EECE 5496 Week 3: Wireless Sensor Network: Design Solution 2: Edge Nodes for Data Processing

September 12, 2025

1 Background

In Week 2 you conducted a black-box study of an ad hoc wireless sensor network wherein each sensor was connected to every other sensor within its communication radius. One of the results of this study was the average propagation delay experienced by sensor events relative to the specified communication range of the sensor.

Recall that the input parameters were the number of sensors to be considered, the communication radius, and the number of events generated by the network.

In this week's project, we extend the design of the wireless sensor network (WSN) by including in the network a set of **edge computing** nodes. These nodes are responsible for preprocessing the data generated by each sensor and generating information to the decision-making unit further down the network.

There are two phases for this project:

1. **Phase 1**: Conduct a BB analysis of average processing delays experienced by sensor events at each edge node. This will include an I/O analysis that will help you understand how the system responds to the input parameters that you specify.

    - Two stakeholder needs must be considered: (1) All sensors must be connected to at least one edge node; (2) The average delay at each of the edge nodes must be less than $d_{e_{avg}} < \epsilon$ where ϵ is a value that you can specify.

2. **Phase 2**: Conduct a white-box analysis of the WSN by modeling a subset of conceptual subsystems involved in the BB study. The two subsystems to consider are:

    - Sensors on Soldiers
    - Edge Nodes that Process the Sensor Workload

2 Description of WSN

Fig. 1 shows the randomly distributed sensor nodes and the location of four edge nodes $E_i, i = 1, 2, 3, 4$ configured in the MATLAB code provided for the BB study. Note that if some sensors are outside the radius of an edge node, their events will not be processed. The nodes without coverage are colored grey. By increasing the communication radius, or moving the location of the edge nodes or adding more edge nodes, this issue can be addressed. You can change the MATLAB code to investigate this or

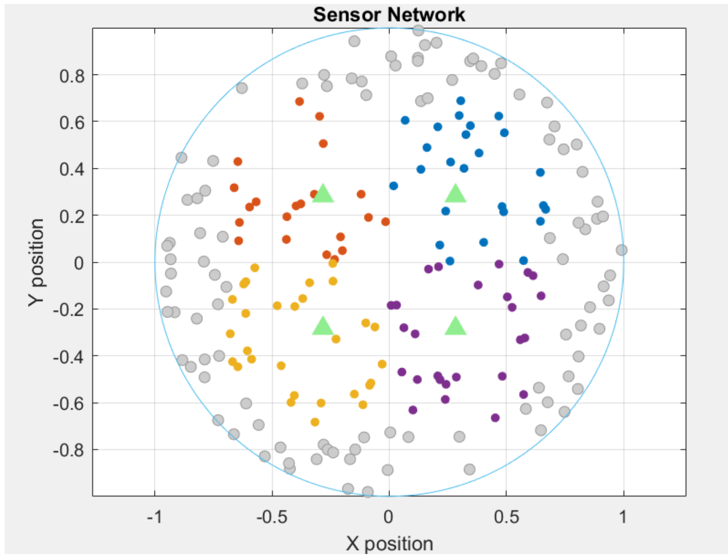

Figure 1: Example of a distribution of $N_s = 200$ sensors with a $C_r = 0.43$, four edge nodes (green triangles). Grey colored sensors are out of range of all edge nodes.

do this in the WB study in Phase 2.

3 Sensor Event Dynamics

Each sensor generates a random sequence of events according to the **Poisson process** that was described in the lecture notes. The parameter λ_s determines the average number of events generated per unit time. This is a user controllable parameter. Assume all sensors operate at this same rate.

Each of the edge nodes receives events from sensors that are within the specified communication radius. If N_i represents the number of sensors that communicate with E_i, then the average rate of events that arrive at E_i in unit time will be $\lambda_s \times N_i$.

4 Computational Workload Generated by Sensor Events

Each sensor is assumed to generate a workload that takes a random amount of time to process. The **exponential random variable** described in the lecture notes will be considered to model these random times. Denote by μ the average number of events that are processed in unit time by each of the nodes - μ is also known as the **service rate**.

