A Novel Optimization Method to Mitigate Congestion in Edge Computing using Tabu Search Algorithm

MSc Research Project
Cloud Computing

Prajakta Balu Mhatre
Student ID: x21136530

School of Computing
National College of Ireland

Supervisor: Dr. Rashid Mijumbi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Prajakta Balu Mhatre
Student ID:	x21136530
Programme:	Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Rashid Mijumbi
Submission Due Date:	15/12/2022
Project Title:	A Novel Optimization Method to Mitigate Congestion in Edge Computing using Tabu Search Algorithm
Word Count:	6589
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Novel Optimization Method to Mitigate Congestion in Edge Computing using Tabu Search Algorithm

Prajakta Balu Mhatre
x21136530

Abstract

The system architecture for edge computing (EC) has changed drastically in recent years. However, the network protocols are frequently diverse, which aids in reducing latency. Radio gadgets are currently spreading rapidly, including cellphones, intelligent cars, body-warn tools, and sensing equipment. As a result, there has been a significant increase in wireless traffic, which might cause congestion in the main network and put a lot of load on the transmission channels. Furthermore, the need for real-time data processing and aggregation is driven by a wide range of characteristics and strategies, such as computer simulation software and smart transportation. The network traffic disrupts packet transmission, resulting in low throughput and significant energy usage. These factors reduce the infrastructure's limited bandwidth and quality of service (QoS). Recent research shows that the existing algorithm shifts traffic from high-energy nodes to low-energy nodes when a node becomes congested, which causes packet loss, uneven energy consumption, and low throughput. In the proposed research we will calculate the Packet delivery rate (PDR), and if the PDR is less than the threshold, it will remove the unfit solution and using the Tabu search algorithm find an alternate route. The simulation is performed for a specific time interval in Network Simulator 2. The research shows that the proposed algorithm is better in terms of throughput, energy efficiency, and energy consumption compared to several existing algorithms.

1 Introduction

Edge computing is a distributed technological (IT) architecture Fig.1 where customer data is handled at the network's border as close to the primary sources as is feasible. Data, which provides invaluable, sophisticated analytical capabilities and facilitates real-time control over critical corporate operations, is the beating heart of today's businesses. The amount of information that can be consistently gathered through sensors and IoT devices running in real-time in remote areas and challenging operating circumstances is enormous, and it is now literally available to businesses all over the world. However, this cloud-based avalanche is also changing how businesses should handle data processing. The traditional computing architecture, which is built on decentralized storage and the open internet, is not well suited to transporting streams of continually growing real-world information. Such operations might be affected by bandwidth limitations, unplanned network failures, and traffic issues (Hull et al. 2004). Edge computing technology is being used by businesses to solve these data issues. Edge computing moves a portion of the storage and computation capacity closer to the real data source and distant from the data center network.

