent receives a message successfully, it decodes and records the source ID in the message, and transmits a beacon (a bit is OK) as an acknowledgement on the channel in the second sub-slot.
   - if the channel is idle, this means there is a chance to transmit successfully and it multiplies its transmitting probability by a factor $(1 + \epsilon)$ (no larger than the pre-defined constant ζ).
   - if the agent detects collisions, it divides its transmission probability by a factor $(1 + \epsilon)$.
2) The agent is in *transmit* state in the first sub-slot:
   - if the agent detects beacons in this sub-slot, this

- means its previous message has been successfully received by its nearby peers, and it keeps listening in all the rest first sub-slots of this round, which is called quiet state.
- if the agent detects nothing in this sub-slot, it means its previous message failed to propagate due to simultaneous transmissions. Thus it divides its transmitting probability by a factor $(1+\epsilon)$.

Factor $(1+\epsilon)$ is a pre-defined constant to adjust the transmitting probability adaptively. In the end of a complete round, if there is no peer detected in this whole round, which indicates the encounter process is finished, the agent turns to the detecting stage .

C. Analysis of the protocol

When an encounter happens, an agent in detecting stage will switch to connecting stage very soon. We derive this conclusion from the following three lemmas.

Lemma 2: Consider any two agents u_i and u_j in detecting stage. In each period T_0, they will turn on the radio in the same slot at least once.

Proof: Assume the time drift between u_i and u_j is δ_{ij} (mod T_0). In the RDS constructed under T_0, there exists at least one ordered pair (a_i, a_j) such that $a_i - a_j \equiv \delta_{ij}$ (mod T_0). Thus the a_i^{th} slot in a period of u_i is exactly the a_j^{th} slot in a period of a_j^{th} and both of them turn on the radio in this slot according to Alg. 2 Line 4, which completes the proof. ■

Lemma 3: Consider k agents all in the detecting stage at the beginning. Given a small enough constant η, an agent in detecting stage will turn to the connecting stage in $O(\theta^{-2})$ slots with high probability.

Proof: By Lemma 2, an agent can turn on the radio in the same slot with any other peer at least once during a period of T_0. The probability it detects a peer in a period of T_0 is at least $Pr \geq 1 - \frac{1}{2^{k-1}}$. Hence given a small enough constant η, it holds with high probability that an agent will turn to connecting stage in $\frac{\ln \eta}{\ln \frac{1}{2^{k-1}}}$ periods, which is $\frac{\ln \eta}{\ln \frac{1}{2^{k-1}}} \lceil \frac{9}{4\theta^2} \rceil$ slots in total. ■

Lemma 4: An agent turns to the connecting stage will increase the probability that other peers in the detecting stage to detect peers.

Proof: An agent in the detecting stage will transmit or listen in every first sub-slot which can help peers to detect it, while it only turns on the radio a fraction of time in the detecting stage. ■

According to Lemma 9 and 10, every agent will turn to the connecting stage in $O(\theta^{-2})$ slots with high probability.

Next, we prove that with constant probability, there are one agent switch to the quiet state in each slot.

Lemma 5: Consider a set of l agents, $u_1, u_2, \ldots, u_l$. For an agent u_i, it transmits with probability $0 < \omega(u_i) < \frac{1}{2}$. Let p_0 denote the probability that the channel is idle in a slot; and p_1 denote the probability that there is exactly one transmission in a slot. Then $p_0 \cdot \sum_{i=1}^{l} \omega(u_i) \leq p_1 \leq 2 \cdot p_0 \cdot \sum_{i=1}^{l} \omega(u_i)$

Lemma 6: With $a_i \in [0, \frac{1}{2}]$ for $i = 0, 1, \ldots,$ it holds that

$$4^{-\sum_i a_i} \leq \prod_i (1 - a_i) \leq e^{-\sum_i a_i}. \quad (1)$$

The proof of Lemma 12 and Lemma 13 are omitted here; inspirations are drawn from [27] and [16].

Lemma 7: For a slot $T > 0$ with $|S_t| \geq \log k$, if there exist constants $\alpha_1, \alpha_2 \geq 1$ such that $\alpha_1 \leq \sum_{u \in S_t} \omega_t(u) \leq \alpha_2$, then with constant probability there is one active node switching to the quiet state in each slot.

Proof: By Lemma 13 and 13, in each slot, the channel is idle with probability at least $4^{-2\alpha_2}$, and there is exactly one transmission on the channel with probability at least $\alpha_1 \cdot 4^{-2\alpha_2}$. Thus, it holds that with constant probability (given α_1 and α_2), there are one agent switch to the quiet state in each slot. ■

V. Analysis of the Encounter Protocol

In this section, we prove that with a clique of k agents, our protocol guarantees each agent can record all the peers in $O(k)$ slots with high probability. Note that, k is not known in the execution of the protocol but just for analysis in this section. Formally, this result is derived from the following theorem.

Theorem 1: Consider an encounter process of k agents (k is a integer $k \geq 2$) :

(1) An agent in detecting stage will switch to the connecting stage in $O(\theta^{-2})$ slots with high probability. Recall that θ is the pre-defined duty cycle and can be seen as a constant.
(2) When all the agents are in the connecting stage, all of them will transmit successfully in a first sub-slot and turn to the quiet state in $O(k)$ slots with high probability.

