Battery charge scheduling for SUAV mobile ad-hoc network. A bio-inspired approach

Mauro Belgiovine

March 29, 2017

1 Introduction

The persistent area coverage problem using Small Unmanned Aerial Vehicles (SUAVs) has been addressed using different approaches in [1]. In this work, I wanted to further explore strategies based on a distributed, stimulus-based approach to define the charging and discharging activity schedule during the network lifetime.

2 Scenario and network configuration

The network is modeled as described in [1], if not stated differently. For these experiments, we considered a mobile ad-hoc network formed by Small Unmanned Aerial Vehicles (SUAVs) with the following features:

- For simplicity, each SUAV is assumed to be in a fixed position, since the positioning for the optimal area coverage has already been solved.
- The area A to cover is divided into hexagonal cells, where each SUAV is considered to be both at the center of his cell and on a vertex of the adjacent cells [3], in order to maximize the area covered by the SUAV swarm.
- We consider N SUAVs and M charging stations, with $M << N$.
- We define a fraction k of the total area to be persistently covered as a requirement of our network activity.
- Each SUAV can communicate with its neighbors only (the ones at 1-hop distance).

Each SUAV energy level E is expressed as a real value between 0 and 1, representing the fraction of the residual energy with respect to full charge battery level. The network lifetime is discretized into time slots (T_{slot}). Each SUAV's energy is affected by a discharging factor α when is in active state and a charging factor β when is not, both constant and relative to a single time slot. At each T_{slot} (in this case, $T_{slot} = 1s$) each SUAV energy level i at time j is updated using this expression:

$$\begin{aligned} E(i,j) = E(i,j-1) - (\alpha \cdot T_{slot} \cdot s(i,j)) + (\beta \cdot T_{slot} \cdot (1 - s(i,j))) \\ -(OP(h) \cdot s(i,j-1) \cdot (1 - s(i,j))) \\ -(OP(h) \cdot s(i,j) \cdot (1 - s(i,j-1))) \end{aligned}$$

Where $s(i,j)$ is the SUAV's status at time j (1 for discharge or 0 for charge state) and OP(h) is the ascending/descending operation $OP(h) = \gamma \cdot h$, where h is the height expressed in meters and γ is a discharge factor per meter.

After updating it, each SUAV sends to neighbors a message containing its energy level: in this way, each SUAV is aware of the energy level of all his active neighbors.

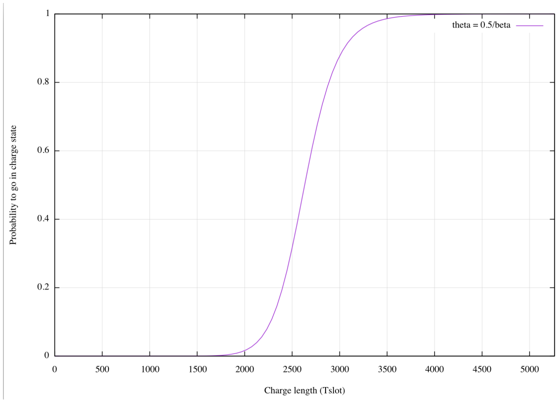

Figure 1: Basic threshold response function with $C = 0.5$, $\beta = 0.00019$, $n = 15$

3 Response-threshold function

The response-threshold function [2] proposed assigns at each active SUAV a probability to go into charge state every time the energy level is updated. It depends on three variables:

- $S(i)$, the actual stimulus intensity;
- θ, the response threshold expressed in units of stimulus intensity;
- n, a parameter used to adjust the steepness of the curve.

The probability function is then defined like this:

$$P(i) = \frac{S(i)^n}{S(i)^n + \theta^n}$$

In this case, I've chosen to model the stimulus as the charging time necessary to fully recharge the battery from its current energy level: the higher the time needed to go into charge state, the higher the stimulus. Thus, the stimulus is defined like this:

$$S(i) = \lfloor \frac{1 - (E(i,j) - 2 \cdot OP(h))}{\beta} \rfloor$$

which is the number of T_{slot} necessary to fully recharge the battery from the current energy level (taking into account also the descending and ascending operations).

