

Second Homework

EECE 5155 - Wireless Sensor Networks and the Internet of Things

Instructor: Prof. Francesco Restuccia

In this homework, we will use the OMNeT++ network simulator to evaluate the performance of the *Dual Beacon Discovery* (2BD) protocol for Wireless Sensor Networks with **Mobile Sinks** (WSN-MSs) discussed during class.

Algorithm. In the 2BD algorithm, sensors are assumed to switch between **different duty cycles**, according to a **hierarchical approach**. In particular, during the discovery phase, sensor nodes operate most of the time with a **low duty cycle** to save energy, and switch to a **high duty cycle** only when they perceive that the **MS is about to enter their transmission range**. Information about the actual MS location is provided to sensor nodes by the MS itself, through a **periodic emission of two different beacon messages**, namely *Short-Range Beacons* (SRBs) and *Long-Range Beacons* (LRBs).

SRBs and LRBs are transmitted in an **interleaved way**, both with the **same period $2 \cdot T_{bi}$** such that T_{bi} is the overall beacon period. However, they are transmitted with **different transmission power**. Specifically, SRBs are transmitted with the same transmission-power level used during the communication phase for data exchange. Thus, they can be received within an area defined as **communication area**, and are aimed at notifying the sensor node the MS is within its transmission range, thus data exchange can actually take place. Conversely, LRBs are transmitted using a higher power than SRBs, and are used to inform the sensor node that the MS is approaching the communication area, and a contact could potentially occur shortly. Given the higher transmission power, LRBs can be received within a (much) **larger area** than the communication area, defined as **discovery area**. During the discovery phase, the sensor node is initially in the *LRB-Discovery* state, and wakes up periodically (using a low duty cycle δ_L) in order to check for possible beacons from the MS. Upon receiving a LRB, the sensor node transits to the *SRB-Discovery* state, increases its duty cycle to the high duty cycle value δ_H , and waits for a SRB. To avoid wastage of energy, if a valid SRB is not received within a pre-defined **timeout**, the sensor node transits back to the **LRB-Discovery** (or *Sleeping*) state. Whenever a SRB is received (both in LRB-Discovery and SRB-Discovery state), the sensor node enters the **Data Transfer state**, **increases the duty cycle** to 100%, and starts exchanging data with the MS. After transferring all its data, the sensor node transits to the LRB-Discovery state again in order to detect the next contact.

First Question [60pt]

Design and implement a module named **SensorNode2BD** that uses the 2BD protocol to discover and transmit data to a **MobileSinkNode2BD** module. The MS uses a simple algorithm that will take care of **updating its position** over time. In details, let us assume that positions are defined in a Cartesian coordinate system. The sensor node is positioned at $\mathcal{O} = (0, 0)$. At the beginning, the MS is positioned at **$\mathcal{S} = (x_s, y_s)$** , respectively. By assuming a fixed **speed v** and a destination **$\mathcal{E} = (x_e, y_e)$** , the MS's position is updated once every **Δ** seconds.

For the sake of simplicity, we will simulate wireless transmissions between the MS and the sensor node by using a **WirelessChannel** module. This module will take care of simulating the wireless channel between the MS and the sensor node. Specifically, the interactions between the three modules will be as follows. Packets/beacons sent from/to the sensor node will be **first received by WirelessChannel**, which will subsequently decide to forward the beacon/packet from/to the MS/sensor node according to a packet

loss probability $p(d)$ that depends on the distance between the MS and the sensor node. In details, by defining

$$p = \frac{d}{4 \cdot R} \quad (1)$$

where d is the Euclidean distance of the MS from the origin (i.e., the position of the sensor node) and R is the discovery range. You may assume that p is always equal to 1 in case the MS is outside the discovery range.

Figure 1 shows the interactions described above.



Figure 1: Interactions between *WirelessChannel*, *SensorNode2BD* and *MobileSinkNode2BD*.

The sensor node transfers data to the MS using a Stop-and-wait Automated Repeat reQuest (ARQ) protocol, identical to the one discussed during in class. You should also consider a propagation delay σ in your protocol design. The static node assumes that the MS has exited the contact area when it misses 3 consecutive ACKs.

Implement the following performance metrics:

- *Average Discovery Ratio* (DR): defined as the ratio between the number of times the MS is correctly discovered and the total number of MS passages;
- *Average Throughput* (T): defined as the average number of bytes sent to the MS during each passage;
- *Average energy spent in Discovery Phase per passage* (E-DP): defined as the average value of the energy consumed during the discovery phase per passage. You should only consider energy consumption during reception (RX) mode;
- *Average energy spent in Data Transfer Phase per passage* (E-DTP): defined as the average value of the energy consumed during the data transfer phase per passage. You should consider energy consumption during reception (RX) and transmission (TX) mode only.

Second Question [40pt]

If not specified otherwise, the simulation parameters to be used are as follows: beacon period (T_{BI}) 100 ms, MS speed 40 Km/h, discovery range (R) = 100m, communication range (r) 50 m, low duty cycle (δ_L) 0.3%, high duty cycle (δ_H) 3%, P_{TX} = 52.2 mW, P_{RX} = 56.4 mW, Δ = 1ms, σ = 10ms, packet duration (beacons, acknowledgement and data) P_D = 4ms. Assume the time the sensor node stays on during a duty cycle period is T_{ON} 100ms (the same as the beacon period). For the purpose of computing the throughput, assume that the data packet size is 133 bytes. For the mobility, assume that the MS moves on a line that is parallel with the x-axis (i.e., linear mobility), with a distance from the sensor node (y) equal to 15 m. Assume that the MS's starting and ending positions are 1 meter outside the discovery range.

