**481 Simulation Project Report**

**Introduction:**

For this project we were tasked to create a simulation that mimics network communication between and gateway node and sensors. These sensors use Dijkstra’s algorithm to find the least cost path (best) and use that path to transmit packets between the nodes to the gateway node. During this transaction there are other factors that help drain the nodes’ energy levels including a target node that travels at a rate between 0.1 to 0.5 seconds and travels uniformly using cartesian coordinates and line equation. If this target node is in range of any of the active sensors they should connect until the target is out of range. All nodes should look for the best (least energy) path and once established starts transmitting data immediately in order to drain the nodes’ energy levels. If nodes are not transmitting, then they are able to sleep for T seconds as varies for testing purposes. During the simulation nodes will constantly switching on and off in order to preserve energy and prolong the life of the simulation. Once the first node runs out of energy (dies) then the simulation ends. All of these factors affect simulation time and this experiment will help us understand how the number of sensors, seconds between switching to sleep mode and target arrival rate affect the amount of energy used in a network system.

**How The Simulation Works:**

My implementation of this project was in C++ and there are a few components that tell the user what is going on while the simulation is running. When the program begins the user will see a message noting that the simulation has begun with a time stamp. You will then see indicators that transmission, sensing and target movement has begun as well. Then the locations of the gateway node as well as every sensor node will be listed. The simulation will then show all available routes and their corresponding distance between themselves and the gateway node. Under each path there will be a note that either says that the path is invalid or that the path can reach the gateway node. If a path ends in a -1 that also means that the path is invalid as well. It is important to note that since my simulation shows every node’s path to the gateway node it is possible to see repeat in paths as they are built. Once the best routes are established then nodes will start transmitting data as this happens each node that is awake and has a valid path at the time of transmission will have their energy depleted incrementally using the equations given in the document. So, you will see multiple nodes and their energy numbers dropping until one of them dies. While this is happening the target nodes began their path across the grid and transmitting data to sensor nodes in areas of interest. Although this is not directly seen from the simulation outputs there are functions that ensure this process. The simulation ends when the first node dies and is noted with a time stamp and indications that transmission, sensing and target nodes are all turned off. In order to test the necessary performance metrics you can directly change the variables inside simulator.cpp and is noted with a comment.

**Graphs and Data:**

**Δ Sensor Count: 100 Sensors Δ Sensor Count: 150 Sensors**

Constants: T = 5, Target Arrival rate = 0.3 Constants: T = 5, Target Arrival rate = 0.3

A grid with many dots

Description automatically generatedA grid with many dots

Description automatically generatedSimulation Time: 10 seconds Simulation Time : 15 seconds

**Δ Sensor Count: 200 Sensors Δ Sensor Count: 250 Sensors**

Constants: T = 5, Target Arrival rate = 0.3 Constants: T = 5, Target Arrival rate = 0.3

A grid with many colored dots

Description automatically generatedA grid with many colored dots

Description automatically generatedSimulation Time: 22 seconds Simulation Time : 26 seconds

|  |  |
| --- | --- |
| **Δ Sensor Count** | **Simulation**  **Time (seconds)** |
| 100 | 10 |
| 150 | 15 |
| 200 | 22 |
| 250 | 26 |

A graph with a line going up

Description automatically generated

**Δ T seconds**

Constants: Sensor count = 100, Target Arrival rate = 0.3

|  |  |
| --- | --- |
| **Δ T seconds** | **Simulation**  **Time (seconds)** |
| 5 | 6 |
| 6 | 10 |
| 7 | 10 |
| 8 | 10 |
| 9 | 12 |
| 10 | 12 |

A graph with a line going up

Description automatically generated

**ΔTarget Arrival Rate**

|  |  |
| --- | --- |
| **ΔTarget Arrival Rate** | **Simulation Time (seconds)** |
| 0.1 | 6 |
| 0.2 | 10 |
| 0.3 | 10 |
| 0.4 | 5 |
| 0.5 | 5 |

Constants: Sensor count = 100, T = 5 seconds

A graph with a line going up

Description automatically generated

**Performance Metrics Analysis**

According to my simulation the average running time of all of my data from my charts was 11.3 seconds. Run time of the simulation begins when you first start the program and ends when the first node in the simulation dies. Through all the test the variable that seemed to have the most effect on the runtime of the simulation was the number of sensors inside the simulation as it had the greatest difference in runtime of 16 seconds. Thus, more sensors mean a more prolonged simulation. In terms of networks when the first sensor node dies it may cause disruption between the routes and making a route invalid. As seen through this experiment if there are more sensors within a networking system there will be more instances of varying routes that can be used in order to transmit data to the gateway node. This would prolong the longevity of the system and allowing the more data packets to be sent.