If we consider a SUAV as a single entity, we want it to reach a high probability for going into charge state after what we consider a "critical" battery level, that we indicate with a constant C. Since we have to define this threshold in terms of $S(i)$ units, we can define it like this:

$$\theta = \frac{C}{\beta}$$

For example, $\theta = \frac{0.5}{\beta}$ indicates the number of T_{slot} necessary to fully recharge the battery when the energy level is 50%. Fig. 1 shows the threshold response function using the parameters just described: as expected, when $S(i) << \theta$ the energy level is still acceptable so the probability is close to 0; on the other hand, when $\theta << S(i)$, the charge time is higher than the threshold and so the probability becomes close to 1. When $S(i) = \theta$, the probability is exactly 50%.

3.1 Dynamic threshold

Since each SUAV communicates with other vehicles at 1-hop distance, we can consider the neighbors of each node as a sub-group of the network. Being displaced in an hexagonal mesh, each SUAV could have up to 6 neighbors. Knowing our energy level and the ones of our neighbors, each node has a local knowledge of who is more in need to go into charge state: ideally, we want vehicles with less energy in the group to go into charge state earlier with respect to the others.

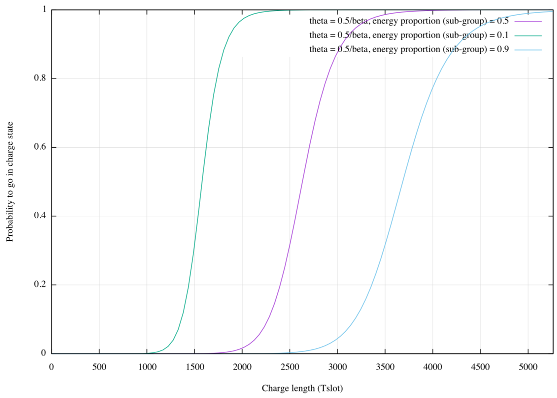

Figure 2: Dynamic threshold response function with $C = 0.5$, $\beta = 0.00019$, $n = 15$. Each curve represents the P(i) function varying the energy proportion of the SUAV with respect to its neighbors.

Having $P(i)$ determining the probability of going into charge state, tweaking the response threshold θ we can alter the responsiveness of each node and make it give a higher probability to vehicles that hold the lowest proportion of the energy in the group (lowering θ) and a lower probability to the ones holding a higher proportion (increasing θ). Thus, we can compute it at runtime using this function:

$$\theta = \frac{C}{\beta} + (\frac{C}{\beta} \cdot (energyProportion(|NE|) - 0.5)$$

where $|NE|$ represents the set of active neighbors cardinality and

$$energyProportion(|NE|) = \frac{E(i,j)}{|NE| + 1}$$

which is the energy proportion of SUAV i with respect to the total energy in the group if every node was in full charge. In this way, we are adding or subtracting up to half the threshold value, proportionally to the quantity of energy hold by the node with respect to its neighbors, so that high energy proportion nodes have more chances to go into charge state later and low proportion ones go earlier. Fig. 2 shows how the energy proportion affects $P(i)$.

4 Charge time duration and activity scheduling

After assigning a probability $P(i)$, a critical aspect of the charge activity scheduling it's deciding for how many time slots each SUAV has to go into charge state. The network lifetime lasts until a certain ratio k of the total area is covered by SUAVs activity and no vehicle is dead. For simplicity, we consider $A = N \cdot Cov(h)$, since each vehicle is assumed to be at the same height and cover the same amount of area.

We can choose M so that $\frac{N-M}{N} > k$: in this way the number of living SUAVs will always cover the required k ratio of A.

