Random Graph Model to Simulate and Study Cascading Failures in Internet of Things

MTH 565 Project Report

Vikram Vijayakumar

Abstract:

The Internet of Things (IoT) represents a highly interconnected network where communication between devices facilitates diverse applications such as healthcare, smart cities, and industrial automation. However, the complex interconnectivity of IoT networks makes them vulnerable to cascading failures, where the failure of a single node can trigger a chain reaction, leading to total system failure. This study uses the random graph model to simulate and analyze cascading failures in IoT systems, focusing on Wireless Sensor Networks (WSNs), a foundational IoT component. By constructing and simulating a small world network in MATLAB, this study evaluates the effects of clustering, path efficiency, node energy constraints, and load redistribution on network stability.

The study demonstrates how network resilience is influenced by factors such as node energy capacity, communication range, and overall network density. The findings highlight that higher density and short diameters reduce the likelihood of cascading failures, emphasizing the value of small world properties in designing a robust IoT system. This work provides critical insights into developing strategies to mitigate cascading failures, offering a scalable framework for enhancing IoT network reliability in mission-critical applications.

Introduction:

The Internet of Things (IoT) is a vast, interconnected ecosystem where devices communicate in real time, providing diverse applications from smart cities and environmental monitoring to healthcare and industrial automation. However, the interconnectivity of IoT also makes it significantly vulnerable to a phenomenon known as cascading failure. In an IoT network, a single node's failure can often trigger a domino effect, destabilizing the entire system. To address this challenge, this project explores using a random small world network model to simulate and analyze cascading failures within IoT networks, with a particular focus on wireless sensor networks (WSNs). WSNs are foundational components of IoT which are highly interconnected systems critical to real-time data collection and understanding their failure propagation is vital for improving the overall resilience of IoT systems.

Small world networks are defined by their high clustering coefficients and short path lengths, features that balance local interconnectedness and global reachability. These properties make the small world network model valuable for modeling IoT, where devices must efficiently communicate with one another. In this project, the random graph model will represent IoT's complex connectivity patterns, showing the potential for robust communication and their vulnerability to cascading failures. This model can provide insights into managing IoT's scalability and reliability by simulating failure scenarios and offering strategies to mitigate cascading effects.

We could have potentially used a known small world network model and used their characteristics to define the IoT system, but there are multiple limitations in the system that can be implemented if we had to use the rules of a standard small world network model. Also, the model developed is expected to have less clustering coefficients and therefore it will not be justified to call it a small

world network. However, we will try to study the characteristics of the networks to see if it has small world network properties and see the measures can be done to achieve this. We have answered a few questions on the IoT system's relation to small world networks particularly in determining if the system is robust (before or after failure) and if the IoT network actually represents a small world model.

Literature review:

The concept of small-world networks has garnered significant attention in the study of Internet of Things (IoT) systems, primarily due to the IoT's reliance on high interconnectivity and network robustness. Small-world networks offer features such as high clustering and short average path lengths, which are advantageous for developing efficient, resilient IoT infrastructures.

Simulating Watts-Strogatz and Newman-Watts small-world models [3] have demonstrated the potential of these structures to enhance routing and clustering in wireless networks, validating the small-world phenomenon in distributed IoT network environments. Through this work, insights were gained on average node degree, path length, and routing protocol effectiveness, showing how small-world properties can be applied to improve wireless network resilience and performance.

The Social Internet of Things (SIoT) model [4] further integrates small-world concepts, emphasizing efficient service discovery among IoT devices. This model uses a hop-based service query to foster navigability within the network, effectively mirroring real-world small-world structures and achieving efficient service execution across devices.

In recent studies, various adaptations of small-world network models have been explored to optimize IoT applications and address IoT-specific challenges, such as high-density device communication and failure resilience. For instance, the Small-World SSDNet [6] model leverages both small-world and super-dense features to support proximity-based services and fog computing applications. This approach aims to maintain efficient device-to-device communication while navigating the increased complexity introduced by 5G and IoT devices in community-scale networks. A Greedy Model with Small World properties (GMSW) [5] has been proposed to enhance robustness in heterogeneous sensor networks within IoT. This model incorporates a shortcut-adding strategy that significantly enhances network resilience by linking super-nodes with superior hardware, resulting in reduced latency even under random or targeted failures.

These studies collectively underline the importance of small-world network properties in advancing IoT network design, with small-world architectures providing a promising foundation for improving resilience, efficiency, and navigability in various IoT applications.

Work performed:

Modelling an IoT system's network in MATLAB involves designing a random graph network model with approximately 100 vertices (nodes) and edges connected at random between nodes within their range. We have replicated the clustering and path efficiency found in real IoT networks. In this scale, the focus on short path lengths and high clustering provides insights into small-world properties, capturing both the network's efficiency and its potential failure points.

The project uses MATLAB to simulate a random graph, representing each node as an IoT device with defined load capacity and energy constraints. The steps include:

- Constructing a random graph model to maximize using the following data,
    - Model – G(n, p), we may assume it is an Erdos-Renyi graph but there will be a small additional clause to it. Hence this random graph will not come under any specific model. However, we will compare it with Erdos-Renyi model and regular lattice.
    - No. of. nodes, n = 10 and Edges are added at random in the initial graph.
    - Set energy utilization to 80% to trigger node failures.
    - P = 0.3, we will vary this to increase clustering and observe its effects. We will also limit the range of nodes to 30 to add additional challenge and study it. The range is limited to 30 to resonate with characteristics of a network router.
    - We will simulate another graph without the probability factor, Gt(100) but the nodes will be limited to a range of 30.
    - Finally compare it to Erdos-Renyi model and a regular lattice to observe the characteristics and see the possibilities of increasing closeting in the network, to try and achieve small world characteristics.
- We will observe failure propagation in both G(n, p) and Gt(n) models.
- Implementing load redistribution mechanisms to observe the propagation of node failures and potential cascading effects.
- Analyzing the effects of energy utilization in the nodes.
- Tracking metrics such as Node Capacity, Edge Length, diameter, density, Clustering Coefficient, and Energy Constraints to study their impact on failure and resilience.
- Analyzing how clustering and edge lengths(range) impact network stability and developing adaptive algorithms to enhance IoT resilience against cascading failures.

