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

Abstract—**Encounters between wild animals mark of the same or different species mark important events, 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, transmit identification packets periodically, and receive and record packets from others. However, non-trivial protocols for this problem have not been studies, and neither did the fundamental tradeoff between minimizing the power consumption and reducing the encounter latency. Our insight is that 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.**

In this paper, we propose a robust encounter registration protocol that with high probability accomplishes the encounter registration in $O(k)$ slots time, assuming k is the number of encounter tags. Our protocol consists of two process: detect other tags with low energy consumption; and make connections with the detected tags to record each other's ID. We analyze the performance of this protocol under a formulated radio model and carry out a number of experiments to validate this model. To the best of our knowledge, this is the first practical protocol for a real encounter-registration system.

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. It consists 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 on 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 that the transmitting tag is in close proximity to the receiving tag. Recent systems record each packet with a received signal strength indication (RSSI) [9], which helps 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.

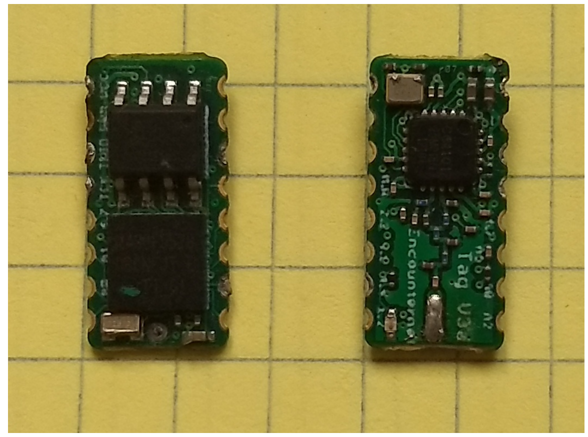


Fig. 1. Printed circuit boards for Encounternet tags [7]. Lines are 5mm apart. Complete tags require the addition of miniature batteries, whip antenna, and coating. The small batteries make effective protocols important.

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. The key problem 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. Paradoxically, highly-efficient systems with good time synchronization and short activity periods suffer more from this problem than less efficient systems in which the clocks of tags are not synchronized tightly enough to transmit simultaneously. This is not a theoretical problem; 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 formulate the problem model and propose a robust protocol for encounter registration problem with theoretical analyses.
2) We conduct experiments and extensive simulations. Our evaluation results show that, compared to baseline methods, our protocol achieves lower latency, higher encounter registration rate and 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 and three encounter cases.

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 experiments for model validation in Section V and simulations for protocol validation in in Section VI. The paper is concluded in Section VII.

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 battery. The system has been commercially used extensively. For example, it is used to study the possibility of disease transmission between cattle and wild badgers [4].

The next generalization was a system called *Enounternet* [2], [7], which has been available commercially for a few years but is no longer available. 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]. Another innovation of the Encounternet system is the recording of RSSI with every reception report, allowing researchers to estimate the distance of each encounter.

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 otherwise.

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 [10], [11], neighbor discovery problem [12]–[14] and information exchange problem [15]–[17].

Minimum dominating set problem [10], [11] 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. Many algorithms [12]–[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. A common solution to these problems is to control the transmission probability of the node in the network, making it optimal to transmit successfully. Some deterministic methods, such as quorum system [21] and prime number, are also adopted to help synchronization in the protocol design. The major difference between the problems above and the encounter registration problem in this paper is the mobility of the wild animals. It makes the problem complicated that whether an encounter is happening at the moment is unknown.

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 for theoretical analyses 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 $2\hat{t_0}$ [22], [23] , where $\hat{t_0}$ is assumed to be sufficient 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 [24].