4.1 Fixed charge length

The worst case scenario is when we have $M = 1$: in this case, the best way to extend the network lifetime is to send every SUAV into charge state before anyone dies, maximizing the charging time. If we assume the initial energy equal to 1.0, every SUAV will die in $D = \frac{1}{\alpha} T_{slot}$, so each SUAV has to go in charge state for

$$C_t = \frac{D}{N + 1}$$

time slots ($N + 1$ is because we have to consider an initial discharge phase before sending any SUAV into charge state). The amount of lifetime extension will depend on the ratio $\frac{\beta}{\alpha}$: if $\frac{\beta}{\alpha} = 0.25$, the life of each SUAV will be extended of exactly $\frac{C_t}{0.25}$ time slots.

In case $M > 1$, it's easy to see that this approach is still valid: having more charging stations available, more than one SUAV could go into charge state at the same time, leading to a higher number of charge activity during a SUAV's lifetime (roughly a factor M).

Note that in this case each SUAV needs to know N, the number of vehicles in the network, which seems to be counter intuitive since this is supposed to be a fully distributed apporach. In this problem instance, N never changes and this information can be shared in the MANET before starting its activity, without altering the decentralized policy.

4.2 Variable charge length

Based on the previous strategy, we can further optimize the time each SUAV goes into charge state depending on the discharge length of each neighbors. Considering our critical battery level to be $CL = 0.2$, we use this simple rule to define the charge length:

$$chargeLenght(i) = \begin{cases} min(C_t, minDischargeTime(NE)), & \text{if } E(i,j) > CL \\ C_t, & \text{otherwise} \end{cases}$$

where $minDischargeTime(NE)$ returns the minimal discharge length in the set of active neighbors NE and the discharge time D_t for each SUAV is computed like this:

$$D_t = \begin{cases} \lfloor \frac{E(i,j) - OP(h)}{\alpha} \rfloor, & \text{if } E(i,j) > (CL + OP(h)) \\ 0, & \text{otherwise} \end{cases}$$

In this way, each SUAV will get out of charge state before the SUAV with less energy in the sub-group goes under the critical energy level, furthermore optimizing the charge activity scheduling.

4.3 Evenly distributed charge activities

The use of a probability $P(i)$ to decide when a SUAV has to go into charge state it's a stochastic process, so it could happen by chance that a SUAV goes into charge state more frequently, especially when they have a similar $P(i)$: this could make vehicles die before others because of unbalanced time spent into charging state.

To make charge activities more evenly distributed among vehicles in the network, we make each SUAV keeping track of the number of charges activities done and sending this information, along with the others, while updating its neighbors: before a SUAV could go into charge state, it has to check if it's a local minimum in terms of number of charge activities done with respect to others in the sub-group (i.e. the number of charges is lower or equal to the minimum in the group).

While this approach doesn't solve completely the problem as a centralized approach would, it guarantees that charge scheduling is evenly distributed in each sub-group and, since each SUAV is in more than one sub-group, this information is propagated to a wider number of vehicles, with the exception of isolated or several-hop distant ones.

5 Network simulation and experiments

In order to test the proposed solution, I used Omnet++ 5.0 to setup a simple simulation as described in the paper so far. The SUAVs parameters are based on the ones used in [1] (they are converted into percent values instead of using kJ or W units) and they're the following:

- $\alpha = 0.00076$ discharge factor (per second)
- $\beta = 0.00019$ charge factor (per second)
- $\gamma = 0.000038$ ascending/descending discharge factor (per meter)
- $h = 30$ (meters) flying height of SUAVS
- $a = \pi/3$ angle of cone projected by each vehicle to determine its coverage

We set $k = 0.5$ so that the network has to cover at least the 50% of the total area (i.e. at least 50% of SUAVs must be active). The timings presented here are expressed in seconds (because $T_{slot} = 1s$) and they represent the average value computed out of 100 simulations per experiment.