We will place two nodes (Node 1 or control node and node 2, which will be the farthest nodes) at fixed points, (0,0) and (100,100) respectively. The rest 98 nodes will be placed at random, and edges will be formed respectively. We will use the Euclidean distance formula, $d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$ to calculate distance between nodes so as to form edges between them within a range of 30.

Our aim is to form a path between the farthest node (node 2) and the control node (node 1). Each node communicates with the control node, which is responsible for handling data by hopping to its nearby neighbors because it is impossible for nodes to communicate directly with the control node in a wireless environment given that the nodes are placed at far distances and have a limited range. The network is designed to take the shortest path between nodes, so as to limit the number of hops. Increased number of hops results in an increased number of packets (unit of data that's transmitted over a computer network) being transmitted.

When a node fails at random, the nearby nodes connected to it, transmitting information, are forced to take an alternative path which is longer than the previous path and might increase the number of hops. This will increase the load on other nodes, which could be vital in connecting the highest

number of nodes and transmitting data and thereby making them susceptible to failure. This process happens for a number of iterations until the network comes to a standstill, which means that the nodes can no longer communicate with the control node (node 1). At this point the network fails completely, and communication is terminated. This scenario is called cascading failure.

We will limit the situation to observe the communication between control node (node 1) and the farthest node (node 2). When there is no path between node 1 and node 2, we will declare the network is dead.

Results:

Firstly, we will simlaute the Graph, **G(n, p)** with 100 nodes and p = 0.3 and obsevre the network effects.

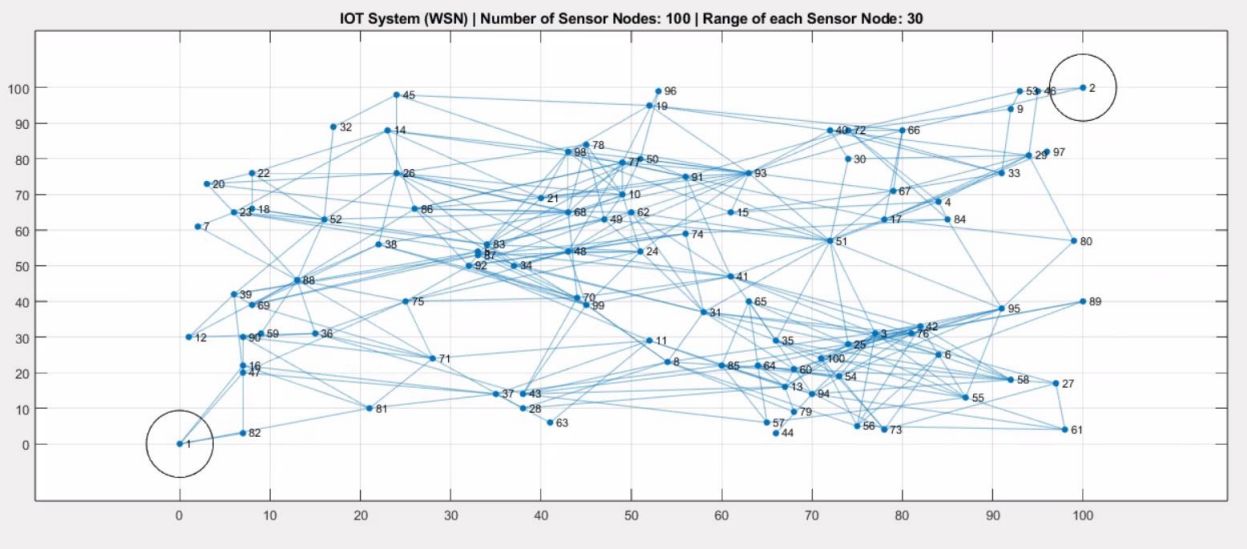

Fig .1. IOT(WSN) system modelled using random graph model, G(n, p).

Fig .1 depicts the simulation of an IoT network, G(n, p) model in MATLAB. The nodes have a range of 30, but here we can see that not all nodes within a range of 30 are connected because we have used a probability factor of 0.3. The only justification for this is to try a different approach of using probability factors and compare it with small world network and random graphs. We will also simulate the same model with Gt(n) model. The random graph in Fig. 1 represents connection of nodes (IoT devices) through edges (internet channel), which is the initial model simulated. The initial graph depicts the following characteristics observed in MATLAB,

Initial Diameter: 8

Initial Density: 0.069899

Initial Clustering Coefficient: 0.064516

These characteristics show that this graph is not qualified to be called a small world network.