Generate the following graphs, by assuming no packet loss:

1. Average DR (expressed as a percentage value between 0% and 100%) as a function of $\delta_L \in \{0.3, 0.5, 0.8, 1.3\}\%$, for two values of R : $R = 200\text{m}$ and $R = 100\text{m}$.
2. Average DR (expressed as a percentage value between 0% and 100%) as a function of $\delta_L \in \{0.3, 0.5, 0.8, 1.3\}\%$, for three values of y : $y = 5\text{m}$, $y = 25\text{m}$, and $y = 35\text{m}$.
3. Average DR (expressed as a percentage value between 0% and 100%) as a function of $\delta_H \in \{3, 5, 7, 9\}\%$, for three values of δ_L : $\delta_L = 0.5\%$, $\delta_L = 0.8\%$, and $\delta_L = 1.3\%$.
4. Average T (expressed in bytes per MS passage) as a function of $\delta_L \in \{0.3, 0.5, 0.8, 1.3\}\%$, for two values of R : $R = 200\text{m}$ and $R = 100\text{m}$.
5. Average E-DP (expressed in mJ) as a function of $\delta_L \in \{0.3, 0.5, 0.8, 1.3\}\%$, for two values of R : $R = 200\text{m}$ and $R = 100\text{m}$.
6. Average E-DTP (expressed in mJ) as a function of $\delta_L \in \{0.3, 0.5, 0.8, 1.3\}\%$, for two values of R : $R = 200\text{m}$ and $R = 100\text{m}$.

Furthermore, generate the same graphs by assuming the packet loss model in Equation (1).

Third Question (Optional) [25pt]

Implement the **Selective Repeat (SR) algorithm** described in details in [1], pages 11-13, and evaluate its performance. The SR algorithm is defined as follows. Upon receiving a SRB from the MS, the sensor enters the data transfer state. While in this state, the static node remains always active to exploit the residual contact time as much as possible. On the other hand, the MS enters the data transfer phase as soon as it receives the first message sent by the static node, and stops beacon transmissions. The sensor node splits buffered data into messages, which are transmitted in groups (i.e., *windows*). The number of messages contained in a window, i.e. the **window size**, is assumed to be fixed and known both at the sender and at the receiver. The static node sends messages in a window back-to-back, then waits for an acknowledgement (ACK) sent back by the MS. The ACK message contains information indicating which individual messages of the corresponding window have been correctly received by the MS.

If the acknowledgement message is lost, the static nodes simply **retransmits** the previous window as is. Otherwise, the static node flushes the messages which have been correctly received by the MS and fetches the new messages to transmit from its local buffer. We will assume that the sensor node has always data to send. Thus, the static node uses all the available contact time and goes to sleep only when the MS is not reachable any more. As in the Stop-and-wait protocol, the static node assumes that the MS has exited the contact area when it misses 3 consecutive ACKs.

Generate the following graphs:

1. Average T (expressed in bytes per MS passage) as a function of $\delta_L \in \{0.3, 0.5, 0.8, 1.3\}\%$, for two values of R : $R = 200\text{m}$ and $R = 100\text{m}$, no packet loss;
2. Average E-DTP (expressed in mJ) as a function of $\delta_L \in \{0.3, 0.5, 0.8, 1.3\}\%$, for two values of R : $R = 200\text{m}$ and $R = 100\text{m}$, no packet loss;

Generate also the same graphs by assuming the packet loss model in Equation (1).

0.1 Important Remarks and Grading Policy

To generate each point of your graphs, you should perform at least **10 independent replicas**, where each replica consists of at least **1000 MS passages**. Your graphs should also show, for each simulation result, **confidence intervals** with 95% confidence level.

The primary target of this homework is to elicit your creativity in writing network simulations. Therefore, some of the specific implementation details have been omitted on purpose, and left for you to decide as you are writing your own simulation code. Furthermore, as explained in class, simulations depend on a lot of factors (i.e., choice of initial seeds, number of runs, etc). Your homework will be evaluated by following these criteria, listed in order of importance:

1. *Functionality* (i.e., does your simulation code compile and run?)
2. *Correctness of implementation* (i.e., is your implementation of the 2BD algorithm correct?)
3. *Correctness of results* (i.e., did you get reasonable results in terms of discovery ratio, throughput, and energy consumption? Do your graphs have confidence intervals? Are those intervals reasonable in terms of span?)
4. *Modularity of implementation* (i.e., is your code understandable, modular, and clean? Did you write enough comments in the code?)

0.2 Submission Instructions

Please submit your assignment using Canvas. You should upload the following files: (i) One .ZIP file containing two (or three, if addressing the third question) subfolders containing your simulation code (including every .INI and .NED files) for each homework question; (ii) A .PDF file containing (a) your simulation results, including confidence intervals, both in graphical and tabular form; and (b) a short description of your simulation results and your interpretation of the results you obtained.

References

- [1] G. Anastasi, M. Conti, and M. Di Francesco, “Reliable and energy-efficient data collection in sparse sensor networks with mobile elements,” *Performance Evaluation*, vol. 66, no. 12, pp. 791–810, 2009. Available at <http://info.iet.unipi.it/~anastasi/papers/peval0.pdf>.