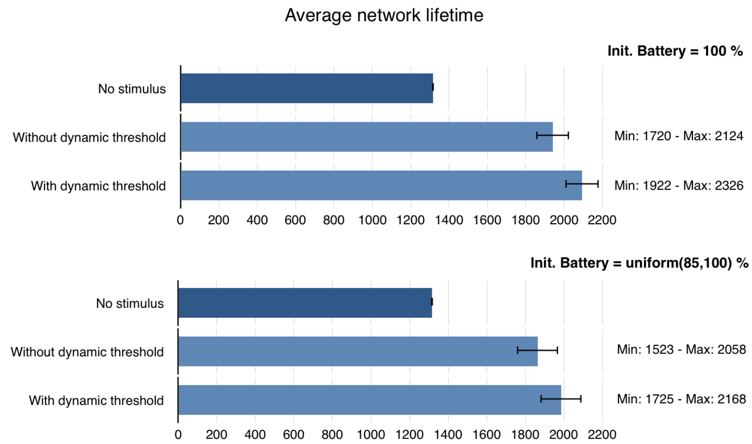

Figure 3:
Mean and std. deviation of the network lifetime (N = 7 and M = 3).

5.1 Impact of dynamic threshold

In this scenario we use a minimal setup with $N = 7$ and $M = 3$. Fig. 3 shows how the lifetime is increased using the response-threshold function with respect to the usual SUAVs lifetime of 1316 s. The dynamic threshold approach explained in 3.1 extends the network lifetime, whether we initialize the SUAVs' battery at full charge or with a random value between 85% and 100%.

Fig. 4 shows the scheduling difference from a sample run with and without the dynamic threshold approach: giving different priorities through responsiveness, the vehicles tends to distribute better the charging activities and better balance the overall energy in the network, instead of trying to go into charge state almost at the same time (due to the identical response-threshold function).

5.2 Lowering the stimulus threshold

So far we computed θ with the critical level set to $C = 0.5$. If we lower C, the overall responsiveness of the network will increase, making SUAVs go into charge state earlier and increase the number of charges each SUAV can do. Fig. 5 shows how the network lifetime get extended lowering θ.

5.3 Comparison with random probability

Fig. 5 shows also a comparison of the previous strategy to a random one: if we assign a uniform random probability for entering into charge state (using fixed charge time policy), we have a highly variable and unpredictable network lifetime, as expected.

5.4 Number of charging station

The network lifetime get extended based on how many times the SUAVs are able to go into charge state: increasing the number of SUAVs to, say, $N = 15$ and leaving $M = 3$, the network lifetime gets lower because SUAVs goes into charge state less times.

Without altering the charging factor parameter, a higher ratio of $\frac{M}{N}$ is the only way to effectively increase the network lifetime using this approach. Fig.7 shows how the average lifetime of the network is increased using a higher number of charging stations, varying $\frac{M}{N}$ ratio from 0.2 up to almost 0.5.

5.5 Impact of variable charge length

Using the approach described in 4.2, the network lifetime is slightly extended. Fig. 7 compares all the average lifetime using both static and dynamic charge length approaches.