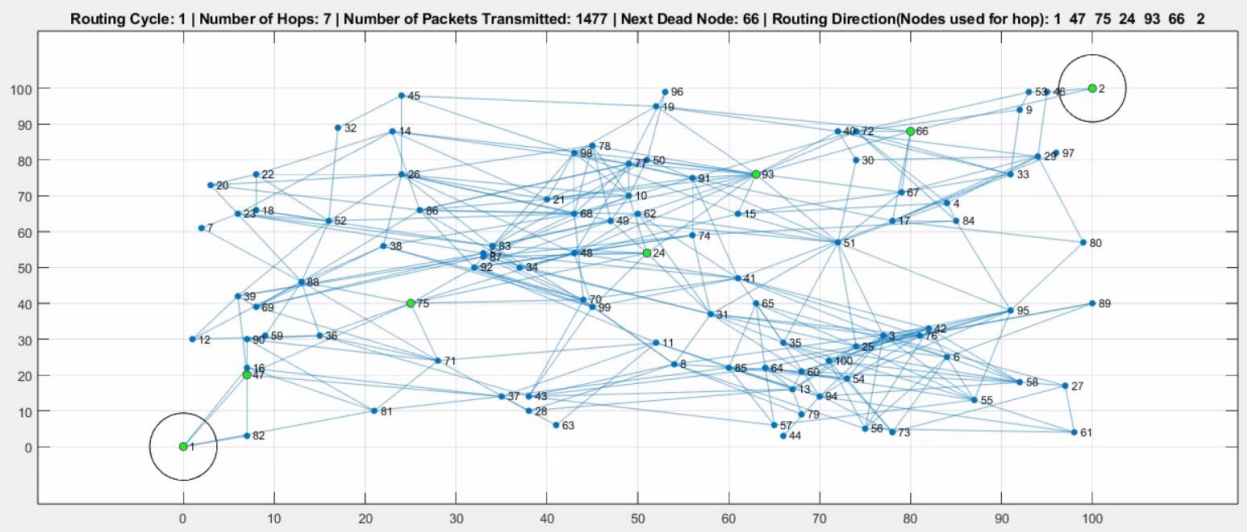

Fig .2. Initial routing cycle showing the Path between control node (Node 1) and farthest node (Node 2)

Fig .2 depicts the path between control node (node 1) and the farthest node (node 2). There are 8 hops between nodes 1 and 2, which denotes a diameter of 7. In MATLAB, we choose a node to fail at random by limiting the power utilization of nodes to 80%. This process will be continued for certain iterations, until the communication between control node and the farthest node fails. In short, the network is alive and functioning until all nodes are able to hop and communicate with node 1. In the first routing cycle, we can see that node 66 will fail and thus the next routing cycle will choose the next shortest alternative path. The following characteristics were observed, Iteration 2 – Diameter = 8, Density = 0.069899 and Clustering Coefficient: 0.064516.

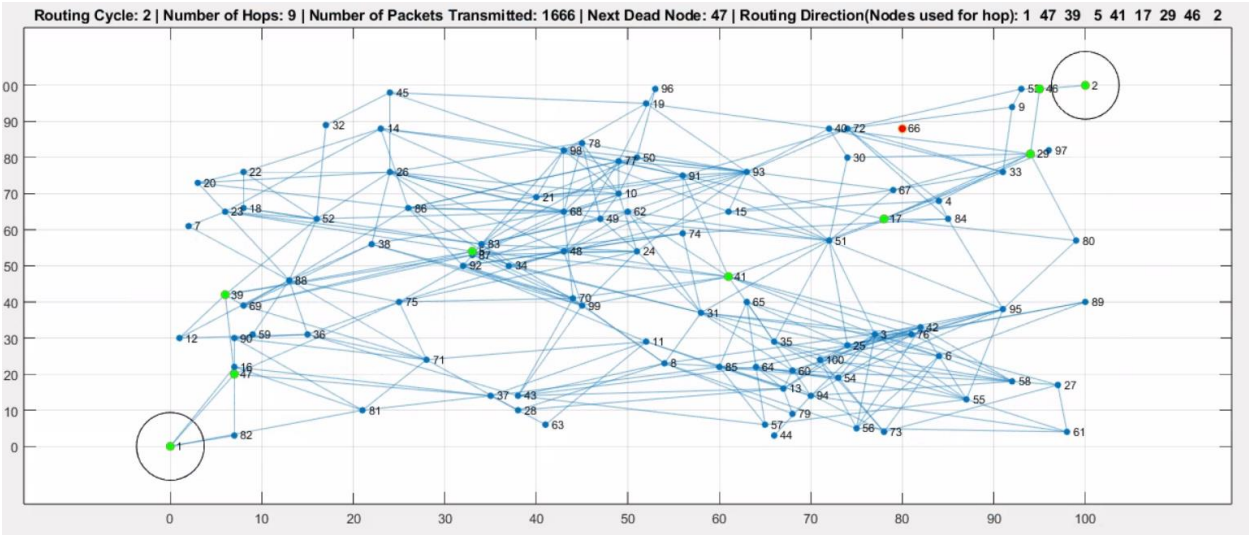

Fig .3. Routing cycle 2 choosing a new path for communication after the failure of node 66.

Fig .3 depicts the new path between control node (node 1) and the farthest node (node 2) after node 66 failed. We can observe that the number of hops has increased from 8 to 9, indicating a change in diameter. The number of packets being transmitted has also increased due to increased path length. This cycle indicates that node 47 will fail. This cycle will continue until there is no path between node 1 and node 2.

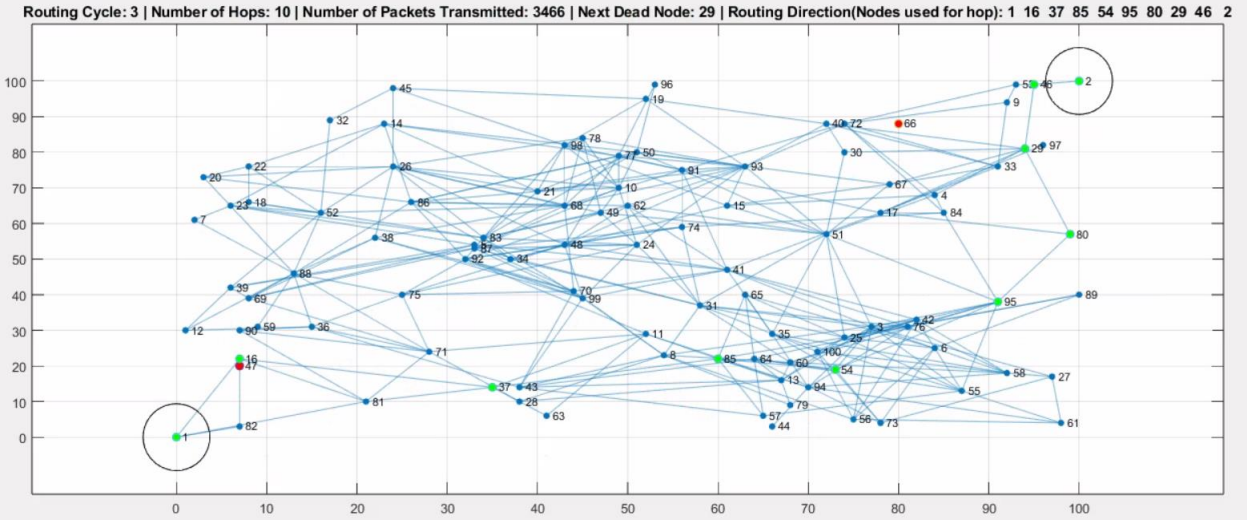

Fig .4. Routing cycle 3 choosing a new path for communication after the failure of node 47.