Duty cycle mechanism. An agent has the capability to turn off the radio to save energy for most of the time, and may only be active (transmitting or receiving) 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 [25], which is part of the 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. Adaptive Wildlife Encounter 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 the 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 very 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) [26]. We use the RDS technique to guarantee that 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 [21]. 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 $d \neq 0$ (mod T), there exists at least one ordered pair (a_i, a_j) such that $a_i - a_j \equiv d$ (mod T), where $a_i, a_j \in D$.

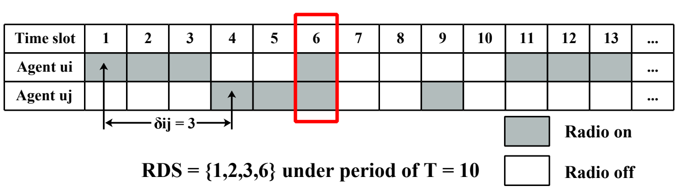


Fig. 2. A example of how RDS works to help synchronization. Consider a period of ten slots and the time drift between two agents u_i and u_j is 3. There exists an ordered pair $(6, 3)$ in the constructed RDS such that $6-3 \equiv 3$ (mod 10). Thus they will determinately turn on the radio at the same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

We now give an example to explain how RDS works to help synchronization. Suppose the duty cycle is set as 0.4, 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: $R := \emptyset$; $\lambda := \lceil\sqrt{N}\rceil$, $\mu := \lceil\frac{\lceil\sqrt{N}\rceil}{2}\rceil$;
2: **for** $i = 1 : \lambda$ **do**
3:     $R := R \cup i$;
4: **end for**
5: **for** $j = 1 : \mu$ **do**
6:     $R := R \cup (1 + j * \lambda)$;
7: **end for**

It has been proved that any RDS must have cardinality $|R| \geq \sqrt{N}$ [26]. 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. 2.

We show the correctness of the construction formally.

Lemma 1: Set $R = \{r_0, r_1, ..., r_{\lambda+\mu-1}\}$ constructed in Alg. 2 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 $a_i - a_j \equiv d$ (mod T_0), an opposing pair (a_j, a_i) exists such that $a_j - a_i \equiv (T_0 - d)$ (mod 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 can be derived. ■

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 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 process (messages containing peers' IDs) to its local log. A successful identification

Algorithm 2 Detecting Algorithm

$T_0 := \lceil \frac{9}{4\theta^2} \rceil$; $\omega_0 := \frac{1}{2}$; $t := 0$;
Invoke Alg. 2 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
  else
    $Sleep$ in the whole slot;
  end if
  $t := (t+1)\%T_0$;
end while

Algorithm 3 Connecting Algorithm

1: $t := 0$; $\omega_t := \zeta$;
2: **while** $True$ **do**
3:   ***In the first sub-slot:***
4:   Transmit a message containing ID with probability ω_t and listen with probability $1 - \omega_t$;
5:   ***In the second sub-slot:***
6:   **if** the agent is in $listen$ state in the first sub-slot **then**
7:     **if** reveive a message successfully **then**
8:       Record the source ID and transmit a beacon;
9:       Set $\omega_{t+1} := \frac{\omega_t}{(1+\epsilon)}$;
10:     **else if** channel is idle **then**
11:       Set $\omega_{t+1} := min\{(1+\epsilon) \cdot \omega_t, \zeta\}$;
12:     **else**
13:       Set $\omega_{t+1} := \frac{\omega_t}{(1+\epsilon)}$;
14:     **end if**
15:   **else**
16:     **if** detect beacons in this sub-slot **then**
17:       $\omega_{t+1} := 0$;
18:     **else**
19:       Set $\omega_{t+1} := \frac{\omega_t}{(1+\epsilon)}$;
20:     **end if**
21:   **end if**
22:   $t := (t+1)$;
23:   **if** $t == \hat{T}$ **then**
24:     **if** no peer is found in this round **then**
25:       Turn to the **detecting stage**;
26:     **else**
27:       $t := 0$; $\omega_t := \zeta$;
28:     **end if**
29:   **end if**
30: **end while**

happens only if the agent is listening and only one peer is transmitting.