We prove these two conclusions in Theorem 1 in Section V-A, and Section V-B respectively. Based on this theorem, we can estimate a duration of the encounter process between any pair of agents u_i and u_j, by checking the records in their local logs. If there are z consecutive records of the other peer, it indicates a duration of $2z\hat{T}t_0$. Recall that $2t_0$ is the real time of a slot and $\hat{T}$ is the number of slots of a round in the connecting stage, which is sufficiently large to satisfy the bound of the encounter protocol while still short enough relative to the time required for an agent to move a short distance.

A. Time bound for detecting stage

When an encounter happens, an agent in detecting stage will switch to connecting stage very soon. We derive this conclusion from the following three lemmas.

Lemma 8: Consider any two agents u_i and u_j in detecting stage. In each period T_0, they will turn on the radio in the same slot at least once.

Proof: Assume the time drift between u_i and u_j is δ_{ij} (mod T_0). In the RDS constructed under T_0, there exists at least one ordered pair (a_i, a_j) such that $a_i - a_j \equiv \delta_{ij} \pmod{T_0}$. Thus the a_i^{th} slot in a period of u_i is exactly the a_j^{th} slot in a period of a_j^{th} and both of them turn on the radio in this slot according to Alg. 2 Line 4, which completes the proof. ■

Lemma 9: Consider k agents all in the detecting stage at the beginning, an agent in detecting stage will turn to the connecting stage in $O(\theta^{-2})$ slots with high probability.

Proof: By Lemma 8, an agent can turn on the radio in the same slot with any other peer at least once during a period of T_0. The probability it detects a peer in a period of T_0 is at least $Pr \geq 1 - \frac{1}{2^{k-1}}$. Hence given a small enough constant η, it holds with high probability that an agent will turn to connecting stage in $\frac{\ln \eta}{\ln \frac{1}{2^{k-1}}}$ periods, which is $\frac{\ln \eta}{\ln \frac{1}{2^{k-1}}} \lceil \frac{9}{4\theta^2} \rceil$ slots in total. ■

Lemma 10: An agent turns to the connecting stage will increase the probability that other peers in the detecting stage to detect peers.

Proof: An agent in the detecting stage will transmit or listen in every first sub-slot which can help peers to detect it, while it only turns on the radio a fraction of time in the detecting stage. ■

According to Lemma 9 and 10, every agent will turn to the connecting stage in $O(\theta^{-2})$ slots with high probability.

B. Time bound for connecting stage

Consider the k agents all in the connecting stage. We analyze the upper bound for all the agents in the clique to transmit successfully. Note that, an agent transmitting successfully will be recorded by all the other peers in the clique, and then it switches to the quiet state.

Denote S_t as the set of agents which have not switched to the quiet state in slot t, and $|S_t|$ is the cardinality. Thus the upper bound of the protocol is the maximum slots for a clique to turn to $|S_t| = 0$.

In the following, we first show the upper bound of slots it takes from the beginning to $|S_t| \leq \log k$ in Lemma 11. Then we show the upper bound it takes to $|S_t| = 1$ in Lemma 21. Finally we present that it takes for all agents to successfully transmit (i.e, $|S_t| = 0$.) in $O(k)$ in Lemma 22.

Lemma 11: There exists $T_2 = O(k)$ it holds with high probability that $|S_{T_2}| \leq \log k$.

To prove this lemma, we need to introduce and prove some small lemmas at first. We review two useful lemmas as follows.

Lemma 12: Consider a set of l agents, $u_1, u_2, \ldots, u_l$. For an agent u_i, it transmits with probability $0 < \omega(u_i) < \frac{1}{2}$. Let p_0 denote the probability that the channel is idle in a slot; and p_1 denote the probability that there is exactly one transmission in a slot. Then $p_0 \cdot \sum_{i=1}^{l} \omega(u_i) \leq p_1 \leq 2 \cdot p_0 \cdot \sum_{i=1}^{l} \omega(u_i)$

Lemma 13: With $a_i \in [0, \frac{1}{2}]$ for $i = 0, 1, \ldots,$ it holds that

$$4^{-\sum_i a_i} \leq \prod_i (1 - a_i) \leq e^{-\sum_i a_i}. \tag{2}$$

The proof of Lemma 12 can be referred to [27] and the proof of Lemma 13 can be referred to [16]. Based on these two lemmas, we get the following conclusion.

Lemma 14: For a slot $T > 0$ with $|S_t| \geq \log k$, if there exist constants $\alpha_1, \alpha_2 \geq 1$ such that $\alpha_1 \leq \sum_{u \in S_t} \omega_t(u) \leq \alpha_2$, then with constant probability there is one active node switching to the quiet state in each slot.

Proof: By Lemma 13 and 14, in each slot, the channel is idle with probability at least $4^{-2\alpha_2}$, and there is exactly one transmission on the channel with probability at least $\alpha_1 \cdot 4^{-2\alpha_2}$. Thus, it holds that with constant probability (given α_1 and α_2), there are one agent switch to the quiet state in each slot. ■

Next we show that after all the agents turn to the connecting stage, the summation of the total agents' transmission probabilities will go between α_1 and α_2 soon, where α_1 and α_2 are constants defined in Lemma 14.