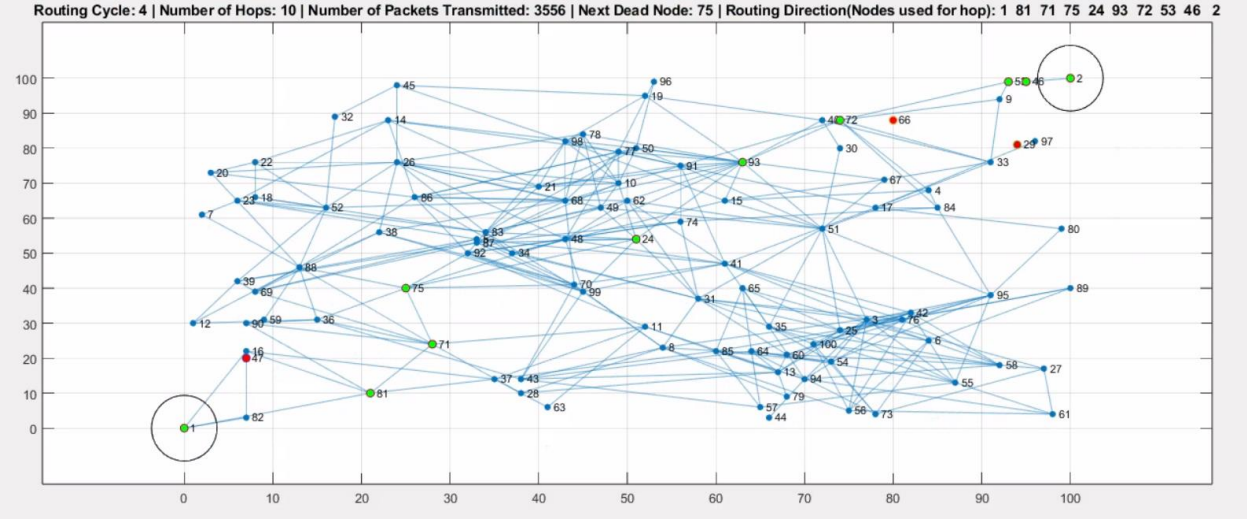

Fig .5. Routing cycle 4 choosing a new path for communication after the failure of node 29.

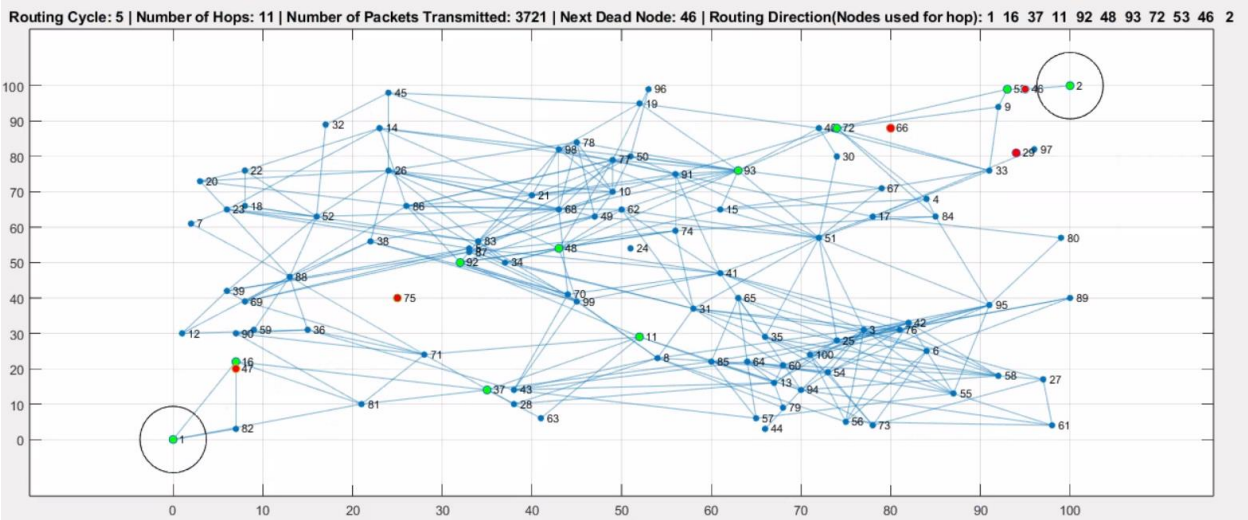

Fig .6. Routing cycle 5 choosing a new path for communication after the failure of node 75.