5 Utilization of Edge Nodes

The utilization of the edge node is a performance metric that determines how busy each node is. For example, a node that operates at 65% utilization is idle 35% of the time. The trade-off that must be considered here is the time that each event has to wait for its workload to be processed versus increasing efficiency of the edge nodes. Events arriving at each node are served in a first-in-first-out process. When the processor is busy, events will queue for service. Higher the utilization of the processor, higher will be the average delay experienced by events processed at that node.

The schematic in Fig. 2 shows a model of the edge node that includes an infinite waiting room or buffer for each event to queue if the processor is busy. This system includes one processor that serves the arriving events on a first-come-first-serve (FCFS) basis. Other scheduling algorithms can be considered to optimize performance but this is the simplest one.

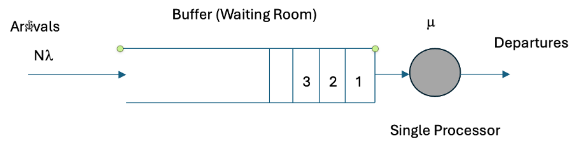

Figure 2: Schematic of the edge node queue and service.

The utilization or efficiency of each edge node is represented by the **design parameter** ρ which is given by:

$$\rho_i = \frac{Average\ Arrival\ Rate}{Average\ Processing\ Rate} = \frac{\lambda N_i}{\mu} \tag{1}$$

For the system (i.e. edge node queuing system) to be stable, $0 < \rho < 1$.

This queue model of Poisson arrival rates and exponential service times with a single processor is referred to as a $M/M/1/\infty$ system in the queueing literature. The M stands for Markovian random processes that are memoryless events - in other words the times of arrival of successive events are independent and the processing times of events are also independent. The first field in the notation $M/M/1/\infty$ represents the type of arrivals, the second field represents the type of service, the third is the number of processors and the last is the amount of waiting space in the queue.

Note that if you want the system to operate at a required value of ρ, you need only specify λ_s and ρ. The value of μ can be calculated to satisfy the stability condition.

The rest of this document describes the process of a black-box (BB) analysis with SysML using MATLAB code that provides theoretical values of average waiting times and average length of the queue at the edge nodes for the $M/M/1$ queue.

6 Understanding the MATLAB model

The code performs the following actions:

- Each sensor is assigned to its nearest edge node that is within its communication radius.
- Each sensor generates events randomly according to a **Poisson process** with rate λ_s.
- Knowledge of number of sensors connected to an edge node, enables the system to prescribe the node's processing rate μ such that its utlization is less than one. The average event arrival rate at node i : $\lambda_i = N_i \times \lambda_s$
- Knowledge of the event arrival rate at each node, $\lambda_i, i = 1, 2, ..$ and the processing rate $\mu_i, i = 1, 2...$ allows computation of average delay at the node. The average delay is $d_{i_{avg}} = \frac{1.0}{\mu_i - \lambda_i}$.

6.1 Supplemental Materials

The MATLAB file `WSN_mm1_theory.m` is the code that computes the performance metrics defined above. Run it first and familiarize yourself with the performance of the system. Note: Change the input parameters and observe changes in the outputs.. Bring up MATLAB environment and open the script.

Run the code in MATLAB console:

[results] = WSN_mm1_theory(200,0.43,0.03,0.8)

Examples of outputs produced are shown. The parameters used in these examples are: (i) N_s : 200, (ii) C_r : 0.43, (iii)λ_s : 0.03, (iv) $\rho_{target} = 0.8$, The number of edge nodes was fixed to be 4.

Figure 3 shows the number of sensors out of 200 that are connected to each edge node.

Figure 4 shows the cumulative distribution function (CDF) of the time spent by an event at one of the nodes.

