

# $\kappa$ -FSOM: Fair Link Scheduling Optimization for Energy-Aware Data Collection in Mobile Sensor Networks

Kai Li<sup>1</sup>   Branislav Kusy<sup>2</sup>   Raja Jurdak<sup>2</sup>   Aleksandar Ignjatovic<sup>1</sup>  
Salil S. Kanhere<sup>1</sup>   Sanjay Jha<sup>1</sup>

<sup>1</sup> University of New South Wales, Australia  
{kail, ignjat, salilk, sanjay}@cse.unsw.edu.au  
<sup>2</sup> Autonomous Systems Lab, CSIRO ICT Centre, Australia  
{brano.kusy, raja.jurdak}@csiro.au

**Technical Report**  
**UNSW-CSE-TR-201326**  
**September 2013**

THE UNIVERSITY OF  
NEW SOUTH WALES



School of Computer Science and Engineering  
The University of New South Wales  
Sydney 2052, Australia

## Abstract

We consider the problem of data collection from a continental-scale network of mobile sensors, specifically applied to wildlife tracking. Our application constraints favor a highly asymmetric solution, with heavily duty-cycled sensor nodes communicating with a network of powered base stations. Individual nodes move freely in the environment, resulting in low-quality radio links and hot-spot arrival patterns with the available data exceeding the radio link capacity. We propose a novel scheduling algorithm,  $\kappa$ -Fair Scheduling Optimization Model ( $\kappa$ -FSOM), that maximizes the amount of collected data under the constraints of radio link quality and energy, while ensuring a fair access to the radio channel. We show the problem is NP-complete and propose a heuristic to approximate the optimal scheduling solution in polynomial time. We use empirical link quality data to evaluate the  $\kappa$ -FSOM heuristic in a realistic setting and compare its performance to other heuristics. We show that  $\kappa$ -FSOM heuristic achieves high data reception rates, under different fairness and node lifetime constraints.

# 1 Introduction

Recent advances in embedded systems and battery technology have enabled a new class of large-scale mobile sensing applications. Consider a swarm of micro-aerial vehicles fitted with a variety of sensors that can achieve fine-grained three-dimensional sampling of our physical spaces, thus enabling a variety of new applications such as urban surveillance, disaster recovery and environmental monitoring [25, 3, 21]. It is now possible to monitor individual movement patterns of wildlife alongside the various aspects of their environment [8, 4, 2, 1, 9]. In a typical mobile sensing scenario, sensor nodes mounted on a carrier (e.g., vehicle or animal) collect numerous sensor readings while in transit. The nodes ultimately arrive back at a known rendezvous point (e.g., command center or animal pen), often as a large swarm and remain there for an extended period of time. The data stored on each sensor node is offloaded to a base station (BS) during this time.

A number of considerations make the data collection non-trivial. First, the number of nodes are quite large (several hundreds) and while the nodes normally arrive back in large groups, their exact arrival sequence is often unknown. Second, the sensor nodes typically have low residual energy levels after being out in the field for an extended period and limited bandwidth due to their weight and size limitations. It is thus critical to maximize the data collection in such a way that data can be collected from each node before its residual energy is exhausted. Third, the quality of the wireless channel between each node and the BS may vary with time. Having a node transmit during instances when the channel quality is poor is likely to result in packet reception errors, which in turn would require retransmissions and thus increase energy expenditure. Fourth, data should be harvested from all the nodes in a fair way. In particular, the amount of data collected from each node should be greater than a certain application-specific threshold. This is important to maximize the accuracy of data analysis, for example, in the context of mobility modeling and population characteristics for wildlife monitoring. The above considerations suggest the adoption of a scheduled transmission protocol to enable radio duty cycling at each node and to reduce the chance of packet collisions (and subsequent retransmissions). To the best of our knowledge, the issue of maximizing data collection while simultaneously considering all of the aforementioned constraints has not been examined in the context of large-scale mobile sensor network (MSN).

Conventional scheduling such as the one employed in IEEE 802.15.4 [6] are based on First Come First Served (FCFS), which we refer to as *batch processing*. Batch processing has limited performance in real-world conditions with irregular radio channels and limited bandwidth. Any node with poor link quality occupies the channel due to retransmissions, while the nodes with higher link quality have to wait. Finally, batch processing does not support data collection fairness, potentially downloading a large amount of data from a small subset of nodes.

We consider the scheduling problem in the context of a real-world application for monitoring flying foxes (also known as fruit bats). Flying foxes typically swarm out in search of food at night and flock back to roosting camps during the daytime. A typical roosting camp can consist of hundreds to tens of thousands of animals [20]. Recent work [10] aims to collect fine-grained spatiotemporal data about their movement patterns and environmental surroundings by attaching a sensor collar to these animals. The embedded sensors record the flight and

biological data such as GPS, temperature and air pressure while the bats are out and about. The data is offloaded to a BS in the roosting camp when the bats flock back during the daytime. Fig. 1.1(a) and 1.1(b) depict a typical roosting camp and the sensor collar attached to the animal.

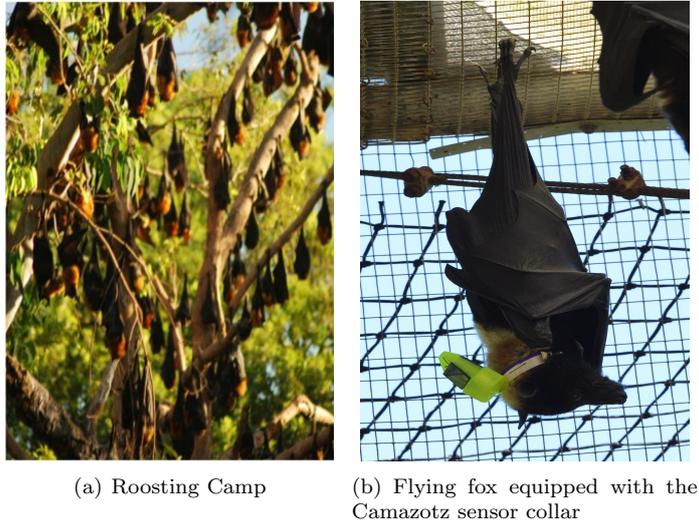


Figure 1.1: Motivating Application: Flying fox monitoring

In this paper, we propose  $\kappa$ -fair scheduling optimization model ( $\kappa$ -FSOM) to maximize data harvesting in a large-scale MSN.  $\kappa$ -FSOM schedules transmissions based on both the link quality and the residual energy of each node. It also guarantees that a certain application-specific amount of data is collected from each node. We first show that this optimization problem is NP-complete. Next, we propose a heuristic algorithm to optimize the scheduling in linear time. The  $\kappa$ -FSOM heuristic prioritizes the nodes for scheduling based on a ratio of the link quality and residual energy. This enables the nodes with the lowest energy reserves and the best chance of achieving successful transmissions to transfer their data first. In addition, we develop a states transition model in two steps to address the fairness criteria and maximize overall network goodput.

While we use the bat monitoring application as a case study, the proposed optimization model and heuristic are application-agnostic and hence applicable to a wide variety of large-scale mobile sensing scenarios with delay tolerance.

The rest of paper is organized as follows: Section 2 describes related work on link scheduling and optimization. We discuss the network configuration and provide details about the MAC protocol in Section 3. Section 4 formulates the transmission scheduling optimization model and the associated constraints. In Section 5, we show that the optimization problem is NP-complete and introduce the  $\kappa$ -FSOM heuristic algorithm. In Section 6, we show simulation results to demonstrate the performance of the  $\kappa$ -FSOM heuristic algorithm and compare it with state-of-the-art. Finally, the paper is concluded in Section 7.