Lemma 15: For a slot with $\sum_{u \in S_t} \omega_t(u) = \alpha$, it holds that $Pr[\sum_{u \in S_t} \omega_{t+1}(u) \leq \alpha \cdot \frac{3}{4}] \geq \frac{7}{8}$ for large enough α.

Proof: The probability that there are more than one agents transmitting in slot t is at least $1 - exp\{-\alpha\}$ according to Equation (2). All the agents will halve their transmission probabilities if the channel is not idle in slot t. Denote X as the random variable that indicates the value of $\sum_{u \in S_{t+1}} \omega_{t+1}(u)$. We get,

$$Pr[X = \sum_{u \in S_t} \frac{\omega_t(u)}{2}] \geq 1 - exp\{-\alpha\}$$

which is at least 7/8 when $\alpha > 2.1$. Hence,

$$X \leq \frac{7}{8} \cdot \frac{1}{2} \cdot \alpha + \frac{1}{8} \cdot 2\alpha < \frac{3}{4} \cdot \alpha.$$

Therefore, it holds with high probability that for large α, $Pr[\sum_{u \in S_t} \omega_{t+1}(u) \leq \alpha \cdot \frac{3}{4}] \geq \frac{7}{8}$. ■

Lemma 16: There exists a constant $\hat{\alpha_2} > 1$, such that among $\gamma \log k$ slots (not necessarily consecutive) with $\sum_{u \in S_t} \omega_t(u) \geq \hat{\alpha_2}$ and sufficiently large $\gamma > 0$, there are at least $\frac{3}{4}\gamma \log k$ slots with $\sum_{u \in S_{t+1}} \omega_{t+1}(u) < \frac{3}{4} \sum_{u \in S_t} \omega_t(u)$, with probability $1 - O(k^{-1})$.

Proof: Let $T := \gamma \log k$, and X_t be the random variable that indicates the value of $\sum_{u \in S_{t+1}} \omega_{t+1}(u) / \sum_{u \in S_t} \omega_t(u)$. Then by Lemma 15, it holds that $Pr[X_t \leq \frac{3}{4}] \geq \frac{7}{8}$. Let Y_t be the binary random variable that takes value 1 if $X_t \leq \frac{3}{4}$. Note that given $\sum_{u \in S_t} \omega_t(u) \geq \hat{\alpha_2}$, $E[Y_t] \geq \frac{7}{8}$ always hold. Hence, $E[\sum_{t=1}^{T} Y_t] \geq T \cdot \frac{7}{8}$, and it holds that $Pr[\sum_{t=1}^{T} Y_t \leq T \cdot \frac{3}{4}]$ by the Chernoff bound. That is with probability $1 - O(k^{-1})$, there are at least $T \cdot \frac{3}{4}$ slots t with $\sum_{u \in S_{t+1}} \omega_{t+1}(u) / \sum_{u \in S_t} \omega_t(u) \leq 3/4$, which completes the proof. ■

Lemma 17: There exists a constant $\hat{\alpha_2} > 1$, such that during any period of $\gamma \log k$ slots with sufficiently large $\gamma > 0$, the probability that within the considered period there is a slot t with $\sum_{u \in S_t} \omega_t(u) \leq \hat{\alpha_2}$ is $1 - O(k^{-1})$.

Proof: Denote $T := \gamma \log k$ and the period of T starts from slot t_0. By Lemma 16, with probability at least $1 - On^{-1}$, it holds that

$$\sum_{u\in S_{t_0+T}} \omega_{t_0+T}(u) \geq \sum_{u\in S_{t_0}} \omega_{t_0}(u) \cdot (\frac{3}{4}\cdot\frac{3}{4}\cdot\frac{3}{4}\cdot 2)^{\frac{T}{4}} = \sum_{u\in S_{t_0}} \omega_{t_0}(u) \cdot (\frac{27}{32})^{\frac{T}{4}}.$$

Since $\sum_{u\in S_{t_0+T}} \omega_{t_0+T}(u) < k$ and $T = \gamma \log k$, we know that $\sum_{u\in S_{t_0+T}} \omega_{t_0+T}(u)$ is at most $\hat{\alpha_2}$ for large enough γ. ■

Lemma 18: There exists a constant $\hat{\alpha} \geq 0.01$ such that for any time t with $\sum_{u\in S_t} \omega_t(u) = \hat{\alpha}$, it holds that

$$Pr[\sum_{u\in S_{t+1}} \omega_{t+1}(u) \geq \hat{\alpha}\cdot\frac{4}{3}] \geq \frac{7}{8}. \quad (3)$$

Proof: Denote X as the random variable that indicates the value of $\sum_{u\in S_{t+1}} \omega_{t+1}(u)$. The probability that there are no transmissions in slot t is at least $4^{-\hat{\alpha}}$. Hence,

$$Pr[X = \sum_{u\in S_t} 2\cdot\omega_t(u)] \geq 1 - 4^{-\hat{\alpha}}$$

which is at least 7/8 when $\hat{\alpha}$ is close to 0.01. Hence,

$$X \geq \frac{7}{8}\cdot 2\cdot\hat{\alpha} + \frac{1}{8}\cdot\frac{1}{2}\hat{\alpha} > \frac{4}{3}\cdot\hat{\alpha}.$$

