An Algorithm for Energy Optimization and AI-Facilitated Self-Adaptiveness in Wireless Networks

A Project Report

Submitted to the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Information Technology

by

Adwaid M (TRV19IT005)

Ratan Thomas David (TRV19IT043)

Ron Regi Zacharia (TRV19IT045)

Roshan Aji Cherian (TRV19IT046)

DEPARTMENT OF INFORMATION TECHNOLOGY

GOVERNMENT ENGINEERING COLLEGE, BARTON HILL,

THIRUVANANTHAPURAM-695035

KERALA

May 2023

DEPARTMENT OF INFORMATION TECHNOLOGY

GOVERNMENT ENGINEERING COLLEGE, BARTON HILL,

THIRUVANANTHAPURAM-695035

2022 - 23

CERTIFICATE

This is to certify that the report entitled **An Algorithm for Energy Optimization and AI-Facilitated Self-Adaptiveness in Wireless Networks** submitted by **Adwaid M** (TRV19IT005),
Ratan Thomas David (TRV19IT043), **Ron Regi Zacharia** (TRV19IT045) & **Roshan Aji Cherian** (TRV19IT046) to the APJ Abdul Kalam Technological University in partial fulfilment of the B.Tech. degree in Information Technology is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Suryapriya S
(Project Guide)
Assistant Professor
Department of Information Technology
Government Engineering College,
Barton Hill, Trivandrum

Dr. Haripriya A.P.
(Project Coordinator)
Associate Professor
Department of Information Technology
Government Engineering College,
Barton Hill, Trivandrum

Prof. Manju R
(Project Coordinator)
Assoicate Professor
Department of Information Technology
Government Engineering College,
Barton Hill, Trivandrum

Dr. Vijayanand K.S

Professor and Head
Department of Information Technology
Government Engineering College,
Barton Hill, Trivandrum

DECLARATION

We hereby declare that the project report **An Algorithm for Energy Optimization and AI-Facilitated Self-Adaptiveness in Wireless Networks**, submitted for partial fulfillment of the requirements for the award of the degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Prof. Suryapriya S

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Adwaid M
Ratan Thomas David
Ron Regi Zacharia
Roshan Aji Cherian

Trivandrum
24 May, 2023

Abstract

The number of mobile devices in use around the world has grown significantly in recent years. However, the rapid increase in the number of wireless devices in use poses us with a lot of network problems which should be resolved to prevent any major network bottlenecks in the future. One of the disadvantages of wireless networks is that they consume a significant amount of energy for the transmission of data across the network. Greater the distance between the nodes more will be the cost involved in transmitting data across the nodes. In this report, we propose a system that provides intelligent and energy-efficient routing; meeting the demand requirements. Incorporating AI makes the system dynamic by making it self-adapting and self-learning. We use a constraint-based model to achieve energy-efficient routing.

Acknowledgement

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to Dr. Vijayanand K.S, Head of Department, Information Technology, Government Engineering College, Barton Hill, for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to the Dr. Haripriya A.P., Department of Information Technology, Government Engineering College, Barton Hill, Trivandrum for the support and co-operation. We also would like to thank Prof. Manju R, Department of Information Technology, Government Engineering College, Barton Hill, Trivandrum and Prof. Manju R, Department of Information Technology, Government Engineering College, Barton Hill, Trivandrum for their extended support.

We would like to place on record our sincere gratitude to our project guide Prof. Suryapriya S, Assistant Professor, Information Technology, Government Engineering College, Barton Hill, for the guidance and mentorship throughout this work.

Finally, we thank our families, and friends who contributed to the successful fulfilment of this project work.

Adwaid M

Ratan Thomas David

Ron Regi Zacharia

Roshan Aji Cherian

Contents

List of Figures

Chapter 1

Introduction

With the advent of 5G and other reconfigurable wireless networks, the internet is expanding and changing expeditiously. As a result of this expansion, the magnitude of network traffic and energy utilization have risen. Although many attempts to rectify these issues were employed, most of the ideal routing solutions in use today are not very intelligent. Artificial Intelligence has expedited automation and has also facilitated adaptive self-configurable networks. The intelligent distribution of network resources has been enabled by the strength of AI and software-defined networking (SDN). SDNs, which additionally permit centralized control, have allowed for the separation of the data and control planes. As the traffic increases, it's essential to reduce the effort of network administrators and hand off the routing to a smart system. The network will be able to self-adjust and make decisions without human involvement with the use of intelligent routing. We must also overcome the difficulty of creating a network resource management system that uses less energy and traffic.

In order to achieve the rigorous requirements in a real-world deployment, we suggest an AI-Assisted energy-effective routing solution. We transform the more straightforward energy consumption model from the multi-constraint energy efficiency optimisation model. Then, in order to support this paradigm, we suggest a simple heuristic algorithm. As a result, intelligent and energy-efficient routing is designed to achieve maximum energy efficiency.