The collision detection (CD) mechanism is incorporated in this stage to increase of efficiency. This mechanism enables the listening agent to notify the transmitting peers of the transmission outcomes, and thus they take measures to reduce the collisions if not successful.

In this stage, every $\hat{T}$ slots consists of a round. Each agent repeats the operations in Alg. 3 round by round and it turns to the connecting stage when it cannot find any peer in a complete round, as the operation in Line 25.

As discussed in section III, $\hat{T}$ is relatively short in real world, thus the communication connectivity stays stable in a round. However, due to the dynamic movements of agents, the communication connectivity may change from round to round, so all the parameters will be initialized at the beginning of each round and adaptively adjusted later according to the transmission outcome.

Each slot is divided into two sub-slots. Agents execute transmission or reception in the first sub-slot, and in the second sub-slot take actions responding to the outcome of the previous sub-slot (success/fail to transmit/receive a message).

Since the number of the nearby peers is unknown to each agent, the transmitting probability is initially set as ζ, which is a pre-defined constant.

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 predefined 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. For simplify, we set $(1+\epsilon) := 2$ for analysis in the next section. 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 stage switch

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, 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 3 and 4, every agent will turn to the connecting stage in $O(\theta^{-2})$ slots with high probability.

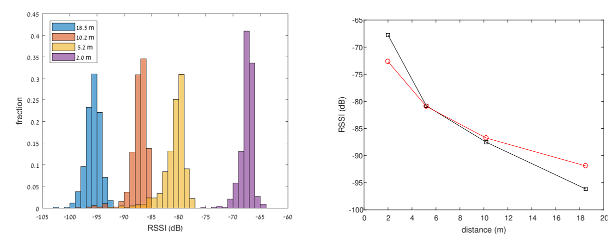
V. Experiments for model validation

In this section, we carried out a number of experiments in order to validate the radio model.

All the experiments were carried using 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 we used the built-in helix antenna printed on the boards. This antenna is fairly inefficient, resulting in about 4x to 8x degradation in communication range. We used it because the shorter ranges make testing easier and also because antennas on small wildlife tags tend to be inefficient due to the tiny ground plane. The chips that we tested are widely-used in miniature wildlife tags, and they essentially represent a modern version of the chips that were used on the original Encounternet tags.

In all 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 tests one board transmitted 1.87 packets per second (intervals of 500 ms between packets). Packets consisted 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 are 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 to transmit a CW carrier at the same frequency. The three boards were at distances 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.



(a) (b)

Fig. 3. Figure (a) depicts RSSI of received packets at different distances between the transmitter and the receiver. Figure (b) describes mean RSSI at different distances in black curve and a least-square fitting of the data to a 6 dB attenuation when the distance doubles in red curve.

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 pointed 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 3 (a). At each distance we measured the fraction 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 3 (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.

VI. Simulations for protocol validation

In this section, we evaluate the performance of our protocol by extensive simulations, and results show our protocol achieves 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 Table I.

TABLE I
Parameter Settings

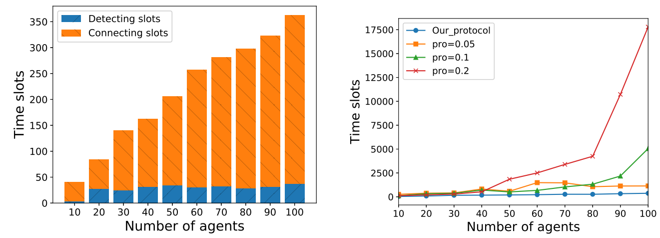
Notation	value	Description
$2\hat{t_0}$	20 ms	The time length of a synchronized slot in reality.
D	20 m	The radio range of tags.
ω_0	0.5	Fixed transmitting probability in Detecting Stage.
ω_t	0.5	Initial transmitting probability in Connecting Stage.
ζ	0.5	Upper bound of the transmitting probability in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