## 2 Related Work

In this section, we review the literature on link scheduling and optimization in wireless networks. To solve different optimization goals, recent work considers throughput, energy consumption or time delay.

Extensive studies have been conducted on link scheduling in cellular networks. In [19], the link quality is predicted by an application framework which tracks the direction of travel of mobile phone at the BS. They develop energy-aware scheduling algorithms for different application workloads such as syncing or streaming. Some scheduling optimizations which consider multicast [13], quality-of-service assurance [26] and fair relaying with multiple antennas [12] are proposed to achieve optimal delay, capacity gain or network utility.

A vast majority of related work has focused on addressing the scheduling problem in the context of multi-hop networks [28, 11, 18]. However, the notion of fairness in multi-hop networks focuses on fair allocation of time slots among the links in each super frame, which is different from the fairness in data collection of MSN.

A scheduling for maximum throughput-utility in the single-hop networks with the constraint of network delay is presented in [17]. It establishes a delay-based policy for utility optimization. The policy provides deterministic worst-case delay bounds with total throughput-utility guarantee. The author in [16] proposes an opportunistic scheduling algorithm that guarantees a bounded worst case delay in single-hop wireless networks. However, those scheduling algorithms are not applicable in MSNs, because they do not consider the constraints of energy and fairness of collection. In [23], a sensing scheduling among sensor nodes is presented to maximize the overall Quality of Monitoring utility subject to the energy usage. The scheduling algorithm maximizes the overall utility which is to evaluate quality of sensor readings based on the greedy algorithm.

For body sensor network, Sidharth, *et al.* focus on polling-based communication protocols, and address the problem of optimizing the polling schedule to achieve minimal energy consumption and latency [15]. They formulate the problem as a geometric program and solve it by convex optimization.

To the best of our knowledge, there is no research focusing on link scheduling optimization for fair data collection in large-scale single-hop MSNs. The recent work in the literature is not applicable because they do not optimize the scheduling with the requirements of both energy consumption and data reception fairness.

The key difference of our work over previous scheduling optimization is that for a single-hop MSN which includes a large number of nodes, data collection is maximized in a fair way before they run out of energy. We formulate the transmission scheduling optimization model in Section 4.

## 3 Network Configuration and MAC Protocol

In this section, we first provide an overview of the network setup in the context of the bat monitoring application. Next, we propose extensions to IEEE 802.15.4 MAC protocol to improve its performance under our specific constraints.

### 3.1 Network Configuration

As depicted in Fig. 1.1(b), each bat is tagged with a collar that houses the Camazotz sensor node [10], a custom-designed light-weight sensing platform. The node embeds a GPS receiver, a three-axis accelerometer, air pressure sensors and a microphone. The node is powered by a battery and includes a solar panel for harvesting energy. The node will collect numerous sensor readings while the bat is in transit. The sensor readings are stored in a Secure Digital (SD) flash card. The bats are nocturnal hunters and are known to travel long distances (20 km in one night on average) in search for food. Typically the bats return to the roosting camps during the day as a swarm and remain there before heading out again at night. On occasion, individuals are known to be away for several days (up to several weeks) before returning back to the camp. Thus the total data payload on each node can vary up to a few MB.

We also deploy a number of powered BSs located at animal congregation areas. The stored data is offloaded to the BS via single-hop communication. Both the Camazotz node and the BS use the CC1101 radio, which uses the 915 MHz frequency band for communication. The BS is equipped with 3G connectivity to transfer the collected sensor readings to a central repository.

### 3.2 MAC Protocol

While the focus of this paper is on optimizing the data collection process, a secondary consideration is also the choice of the MAC protocol to be employed. As a support to  $\kappa$ -FSOM, the proposed MAC protocol gathers the node's information as the input to  $\kappa$ -FSOM at first. Then, the BS informs all the nodes the optimized scheduling which is output by  $\kappa$ -FSOM. Finally, the node transmits data in the scheduled time. Rather than reinventing the wheel, we propose to use a MAC that is heavily influenced by the widely used IEEE 802.15.4 MAC protocol [6]. In particular, the beacon mode of IEEE 802.15.4 super frame contains contention access period (CAP), contention-free period (CFP) and inactive period. Using 802.15.4 super frame the BS needs to transmit a beacon at the beginning of the frame periodically. The node competes for the channel to transmit data by random access in CAP after it receives the beacon. The slots in CFP are allocated to the node which competes the channel successfully if available. However, 802.15.4 MAC protocol is not feasible for scheduling optimization of MSNs. First, at the beginning of super frame, the BS does not have any information (PRR, energy, data size, etc) about the nodes. Thus, the time slots allocation in CFP is simply FCFS. However, the information from the nodes is basic element for  $\kappa$ -FSOM which is presented in Section 4. Second, the node competes the channel in CAP only when it receives the beacon. If the node misses the beacon due to the poor link quality, it has to keep radio on in order to get the beacon in the future frames. The node consumes much energy on idle listening. Even worse, the node which misses the beacon has no chance to compete for the channel no matter how small the energy or how good the link quality is. As a result, IEEE 802.15.4 MAC protocol does not achieve the fairness of data reception and energy consumption constraints.

The super frame used for  $\kappa$ -FSOM is composed of random channel access period (RCAP) and scheduled data transmission period (SDTP) (see Fig. 3.1). The two periods interchange periodically and are synchronized by the BS. Sensor

nodes do not keep track of the schedule while away from the BS, they only participate when in the range of the BS. The purpose of RCAP is that the BS gathers all the nodes' information (current energy levels and the PRR values) and makes a schedule for their data transmissions in SDTP. How the BS makes the transmission schedule is discussed in Section 5.

The BS keeps the radio on all the time. The radio of Camazotz is switched on and off by different duty cycles so that the node wakes up at different time points to avoid the initial channel access collision. Once a node wakes up in the RCAP, it turns on the radio to check the channel whether busy or idle through *Carrier Sensing (CS)*. If the channel is idle, the node transmits the BS a *Hello* packet which includes ID of the node, time stamp and current energy. If collision of *Hello* transmissions happen or the node detects the channel is busy, then it backs off in a manner similar to CSMA and turns off the radio during the backoff period. For the input of  $\kappa$ -FSOM, the BS calculates PRR from the RSSI of receiving the *Hello* packet and then replies a *HACK* which contains a packet sequence number and time of broadcasting Scheduling ACK (*SACK*) packet. The *SACK* includes IDs of the nodes which are scheduled to transmit and their allocated time slots. The schedule bundled in *SACK* is output by  $\kappa$ -FSOM which will be presented in the Section 4. After the node receives the *HACK* packet for it, it turns off the radio to sleep until the time to receive the *SACK*. With a large number of nodes, some of them may fail to communicate with the BS in RCAP of some frames. However, those nodes consume tiny energy due to a long sleeping time. In the SDTP, the time is partitioned to a certain number of slots with the purpose of node's scheduled transmission without collision.

In RCAP, the schedule which defines a time allocation  $\Delta T_i$  ( $\Delta T_i = Tend_i - Tstart_i$ ) for the node's transmission is sent to all the HACKed nodes through the *SACK* message. Note that different nodes can be allocated different transmission length, namely,  $\Delta T_{i1} \neq \Delta T_{i2}$  ( $\forall i1, i2 \in [1, N]$ ). Thus, the node is able to transmit multiple packets in  $\Delta T_i$ . After the nodes receive the *SACK*, they go to sleep until the time of  $Tstart_i$ . The nodes sleep permanently once either  $E_i$  is smaller than  $E_{td}$  or they have finished the transmission of all the data packets. Moreover, we set the guard interval between two nodes' data transmission time  $\Delta T_{i1}$  and  $\Delta T_{i2}$  to be smaller than the duration of CS,  $Tend_{i1} - Tstart_{i2} < T_{CS}$ . Any node arriving in SDTP detects the channel is busy, so it backs off the wakeup.