Chapter 2

Literature Review

2.1 Existing Systems

Many attempts have been made to reduce the constantly increasing traffic and also to reduce energy consumption. Some of the approaches concentrated to meeting diverse demands and others concentrated on improving performance and efficiency to reduce energy consumption.These researches make use of various algorithms and schemas. Intelligent routing is also another aspect that has been researched.

2.1.1 Energy-Efficient Routing in NOMA

Zhang et al.,2019 [1] in their paper have presented an approach using Non-Orthogonal Multiple Access (NOMA) technology and Heterogenous Networks. NOMA uses orthogonal sub-carriers. By using successive interference cancellations, it enables multiple users to be multiplexed on the same frequency band. Numerous desirable potential advantages, including increased spectrum efficiency, decreased latency with high reliability, and extensive connectivity, are provided by NOMA. A user with poorer Channel State Information (CSI) is given more power than a user with better CSI, claims NOMA. This ensures access for unreliable users. Heterogeneous networks (HetNet) is a term used for modern mobile communications networks. A modern mobile communications network consists of a combination of different cell types and different access technologies. Macro cells are used to provide coverage. Pico cells and microcells are used to enhance capacity in busy areas, such as train stations, shopping malls, and city centres. Femtocells and Wi-Fi are used at the office and at home. [1]presents the situation when perfect Channel State Information (CSI) is available. When perfect CSI is available matching theory is used to match the user equipment to the corresponding sub-frequency bands. Therefore, a suboptimal user scheduling scheme can be designed for this system by using a matching theory. The overall system's energy efficiency can be expressed as the maximisation of energy effectiveness depending on the restrictions of the maximum transmission power, the minimum user data rate, and the maximum number of users that can be allocated on one sub-band.

2.1.2 Energy Efficient Routing in Non-Orthogonal Multiple Access Systems

[2] proposes a hybrid scheme for supporting diverse wireless services combining the concepts of non-orthogonal multiple access networks and orthogonal frequency division multiplexing is mentioned in this paper. The joint resource management of user clustering (UC) and power allocation is investigated in detail in this paper. Under two different power consumption cases, the optimal resource allocation (Opt-RA) algorithm is developed with the help of converting the original mixed integer nonlinear programming (MINLP) problem to decoupled problems. For practical implementation, the heuristic resource allocation (HeurRA) algorithm is proposed, which includes a low-complexity user clustering algorithm based on the candidate search-and-allocation approach.

2.1.3 Network Slicing using Deep Reinforcement Learning

Wang et al., 2019 [3] present a system in which Deep Reinforcement Learning is used for optimizing resource scheduling thereby improving the routing efficiency of a network. The concept of network slicing has been discussed in this paper. Using network slicing the entire network can be divided into multiple independent networks each having its functions. Modern-day applications produce large amounts of data and the concept of network slicing has tremendous scope in such situations. The authors of the paper aim at better resource utilisation by incorporating Machine

Learning into network slicing algorithms. The proposed system can allocate resources effectively to multiple network slices simultaneously while maintaining a good QoS. Deep reinforcement learning is the machine learning model used to achieve this. This method has proven to be better than other random, greedy and heuristic approaches.

2.1.4 Neural Network Approach for QoS Routing

As the traditional intra-domain routing methods are environment-specific or metric-specific, they lack flexibility in the sense that they aren't able to modify rules when exposed to different metrics. Moreover, these methods are inefficient in gathering valuable information from massive network flows and network status; thus limiting their performance. Although learning methods such as deep learning, reinforcement learning, etc. have been applied to several systems such as SDN-IoT systems and virtual network embedded systems, they have only been able to extract partial network patterns. [4] proposes a system that makes use of vectors to identify devices. Encoding network units into vectors gives us a universal representation and can be directly used in a shallow neural network to predict attributes of links. By integrating these vectors into a neural network, the system becomes more flexible and different QoS demand routings can be devised by extracting desired features. Complex routing and planning of paths are simplified to simple vector calculations. Validation of performance is done by applying the proposed system to handle basic routing and on the delay constraint least cost path problem.

2.1.5 Energy Effective routing in D2D networks

Future networks will need to increase system capacity, spectral efficiency, energy efficiency, and other factors to meet the high demands that wireless communication has entered into. The Internet of Things (IoT) will be created with millions of connected devices. Since these devices require a significant amount of energy for data collection and transmission, numerous studies have been devoted to improving and analysing energy efficiency in D2D communication networks. The mode selection for energy-effective D2D communications has been taken into account in [5], allowing D2D users to perform better. D2D users have an additional level of freedom to improve their performance by means of mode selection between cellular and D2D modes.

