Implementing true-fairness based Cluster Head Selection in Wireless Sensor Networks

Heerok Banerjee
Dept. of Mathematics & Computer Science,
University of Antwerp, Belgium
Heerok.Banerjee@student.uantwerpen.be

Summary

Traditional LEACH and variants of LEACH protocols use a normal distribution to randomly select a cluster head. Arguably, this yields a skewed energy distribution which deteriorates the overall network lifetime. You will use `Petri Nets` to develop a true-fairness based `cluster head` selection and analyze empirical results to support your model. You will start with a small model and incrementally extend the network topology, keeping in mind the necessary `WSN` processes.

Instructions

*Github Classroom

First you need to link your `github account` with the repository for this project [1]. Next, I have created a public repository [2] which you should use to share your work with me. This will allow you to incrementally progress with your work and employ version control. Use the link given in [2] and accept the assignment to get started with your work.

Github Template Repository

There is another repository which will be used to host your final work after the end of this program. But for now, I will add references and examples after each lecture sessions. You can use this repository to fetch latest examples that will be relevant to your current assignments.

Report

A report is mandatory for this assignment. For the purposes of this program, I strongly recommend that you use `overleaf` to create a LaTeXreport and publish it on `ResearchGate`. You will first need to create a `Research Project` and then add a `'Report'` to our RG webpage. Following which, you can publish your work which will be publicly visible to your and the lab's followers.

Assignment

In this assignment, you will build a `Petri Net` model to resemble a `true-fairness` based implementation of a `WSN`. The term `true-fairness` with respect to `WSN` means a networked system which allocates equal probabilities for each sensor node to trigger a process (that could be transmission, aggregation, computation, link setup, etc.). So in your case, you are asked to build a model that accounts `sensor nodes, cluster heads and a single base station`, and model their network activities such that every participant is allocated an equal chance of triggering a process. A good starting point is given in [3] in the \example_PN\ChatRoom example.

Your basic setup should is given below. Read the text carefully and pay extra attention while modelling:

1. `Sensor Nodes:` You should model sensor nodes with all of their embedded capabilities. Firstly, the primary function of sensor nodes are to sense physical/environmental parameters and report to `BS`. So you need to model transmission of data from sensor nodes to `CHs`. Additionally, sensor nodes can be selected as `Cluster Heads (CHs)` and therefore, you must also model `Aggregation`. `Note: Your Petri Net model should account atleast 20 sensor nodes.`

2. `Cluster Head (CHs):` CH should not be separately modelled (therefore, no hierarchy). You should simply insert the functionalities that a `CH` posses in `sensor nodes`. Secondly and most significantly, the CH selection procedure should be deterministic and implement `true-fairness`. `Note: Your Petri Net model should account atleast 4 clusters and therefore, 4 CHs.`

3. `Base Station (BS):` A `BS` resembles an monitoring & control centre to gather statistics on the received information. Therefore, there should be a single `BS` in your setup. A `BS` should be able to trigger a `Aggregation` cycle wherein the `CHs` must forward the aggregation data. Remember, in the `aggregation` cycle, sensor nodes must not perform their networking activities.

4. `*Buffering:`You also need to model buffers for `CHs`, so that the aggregated information can be stored locally. During `Aggregation` cycle, this buffered data must be forwarded directly to the `BS` (in a fair-way). Since `sensor nodes` are equipped with extremely limited hardware, you should model a `K-bounded Buffer`, such that a limited amount of data can be captured. `Sensor nodes` can have different `K-boundness` and it is up to your choice. Additionally, the `BS` should also have a buffer but with unlimited capacity.

As for inter-network routing/cluster routing, this is beyond the scope of our research. But for precision in your analysis,you are free to draw assumptions as per your choice.

1 Analysis & Experiments

1. Test the true-fairness of your model and comment about the fairness scheme of your model by drawing conclusions from the analysis results?

2. Adjust your model such that the true-fairness implementation is removed. Then, compare your results?

3. Observe and estimate the average probability of each node to be selected as `CH`?

4. Use the Energy Consumption table given in [4], and attempt to model energy consumption from your Petri Net model? `Note: This is optional and not mandatory`

5. Simulate your PN model for `10 iterations` and observe the probability of each node to be selected as `CH`? Then, Simulate your PN model without true-fairness for `10 iterations`. Compare and plot the results.

References

[1] Link for Github Classroom- https://classroom.github.com/classrooms/62105038-modeling-of-networked-systems-performance-evaluation-petrinet-analyses/roster

[2] Invite Link for Assignment- https://classroom.github.com/a/So7TjtLH

[3] Examples & Template Repo- https://github.com/smartlabs-srm/wsn-PetriNets

[4] Banerjee, Heerok, S. Murugaanandam, and V. Ganapathy. "Low-Energy Aware Routing Mechanism for Wireless Sensor Networks." International Journal of Engineering Research in Computer Science & Engineering (IJERCSE) 5.1 (2018): 112-117. [link]