SDTP is driven by the schedule calculated by the  $\kappa$ -FSOM heuristic. The nodes find their transmission slot (*DATA* slot) within the super frame and only transmit during their scheduled time to prevent interference. The length of the *DATA* slots is selected by the scheduler and will typically allow for multiple packet transmissions. We use guard intervals to prevent packet collisions due to time-synchronization errors. With a large number of nodes, some of them may fail to communicate with the BS during RCAP. However, these nodes consume limited energy due to a long sleeping time during the SDTP.

According to the super frame structure, the node wakes up only when  $\Delta T_i$  is its allocated transmission time slots. It sleeps in the others' time slots. Specifically, this structure avoids the energy consumption on idle listening for the nodes and reduce the potential transmission collisions.

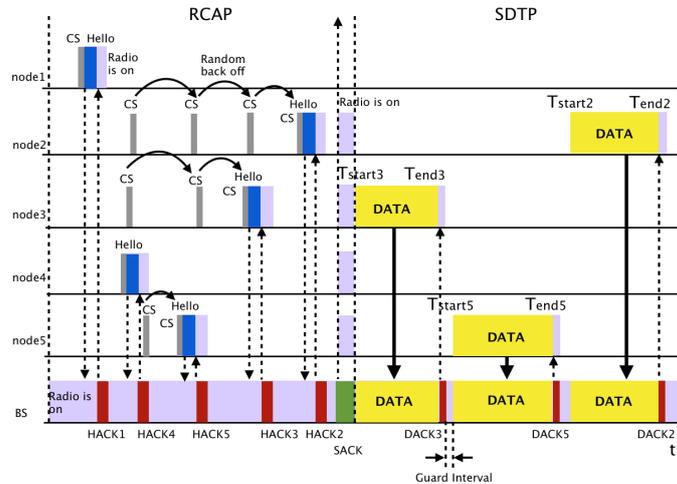


Figure 3.1: The timing relationship when five nodes communicate with the BS in RCAP and SDTP.  $Tstart_i$  and  $Tend_i$  stand for the starting and ending time of node  $i$ 's data transmission respectively.

## 4 Transmission Scheduling Optimization Model

Next, we present an abstract generalizable model of the network, which is used for the optimization model presented in this section. We assume that there are  $N$  nodes that directly communicate with the BS using single-hop communication. The nodes typically arrive in large groups but their exact arrival sequence is unknown. The residual energy of a node  $i$ , when it arrives at the camp is denoted by  $E_i^0$ . In order to prevent a node from completely depleting its battery, we assume that a node powers down if the residual energy goes below a certain threshold  $E_{td}$ .

In this paper, a node in such a state is referred to as a *dead node*. The wireless channel between each node and the BS is typically influenced by a variety of environmental factors and the motion of the node. The channel variability in turn influences the Packet Reception Rate (PRR) of the node. We estimate the PRR as a function of empirically collected RSSI traces from a real testbed as outlined in Section 6.

On the basis of Section 3, the BS aggregates the nodes and channel information in the RCAP in order to schedule the transmissions. In this section, we explain the basic notations and propose the scheduling optimization model under the constraints of reception fairness and node's remaining energy. We formulate the scheduling optimization as an Integer Linear Programming (ILP) problem.

### 4.1 Problem Formulation

According to the super frame as shown in Fig. 3.1, we divide the SDTP to a number of slots  $S$ , where,  $S = \sum_{i=1}^N \Delta T_i$ . Time slot  $j$  ( $j \in [1, S]$ ) is allocated by the BS to only one node's transmission for the purpose of avoiding collisions. Therefore, the allocated time  $\Delta T_i$  of the node  $i$  contains multiple time slots in

one super frame.  $\kappa$ -FSOM calculates optimal solutions for multiple frames so that the schedule is optimized globally.  $F$  is defined as the total number of super frames needed for all the nodes to finish their data transmissions. The sequence number of frame is denoted as  $f$  ( $f \in [1, F]$ ). We assume the residual energy when node  $i$  arrives at the camp is  $E_i^0$  ( $i \in [1, N]$ ). The PRR is indicated by  $q_i^f$ , where  $q_i^f \in [0, 1]$ . Additionally,  $q_i^f$  may change from one frame to the next due to the time-varying channel. We assume  $q_i^f$  does not change during the super frame since the flying foxes are not highly mobile in the camp. The data payload stored on each node is represented by  $\lambda_i$  and the fairness coefficient is  $\kappa$  where  $\kappa \in (0, 100\%]$ . Thus, the data reception fairness ensures that the number of data packets the BS collects from each node is not less than  $\kappa \cdot \lambda_i$ . We define the boolean variable  $x_{ij}^f$  as a transmission indicator for node  $i \in [1, N]$  associated with the slot  $j \in [1, S]$  in the super frame  $f \in [1, F]$ .  $x_{ij}^f = 1$  means node  $i$  has  $j$ th slot reserved for transmission in frame  $f$ .

The number of data packets received by the BS in a super frame is defined as  $\gamma_f$ , where

$$\gamma_f = \sum_{i=1}^N \sum_{j=1}^S x_{ij}^f \cdot q_i^f, (f \in [1, F]) \quad (4.1)$$

Similarly, for all super frames, the data received by the BS from any node  $i$  is defined as  $\alpha_i$ , where

$$\alpha_i = \sum_{f=1}^F \sum_{j=1}^S x_{ij}^f \cdot q_i^f, (i \in [1, N]) \quad (4.2)$$

The energy consumption of nodes arises from the transmissions in RCAP and SDTP as shown in Fig. 3.1. In this paper, we let  $e_{tx-hello}$ ,  $e_{rx-hack}$  and  $e_{rx-sack}$  be the energy consumption of transmitting one *Hello* packet, receiving one *HACK* and one *SACK* of the nodes, respectively. The  $e_{tx}$  represents energy consumption of transmitting one data packet. Due to the tiny energy consumption of carrier sensing compared to transmitting and receiving packets [5], we neglect the same in our model. The energy consumption of node  $i$  in the RCAP is  $\check{E}_A$ , where

$$\check{E}_A = e_{tx-hello} + e_{rx-hack} + e_{rx-sack} \quad (4.3)$$

We next define  $\check{E}_{Di}$  as the energy that node  $i$  consumes on data transmission in all super frames, where

$$\check{E}_{Di} = \sum_{f=1}^F \sum_{j=1}^S x_{ij}^f \cdot e_{tx}, (i \in [1, N]) \quad (4.4)$$