2.1.6 AI-Assisted Energy-Efficient and Intelligent Routing for Reconfigurable Wireless Networks

As wireless networks evolve and become increasingly commonplace, the volume of network traffic and network energy consumption also significantly increases. A routing algorithm based on the concepts of Artificial Intelligence is proposed in this paper which helps in energy-effective routing of packets. The developed AI model also helps in achieving a better quality of service. To facilitate intelligence and autonomy for the algorithm, the authors convert the problem into an optimisation mathematical model. This routing algorithm based on AI provides the feature of dynamically modifying link weights helping in energy savings. The whole system is implemented on a software-defined networking (SDN) platform.

Chapter 3

System Development

3.1 Proposed System

Here, we present a system in which concepts of SDN can be used for the allocation of resources. SDN consists of the control plane and the data plane, each of which performs crucial functions when the user requests a service. The control plane performs the function of finding the energy-efficient path through which the data must be transmitted. The data plane configures the data flow table accordingly and finalises the forwarding process. Hence, the control plane is used to cater to different requests based on detailed policies.

To achieve network intelligence we use the AI theory to enhance routing adaptability. The software-defined networking idea is used to achieve this goal from a network-level perspective. We use the AI method to achieve AI-assisted intelligent routing for re-configurable wireless networks, via a heuristic approach. We let the AI algorithm perceive the external network environment, as well as plan and make decisions to guide future actions in the network. This process repeats and constructs a continuous loop to allocate network resources effectively. This intelligent routing algorithm is implemented in the control plane of the SDN. We use a recurrent multi-layer perceptron network in the control plane to achieve intelligent routing. Multi-layer recurrent perceptron network has powerful learning and generalizing ability and is used to obtain the link weight update in the network. The inputs to train the AI model are The energy efficiency of the network, the network link-weight matrix, and the current end-to-end request vector. The output from the AI model is an updated network link-weight matrix to facilitate intelligent routing. We also implement a link weight updation scheme to accommodate the new link-weight matrix.

The cognitive process implemented in the system is illustrated in Fig.3.1 [5]. Learning occurs as a result of accumulated past actions and the utilization of observed knowledge to engage in long-term reasoning processes. Reasoning entails the act of making decisions aimed at enhancing the energy efficiency of a network. Within the cognitive loop, there exist six components: perception, planning, decision-making, action, learning/reasoning, and external communication networks. The AI system perceives its surrounding networks through a sensing module and makes decisions based on the acquired network information. Simultaneously, the sensing module stores valuable network information to support the learning/reasoning module. This learning/reasoning module assimilates useful network information to aid future decision-making by the decision module. The planning module identifies actions based on the perceived network information. The decision module then selects actions considering the outcomes of previous steps.

Graph theory is used to develop the cloud computing network for re-configurable wireless networks as a graph. Let G(V,Z,W) be a graph representing the network, where V is the set of nodes in the network, Z is the set of links between the nodes and W represents the weight of each link. According to [5], the problem can be expressed as an optimization problem as follows

seek an optimal path p for any request r

s.t.

$$\min \eta_{EE} = \sum_{l_y \in L} E(x_{ij}) y_i / \sum_{l_{ij} \in L} x_{ij} \tag{3.1}$$

$$x_{ij} \leq C_{ij} \tag{3.2}$$

Here, the Eq 3.1 is the first constraint which denotes the minimum energy efficiency and Eq. 3.2 is the link load constraint. Energy efficiency is defined as the sum of the energy consumption per bit for each link in the network.

The energy-efficient optimization [5] model can be constructed as follows

$$\min \eta_{EE} = \sum_{l_y \in L} E(x_{ij}) y_i / \sum_{l_{ij} \in L} x_{ij} \tag{3.3}$$

s.t.

$$\sum_{i,s,d \in N} f_{ij}^{sd} - \sum_{i,s,d \in N} f_{ji}^{sd} = \begin{cases} r_{sd} & \forall (s,d), j = s \\ -r_{sd} & \forall (s,d), j = d \\ 0 & \forall (s,d), j \neq s,d \end{cases} \tag{3.4}$$

$$x_{ij} = \sum_{s \in N} \sum_{d \in N} f_{ij}^{sd} \lambda_{p_{sd} l_{ij}} \tag{3.5}$$

$$x_{ij} \leq \alpha C_{ij}, i, j \in N \tag{3.6}$$

$$\alpha \in (0,1) \tag{3.7}$$

$$y_{l_{ij}} = \begin{cases} 1 & \text{if link } l_{ij} \text{ is active} \\ 0 & otherwise \end{cases} \tag{3.8}$$

$$\lambda_{p_{sd} l_{ij}} = \begin{cases} 1 & \text{if path } p_{sd} \text{ uses link } l_{ij} \\ 0 & otherwise \end{cases} \tag{3.9}$$

Eq.3.3 is the energy efficiency optimization objective function for the entire network. Eq.3.4 describes the traffic conservation in the network. Eq.3.5 describes the relationship between link loads and end-to-end requests, Eq.3.6 describes the traffic constraints on a link Eq.3.7 is the link utilization ratio threshold, where $\alpha \in (0,1)$. Eq.3.8 determines if the link is On or OFF and Eq3.9 is a value which determines if a path uses a link in the network.