The average delay per packet can be affected differently by the different factors from this experiment. The number of sensors can determine the number of possible routes a packet can take at any given time. With that being said the more nodes there are the greater chance of there being a nearby node to forward to the next hop in the route which in general would reduce overall transmission delay. However, a downside would be higher energy consumption which may cause nodes to die off more quicky. The change in T for how long a node stays in sleep mode is another factor in average delay per packet. Shorter sleep times allows for higher probability of a nearby sensor being able to transmit data which increases packet forwarding rate, but a trade-off would be faster energy depletion of the nodes as sleep allows nodes to preserve their energy so less sleep means less energy preservation. This was demonstrated in my experiment as T increased so did the time before the first node lost all its energy. The last variable that we tested that directly affects average packet delay was the target node arrival rate. Since the target node was used in order to deplete the energy of the nodes. When the target was in range of any of the sensor nodes while the sensor node was awake, they would transmit to each other which would drain the sensor’s energy levels. As the target node rate increases there is lesser chance that a sensor node may be awake and able to forward a packet towards the moving target. Not only that but considering the sensing distance is 140 meters as the target moves it may still be in range even after switching coordinate points. The network topology would be constantly changings making it more susceptible to packet loss.

Network throughput can be affected by all 3 variables tested during this experiment. Increasing the number of sensor nodes would generally mean more data packets can be created and transmitted. The change in T, which is the amount of time a sensor node remains in sleep mode, can affect how many full packets are transmitted. Frequent transmissions between nodes turning on and off can cause a great deal of packet loss through routes if a node happens to transmit as another node is transitioning to sleep mode. Therefore, increasing the sleep times may decrease network throughput. The arrival rate of a targets may strain the network as there would be more targets entering areas of interests which would cause sensor nodes to constantly establish who they want to transmit to and be able to adapt to the constantly changing system.

As aforementioned targets are constantly moving within the system and are used to drain the energy of sensor nodes as they transmit data whenever they are in range of one another. Since sensor nodes are constantly switching between sleep and awake mode it is possible that a target may cross areas of interest while a sensor is either asleep or transitioning which would cause a loss in data transmission. However, with fewer sensor nodes there is a lesser chance that a target node can find another node within its route to transmit data to. So in order to create a network with a quicker target detection rate there needs to be a greater number of packets and more uniform sleep times in order to create the most efficient network in terms of this metric.

**Error Analysis:**

In reference to my version of the project I believe that the trend of my simulation relative to target arrival time is off. My results are bidirectional as target arrival time increases. As target arrival rate decreases the simulation time should increase as there would be less activity exchanged between sensor nodes and targets, therefore node life expectancy should be prolonged.

**Conclusion:**

My simulation has been able to capture how varying factors such as number of sensor nodes, T time for sleep and target arrival rate may affect simulation time, average delay per packet, network throughput and target detection time. Through this experiment I learned that increasing the number of sensors and T time create a more time efficient network system as it prolongs the simulation time and increases the transmission rate of data while keeping the packet loss to a minimum. However, a downside to increasing these would be the frequent loss in energy that each node may experience from the increase in transmission rate. In conclusion the design of a network depends on balancing the number of sensor nodes as well as the times that a sensor must remain asleep. Target arrival rate should be adjusted in order to meet the requirements of the network which include factors such as energy consumption, network throughput, accuracy and packet loss and target transmission rate. In conclusion these main factors can greatly affect the efficiency of a network, but in general you must take into account everything and how it aligns with the goals of the network system you are attempting to implement.