## 4.2 Optimization Model

Based on the notations in the problem formulation, we formulate  $\kappa$ -FSOM for finding the optimal schedules as follows:

$$\begin{aligned} & \text{maximize} && \sum_{f=1}^F \gamma_f \\ & \text{subject to} : && E_i^0 - \sum_{f=1}^F (\check{E}_A \cdot \varphi_i^f) - \check{E}_{Di} \geq E_{td}, \quad (i \in [1, N]) \end{aligned} \quad (4.5)$$

$$\alpha_i \geq \kappa \cdot \lambda_i, \quad (i \in [1, N], \kappa \in (0, 1]) \quad (4.6)$$

$$\alpha_i \leq \lambda_i, \quad (i \in [1, N]) \quad (4.7)$$

$$x_{ij}^f \leq 1, \quad (i \in [1, N], j \in [1, S], f \in [1, F]) \quad (4.8)$$

$$\sum_{i=1}^N x_{ij}^f \leq 1, \quad (j \in [1, S], f \in [1, F]) \quad (4.9)$$

$$\begin{aligned} \lambda_i - \sum_{g=1}^f \sum_{w=1}^j x_{iw}^g \cdot q_i^g &\geq v_{ij}^f, \\ (i \in [1, N], j \in [1, S], f \in [1, F]) \end{aligned} \quad (4.10)$$

$$v_{ij}^f \geq v_{ij'}^f, \quad (j' \geq j, j \in [1, S]) \quad (4.11)$$

$$v_{ij}^f \geq v_{ij'}^g, \quad (g \geq f, j' \geq j, j \in [1, S], f \in [1, F]) \quad (4.12)$$

$$\sum_{a=1}^{F-f} \varphi_i^{f+a} \leq v_{ij}^f, \quad (i \in [1, N], j \in [1, S]) \quad (4.13)$$

$$x_{ij}^f \leq \varphi_i^f, \quad (i \in [1, N], j \in [1, S], f \in [1, F]) \quad (4.14)$$

Objective function of the optimization model is to maximize  $\gamma_f$  of all super frames. Constraint (4.5) specifies the minimum remaining energy to be above  $E_{td}$ . A node stops accessing the channel after all its data has been transmitted or constraint (4.5) is violated. Consequently, it does not waste energy in RCAP in subsequent super frames. For this purpose,  $\varphi_i^f$  is defined as an indicator of RCAP in a super frame for the node. If the node  $i$  does not compete for the channel in the RCAP of frame  $f$ ,  $\varphi_i^f$  is equal to 0.  $\sum_{f=1}^F (\check{E}_A \cdot \varphi_i^f)$  indicates the energy consumption of the node in the RCAP of all super frames.

Constraint (4.6) guarantees that the BS receives sufficient data packets to meet the fairness requirement. Constraint (4.7) limits the value of  $\alpha_i$  by the total payload  $\lambda_i$ . Constraints (4.8) and (4.9) specify that at any data transmission time slot only one node communicates with the BS to prevent transmission collisions.

The only unknown is the total number of super frames during which a node is required to transmit. In other words,  $\varphi_i^f$  is not known. To determine  $\varphi_i^f$ , we define a variable  $v_{ij}^f$  for node  $i$  at any slot  $j$  of frame  $f$ .

Accordingly, constraint (4.10) presents whether node  $i$  has stopped the data transmission or not.  $\sum_{g=1}^f \sum_{w=1}^j x_{iw}^g \cdot q_i^g$  is the total received packets until the current slot  $j$  of frame  $f$ . If the amount of data packets received from node  $i$  matches the size of payloads  $\lambda_i$ ,  $v_{ij}^f$  is equal to 0. Constraints (4.11) and

(4.12) ensure the future slots  $j'$  and frames  $g$  have  $v_{ij}^f = 0$  if  $\lambda_i$  packets have been received from node  $i$ . Constraint (4.13) guarantees all  $\varphi_i^f$  of the future super frames is 0 if  $v_{ij}^f = 0$ . As a result, the remaining energy of node  $i$  which is restricted by the RCAP indicator  $\varphi_i^f$  stops decreasing in constraint (4.5). Constraint (4.14) ensures that the node  $i$  stops data transmission if  $\varphi_i^f = 0$ .

## 5 $\kappa$ -FSOM Heuristic Algorithm

In this section, we first show that  $\kappa$ -FSOM is NP-complete. Next, a  $\kappa$ -FSOM heuristic algorithm is proposed to approximate the optimal solution.

Maximizing the collected data presented in  $\kappa$ -FSOM is a typical 0-1 Multiple Knapsack Problem (MKP) [14]. We reduce an instance of a MKP to our scheduling optimization problem by assigning  $\Delta T_i$  to each knapsack. Therefore, the capacity of the knapsack is equal to  $\Delta T_i$ . The items to be put in knapsacks are data packets whose size is prorated by  $q_i^f$ . The parameters of the energy and fairness conditions (constraint (4.5) and (4.6)) are chosen so that they are satisfied by any placement of items. In this way, optimal placement of items in knapsacks is reduced to such an instance of our scheduling problem. Since the problem is obviously an NP problem, this shows that our scheduling problem presented in the Section 4 is NP-complete.

We propose a heuristic algorithm to approximate the optimal solution of  $\kappa$ -FSOM. Due to the effect of  $E_i^f$  and  $q_i^f$  to the schedule making, a ratio of the link quality and remaining energy of the node  $i$  is denoted as  $\eta_i^f$ , where

$$\eta_i^f = \frac{q_i^f}{E_i^f}, \forall i \in [1, N], \forall f \in [1, F] \quad (5.1)$$

Accordingly,  $E_i^f$  is obtained by

$$E_i^f = E_i^0 - \sum_{f'=1}^f (\check{E}_A \cdot \varphi_i^{f'}) - \sum_{f'=1}^f \sum_{j=1}^S x_{ij}^{f'} \cdot e_{tx} \quad (5.2)$$

The motivation of calculating  $\eta_i^f$  is to prioritize the nodes based on both the link quality and remaining energy. The  $\kappa$ -FSOM heuristic gives a high transmission priority to the node with larger  $\eta_i^f$ . This method achieves large data reception because for the nodes with the same  $q_i^f$ , the node with the smallest  $E_i^f$  gets higher transmitting priority. Similarly, for the nodes with the same  $E_i^f$ , one with higher  $q_i^f$  has higher priority.

In our heuristic, the node works in three states, Access & Data transmission (AD), NonAccess (NA) and NonData (ND). In AD state, the node competes for the channel in RCAP and transmits data in SDTP as shown in Fig. 3.1. In NA state, the node neither accesses the channel nor transmits data but only receives the *SACK* packets for the purpose of saving energy in the super frame. More importantly, none of the nodes which are in the NA state transmit data given that no time slots are allocated to them. This helps more nodes achieve fairness. In ND state, the node does not turn on the radio and remains in sleep mode.

---

**Algorithm 1**  $\kappa$ -FSOM Heuristic Algorithm

---

```
1: nodes are in AD state and compete the channel
2: The BS calculates  $\eta_i^f$  for the node  $i$ ,  $\forall f \in [1, F]$ 
3: The BS sorts the nodes by  $\eta_i^f$ , then  $\eta_i^f \geq \eta_{i'}^f, (i \neq i', i' \in [1, N])$ 
4: The BS schedules the node  $i$  to transmit
5: if  $\alpha_i \geq (\kappa \cdot \lambda_i)$  then
6:   The node  $i$  goes to NA state
7:   The BS schedules the next one to transmit
8: else
9:   The node  $i$  remains in AD state
10: end if
11: if every node has  $\alpha_i \geq (\kappa \cdot \lambda_i) \quad \forall i \in [1, N]$  then
12:   All the nodes transfer to AD state
13:   The BS calculates  $\eta_i^f$  for each node
14:   The BS sorts the nodes by  $\eta_i^f$ , then  $\eta_i^f \geq \eta_{i'}^f, (i \neq i', i' \in [1, N])$ 
15:   if  $E_i \geq E_{td}$  then
16:     The BS schedules the node  $i$  to transmit
17:   else
18:     The node  $i$  changes state to the ND
19:     The BS schedules the next one to transmit
20:   end if
21:   if  $\alpha_i < \lambda_i$  then
22:     The node  $i$  remains in AD state
23:   else
24:     The node  $i$  changes state to the ND
25:   end if
26: end if
```

---

The  $\kappa$ -FSOM heuristic develops two steps to maximize the data reception with  $\eta_i^f$ . It is implemented as shown in Algorithm 1.