The AI-assisted routing system is developed in two stages:

- Firstly, an algorithm for solving the routing problem using a greedy approach is developed. This algorithm is the Energy Efficient Integral Wireless Routing Algorithm(EEIR).
- Secondly, the results obtained from the EEIR algorithm is enhanced by feeding the outputs from the greedy algorithm to a recurrent multilayer perceptron network.

3.2 Technologies Used

The following technologies were used to develop the application :

1. Python 3.x
2. NetworkX: A versatile Python module to construct and visualize graphs
3. TensorFlow: A library developed by Google for creating and running machine learning models using data flow graphs and tensors
4. Tkinter: A GUI library for Python

3.3 Initial Data Parsing

The network topologies on which the developed algorithm should run are sourced from *sndlib.zib.de* which is a repository consisting of a range of topologies of varying sizes and demands on which the algorithm can be run. The topologies are available to download as text files. These text files consist of a set of nodes, the properties of the nodes, the links between the nodes and a set of demands routed through the network. This text file must be imported into the Python environment, and all of the attributes mentioned above must be pushed into suitable data structures. Basic file parsing techniques are used to perform this.

3.4 GUI Development

The *Tkinter* module within Python is used to develop a GUI to display the graph topologies at various stages of the algorithm. An image viewer screen is designed with scrolling functionality to view various outputs that have been generated by the algorithm. The grid mechanism with Tkinter has been used to place various components on the screen. A dropdown box is provided to select the topology on which the algorithm has to be run.

3.5 NetworkX Graph Plotting

NetworkX is a library within Python for creating, manipulating and studying the structure and dynamics of complex networks. All the graph printing functions are defined in a separate Python module, because of its repeated use in various stages of the algorithm. The graphs are plotted using *Matplotlib* and they are simultaneously saved to the results folder of the application. These images are later used by the GUI to display the results.

3.6 Development of the EEIR Algorithm

he Energy Efficient Integral Routing Algorithm is an algorithm that aims to minimize power consumption by turning off unwanted links. The EEIR algorithm uses Dijkstra's algorithm as a base for finding the set of shortest paths for all the demands for a particular topology. Therefore the input for the EEIR algorithm is the graph representing the topology and the demands list. The output produced is the graph having the updated transmission rates.

The algorithm first calculates the shortest path for each demand in the network. It then proceeds to deactivate unused links and updates the set of active links. Next, the EEIR method selects a link with the highest remaining bandwidth and attempts to deactivate it, while ensuring that all demands can still be accommodated. If turning off a link affects any traffic demand, the algorithm generates a set of alternative shortest paths to reroute that demand. However, if a link cannot be deactivated without violating the demand requirements, it is added to a set of links that must always remain active. Overall, the algorithm consists of three main steps as described above.

Step 1: for each demand D, EEIR generates a set of shortest paths $SP_{(s,t)}$. Dijkstra's algorithm is used for this and the result is stored in S. The criteria for selecting the shortest path is the minimum number of hops. Bandwidth d_{st} is also reserved along the shortest path SP_{st}.

Step 2: Maximum amount of links are turned off by the algorithm in this step, keeping into consideration the routing demands. The total bandwidth flow $l_{(u,v)}$ on each link is calculated. The transmission rate which just satisfies the total flow is taken as $z_{(u,v)}$. The residual capacity is calculated by subtracting the total flow from the transmission rate and removing unused links i.e. if the total flow through a link is zero then the link is removed from ε_{ON} and the transmission rate for the link is set to zero.

Step 3: EEIR selects a link with highest residual capacity$r_{(u,v)}$. The Reroute() function, which is a greedy algorithm is run to find the feasibility of bringing down the transmission rate of a particular link. The aim is to reduce the transmission rate and reroute multiple demands within the system. The selection of demands for rerouting depends on the excess rate $e_{(u,v)'}$ above the degraded link rate $z^{'}_{(u,v)'}$. If it is possible to reroute the demands, the transmission rate of the link $(u,v)'$ is reduced by one level, and any link $(u,v)'$ with $z^{'}_{(u,v)'} = 0$ is removed from the set of active links. However, if rerouting the demands is not feasible, the link $(u,v)'$ is stored in the set of fixed links.