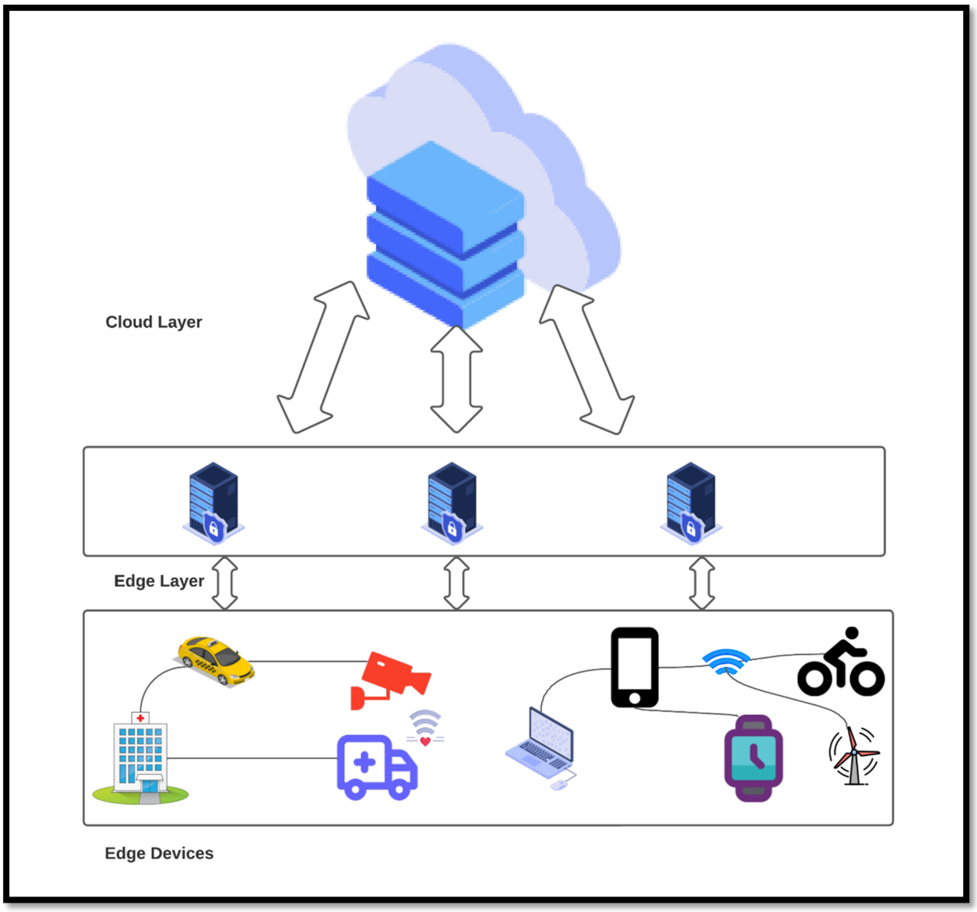

Figure 1: Architecture of Edge computing

Instead of transferring raw data to the central data center for data extraction, that work is now performed where the data is produced, for instance in farming, retail, a massive utility, or across a smart home. Edge computing congestion mitigation is still difficult, despite the massive array of strategies that have already been offered in the past. When traffic volume exceeds the capacity of a node or network, congestion results, producing buffer overflow and interference. To avoid network failure and provide the essential QoS for edge computing, sensor congestion management is required since congestion in sensor networks is mostly unsolved.

1.1 Background and Research Motivation

Congestion in wireless sensor networks causes a buffer overflow, which lengthens the latency. The typical approaches increase the delay by using rate modification to reduce congestion. An AODV-based strategy to prevent congestion from occurring is presented in(Riaz et al. 2019) this work. The present bandwidth usage is evaluated under different conditions by the available bandwidth in the proposed strategy. The source node receives information about the condition of the network and adjusts its data rate as a result. The typical cloud computing approach, the IoT, also has significant shortcomings. Edge computing's reduced latency can improve user program performance and offer a positive application experience. Congestion in the network is resolved at the same time by proportionately reducing the bandwidth burden of each network. The WSN architecture and upper limit for the congestion threshold were then introduced, along with a congestion management method based on cooperative pathways. Applications must choose the best algorithm for specific requirements, and then when new requirements arise, customizing CC algorithms requires a great deal of time and knowledge. The primary component of MOCC is a multi-objective supervised learning architecture that enables CC to proactively discover connections across various application needs and the related best control strategies.(Ma et al. 2022). However, there are still certain gaps in the study that has previously been carried out utilizing such methods as data loss, power usage, and QoS. The Tabu search method is suggested in this research as a means of reducing network congestion and boosting throughput.

1.2 Research Question

Can the proposed methodology choose the best route in edge computing networks to reduce congestion when transmitting data packets from the source to the destination node?

1.3 Research Objectives

The major study objectives are taken into account in order to reduce network congestion during packet transmission.

- **Objective 1:** Applying the Tabu search technique to reduce network congestion
- **Objective 2:** Compare the suggested system to existing protocols in a comparative evaluation.

1.4 Document Structure

This document is made up entirely of the sections that follow. Part 2 covers the most relevant studies and research in the field of edge computing. Section 3 methodology goes into further detail about our strategy for the AODV algorithm and Tabu search. With the use of a flowchart, Section 4 outlines the design requirements of the suggested research. The suggested Tabu search algorithm's step-by-step implementation is described in Section 5. Moreover, Section 6 consists of experimental results. Lastly, conclusions and future work, are included in the summary of Section 7.