At first step, all the nodes work in AD state and the BS schedules the node  $i$  ( $i \in [1, N]$ ) which has maximum  $\eta_i^f$  to transmit data at first in each super frame. The BS records the number of data packets from the node. Once the node  $i$  meets the fairness of data reception (constraint (4.6)), it transfers the state from the AD to the NA. The benefit of NA state is to reduce the channel competitions since the number of nodes competing the channel is decreased. Certainly, after the first step, all the nodes have at least  $(\kappa \cdot \lambda_i)$  number of data packets being transmitted successfully and the fair reception of data is achieved. At the second step, all the nodes change the state from the NA to AD. Then, the BS schedules the node with largest  $\eta_i^f$  to transmit first. To maximize data reception, the node  $i$  works in the AD state until either the constraint of (4.5) or (4.7) is no longer holds. Moreover, if the constraint of (4.5) or (4.7) is not fulfilled by the node  $i$ , its state is transferred to the ND. By using this approach, the number of data packets collected by the BS is maximized, meanwhile, the energy and fairness requirements are both achieved. Operationally, the working states vary between the two steps. Transition graph is shown in Fig. 5.1.

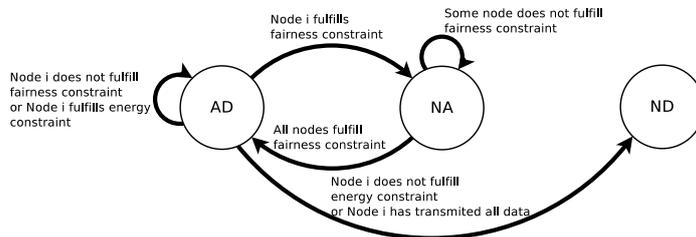


Figure 5.1: The working states transition of node  $i$

Table 6.1: Parameters of the nodes

|  |          |           |
|--|----------|-----------|
| <b>Maximum number of bats</b>                | $N$      | 300       |
| <b>Working temperature</b>                   | $T_A$    | 25 °C     |
| <b>Working frequency</b>                     | $Freq$   | 915 MHz   |
| <b>Supply voltage during radio operation</b> | $V_{cc}$ | 3 V       |
| <b>Transmitting current</b>                  | $I_{tx}$ | 35 mA     |
| <b>Receiving current</b>                     | $I_{rx}$ | 15 mA     |
| <b>Remaining energy threshold</b>            | $E_{td}$ | 1.67 mJ   |
| <b>Bit Rate</b>                              | $R_b$    | 19.2k bps |

## 6 Simulation and Performance Evaluation

Given optimal schedules from  $\kappa$ -FSOM in AMPL, it is observed how the performance of heuristic algorithm approximates them. Moreover, we study the performance of the  $\kappa$ -FSOM heuristic algorithm in both static and dynamic scenarios which are used to evaluate the behavior of flying foxes by MATLAB. A series of experiments had been implemented in a previous work within the same project to obtain the RSSI [10]. The RSSI trace from the Camazotz testbed attached to the flying foxes is imported to the simulations, which provides an environment to conduct repeatable simulations based on empirical data. Finally, we evaluate the impact of the fairness coefficient  $\kappa$  on network performance.

### 6.1 Simulation Configuration

The data collection network in the simulation is composed of one BS and  $N$  nodes ( $N \in [10, 300]$ ) which are randomly distributed within the open camp. Based on hardware setup of the Camazotz, the node communicates with the BS using CC1101 radio transceiver which provides a GFSK communication in the 915 MHz band. According to features of the CC1101 radio [7], configuration of the nodes is outlined in Table 6.1. A data packet which contains time, GPS and biological information has 32 bytes. The length of one *Hello* packet is 10 bytes. Equally, *HACK* and *SACK* have the same length as *Hello*. Therefore, we have

$$e_{tx-hello} = V_{cc} \cdot I_{tx} \cdot \frac{10 \times 8}{R_b} = 0.03mJ \quad (6.1)$$

$$e_{rx-hack} = e_{rx-sack} = V_{cc} \cdot I_{rx} \cdot \frac{10 \times 8}{R_b} = 0.01mJ \quad (6.2)$$

$$e_{tx} = V_{cc} \cdot I_{tx} \cdot \frac{32 \times 8}{R_b} = 0.1mJ \quad (6.3)$$

According to the energy initialization of sensor nodes in simulations [27],  $E_i^0$  in this work is given by a normal distribution with the mean value of 50 Joule. Fig. 6.1 shows the values of  $E_i^0$  when  $N = 300$  as an example. The actual energy of Camazotz supports several days. However, in our simulations, the value of  $E_i^0$  is given purposely so that some dead nodes which run out of energy will be observed among different scheduling algorithms. The RSSI trace is recorded by the Camazotz whose sampling rate is 4 samples per second. Thus, the node has 14400 RSSI samples for one hour. Fig. 6.2 depicts a 780 seconds segment which includes 3120 RSSI points. The variation of RSSI is known that most of samples are between -70 dbm and -85 dbm. A few of them are higher than -70 dbm and smaller than -85 dbm. In this paper, we convert the RSSI to PRR for the  $q_i^f$  by the experimental results of PRR-RSSI relationship [22].

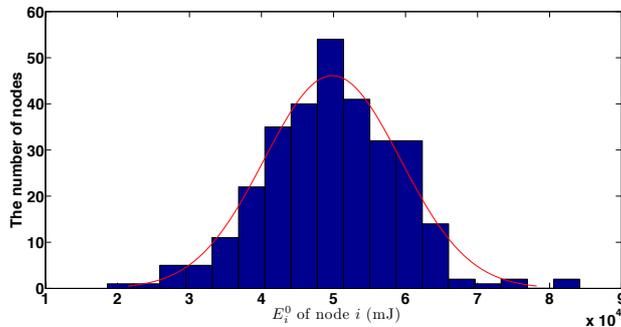


Figure 6.1: Initial energy of the nodes

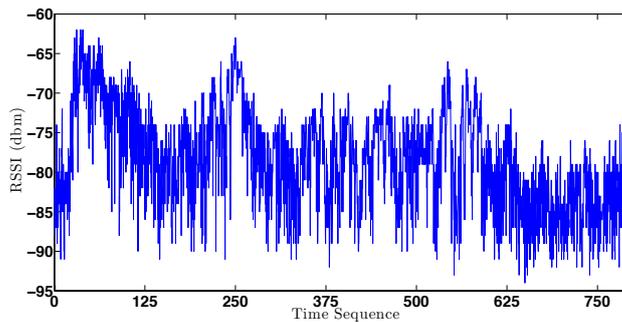


Figure 6.2: RSSI trace of the node by the Camazotz on the flying fox

We evaluate three performance metrics: the total number of data packets received by the BS (data reception), the number of fair nodes and dead nodes. Specifically, the *fair nodes* denote the number of nodes such that node  $i$  fulfills  $\alpha_i \geq \kappa \cdot \lambda_i$  (the fairness constraint (4.6)). At first, we test the  $\kappa$ -FSOM heuristic based on the amount of data a node gathers during one day. We compare the performance of heuristic with  $\kappa$ -FSOM optimal solutions when each node

carries the data payload of 80 KB (daily based). Then, we simulate the  $\kappa$ -FSOM heuristic algorithm in the static and dynamic scenarios. The performance in both scenarios provide support to the proposed heuristic. In the static scenario, we assume all the nodes are in the camp from the start of experiment to the end. In the dynamic case, the nodes arrive back at the camp at different times. Since the number of nodes communicating with the BS in a short time is small, we increase the data payload to 300 KB in order to explore the limits of the scheduling algorithms. For this reason, a node occupies the channel longer while more nodes enter the camp in the dynamic scenario.