3.7 Development of the cognitive learning process

AI concepts are used to further maximize energy efficiency in the routing process. Recurrent multilayer perception networks are used to achieve this. The multilayer perception is implemented over here using LSTM which are used to solve problems that involve sequential data. The figure shows the links weight update scheme where W stands for the weight vector, $\tilde{R}$ represents the end-to-end request vector consisting of z value and e value. $\hat{W}$ represents the updated link weight matrix. We obtain the

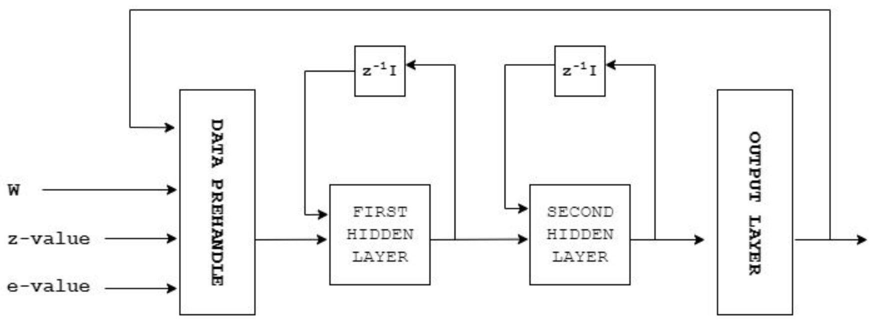

Figure 3.1: Link Weight Update Scheme

following equation:

$$\hat{W} = h(W, \tilde{R}) \tag{3.10}$$

where $h(.)$ represents the map from input to output. We use the TensorFlow package provided by Google to develop the multilayer perceptron network. The function takes the weight list, z value and e value as the inputs. These are the lists of data that will be used to train the neural network. This data is split into training and testing datasets. The neural network itself is designed using TensorFlow's *RNN* layer with an *LSTM(Long Short-Term Memory)* cell. LSTM is particularly useful for solving problems that involve sequential data. LSTMs are able to store past information that is important and forget the information that is not, making them very effective for sequence prediction problems. The output of the neural network is calculated using a dense layer with one unit. The loss function used to train the model is mean squared error, and the optimizer used is Adam. The loss function calculates the overall error of the model and the optimizer updates the weights of various attributes of the model to minimize the losses. The Mean Squared Error (MSE) loss function is commonly used for regression problems, where the goal is to predict a continuous value. The model is trained for 3 epochs using the training data.

Chapter 4

Results and Discussion

4.1 Landing Page

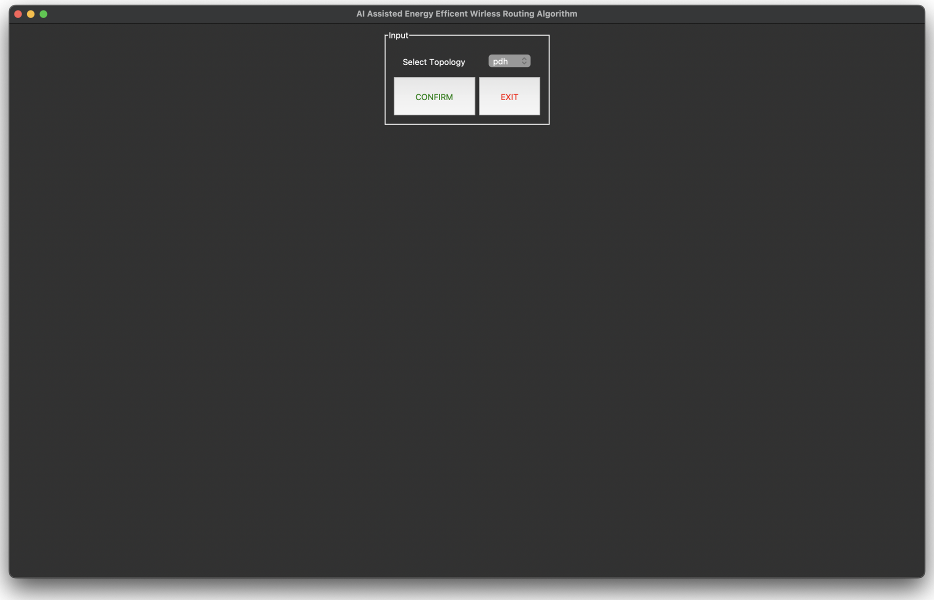

Figure 4.1: Landing Screen

The landing page has been developed using Tkinter. A drop-down menu is provided to select the topology on which the algorithm has to run. pdh topology is selected by default topology because of the minimal number of nodes and demands it has. pdh is the network representation of a highly connected industrial area covering 11 industrial regions. This topology has 24 demands. Pressing the submit button runs the algorithm on the topology. The user can exit the application by using the exit button.