2 Related Work

2.1 Congestion Control (CC) in Edge Computing:

Both cable and remote communication networks experience congestion, which is a serious problem. To reduce congestion, many new techniques are used to improve the network, data connections, and transport levels. Traditional congestion control techniques are ineffective for layer four (the transport layer) protocols when used with congestion control algorithms for multimedia streaming.

Based on the power stability and cyber, this study (Zhang et al. 2022)established a new approach for establishing an EC node to fulfil the decentralized EC needs of the cyber-physical distribution system (CPDS). The best assignments of the CPDS network edge computing nodes is determined using the maximum entropy principle after the development of the CPDS information transmission stability calculation model, CPDS information-power hybrid entropy calculation model, and CPDS power stability calculation model. The simulation results showed that the devised method could effectively regulate the distribution network while reducing congestion and improving response times. To increase the throughput of the energy efficiency (Xiao et al. 2022) introduced the CC algorithm using cooperative routes for WSN (wireless sensor network). On the framework of EECCA (energy efficiency congestion control algorithm), this technique then attempted to investigate the problems of MPTCP (multipath traffic control problem) using the CC method and evaluate the outcomes. The study's findings showed the superiority of the EECCA over any system since it increased output by up to 52.6% and increased energy efficacy by up to 47.3%, both of which helped conserve energy. A CMAC (cooperative multi-agent actor-critic)-DRL (deep reinforcement learning) approach was developed in different research (Zhang, Zhou, Lu & Fujita 2021) using an EC model. The weight distributions of local agents were considered for the overall traffic structure through the use of a cooperative approach. The coordination of the MLAs (Multiple Local Agents) was used to carry out the optimal solution. The results of the research show, when compared to alternative approaches for reducing traffic congestion, the developed technique worked more successfully. A strategy for analyzing transmission routes with traffic and mitigating it by employing interest forwarding control at edge nodes was predicted by (Qin et al. 2020) as part of the EACC (edge computing aided congestion control) project. To document the path's state of congestion, a unique field was added to each data packet. The goal of this technique was to make a local MDP (Markov Decision Process) problem involving interest packet forwarding into a control issue based on local user request data and path congestion information that was returned. NDP (neuro-dynamic programming) was used to finally address this problem. A simulation of the proposed technique was performed using ndnSIM. The simulation's results demonstrated the viability of the proposed approach.

An adaptive online decision method is proposed, and deep reinforcement learning (DRL) was created at a steady and quick pace in order to fully comprehend the best plan and mitigate congestion (Xie et al. 2019). Additionally, a method for enhancing the efficiency of the data that the online learning provided was proposed. It compares the methodology to other approaches, such as the conventional static IW setting, an existing IW setting, Smart IW, and a methodology created for search engines. It may efficiently reduce the average FCT while producing minimal congestion. It also adjusts to shifting congestion. This method helped to reduce FCT congestion, at the very least. Additionally, this can change IW in response to shifting network parameters. This study also demonstrates the efficiency of various factor settings and policy designs.

For broadcasting the packets of data along many routes, (Kumar et al. 2020)built the MP-PS (Multipath Packet Scheduling) method in this paper. Using the probability of a path scheduling delay (PSD), MPS was employed to schedule. For mitigating congestion, a multipath congestion control method was used. Based on buffer capacity and PDR, an explicit response was implemented for the rate of data transmission in this case. The improved performance and lower packet drop were given top priority in the designed approach. The simulation findings proved that the built-in model outperformed conventional techniques in terms of buffer acknowledgement and throughput.

2.2 Energy efficient Task offloading in Mobile Edge Computing (MEC):

Among the most smart technique for governing and monitoring the environment is WSN, which can self-organize and gather data from the device without the help of infrastructure. There are many problems with this WSN, with power consumption being a major one. If the energy available to the sensor nodes is insufficient or there is an energy imbalance, grid performance measures may be obscured.