To evaluate the performance of  $\kappa$ -FSOM heuristic algorithm in the static and dynamic scenarios, two Greedy scheduling algorithms and FCFS algorithm are constructed in the numerical investigations. Because two basic elements used in  $\kappa$ -FSOM are the remaining energy represented by  $E_i^f$  and link quality  $q_i^f$  of node, the Greedy scheduling algorithms are formulated by them. The first Greedy algorithm is called Low Energy Greedy (LEG), namely, the transmission schedule is based on the  $E_i^f$  of node. Lower  $E_i^f$ , higher priority of transmission at frame  $f$ . High PRR Greedy (HPG) is the second algorithm where the node with higher  $q_i^f$  has higher priority. We compare them with the  $\kappa$ -FSOM heuristic algorithm with  $\kappa = 10\%$ ,  $50\%$  and  $90\%$ .

## 6.2 Simulation Results

### Comparing to optimal schedules

To compare to the optimal schedules shown in  $\kappa$ -FSOM, we assess the performance of the  $\kappa$ -FSOM heuristic algorithm when they operates in ten small-scale networks where the number of nodes is increased from 1 to 10. This initial comparison makes us aware of the performance difference between optimal solutions and heuristic. The node  $i$  carries 80 KB data, so  $\lambda_i = 2500$ . In fact, the comparison is not affected by different  $\kappa$  values, thus we choose  $\kappa=50\%$  for both the optimal schedules and heuristic. The optimal schedules achieve a maximum number of received data packets with the fairness and remaining energy constraints. They are constructed using AMPL and a state of the art ILP solver, Cplex 12.5, in a 2.7 GHz Intel core processor with 8 GB of memory.

Table 6.2 summarizes running time, the number of collected data packets and fair nodes. It is also found that there is no dead node in all tests. On data reception, the  $\kappa$ -FSOM heuristic and optimal solution have the maximum difference which is 719 when  $N = 9$ . On average, the number of packets in our heuristic is less than the AMPL output by around 1.8%. The  $\kappa$ -FSOM heuristic algorithm guarantees exactly the same number of fair nodes as optimal schedules.

### Static scenario

Fig. 6.3 and 6.4 show the performance of these four scheduling algorithms on the data reception and fairness. When there are only 10 nodes in the network, they have pretty similar performance. However, the FCFS, LEG and HPG collect 92.2%, 91.9% and 85.4% less data packets than the  $\kappa$ -FSOM heuristic with the increase of nodes. The number of fair nodes of our heuristic is more than the ones of FCFS, LEG and HPG for 174, 170, 147 nodes when  $\kappa = 50\%$  and  $N =$

Table 6.2: Comparison between the optimal solutions and the  $\kappa$ -FSOM heuristic

| Number of nodes | AMPL (Cplex)           |            |              | $\kappa$ -FSOM heuristic |            |              |
|-----------------|------------------------|------------|--------------|--------------------------|------------|--------------|
|                 | Collected data packets | Fair nodes | Running time | Collected data packets   | Fair nodes | Running time |
| 1               | 2499                   | 1          | 1 s          | 2481                     | 1          | 0.029 s      |
| 2               | 4999                   | 2          | 4 s          | 4969                     | 2          | 0.097 s      |
| 3               | 7499                   | 3          | 17 s         | 7469                     | 3          | 0.036 s      |
| 4               | 9998                   | 4          | 50 s         | 9922                     | 4          | 0.042 s      |
| 5               | 12498                  | 5          | 1 m 5 s      | 12477                    | 5          | 0.041 s      |
| 6               | 14998                  | 6          | 5 m 15 s     | 14340                    | 6          | 0.041 s      |
| 7               | 17498                  | 7          | 58 m 47 s    | 17288                    | 7          | 0.05 s       |
| 8               | 19997                  | 8          | 5 h 49 m     | 19792                    | 8          | 0.051 s      |
| 9               | 22498                  | 9          | 17 h 25 m    | 21779                    | 9          | 0.055 s      |
| 10              | 24997                  | 10         | 30 h 5 m     | 24555                    | 10         | 0.072 s      |

300. The reason is that the LEG and HPG make the schedule based on either the  $E_i^f$  or the  $q_i^f$ . The LEG scheduling fails when the low energy nodes have poor link quality since it schedules them to transmit at first. However, although the nodes with high PRR are not scheduled because of low priority, they still consume energy on channel competitions in RCAP. With the HPG algorithm, the nodes with high PRR occupy the SDTP for multiple super frames until they finish the transmissions. It gives rise to a number of dead nodes which have low  $E_i^f$ . Nevertheless, those nodes can potentially gain higher data reception. In contrast, the  $\kappa$ -FSOM heuristic makes the schedule regarding to  $\eta_i^f$ . The first step of our heuristic makes the nodes fulfill fairness constraint (4.6) and the second step is to maximize the data reception. We have shown that it achieves better performance than the greedy and nonscheduled ones.

We find the data reception and fair nodes of FCFS, LEG and HPG do not vary too much from  $N = 100$  to  $300$ . The reason is indicated by Fig. 6.5. It shows the FCFS, LEG and HPG have more dead nodes starting from  $N = 50$ . At the maximum, the difference between them and our heuristic are 171, 154 and 173 nodes. The reason is that most of the node's energy is consumed during the RCAP in FCFS, LEG and HPG, however, they are not scheduled to transmit in the super frames because the high priority nodes have not finished their data transmissions. The BS fails to collect their data before they die.

According to the  $\kappa$ -FSOM heuristic algorithm, we know that  $\kappa$  is a crucial variable which affects the states transition of node  $i$ . The performance of our heuristic varies with different  $\kappa$  value. As shown, they are similar for  $\kappa = 10\%$ ,  $50\%$  and  $90\%$  when  $N$  is 10. From  $N = 50$  to  $N = 300$ ,  $\kappa = 10\%$  performs better than  $50\%$  and  $90\%$ . The reason of this difference is that any node which is scheduled to transmit occupies more super frames when  $\kappa$  is increased due to the fairness constraint (4.6). It makes the other nodes compete the channel in RCAP repeatedly and waste much energy.