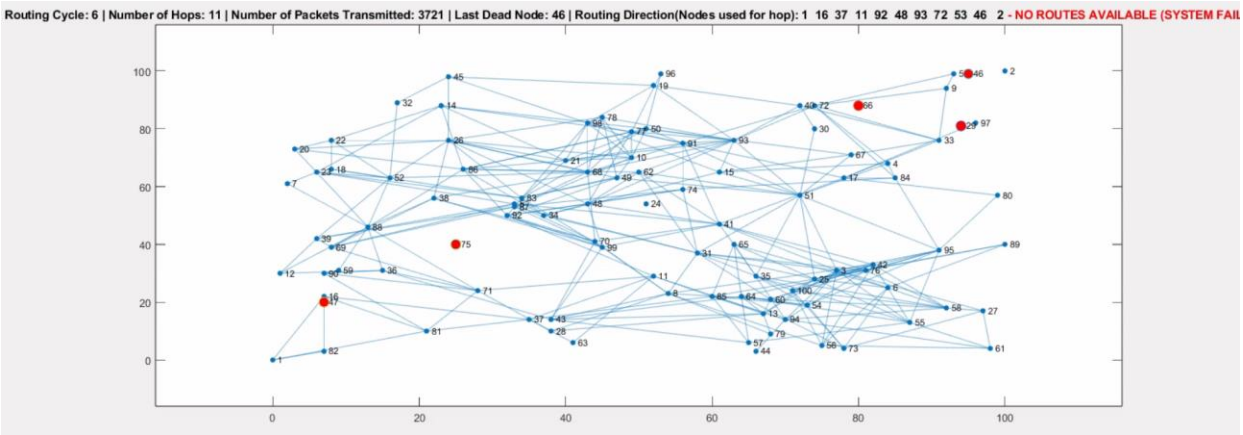

Fig .7. Final Routing cycle denoting network failure after the failure of node 46.

Fig 7. shows the final network routing cycle, indicating system failure. Here when node 46 fails, the communication between control node and farthest nodes is totally cut because there is no alternative path to communicate, implying that the system has failed. The failure of nodes depends on the capacity of information which they can handle.

Final Diameter: 8

Final Density: 0.069899

Final Clustering Coefficient: 0.064516

The number of iterations (routing cycles) will vary at each simulation because this is a random graph. However, when the probability factor was increased, the following characteristics was observed.

P	Diameter		Density		Clustering Coefficient		Number of Iterations
	Initial	Final	Initial	Final	Initial	Final	
0.3	8	8	0.069899	0.056768	0.064516	0.068207	6
0.5	6	8	0.093333	0.058182	0.10357	0.11044	18
0.7	6	6	0.16242	0.11778	0.15744	0.15954	21
0.9	6	6	0.20222	0.11212	0.19427	0.19484	24

Table .1. Values of diameter, density, clustering coefficient for the graph, G(n, p) at different values of p

Table .1 shows the different values of the graph at different values of p. It clearly shows that increasing the probability factor will certainly increase the performance of the network, enhancing reliability and resilience.

Now, we will simlaute the Graph, **Gt(n)** with 100 nodes and obsevre the network effects. We will limit the range of nodes to 30 here as well.

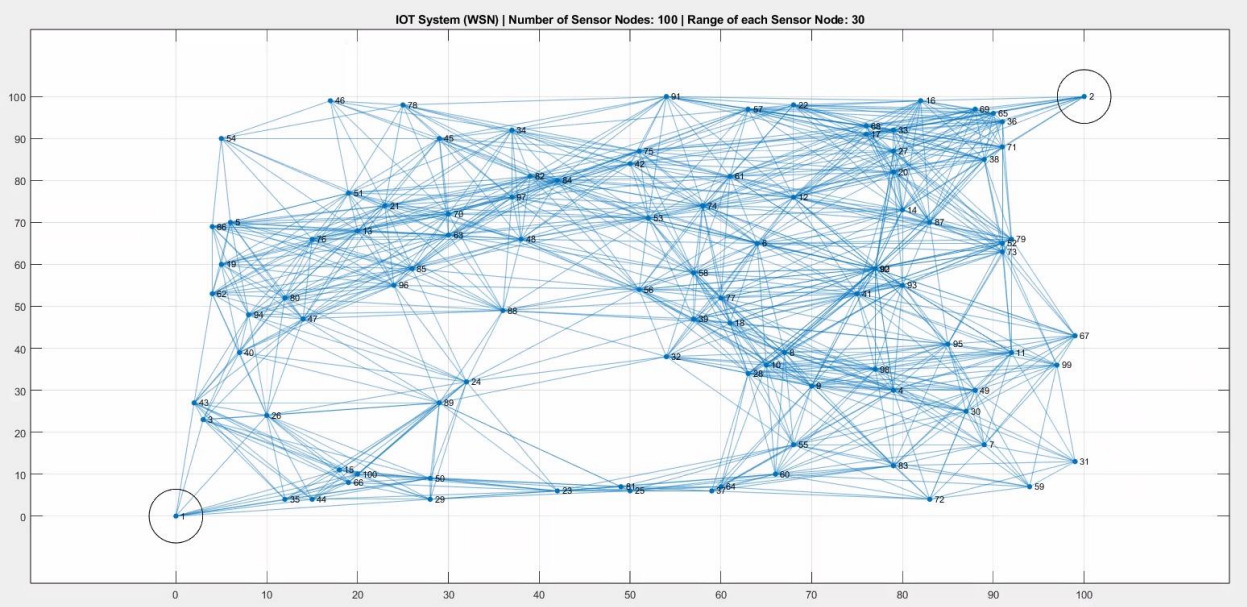

Fig .8. IOT(WSN) system modelled using random graph model, Gt(n).

Fig .8 depicts the simulation of an IoT network using Gt(n) model in MATLAB. The nodes have a range of 30. The random graph in Fig. 8 represents connection of nodes (IoT devices) through edges (internet channel), which is the initial model simulated. The initial graph depicts the following characteristics observed in MATLAB,

Initial Diameter: 6

Initial Density: 0.2002

Initial Clustering Coefficient: 0.21326

These characteristics show that this graph is not qualified to be called a small world network.