An MTO (multi-task offloading) strategy was proposed by (Zhang, Yi & Ma 2021) to increase the device's energy efficiency. Device battery utilization is one of the limiting factors to consider while task unloading. A BDT (binary decision tree) technique was used to jointly optimize the number of jobs, clock frequency structure, and smartphone transmission power to transfer them further. MATLAB was used to carry out the proposed approach. The outcome shows that the proposed approach was able to bring down the energy consumption of devices when compared to existing methods. This study (Mahenge et al. 2022) proposes a task offloading method to reduce energy consumption and to meet capacity and delay demands. Based on MEC, a solution for EETO (energy-efficient task offloading) was described, using a cooperative approach.

(Guo et al. 2019) proposed a technology that connected mobile devices to an edge server and made use of a standard channel of communication. To avoid interfering with the operations of running applications when the communication channel is blocked, the recommended heuristic strategy also allocates the tasks of a previous programme execution to free server cores. Moreover, the offloading problem in the edge cloud computing paradigm was studied using a single edge server. The paper describes task offloading, NCC, pricing systems, and CRSA design as significant study topics and their methodologies. (Zhang et al. 2020) . In the 6G era, the average MEC experiences restricted utilization for computing and broadcasts the resources over the whole network, resulting in ever longer processing and transmission delays, especially for computation-intensive services. A mathematical model demonstrated that, when compared to the conventional methodologies, this strategy was highly effective in reducing latency and increasing processing speed. Furthermore, a 3D DM (three-dimensional decision matrix), which depicts the time moment for offloading the useful work to a particular server, was designed as part of an ideal approach for resolving the offloading issue (Zhu et al. 2019) . To reduce the completion, latency while staying within the energy constraints, an offloading problem was created that took into consideration both the energy consumption and the entire completion latency, which includes the processing delay and the communication delay. Finally, the developed strategy worked well to increase the latency and computation complexity.

2.3 Resource allocation in EC and MCC:

In the context of EC-enabled IIoT (industrial internet of things), (Baranwal et al. 2022) studied a DARA (decentralized auction-based resource allocation) approach. For this, the consortium blockchain and smart contract were deployed to do away with the need for an auctioneer, a dependable middleman. This method helped fulfill the needs of resource allocation, such as seal bidding, etc. The method under investigation helped the edge servers perform the bidding consistently, improving the outcomes to reallocate the resources in polynomial time. This method helped satisfy the demands of resource allocation, such as seal bidding, etc. (Yang et al. 2022) proposed a method for dividing the offloading process into a two-stage model and reversing the order to handle the offloading choice and issue related to resource allocation in order to reduce the complexity of the action and state space. DDPG (Deep Deterministic Policy Gradient Algorithm) and MDP (Markov Decision Process) were proposed as solutions for the edge server load balancing, latency, and transmission energy reduction issues. Then, a genetic algorithm (GA) was introduced for the purpose of finding decisions. The results have demonstrated the recommended approach's stability, flexibility, adaptability, and suitability for actual applications in contrast to alternative approaches.

(Lieira et al. 2021) studied an approach in which TRIAD (5G-IoT resource allocation decision) in edge computing was implemented using the whale optimization algorithm (WOA). The researched approach was compared to the Greedy and Reliable methodologies for analysis. The results showed that this technique was more efficient for serving devices since fewer requests were turned down and fewer devices were banned while being searched. According to the simulation findings, the examined approach might provide more than 26.5% of services while blocking services below 65% and refusing the least number of requests up to 56%. Regarding the JCC subcarrier, RA's power, and computing resource restrictions, (Li et al. 2019) (Li et al., 2019) developed a SO (stochastic optimization) issue that was beneficial for enhancing system efficiency and maintaining queue stability (joint congestion control and resource allocation). Using the Lyapunov optimization method, four drawbacks are separated. Single-factor challenges include resource allocation for smartphones, congestion control, and utility maximization. The combined power and subcarrier allocation problems were solved using alternating and time-sharing strategies. Lastly, A reinforcement-learning-based state-action-reward-state-action (RL-SARSA) method was suggested in (Alfakih et al. 2020) to tackle the resource management problem for selecting one of three offloading alternatives. The crucial economic concerns of edge computing resources have generally been disregarded, despite the fact that great efforts have been devoted to analyzing the allocation of resources. The suggested approach addresses the majority of CPSS issues by offloading to nearby edge servers, which yields the best outcomes in terms of volume, diversity, velocity, and veracity.