The grid system in Tkinter has been used to set the layout of the GUI. (Fig 4.1)

4.2 Total Flow

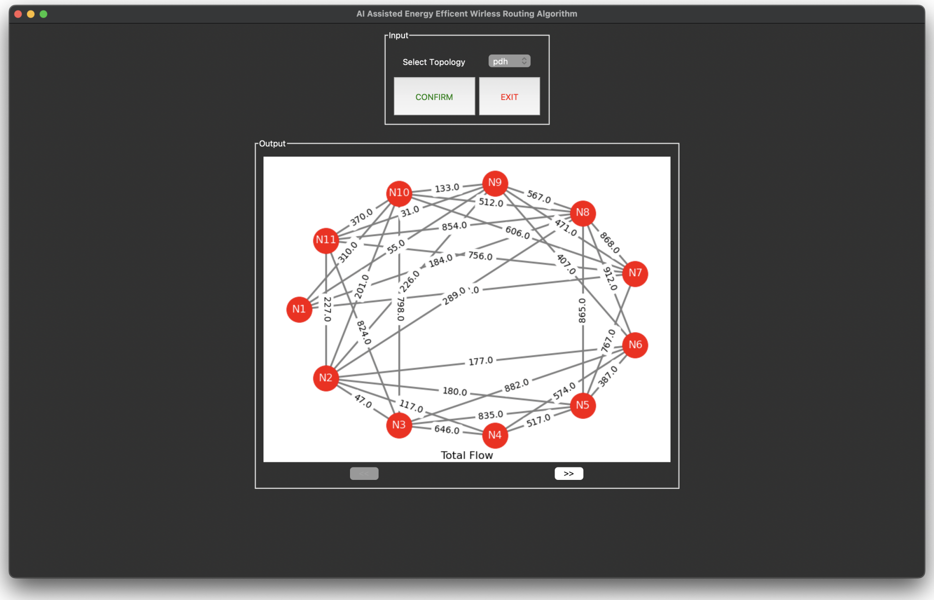

Figure 4.2: Total Flow Graph Plotted using NetworkX

The first graph that the algorithm provides is the graph representing the total flow through each of the links within the graph. This is done by running the Dijkstra algorithm for all possible demands within the graph. The total flow is shown as the link weights in this graph.

4.3 Transmission Capacity (Initial)

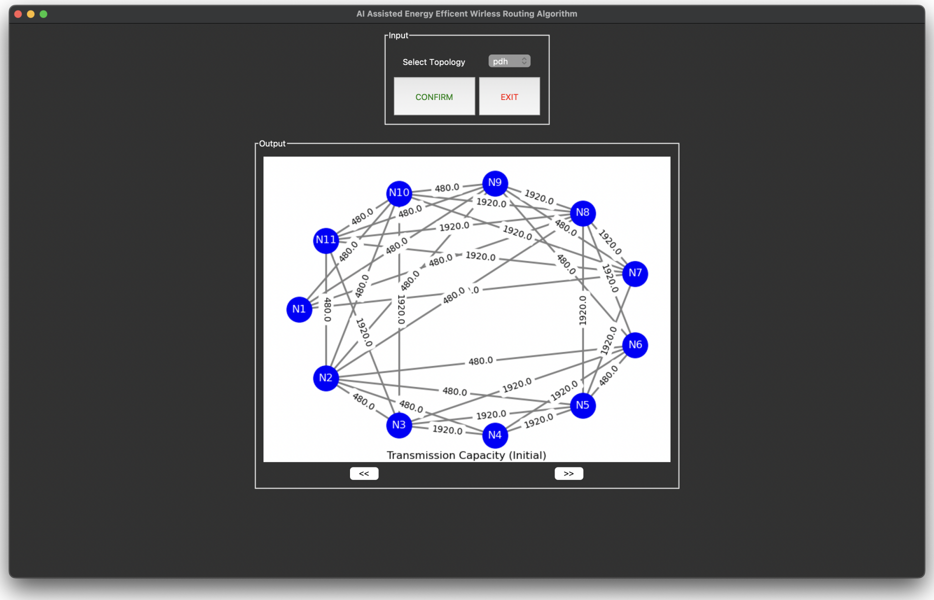

Figure 4.3: Transmission Capacities before running the algorithm

The second graph generated by the algorithm shows the transmission capacity that all the links should maintain so that all the demands of the network are met before the EEIR algorithm is run on the network.