Therefore, it holds with high probability that for small $\hat{\alpha}$ close to 0.01, $Pr[\sum_{u\in S_t} \omega_{t+1}(u) \leq \hat{\alpha}\cdot\frac{4}{3}] \geq \frac{7}{8}$. ■

Lemma 19: There exists a constant $\hat{\alpha_1} > 0$, such that among $\gamma \log k$ slots (not necessarily consecutive) with $\sum_{u\in S_t} \omega_t(u) \leq \hat{\alpha_1}$ and sufficiently large $\gamma > 0$, there are at least $\frac{3}{4}\gamma \log k$ slots with $\sum_{u\in S_{t+1}} \omega_{t+1}(u) \geq \frac{4}{3}\sum_{u\in S_t} \omega_t(u)$, with probability $1 - O(k^{-1})$.

Proof: Denote $T := \gamma \log k$, X_t as the random variable that indicates the value of $\sum_{u\in S_{t+1}} \omega_{t+1}(u) / \sum_{u\in S_t} \omega_t(u)$, and Y_t to be the binary random variable that takes value 1 if $X_t \leq \frac{4}{3}$. Note that given $\sum_{u\in S_t} \omega_t(u) \geq \hat{\alpha_1}$, $E[Y_t] \geq \frac{7}{8}$ always hold. Hence, $E[\sum_{t=1}^{T} Y_t] \geq T\cdot\frac{7}{8}$, and it holds that $Pr[\sum_{t=1}^{T} Y_t \leq T\cdot\frac{3}{4}]$ by the Chernoff bound. That is with probability $1 - O(k^{-1})$, there are at least $T\cdot\frac{4}{3}$ slots t with ■

Lemma 20: Let t_0 be the first slot in which $\sum_{u\in S_t} \omega_t(u)$ drops below $\hat{\alpha_2}$. in the subsequent $T := \tau\cdot\log k$ slots where $\tau > 0$ and k is large enough, the following hold:

(1) there are at least $\frac{3}{4}\cdot T$ slots t with $\sum_{u\in S_t} \omega_t(u) \leq \alpha_2$, where $\alpha_2 > \hat{\alpha_2}$ is a constant.

(2) there are at least $\frac{3}{4}\cdot T$ slots t with $\sum_{u\in S_t} \omega_t(u) \geq \alpha_1\cdot k$, where $\alpha_1 < \hat{\alpha_1}$ is a constant.

With Lemma 14, 15, 16, 17, 18, 19 and 20, we now prove Lemma 11.

Then we show that when the number of agents not switching to the quiet state is less than $\log k$, the process accelerates to only one agent not in quiet state with high probability. This is formulated as the following lemma.

Lemma 21: At time $T_3 := T_2 + \gamma_2\cdot\log k = O(k)$ it holds with high probability that $|S_{T_3}| = 1$.

Proof: After T_2 slots, it holds with high probability that $|S_{T_2}| \leq \log k$. Then it takes at most $\gamma_2\cdot\log k$ slots to keep $\beta_1 \leq \sum_{u\in S_t} \omega_t^u \leq \beta_2$, where β_1 and β_2 are two constants. Afterwards, there is a slot $T_3 := T_2 + \gamma_2\cdot\log k$ such that $|S_{T_3}| = 1$, where $\gamma_2 > 0$ is a large enough constant. Otherwise during the period from T_2 to T_3, with high probability there are more than $\log k$ agents switching to the quiet state, which can not be possible. ■

Finally we need to prove the following lemma.

Lemma 22: For time T_3 when there is only one agent not switching to quiet state, at time $T_4 := 2\cdot T_3 + O(1) = O(k)$ it holds with high probability that this agent successes to transmit.

Proof: Since at time T_3 there is only one agent still attempting to transmit, the transmission probability will get to ζ at time $2\cdot T_3$ if it keeps listening. Note that if agent u transmit with probability ζ for $\log k$ slots, where d then with high probability there exist one slot in which agent u successes to transmit. ■

According to the Lemma 11, 21 and 22, we get the upper bound $O(k)$ of the encounter protocol with respect to the clique size k. Note that, since the round of each agent may not be synchronized, the encounter protocol requires an overlap of at least T_4 time slots in a complete round between each agent. This can be easily solved as we set $\hat{T} := 2\cdot T_4$ in Algorithm 3.

VI. Simulations for protocol validation

In this section, we evaluate the performance of our protocol by extensive simulations, and the results show our protocol can achieve good scalability. We also modeled the mobility and herd characteristics of wild animals, and results validate the efficiency of our protocol.

A. Parameter Settings

To conduct the simulation tests, we carried out some pre-tests to determine the parameters of our protocol. We selected the appropriate values and fixed the parameters as in Table I.