Next, we selected the baseline methods to compare with our protocol. Existing methods 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 $1-p$). 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 fot 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

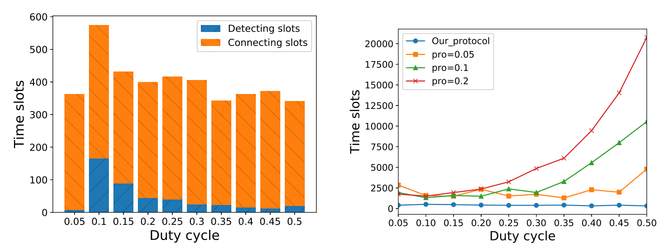
The number of encounter animals in the wild varies from a handful to hundreds. Therefore, scalability is a crucial factor for encounter protocols. In our testbed, we simulated a group of agents to evaluate scalability regarding agent amount and duty cycle.



(a) (b)

Fig. 4. 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. 4, we fix the duty cycle as 0.25 and increase the number of agents from 10 to 100. Fig. 4 (a) illustrates that, the number of slots for the encounter process of our protocol grows as the number of agents is ascending. 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 an listening agent turn to connecting stage in each time slot. When compared to the baseline methods as showed in Fig. 4 (b), our protocol takes the least latency to achieve encounter process, and the time of the other three methods increases markedly as the agent amount grows while that of our protocol still stays in a low lever.



(a) (b)

Fig. 5. 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.

In Fig. 5, we fix the the number of agents as 100 and increase duty cycle from 0.05 to 0.5. Fig. 5 (a) show that 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 see the trend in Fig. 5 (b) that the baseline methods increases the latency as 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.

Fig. 6. Encounter examples of three fundamental cases

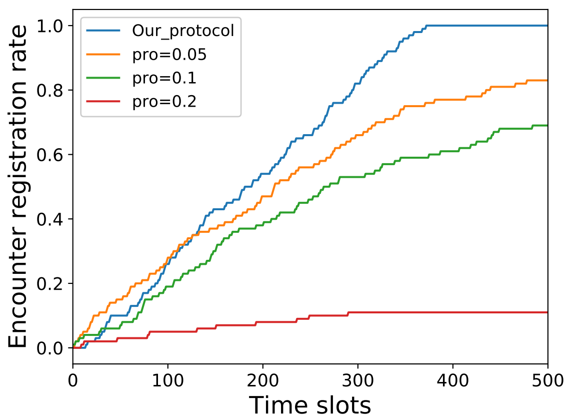


Fig. 7. Encounter process with 100 agents and duty cycle of 0.25. Encounter registration rate increases as time goes on. Our protocol keeps the highest rate during the whole process.

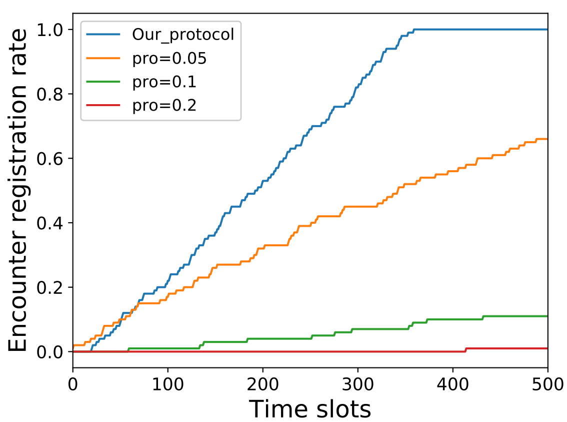


Fig. 8. Encounter process with 100 agents and duty cycle of 0.5. Encounter registration rate increases as time goes on. Our protocol keeps the highest rate during the whole process.