4.4 Transmission Capacity (Final)

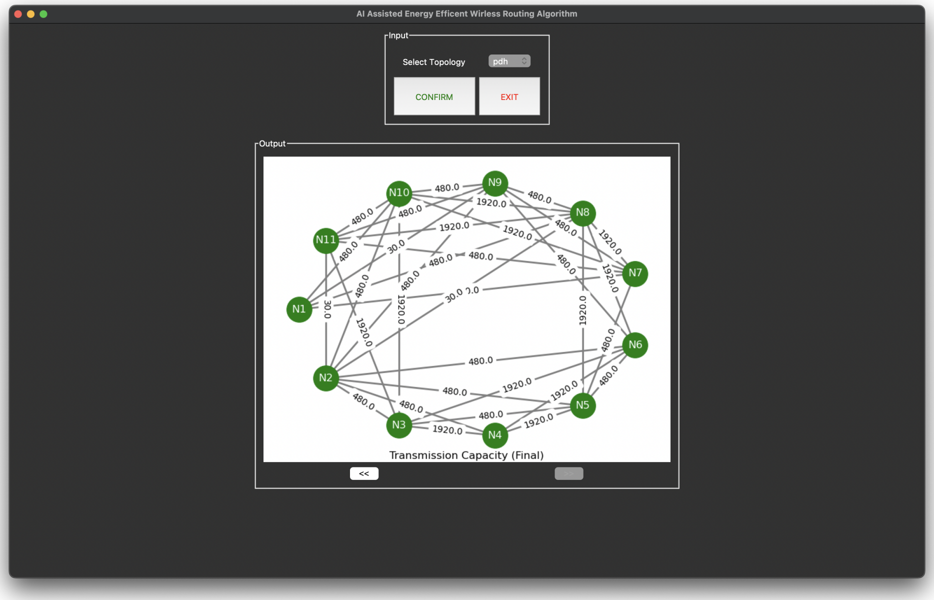

Figure 4.4: Transmission Capacities after running the EEIR algorithm

The third graph generated by the algorithm is the transmission capacities that each link must maintain so that all the demands of the network are met after running the EEIR algorithm.

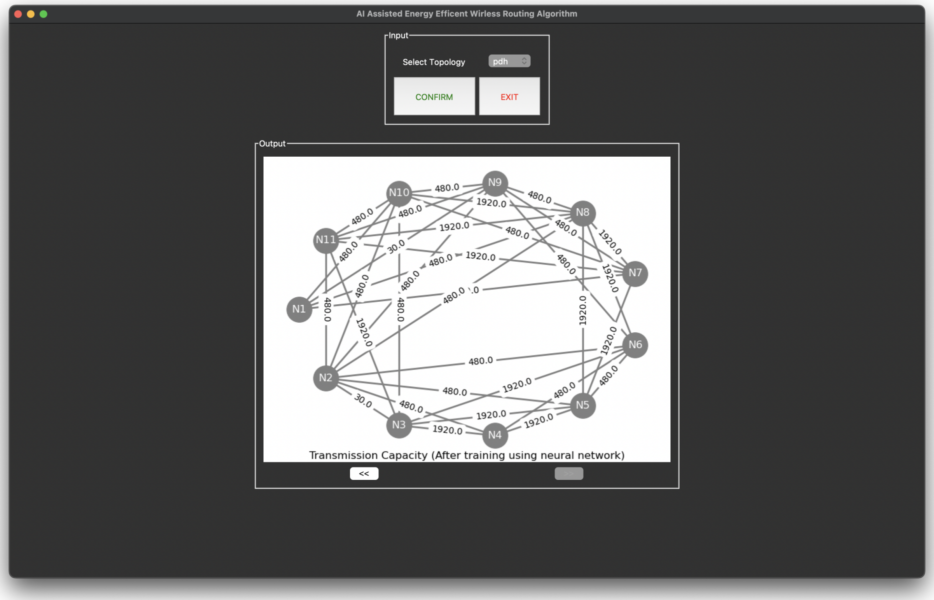

Figure 4.5: Transmission Capacities after training using recurrent neural network

After running the EEIR algorithm on the graph we train a recurrent neural network which helps in further optimizing the transmission capacities for various links within the graph.