TABLE I
Parameter Settings

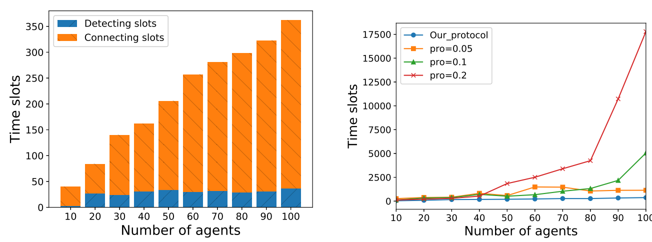
Notation	value	Description
$2\hat{t_0}$	20 ms	The time length of a synchronized slot in reality.
D	20 m	The radio range of tags.
$\hat{T}$	500	The pre-defined period of Connecting Stage.
ω_0	0.5	Fixed transmitting probability in Detecting Stage.
ω_t	0.5	Initial transmitting probability in Connecting Stage.
ζ	0.5	Upper bound of the transmitting probability in Connecting Stage.
ϵ	1.0	Adjustment factor in Connecting Stage.

Next, we selected the baseline methods to compare with our protocol. Existing methods for the encounter registration problem are mainly based on fixed transmitting probability [2], [7] (i.e., agents transmit a beacon with a fixed probability p and listen with $1 - p$). Note that, if p is too large, there would be multiple transmissions in a time slot which results in collisions; if p is too small, there would be no transmissions in a time slot which results in long latency for the encounter process. We tested different values of transmitting probabilities and chose the following three settings for comparison:

- Baseline I: fix the transmitting probability as 0.05.
- Baseline II: fix the transmitting probability as 0.1.
- Baseline III: fix the transmitting probability as 0.2.

B. Scalability

The number of encounter animals in the wild varies from a handful to hundreds. Therefore, scalability is a crucial factor for encounter protocols. In our testbed, we first simulated a group of agents in a stable clique, to evaluate scalability regarding their number and duty cycle.



(a) (b)

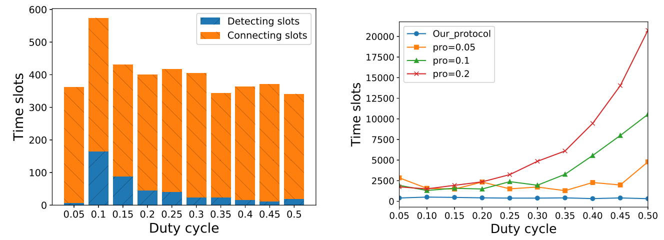
Fig. 3. Time for encounter process increases when the number of agents grows from 10 to 100. Figure (a) depicts the slots for detecting stage and connecting stage of our protocol and figure (b) compares our protocol with the three baseline methods.

In Fig. 3, we fix the duty cycle as 0.25 and increase the number of agents from 10 to 100. Fig. 3 (a) illustrates the number of slots for the encounter process of our protocol grows as the number of agents increases. Particularly, the slots needed in the detecting stage remain steady. This is because although more agents need to be switched to the connecting stage, they meanwhile add to the possibility that a listening agent turns to the connecting stage in each time slot. When compared to the baseline methods as showed in Fig. 3 (b), our protocol has the least latency in the encounter process, and the time of the other three methods increases markedly as the agent number grows while that of our protocol still stays at a low level.

In Fig. 4, we fix the the number of agents as 100 and increase duty cycle from 0.05 to 0.5. Fig. 4 (a) show the time for encounter process of our protocol stays steady when duty cycle varies. Particularly, the slots needed in the detecting stage decreases when duty cycle increases. This is because higher duty cycle increases the probability that an agent is active in each slot. Also we can see in Fig. 4 (b) that the baseline methods increase the latency as the duty cycle rises.

Next, we record the encounter registration rate in each time slot. Encounter registration rate is defined as the proportion of agents that has been recorded.

We set the number of agents as 100, and fix duty cycle as 0.25 in Fig. 6 and 0.5 in Fig. 7. The same trend can be



(a) (b)

Fig. 4. Time for encounter process increases when the duty cycle grows from 0.05 to 0.5. Figure (a) depicts the slots for detecting stage and connecting stage of our protocol and figure (b) compares our protocol with the three baseline methods.

seen that our protocol has higher encounter registration rate all the time and reaches to 1.0 faster than the other methods.

In conclusion, our protocol outperforms the fixed transmitting probability methods and has better scalability.

C. Mobility

The essential difference between a wildlife tracing system and a mobile wireless network is the variability among the animal species in their movement and interaction behavior. The key challenge is that depending on the targeted animals, the moving speed and mobility might vary greatly, and the herd characteristics may also be very different.

In this paper, we consider three explicit animal models of mobility and build the simulation models as they relate to animal movement and interaction. This is the same approach that has been used in mobile wireless networks where human and vehicular mobility are characterized in a similar fashion. Specifically, we model their moving speeds as follows:

- Species I: move at speed of 50 m/s.
- Species II: move at speed of 30 m/s.
- Species III: move at speed of 10 m/s.

Recall that the time length of a synchronized slot in reality is set as 20 ms and the radio range is set as 20 m.

We evaluate our protocol in three simple and heuristic cases, as depicted in Fig. 5.

- Case I: encounter for two agents, e.g., two birds fly towards each other and then fly apart, as depicted in Fig. 5 (a).
- Case II: encounter for a single agent with a group of agents, e.g., a bird fly into a group of birds, as depicted in Fig. 5(b).
- Case III: encounter for two groups of agents, e.g., two groups of birds fly towards each other and then fly apart, as depicted in Fig. 5(c).

1) Encounter for two agents: We can see from Fig. 8 that both agents employing our protocol can record each other regarding all the three species.