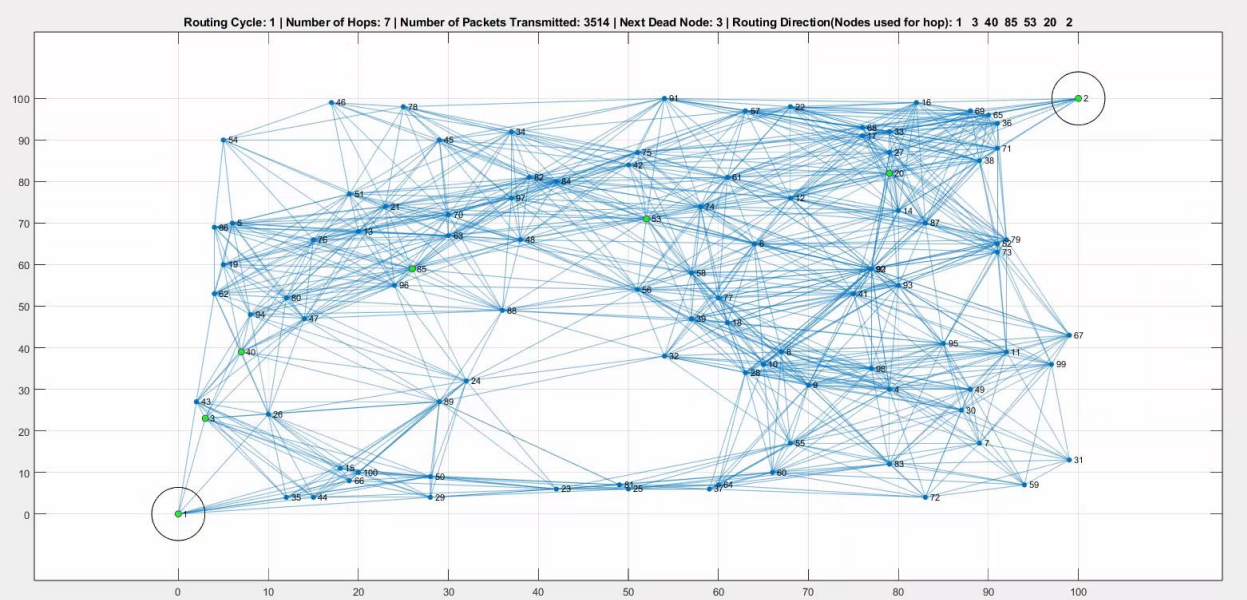

Fig .9. Initial routing cycle showing the Path between control node (Node 1) and farthest node (Node 2)

Fig .9 depicts the path between control node (node 1) and the farthest node (node 2). There are 7 hops between nodes 1 and 2, which denotes a diameter of 6. Similar to the G(n, p) graph, we choose a node to fail at random by limiting the power utilization of nodes to 80%. This process will be continued for certain iterations, until the communication between control node and the farthest node fails. In short, the network is alive and functioning until all nodes are able to hop and communicate with node 1. In the first routing cycle, we can see that node 3 will fail and thus the next routing cycle will choose the next shortest alternative path.

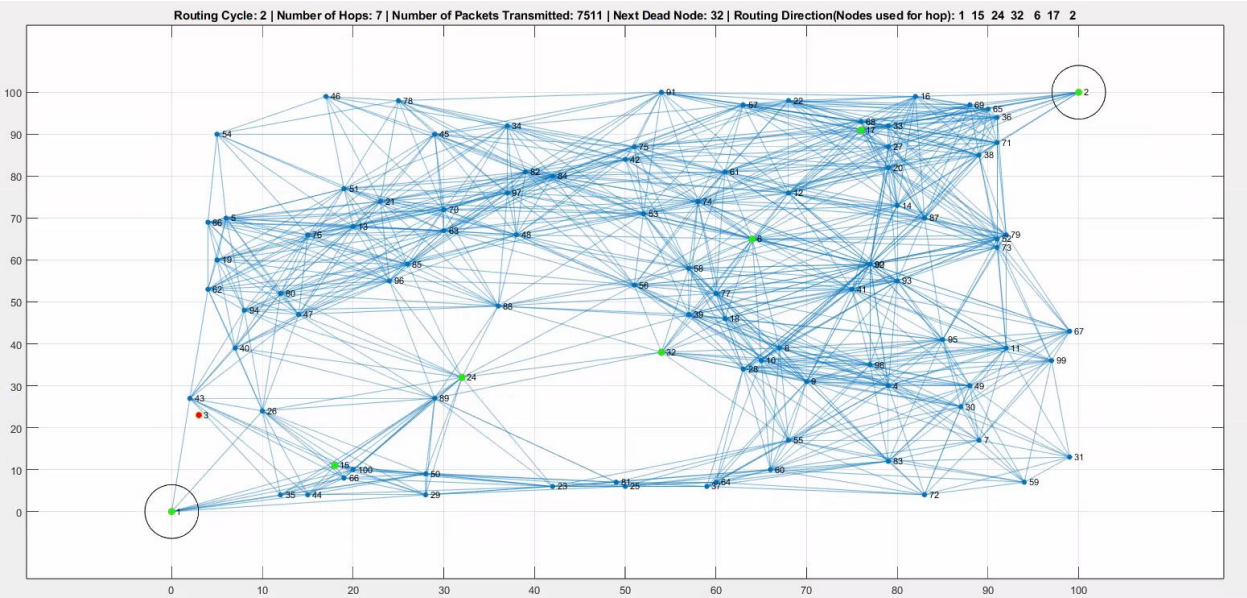

Fig .10. Routing cycle 2 choosing a new path for communication after the failure of node 3.

The cycle will continue until the network comes to a point where multiple nodes have failed and the communication between node 1 and node 2 is cut off. This graph runs for 35 iterations until the network fails.