7 Create the BB SysML model

1. Based on the black-box model you created last week, create a new BB analysis package with the given input parameters.

2. Define the following two stakeholder needs: (1) All sensors must be connected to at least one edge node; (2) The average delay at each of the edge nodes must be less than $d_{e_{avg}} < \epsilon$ where ϵ is a value that you

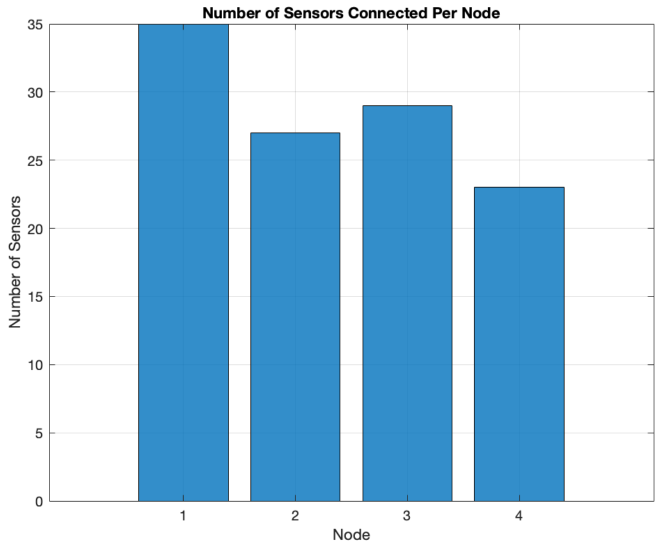

Figure 3: Number of sensors connected to each edge node.

specify.

3. Create a use-case: All sensors are connected to an edge node and experience average delays at nodes less than ϵ. Assume the soldier is the only actor in this use-case.

4. Generate a scenario for this use-case with an activity diagram. Make it simple - You might include just the soldier and the system of interest which is the WSN to depict that every soldier is to be connected to an edge node.

8 White-Box Analysis

Refer to the figure on page 70 of the Magic Grid BOK for an overview on developing the WB analysis. Steps 1

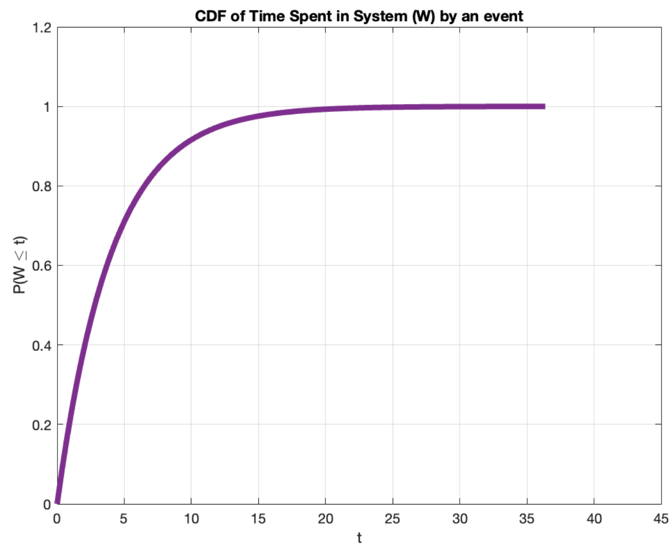

Figure 4: CDF of the time spent by an event (W) in the system

and 2 are completed in the BB analysis.

1. Step 3 on pg. 70: Identify two functional subsystems: (1) Sensors; (2) Edge Nodes.

2. For each of these prescribe required value properties.

3. Sensors are described by: (1) Number of sensors involved; (2) Their x,y positions within the circular region of radius equal to one; (3) communication radius; and (4) sensing rate λ_s

4. Edge Nodes are described by: (1) Number of edge nodes; (2) Their x,y positions in the circular region; (3) The average event rate $\lambda = N_i \times \lambda_s$; (4) Processing rate μ_i

5. Each subsystem will have a MoE: For Sensors: Con-

nection to a Edge Node; For Edge Nodes: Utilization should be less than one.

6. Define the interactions between the two subsystems.

7. Refer to diagram on pg. 72 of the MG BoK for conducting functional analysis that extends the scenario defined in the BB use-case.

8. Follow the three steps for WB Analysis that starts on pg. 73

Positioning Sensors and Edge Nodes at Random Locations: SysML has a *uniform* random number stereotype that can be prescribed to x and y positions ranging from 0 to 1.