Fig. 5. Encounter examples of three fundamental cases

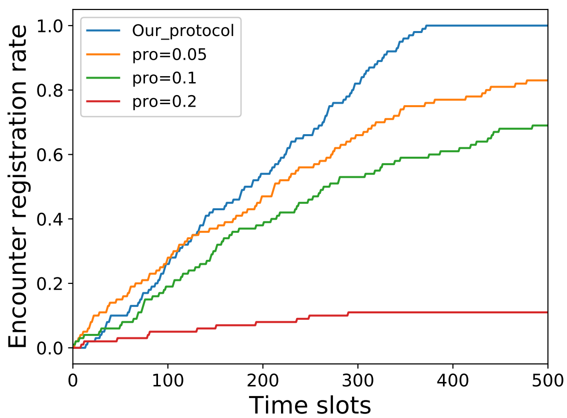


Fig. 6. Encounter process with 100 agents and duty cycle of 0.25. Encounter registration rate increases as time goes on. Our protocol keeps the highest rate during the whole process.

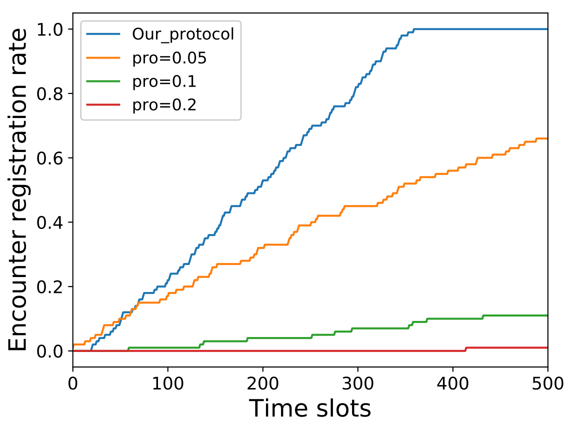


Fig. 7. Encounter process with 100 agents and duty cycle of 0.5. Encounter registration rate increases as time goes on. Our protocol keeps the highest rate during the whole process.

2) Encounter for a single agent with a group of agents: It can be seen from Fig. 9 that our protocol has higher encounter registration rate (the proportion of agents that can be recorded by the single agent) than all the other methods. Particularly, our protocol achieves nearly 100% registration rate regarding species II and species III. This is because these two species move more slowly than species I so that the agents can be recorded in higher probabilities.

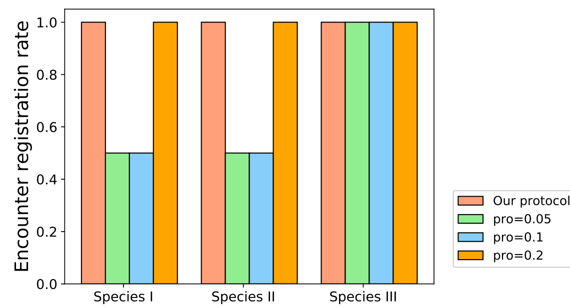


Fig. 8. Encounter for two agents. Our protocol achieves 100% encounter registration rate.

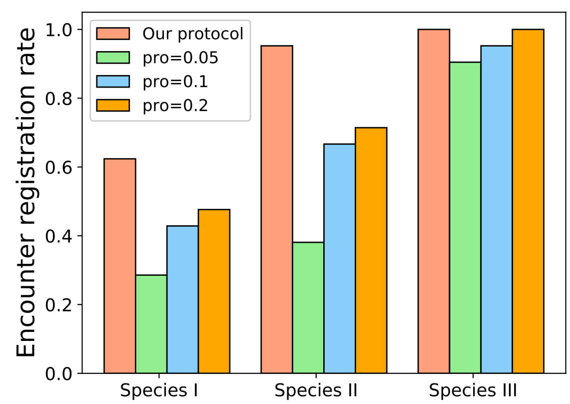
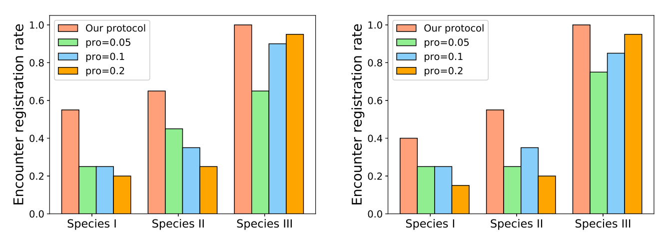


Fig. 9. Encounter for a single agent with a group of agents. Overall, our protocol achieves higher encounter registration rate.



(a) (b)

Fig. 10. Figure (a) shows the proportion of agents in group A can be recorded by the agents in group B, and Figure (b) describes the opposite case. Overall, our protocol achieves higher encounter registration rate.

3) Encounter for two groups of agents: Fig. 10 (a) shows the proportion of agents in group A having been recorded by the agents in group B, and Fig. 10 (b) depicts the opposite case. Overall, our protocol achieves higher encounter registration rate than all the other methods. Particularly, species III has higher encounter registration rate than species II and III. This is because species III moves the most slowly and thus there are more chances for agents to detect and connect their peers.