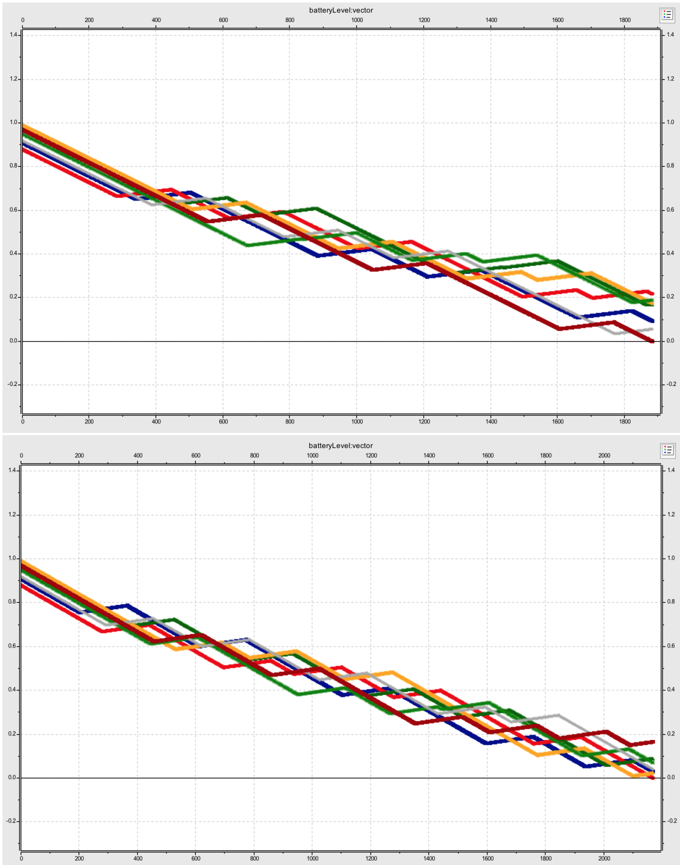

Figure 4:
SUAVs' battery level in the network. Same run without using dynamic threshold approach (up) and using it (down).

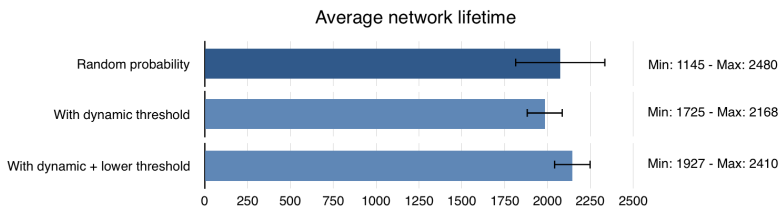

Figure 5:
Comparing the effect of uniform random probability (top), θ computed using $C = 0.5$ (middle) and $C = 0.3$ (down). In both experiments we are using a random value between 85% and 100% as initial battery level for SUAVs.