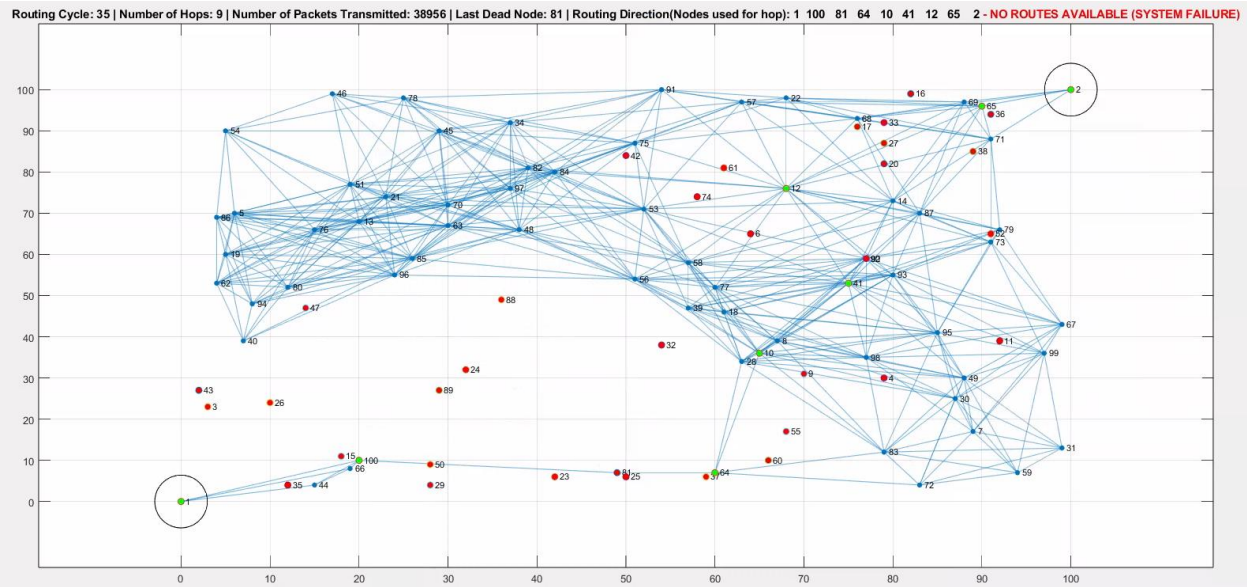

Fig .11. Routing cycle 35 (last cycle before failure) indicating the failure of node 81

Fig .11 indicates that the system has reached a point where communication can no longer happen due to the failure of node 81. The red nodes in the graph show the nodes that failed at each routing cycled. This graph model is much more efficient because it operates much longer than the G(n, p) model.

Final Diameter: 5

Final Density: 0.092525

Final Clustering Coefficient: 0.21861

The number of iterations (routing cycles) will vary at each simulation because this is a random graph. After multiple simulations the number of iterations varies between the range of 28 and 44.

In both graphs, the variation in initial and final values of diameter, density and clustering coefficient is due to the failure of multiple nodes. The more nodes fail, the smaller the graphs become. However, it is not an indicator that the graph is a small world network.

Now, we will compare the graphs with Erdos-Renyi model G(100, 0.3). Using MATLAB when simulating the above graph, we observe the following characteristics.

Density of Graph: 0.29758

Clustering Coefficient: 0.30071

Diameter = 2

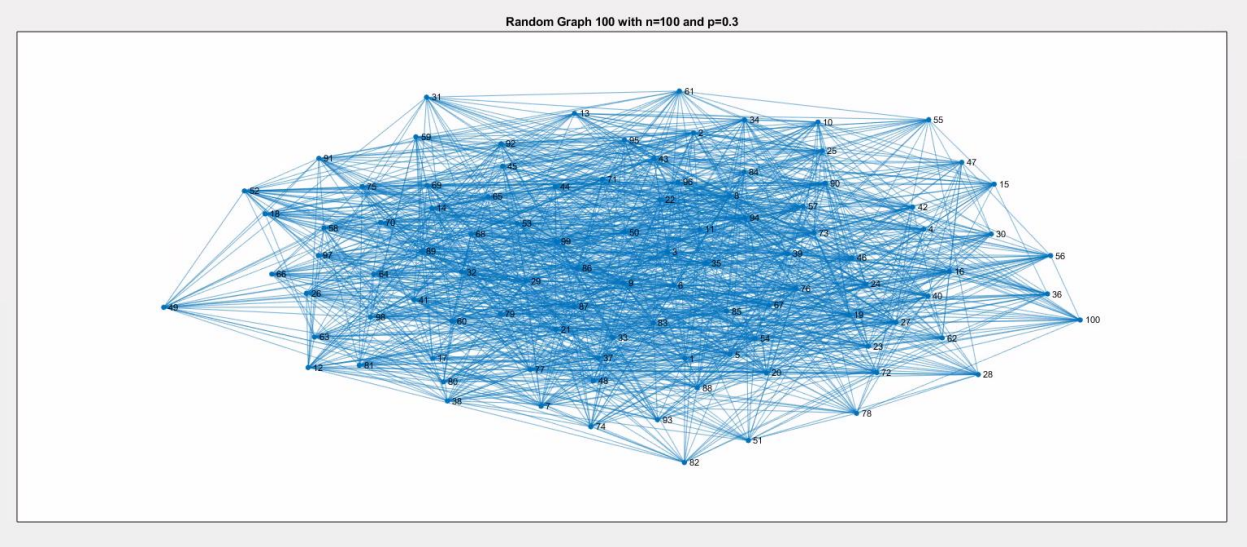

Fig .12. Erdos-Renyi graph, G(100, 0.3)

Fig 12. depicts the simulation of Erdos-Renyi graph, G(100, 0.3). Based on its values of diameter and clustering coefficient, it is clearly seen that this model of graph will be a much more suitable model to simulate a IoT (WSN) network. Although it is much more suitable, it cannot be used practically because the devices are subjected to a network range in real time and hence, they have to follow them. However, if the devices are strategically placed in a way that they are able to communicate with all of the nodes, meaning all nodes are in range then this would be a much more suitable model for IoT network.

Now, we will compare the graphs with Regular Lattice model with 100 modes. We can derive the value of k (nearest neighbors) using the values from Erdos-Renyi model as shown below.