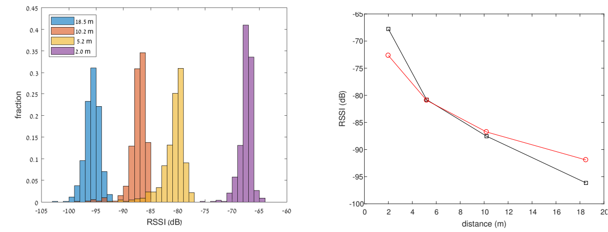
VII. Experiments for model validation

We carried out a number of experiments in order to validate the radio model. All the experiments used LAUNCHXL-CC1350-4 evaluation boards with CC1350 RF microcontrollers by Texas Instruments. We used a quiet frequency in the 434 MHz band, for which the boards are designed, and the built-in helix antenna printed on the boards. This antenna is fairly inefficient, resulting in about 4x to 8x degradation in communication range.

In all the tests we configured the radios for GFSK modulation, 500 kb/s, ±250 kHz deviation, 1243 kHz receive bandwidth, and 10 dBm transmitting power. We used a Windows program called SmartRF Studio version 7 to drive the boards during the tests and to log received packets. In all the tests, a single board transmits 1.87 packets per second (intervals of 500 ms between packets). A packet consists of a 4-byte preamble, 4-byte sync workd, a length byte, 30-byte payload that includes a sequence number, and a 2-byte checksum. The fast transmission rate reduces power consumption (per byte and per packet) and leads to short communication ranges, which is consistent with the goal of registering close encounters.

In a preliminary test, we configured one board to transmit packets, one board to receive them, and a third one to transmit a CW carrier at the same frequency. The three boards were at a distance of 0.4 m from each other. When the CW transmitter was off, virtually all packets were received correctly. When the CW transmitter was on, virtually no packets were received. This verifies that interference indeed blocks the receiver and may prevent packets from being received.

In the main test, we placed the transmitter board at a fixed position, about 1.4 m above ground, and measured receive performance at various distances between 2 and 78 m in an outdoor area with some human activity (but not much). We maintained each test position for at least 5 minutes. The receiver board was also placed about 1.4 m above ground. In all the tests shown in graphs the transmit and receive antennas were pointing at each other. We also performed a few tests with the receive antenna rotated 90 degrees; RSSI values were about 1–2 dB lower but the overall results were similar; this is consistent with the characterization of the antenna, which indicates that it is fairly omnidirectional. The main results are shown in Figure 11 (a). At each distance we measured the fraction



(a) (b)

Fig. 11. Figure (a) depicts RSSI of received packets at different distances between the transmitter and the receiver. Figure (b) describes mean RSSI at different distances (the black curve) and a least-square fitting of the data to a 6 dB attenuation when the distance doubles (the red curve).

of correctly-received packets and the RSSI of each packet. At the four distances reported in the figure, virtually all transmitted packets were received correctly. We can see a clear statistical correlation between distance and RSSI. Figure 11 (b) shows that the RSSI values roughly follow the theoretical rule that stipulates a 6 dB attenuation when the distance doubles. The means shown in this figure are of the center 50% of the packets; the extreme 50% were filtered to remove outliers.

We also tested performance at distances of 53 m and 78 m. At these distances, results were inconsistent. In one test at 78 m, 84% of the packets with mean RSSI of -98 dB. But in another test two hours later, only 2% of the packets were correctly received. At 53 m, no packets were received correctly for over 5 minutes. This location was in a depression, about 3 m lower than both the transmitter and from the 78 m position; this may have contributed to the worse performance. The main conclusion from these long-distance experiments (long given the bit rate) is that packets can sometimes be received at fairly long distances and with RSSI values that typically characterize much shorter distances.

However, the results also indicate that by dropping packets with low RSSI, say below -90 dB for these settings, we can effectively limit the communication range to 20 m or so.

VIII. Conclusions

In this paper, we studied the encounter registration problem in a single-hop network. We proposed a protocol consisting of two stages, namely detecting stage and connecting stage. Our protocol guaranteed that, when an encounter happens, an agent in detecting stage would switch to the connecting stage very soon. We carried out extensive simulations and experiments, and evaluation results showed that our protocol achieved better scalability and were more applicable to mobile wildlife environment. Our protocol has potential to be applicable to various wireless mobile networks.

References

[1] S. Prangle, T. Jordan, C. Hunter, and S. D. Gehrt, “New radiocollars for the detection of proximity among individuals,” Wildlife Society Bulletin, vol. 34, pp. 1333–1344, 2006.

[2] C. Rutz, Z. Burns, R. James, S. Ismar, J. Burt, B. Otis, J. Bowen, and J. S. Clair, “Automated mapping of social networks in wild birds,” Current Biology, vol. 22, no. 17, pp. R669–R671, 2012.

[3] C. Tentelier, J. Aymes, B. Spitz, and J. Rives, “Using proximity loggers to describe the sexual network of a freshwater fish,” Environmental Biology of Fishes, vol. 99, pp. 621–631, 2016.

[4] M. Bohm, M. R. Hutchings, and P. C. L. White, “Contact networks in a wildlife-livestock host community: Identifying high-risk individuals in the transmission of bovine tb among badgers and cattle,” PLoS One, vol. 4, no. 4, p. e5016, 2009.