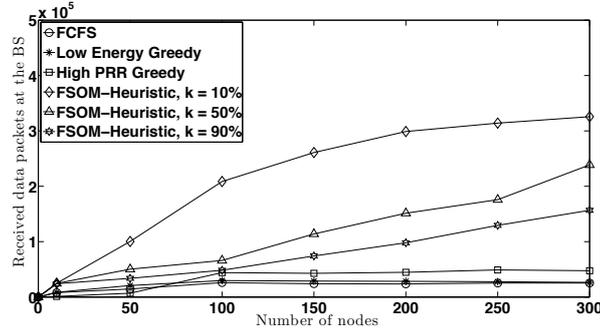


Figure 6.3: The data packets collected by the BS

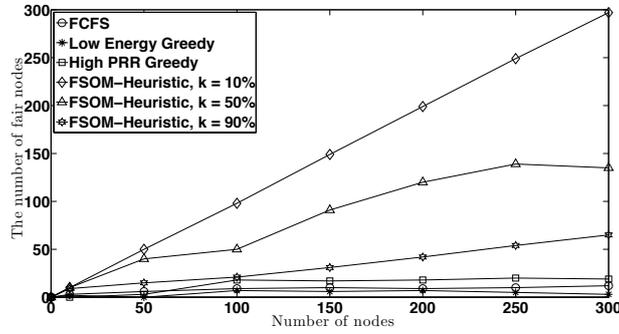


Figure 6.4: Number of fair nodes

### Dynamic scenario

In this set of experiments, we test the scheduling algorithms when nodes fly back as a swarm. Since the arrival pattern of flying foxes in real world is not known, we assume the inter-arrival time of nodes is exponentially distribution which used to model situations involving the random time between arrivals to a service facility [24]. The node has data payload of 300 KB which has been illustrated in the network configuration. From Fig. 6.6, we find that the  $\kappa$ -FSOM heuristic has up to 37.1 times as many collected data packets as FCFS and HPG schedules data packets at most. It outperforms LEG by 5 times as well. The reason is explained by Fig. 6.7 and 6.8. The FCFS, LEG and HPG have less fair nodes and more dead nodes than our heuristic, which means the incoming nodes fail to transmit since the previous node have not finished the transmission. It causes their energy to be depleted very soon. Moreover, in Fig. 6.7, we observe the difference of fairness which is achieved by different  $\kappa$  is smaller. That is because the BS schedules a smaller number of nodes in one super frame in dynamic scenario than the nodes in static scenario. The first step of heuristic is completed faster, hence more nodes achieve fairness constraint in dynamic scenario. Likewise, the number of dead nodes in our heuristic has small difference in Fig. 6.8. Due to the increase of  $\lambda_i$  in this scenario, there are 16 dead nodes with the  $\kappa = 90\%$  in the  $\kappa$ -FSOM heuristic at the maximum.

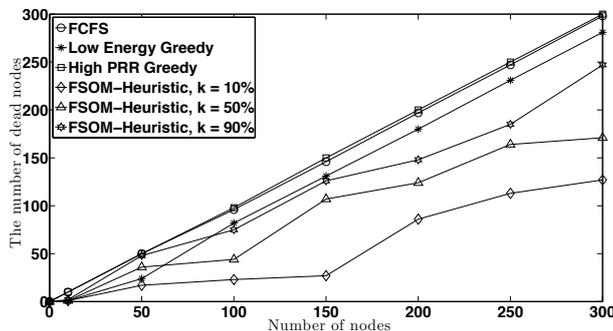


Figure 6.5: Number of dead nodes

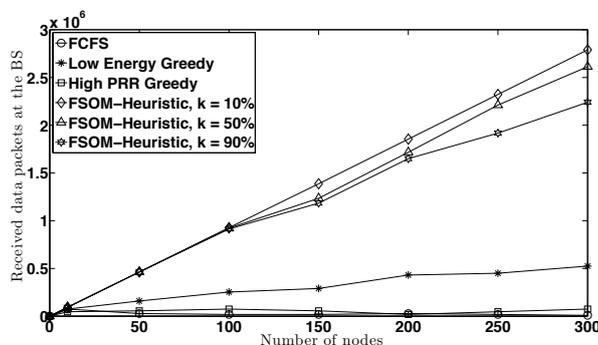


Figure 6.6: The data packets collected by the BS

### 6.3 Effect of Fairness coefficient $\kappa$

Based on the preceding simulations, it is observed that different  $\kappa$  affects the performance of our heuristic. Essentially, the  $\kappa$  decides the fairness level in  $\kappa$ -FSOM. In this experiment, we analyze the impact of  $\kappa$  in the static scenario with 300 nodes. Specifically, the  $\kappa$  is varied from 10% to 100%. The performance of data packets reception, fair nodes and dead nodes are shown in Fig. 6.9, 6.10 and 6.11 respectively.

As it can be seen in Fig. 6.9 the collected data packets do not have too much variation with  $\kappa$  from 10% to 40%. This is because all of the nodes fulfill the fairness constraint (4.6) which is decided by  $\kappa$  in the first step of heuristic. The data reception is maximized in the second step. However, starting from  $\kappa = 50%$  to  $\kappa = 100%$ , the data reception at the BS drops. From Fig. 6.10 and 6.11, we find the number of fair nodes decreases and the number of dead nodes increases at the same time. That means the node has to transmit more data packets as the fairness level is raised. It causes a number of nodes death due to the  $\tilde{E}_A$ . The number of fair nodes is reduced correspondingly.

Generally, we find that the scheduling with smaller  $\kappa$  achieves larger number of fair nodes. However, since the BS gives higher priority to the larger  $\eta_i^f$  node after all nodes satisfy fairness constraint, it does not guarantee most of data can be collected from each node. Therefore,  $\kappa$  changed from 40% to 50% keeps

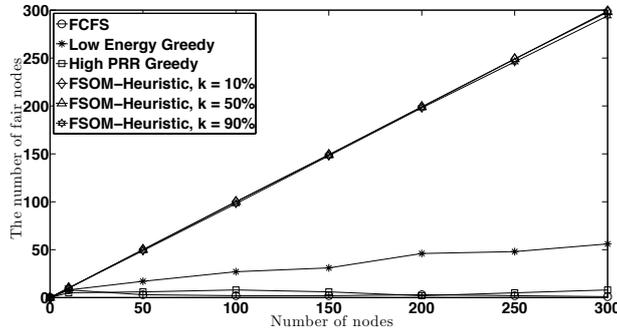


Figure 6.7: Number of fair nodes

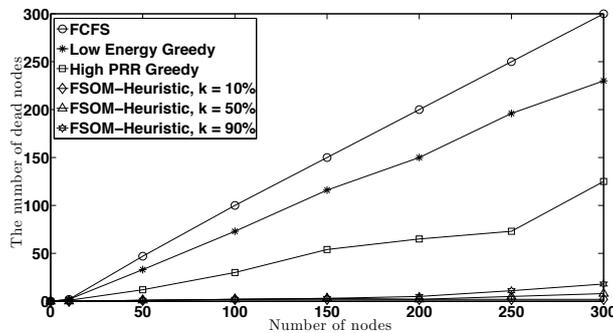


Figure 6.8: Number of dead nodes

a balance between the data reception from each node and total number of dead nodes.

## 7 Conclusion and Future Work

In this paper, we have proposed and evaluated  $\kappa$ -FSOM which is a fair link scheduling optimization model with the objective of maximizing the data reception in the energy-aware data collection of MSN. The super frame structure is developed for the BS to collect data from the nodes. We have proved that the scheduling optimization of  $\kappa$ -FSOM is an NP-complete problem. Therefore, the  $\kappa$ -FSOM heuristic algorithm is proposed to approximate the optimal solution in polynomial time. The  $\kappa$ -FSOM heuristic algorithm schedules the transmissions of data senders with the  $\eta_i^f$  and three working states in two steps. With the application of flying foxes monitoring, we have shown the numerical performance of heuristic algorithm based on the RSSI traced by the Camazotz testbed. We have compared the  $\kappa$ -FSOM heuristic with the optimal schedules of  $\kappa$ -FSOM and presented extensive simulations incorporating both static and dynamic scenarios. Specifically,  $\kappa$ -FSOM provides an optimal scheduling to the data collection in MSNs.