We set the number of agents as 100, and fix duty cycle as 0.25 in Fig. 7 and 0.5 in Fig. 8 respectively. The same trend can be seen that our protocol has higher encounter registration rate all the time and reaches to 1.0 faster than 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 species in their movement and interaction behavior. The key challenge is that depending on targeted animals, the moving speed and mobility might vary, and the herd characteristics are also different depending on the targeted animals.

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 assigned rigorous underpinnings. Specifically, we model their moving speed 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. 6.

- Case I: encounter for two agents, e.g., two birds fly towards each other and then fly apart, as depicted in Fig. 6 (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. 6(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. 6(c).

1) Encounter for two agents: We can see from Fig. 9 that both agents conducting our protocol can record each other regarding all the three species.

2) Encounter for a single agent with a group of agents: It can be seen from Fig. 10 that our protocol has higher encounter registration rate (the proportion of agents can be recorded by the single agent) than any other methods. Particularly, our protocol achieves nearly 100% registration rate regarding species II and species III. This is because these two species move slower than species I so that agents can be recorded in greater possibilities.

3) Encounter for two groups of agents: Fig. 11 (a) shows the proportion of agents in group A can be recorded by the agents in group B, and Fig. 11 (b) describes the opposite case. Overall, our protocol achieves higher encounter registration

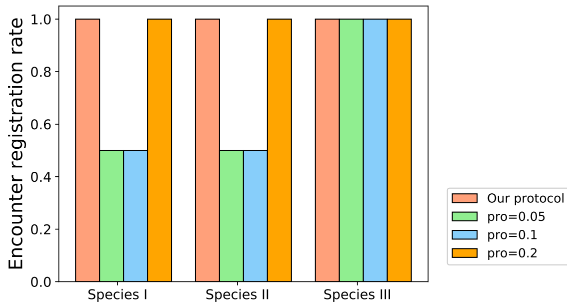


Fig. 9. Encounter for two agents. Our protocol achieves 100% encounter registration rate.

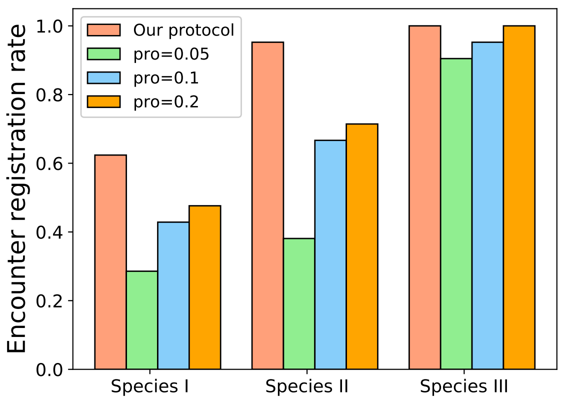
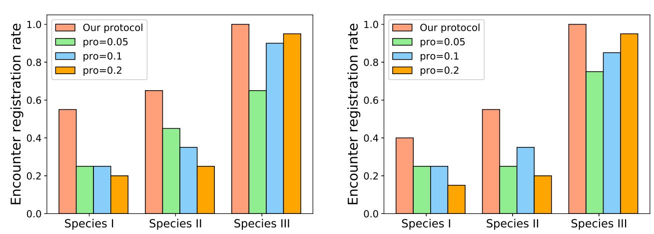


Fig. 10. Encounter for a single agent with a group of agents. Overall, our protocol achieves higher encounter registration rate.

rate than any other methods. Particularly, species III has higher encounter registration rate than species II and species III. This is because species III moves the most slowly and thus there are more chances for agents to detect and connect their peers.



(a) (b)

Fig. 11. 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.

VII. Conclusions

In this paper, we studied the encounter registration problem in a single-hop network of k agents. 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. After all the agents are in the connecting stage, each of them can record all the peers in $O(k)$ slots with high probability. We present concrete analysis of the protocol under the radio model and validate the model by real-world experiments.

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] 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.

[11] 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.

[12] 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.

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.