[5] S. Ripperger, D. Josic, M. Hierold, A. Koelpin, R. Weigel, M. Hartmann, R. Page, and F. Mayer, “Automated proximity sensing in small vertebrates: design of miniaturized sensor nodes and first field tests in bats,” Ecology and Evolution, vol. 6, pp. 2179–2189.

[6] I. I. Levin, D. M. Zonana, J. M. Burt, and R. J. Safran, “Performance of encounternet tags: Field tests of miniaturized proximity loggers for use on small birds,” Plos One, vol. 10, no. 9, p. e0137242, 2015.

[7] D. Mennill, S. Doucet, K. Ward, D. Maynard, B. Otis, , and J. Burt, “A novel digital telemetry system for tracking wild animals: a field test for studying mate choice in a lekking tropical bird,” Methods in Ecology and Evolution, vol. 3, no. 4, pp. 663–672, 2012.

[8] F. Dressler, S. Ripperger, M. Hierold, T. Nowak, C. Eibel, B. Cassens, F. Mayer, K. Meyer-Wegener, and A. Koelpin, “From Radio Telemetry to Ultra-Low-Power Sensor Networks: Tracking Bats in the Wild,” IEEE Communications Magazine, vol. 54, no. 1, pp. 129–135, January 2016.

[9] V. Daiya, J. Ebenezer, S. A. V. S. Murty, and B. Raj, “Experimental analysis of rssi for distance and position estimation,” in International Conference on Recent Trends in Information Technology, 2011, pp. 1093–1098.

[10] M. Bakht, M. Trower, and R. H. Kravets, “Searchlight:won’t you be my neighbor?” in Proceedings of the 18th annual international conference on Mobile computing and networking, 2012, pp. 185–196.

[11] C. Scheideler, A. Richa, and P. Santi, “An o(log n) dominating set protocol for wireless ad-hoc networks under the physical interference model,” in ACM International Symposium on Mobile Ad Hoc NETWORKING and Computing, 2008, pp. 91–100.

[12] J. Yu, N. Wang, G. Wang, and D. Yu, “Review: Connected dominating sets in wireless ad hoc and sensor networks - a comprehensive survey,” Computer Communications, vol. 36, no. 2, pp. 121–134, 2013.

[13] W. Sun, Z. Yang, K. Wang, and Y. Liu, “Hello: A generic flexible protocol for neighbor discovery,” in INFOCOM, 2014 Proceedings IEEE, 2014, pp. 540–548.

[14] L. Chen, R. Fan, K. Bian, M. Gerla, T. Wang, and X. Li, “On heterogeneous neighbor discovery in wireless sensor networks,” vol. 44, pp. 693–701, 2015.

[15] J. I. Capetanakis, “Tree algorithms for packet broadcast channels,” IEEE Trans Information Theory, vol. 25, no. 5, pp. 505–515, 1979.

[16] S. Daum, M. Ghaffari, S. Gilbert, F. Kuhn, and C. Newport, “Maximal independent sets in multichannel radio networks,” in ACM Symposium on Principles of Distributed Computing, 2013, pp. 335–344.

[17] D. Yu, L. Ning, Y. Zhang, H. Jin, Y. Wang, F. C. M. Lau, and S. Feng, “Uniform information exchange in multi-channel wireless ad hoc networks,” in Conference on Autonomous Agents and Multiagent Systems, 2017, pp. 1026–1034.

[18] E. Lebhar and Z. Lotker, “Unit disk graph and physical interference model: Putting pieces together,” in IEEE International Symposium on Parallel & Distributed Processing, 2009, pp. 1–8.

[19] F. De Pellegrini, D. Miorandi, I. Carreras, and I. Chlamtac, “A graph-based model for disconnected ad hoc networks,” Proceedings - IEEE INFOCOM, pp. 373–381, 2007.

[20] C. C. Lee, W. D. Wu, and C. Chao, “Signal-to-interference-plus-noise ratio analysis for direct-sequence ultra-wideband systems,” IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 3, pp. 483–497, 2007.

[21] M. Xu, M. Zhao, and S. Li, “Lightweight and energy efficient time synchronization for sensor network,” in International Conference on Wireless Communications, NETWORKING and Mobile Computing, 2005. Proceedings, 2005, pp. 947–950.

[22] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: a survey,” Network IEEE, vol. 18, no. 4, pp. 45–50, 2004.

[23] W. Zhang, Y. Zhou, M. A. Suresh, and R. Stoleru, “Performance analysis and tuning of coexisting duty cycling wifi and wireless sensor networks,” in IEEE International Conference on Sensing, Communication, and NETWORKING, 2017, pp. 1–9.

[24] X. Yang and N. Vaidya, “On physical carrier sensing in wireless ad hoc networks,” in INFOCOM 2005. Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, 2005, pp. 2525–2535 vol. 4.

[25] W.-S. Luk and T.-T. Wong, “Two new quorum based algorithms for distributed mutual exclusion,” in Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on. IEEE.

[26] D. Peleg and A. Wool, “The availability of quorum systems,” Information & Computation, vol. 123, no. 2, pp. 210–223, 1995.

[27] A. Richa, C. Scheideler, S. Schmid, and J. Zhang, “A jamming-resistant mac protocol for multi-hop wireless networks,” in Twenty-Seventh ACM Symposium on Principles of Distributed Computing, 2010, pp. 45–54.