For future work, we plan to investigate the constraint of relative fairness

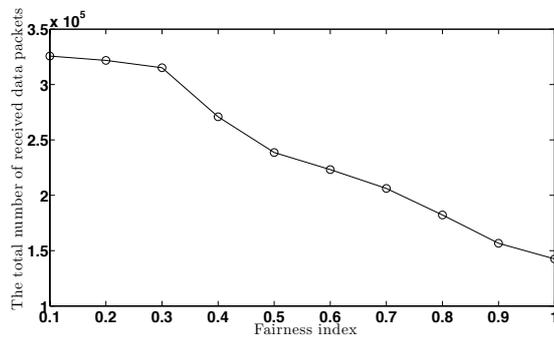


Figure 6.9: The data packets collected by the BS with  $\kappa$

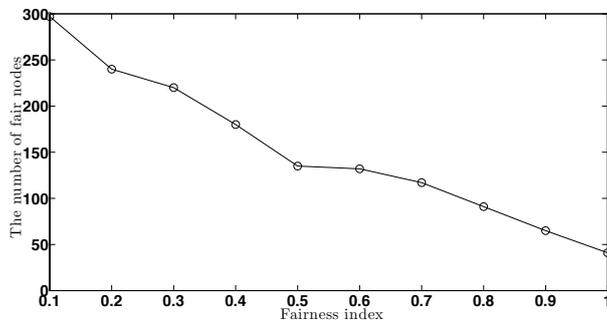


Figure 6.10: Number of fair nodes with  $\kappa$

for data collection of MSNs. The relative fairness which signifies the margin of collected data volumes among the nodes. Meanwhile, we will continue the data collection experiments by using  $\kappa$ -FSOM with the Camazotz for the flying foxes.

## Acknowledgment

This work was supported by the Australian Research Council Discovery Grant DP110104344 and the Batmon Project in CSIROs Sensor and Sensor Networks Transformation Capability Platform. The authors thank Dr. Andreas Reinhardt for his constructive comments for this paper.

## Bibliography

- [1] D. Chabot. *Systematic evaluation of a stock unmanned aerial vehicle (UAV) system for small-scale wildlife survey applications*. PhD thesis, MCGILL UNIVERSITY, 2010.
- [2] Peter Corke, Tim Wark, Raja Jurdak, Wen Hu, Philip Valencia, and Darren Moore. Environmental wireless sensor networks. *Proceedings of the IEEE*, 98(11):1903–1917, 2010.

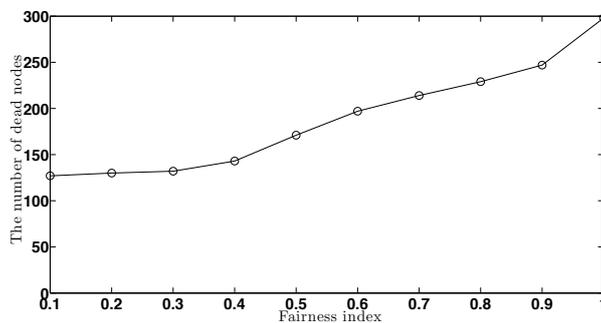


Figure 6.11: Number of dead nodes with  $\kappa$

- [3] Karthik Dantu, Bryan Kate, Jason Waterman, Peter Bailis, and Matt Welsh. Programming micro-aerial vehicle swarms with karma. *ACM SenSys*, pages 121–134, 2011.
- [4] Vladimir Dyo, Stephen A Ellwood, David W Macdonald, Andrew Markham, Cecilia Mascolo, Bence Pásztor, Salvatore Scellato, Niki Trigoni, Ricklef Wohlers, and Kharsim Yousef. Evolution and sustainability of a wildlife monitoring sensor network. *ACM SenSys*, pages 127–140, 2010.
- [5] Sinem Coleri Ergen. Zigbee/ieee 802.15. 4 summary. *UC Berkeley, September*, 10, 2004.
- [6] IEEE 802 Working Group et al. Standard for part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (lr-wpans). *ANSI/IEEE 802.15*, 4, 2003.
- [7] Texas Instruments, April 2013.
- [8] M. Israel. A uav-based roe deer fawn detection system. In *Proceedings of the International Conference on Unmanned Aerial Vehicle in Geomatics (UAV-g)*(H. Eisenbeiss, M. Kunz, and H. Ingensand, eds.), volume 38, pages 1–5, 2011.
- [9] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. *ACM Sigplan Notices*, 37(10):96–107, 2002.
- [10] Raja Jurdak, Philipp Sommer, Branislav Kusy, Navinda Kottege, Christopher Crossman, Adam Mckeown, and David Westcott. Camazotz: multi-modal activity-based gps sampling. *ACM IPSN*, pages 67–78, 2013.
- [11] Mathieu Leconte, Jian Ni, and Rayadurgam Srikant. Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks. *Networking, IEEE/ACM Transactions on*, 19(3):709–720, 2011.
- [12] Yicheng Lin and Wei Yu. Fair scheduling and resource allocation for wireless cellular network with shared relays. *Selected Areas in Communications, IEEE Journal on*, 30(8):1530–1540, 2012.

- [13] Tze-Ping Low, Man-On Pun, Y-WP Hong, and C-CJ Kuo. Optimized opportunistic multicast scheduling (oms) over wireless cellular networks. *Wireless Communications, IEEE Transactions on*, 9(2):791–801, 2010.
- [14] Silvano Martello and Paolo Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley and Sons, Inc., 1990.
- [15] Sidharth Nabar, Jeffrey Walling, and Radha Poovendran. Minimizing energy consumption in body sensor networks via convex optimization. *International Conference on Body Sensor Networks (BSN)*, pages 62–67, 2010.
- [16] Neely. Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks. *IEEE INFOCOM*, pages 1728–1736, 2011.
- [17] Michael J Neely. Delay-based network utility maximization. *IEEE INFOCOM*, pages 1–9, 2010.
- [18] Katerina Papadaki and Vasilis Friderikos. Approximate dynamic programming for link scheduling in wireless mesh networks. *International Journal of Computers and Operations Research*, 35(12):3848–3859, 2008.
- [19] Aaron Schulman, Vishnu Navda, Ramachandran Ramjee, Neil Spring, Pralhad Deshpande, Calvin Grunewald, Kamal Jain, and Venkata N Padmanabhan. Bartendr: a practical approach to energy-aware cellular data scheduling. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 85–96. ACM, 2010.
- [20] Louise A Shilton, Peter J Latch, Adam Mckeown, Petina Pert, and David A Westcott. Landscape-scale redistribution of a highly mobile threatened species, *pteropus conspicillatus* (chiroptera, pteropodidae), in response to tropical cyclone larry. *Austral Ecology*, 33(4):549–561, 2008.
- [21] Arpita Sinha, Antonios Tsourdos, and Brian White. Multi uav coordination for tracking the dispersion of a contaminant cloud in an urban region. *European Journal of Control*, 15(3):441–448, 2009.
- [22] K Srinivasa and P Levis. Rssi is under appreciated. *The Third Workshop on Embedded Networked Sensors (EmNets)*, 2006.
- [23] ShaoJie Tang and Lei Yang. Morello: A quality-of-monitoring oriented sensing scheduling protocol in sensor networks. *IEEE INFOCOM*, pages 2676–2680, 2012.
- [24] Daniel Willkomm, Sridhar Machiraju, Jean Bolot, and Adam Wolisz. Primary user behavior in cellular networks and implications for dynamic spectrum access. *Communications Magazine, IEEE*, 47(3):88–95, 2009.
- [25] Robert Wood, Radhika Nagpal, and Gu-Yeon Wei. Flight of the robobees. *Scientific American*, 308(3):60–65, 2013.
- [26] Dapeng Wu and Rohit Negi. Downlink scheduling in a cellular network for quality-of-service assurance. *Vehicular Technology, IEEE Transactions on*, 53(5):1547–1557, 2004.

- [27] Debdhanit Yupho and Joseph Kabara. The effect of physical topology on wireless sensor network lifetime. *Journal of Networks*, 2(5):14–23, 2007.
- [28] Yaqin Zhou, Xiang-Yang Li, Min Liu, Zhongcheng Li, Shaojie Tang, Xufei Mao, and Qiuyuan Huang. Distributed link scheduling for throughput maximization under physical interference model. *IEEE INFOCOM*, pages 2691–2695, 2012.