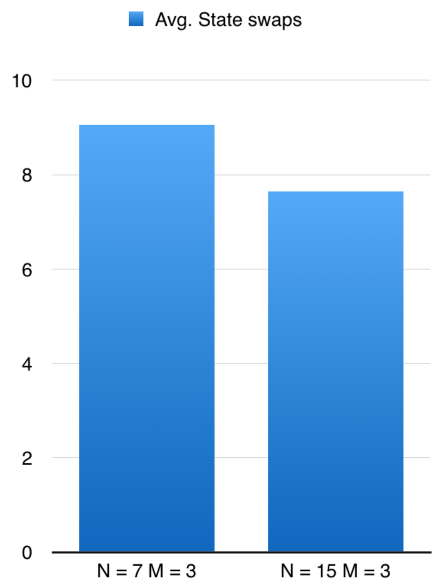

Figure 6: Average number of state swaps (active/charge) per SUAV, varying only N

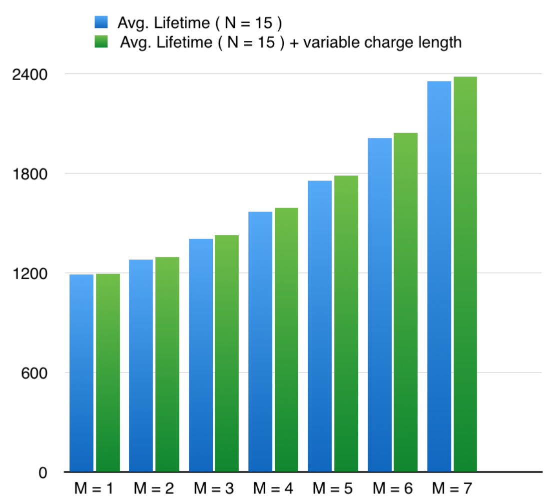

Figure 7: Average network lifetime with $N = 15$ and $M = 1, 2, 3, 4, 5, 6, 7$

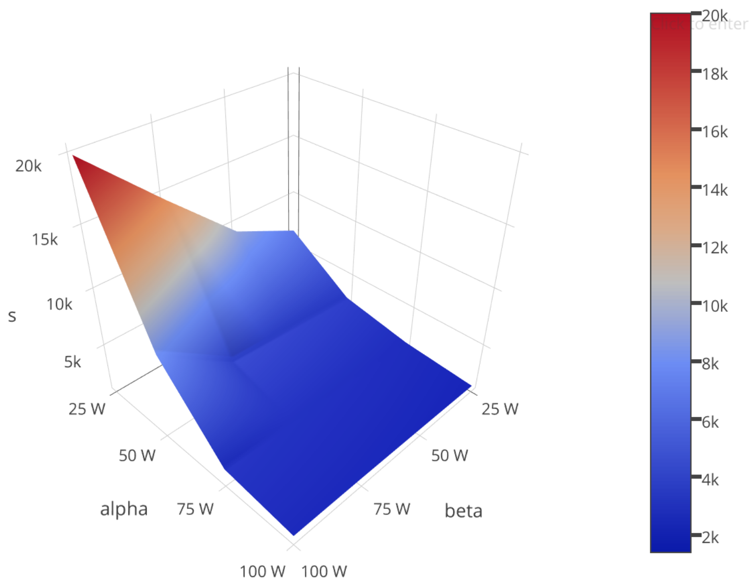

Figure 8:
Network average lifetime as function of α and β, with $N = 15$ and $M = 3$.

5.6 Varying α and β parameters

For the following tests, I used a 20000s time limit on each simulation and a number of 30 repetitions per run. In order to give a better idea of the parameters configuration, here I used α and β expressed in watts instead of the relative percent and considering the full battery charge to be 130 kJ.

Using N = 15 and M = 3, I tried the best scheduling configuration so far (i.e. using dynamic/low threshold, local minimum number of charges, variable charge length) and make the α and β parameters vary over 25, 50, 75 and 100 W to see how they affect the network duration. The configurations where it works the best is when $\alpha = 25W$, so when the discharge factor is pretty low. This is pretty obvious, but it looks interesting how we can reach good results for $\alpha = \beta = 25W$: in this case we reach an average duration of 6767.96 s (almost 112 minutes) and minimum in 30 runs is 6404 s, using only 3 charging station and extending the network lifetime of almost 5 times with respect to the case where we don't use any charging policy. Moreover, while this is still comparable to the results obtained in [1], this approach seems to work even better when $\beta > \alpha$ as shown in Fig. 8:

- with $\beta = 50W$ the average lifetime is 9349.51 s / 155 minutes (minimum is 7601 s)
- with $\beta = 75W$ the average lifetime is 15016.13 s / 250 minutes (minimum is 9334 s)
- with $\beta = 100W$ all the runs are $>$ 20000 s. Specifically, the average is 205783.74 s / 3429 minutes (minimum is 146110 s).

Even if this is an uncommon configuration of charge/discharge factors in a real application, it shows how this approach could reach good performance even without adding more charging stations.

6 Conclusions

In this paper I presented a new approach for scheduling charging activity in the area coverage problem using a MANET composed by SUAVs. This approach is based on a response-threshold function that lets the SUAVs autonomously decide when to go into charge state, using only informations received by immediate neighbors in the network topology.

I showed how the different policies and parameters affects the global network duration, obtaining similar results to the one already obtained in previous work.

References

[1] Angelo Trotta, Marco Di Felice, Kaushik R. Chowdhury, Luciano Bononi, *Fly and Recharge: Achieving Persistent Coverage using Small Unmanned Aerial Vehicles (SUAVs)*, DISI (University of Bologna, Italy) & Northeastern University (Boston, USA)

[2] E. Bonabeau, M. Dorigo and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, 1999

[3] B. Wang, H. Beng Lim, D. Ma. *A survey of movement strategies for improving network coverage in wireless sensor networks.* Computer Communications, 32(1), pp. 1427-1436, 2009.