4.5 Performance Evaluation

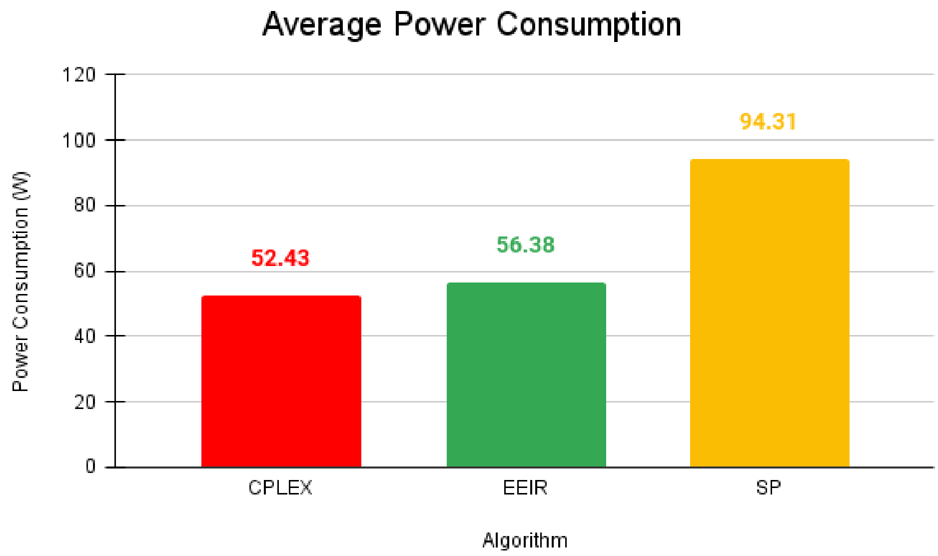

Figure 4.6: Comparison of average network power consumption

Fig 4.6 shows the average power consumption for routing all traffic demands using CPLEX, EEIR and SP algorithms. We find that the average power consumption for SP is higher than both CPLEX and EEIR. SP optimizes the routing based on hop count rather than energy consumption. We also observe that EEIR has a very close performance to CPLEX.

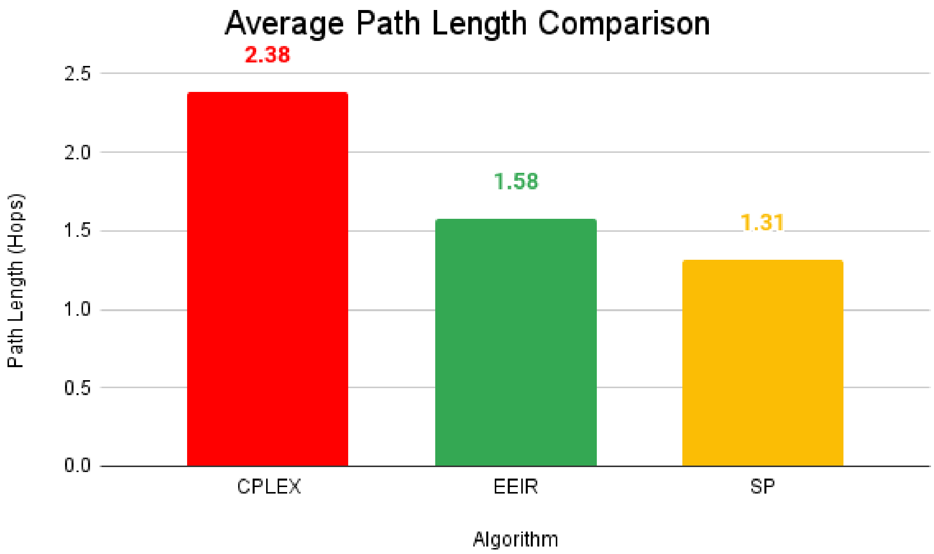

Figure 4.7: Comparison of average path length

Fig 4.7 shows the average path lengths of routes generated by CPLEX, EEIR and SP solutions. SP has the shortest path length since it routes demands based on minimum number of hops.

Chapter 5

Conclusion

In the near future, it is inevitable that the number of networking-capable machines will grow exponentially. Developing a self-configuring and energy-effective routing scheme is therefore essential for proper network administration. We have laid forth a system wherein energy efficiency prioritization and AI theory serve as the foundations for the routing algorithms which is self-learning in nature. This system enables the network management system to dynamically change the link weight in order to decrease the network's energy consumption based on historical data. This system utilizes the EEIR algorithm and a recurrent multi-layer perceptron network to deal with network traffic in a versatile manner so as to conserve energy and reduce congestion due to large packets at peak time. This system is also efficient in different topologies as it is able to adapt to different types of networks based on the specific traffic of the particular network. Overall, the system enables the smooth flow of network traffic in an efficient manner which will reduce the delay in availability of resources and at the same time, will reduce losses due to over-consumption of energy.

References

[1] H. Zhang, F. Fang, J. Cheng, K. Long, W. Wang, and V. C. Leung, "Energy-efficient resource allocation in noma heterogeneous networks," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 48–53, 2018.

[2] J. Shi, W. Yu, Q. Ni, W. Liang, Z. Li, and P. Xiao, "Energy efficient resource allocation in hybrid non-orthogonal multiple access systems," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3496–3511, 2019.

[3] H. Wang, Y. Wu, G. Min, J. Xu, and P. Tang, "Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach," *Information Sciences*, vol. 498, pp. 106–116, 2019.

[4] H. Yao, H. Liu, P. Zhang, S. Wu, C. Jiang, and S. Guo, "An intelligent approach to energy efficient transportation and qos routing," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[5] D. Jiang, Z. Wang, W. Wang, Z. Lv, and K.-K. R. Choo, "Ai-assisted energy-efficient and intelligent routing for reconfigurable wireless networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 78–88, 2021.