Expected degree in Erdos-Renyi, $E = \lceil (n-1)*p \rceil$

When $n = 100$, $p = 0.3$, $E = \lceil 29.7 \rceil = 30$

Therefore, we can use $k = 30$

Using MATLAB when simulating the above graph, we observe the following characteristics.

Density of Graph = 0.30303

Clustering Coefficient = 0.72414

Diameter = 4

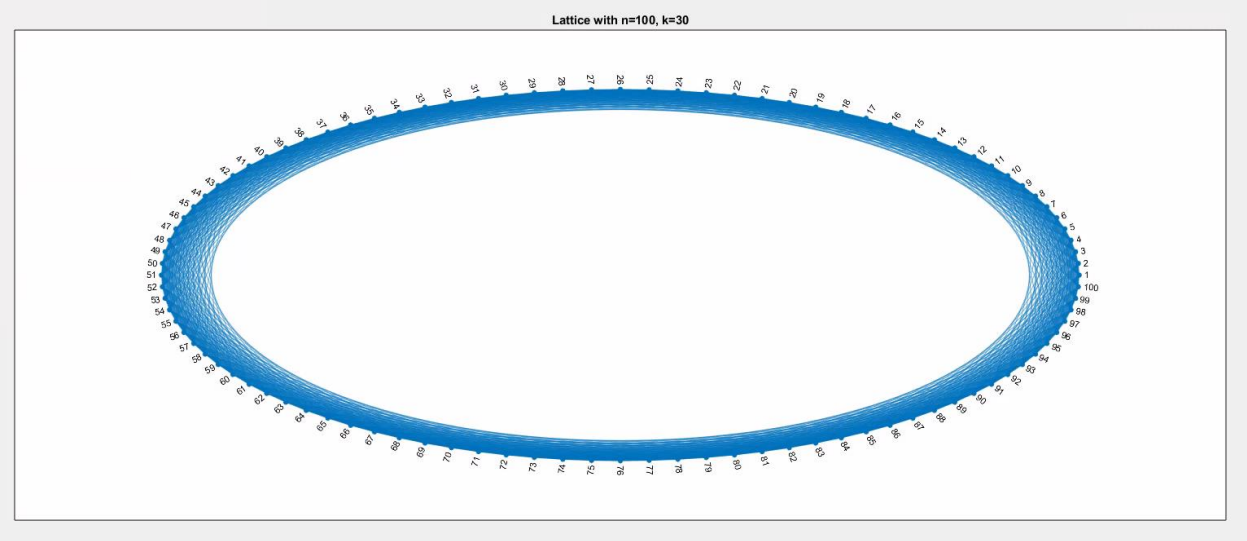

Fig .13. Regular Lattice, n = 100 and k = 30

Fig 12. depicts the simulation of Regular Lattice with n = 100 and k = 30. Based on its values of diameter and clustering coefficient, it is clearly seen that this model of graph will be the best suited model to simulate an IoT (WSN) network, theoretically. Although it is much more suitable, it cannot be used practically because the devices are subjected to a network range in real time and cannot be positioned uniformly. But the regular lattice can qualify as a small world network model.

Therefore, Erdos-Renyi model will be more suited for modelling a IoT network. Although it may not be a small world network, it certainly possesses the qualities to model an IoT network. Considering that in an IoT network, the nodes could be positioned anywhere within an environment, we can follow G(100, 0.3) model to model IoT network and reduce3 cascading failures. We have to ensure that all nodes are within their range.

Apart from the above observations, we were also able to observe the effect of power factors in the network. After increasing the power factor, it is observed that the lifetime of the network is increased (increased number of routing cycles). By also increasing the communication range of each node, we can increase the lifetime of network. However here, we intentionally kept the power factor and the range to a small number so that we can observe how the nodes fail and causes a chain reaction. The average number of hops taken to communicate between two farthest nodes is 7, implying the shortest path length of the network is 7. However, the diameter increases when many nodes fail. When the density of the network is high, it is highly reliable and decreases as density decreases. This random graph model can be used to represent IoT in a small environment like hospitals, buildings, residential apartments etc.

Conclusion:

To model cascading failures in IoT networks, the project used a simulation-based approach, specifically within a WSN context. The random graph model constructed using MATLAB, where nodes (representing IoT devices) are positioned to maximize clustering and minimize path lengths. This experiment simulates node failures and observes how these failures propagate across the network through load redistribution mechanisms.

The study provides valuable insights into the design of resilient IoT networks. By leveraging small world network properties, this project offers strategies to reduce the likelihood and impact of cascading failures. Results highlight how high clustering and short paths in network design can enhance resilience, providing a foundation for developing adaptive load redistribution algorithms.

This study is significant for enhancing the robustness of IoT systems, particularly in mission-critical applications where system reliability is essential. As IoT continues to evolve and expand, developing models that prevent or mitigate cascading failures will be crucial to ensure the dependability and efficiency of these systems.

References:

[1] Sohn, Insoo. "Small-World and Scale-Free Network Models for IoT Systems." *Mobile Information Systems* 2017.1 (2017): 6752048.

[2] Xing, Liudong. "Cascading failures in internet of things: review and perspectives on reliability and resilience." *IEEE Internet of Things Journal* 8.1 (2020): 44-64.

[3] Dong, Ziqian, et al. "An experimental study of small world network model for wireless networks." *2015 36th IEEE Sarnoff Symposium*. IEEE, 2015.

[4] Amin, Farhan, and Gyu Sang Choi. "Advanced service search model for higher network navigation using small world networks." *IEEE Access* 9 (2021): 70584-70595.

[5] Luo, Diansong, et al. "A small world model for improving robustness of heterogeneous networks." *2015 IEEE global conference on signal and information processing (GlobalSIP)*. IEEE, 2015.

[6] Cheng, Wei, et al. "Ssdnet: Small-world super-dense device-to-device wireless networks." *IEEE Network* 32.1 (2017): 186-192.

Appendices: *Link to GitHub(code): https://github.com/Vikramgv/MTH565_Final_Project*