2.4 Tabu search in previous research

Various research projects have been conducted utilizing Tabu search across various networks, including WAN, VANET, and MANET. In a mobile ad hoc network, tabu search is employed to select the optimum route between the transmitter and receiver. The author begins by outlining the most popular operations for the encoding and fix functions. introducing the neighbourhood formation procedure with a termination condition in addition.(Jang 2012). By exploiting multi-channel allocation capabilities, tabu search is used in this study to improve QoS and decrease latency and jitter in VANETs. where there is a system that ranks each communication according to its kind. The scheduling of transmissions also uses tabu search (Ishaq et al. 2018). (Sharma & Panjeta 2022) Beacons and dedicated short-range communication (DSRC) are two forms of communication used by vehicles. When the number of automobiles in a network increases, network congestion occurs as a consequence of the widespread usage of safety alerts. The proposed strategy looks at ways to detect and manage congestion in order to reduce congestion-related problems. In order to minimize network congestion, the proposed method in this research changes the signal transmission rate using the Tabu-search algorithm and makes use of a priority model. In terms of PDR and end-to-end delay, the performance of the proposed system is compared to that of cutting-edge methods.

2.5 Summary of Releated Work

In summary, this study differs from previous research in many ways: first, the majority of prior studies used the multi-path traffic scheduling method to solve the congestion management problem in edge computing. Second, for energy gain, high efficiency, and speed, the study review concentrated on energy-efficient task offloading in mobile edge computing. The allocation of resources in EC and MCC methods is resolved by another proposal. The proposed technique in this research varies from earlier work in numerous ways. The main objective of the current congestion control method is to make the network free of congestion. Algorithmic solutions created to do this lower the amount of energy used by the network. Different types of network challenges, such as connection failure, can be handled via cooperative route algorithms. The existing algorithm shifts traffic from high-energy nodes to low-energy nodes when a node becomes congested, which causes packet loss, uneven energy consumption, and low throughput, but None of the research focused on bandwidth, Packet delivery rates and, or the number of neighbours' parameter using Tabu search algorithm. In the proposed research we will calculate the Packet delivery rate (PDR) and if PDR is less than the threshold, it will remove the unfit solution and using the Tabu search algorithm find an alternate route.

3 Methodology

Effective congestion controls must be designed in order to fulfill edge computing's reliability and QoS needs. The principal objective of this study is to mitigate network congestion utilizing several algorithms, such as AODV and the Tabu search algorithm, which allow for effective data transfer from the source node to the destination node without experiencing any network congestion. Several characteristics can successfully serve as the major metrics in edge computing, which might lead to network congestion. Packet delivery rate, packet size, bandwidth, number of neighbor nodes, and throughput make up these factors.

3.1 Methodological Process

The following processes make up the methodological system: The method is used in this project to generate a virtual environment using a VMware virtual machine. Ubuntu 18.04 is set up in the VMware virtual machine to create the operating environment for the simulation by allocating 4GB of RAM. Before deploying complex and difficult networks in real-world applications today, simulation greatly helps in the analysis of their performance and behavior. Numerous network simulators are readily accessible, and their output resembles real-time implementation as closely as possible. We employed the precisely defined simulator NS2 in this study, and an AWK script was used for performance analysis. The NS2 simulator offers a variety of models, out of which we took into consideration the following models:

- **MAC model:** With the correct frequency, bandwidth, and MAC layer functioning as in IEEE 802.11, the radio properties of the node are represented by the radio model.
- **Traffic Nodes:** Which deliver traffic to the receivers end from the sender, mostly use the CBR and UDP models for traffic.
- **Node deployment model:** It provides node placement and location as a consistent methodology.
- **Energy- model:** It includes power supply, memory, and computing power, among other things

3.2 Steps of Methodology

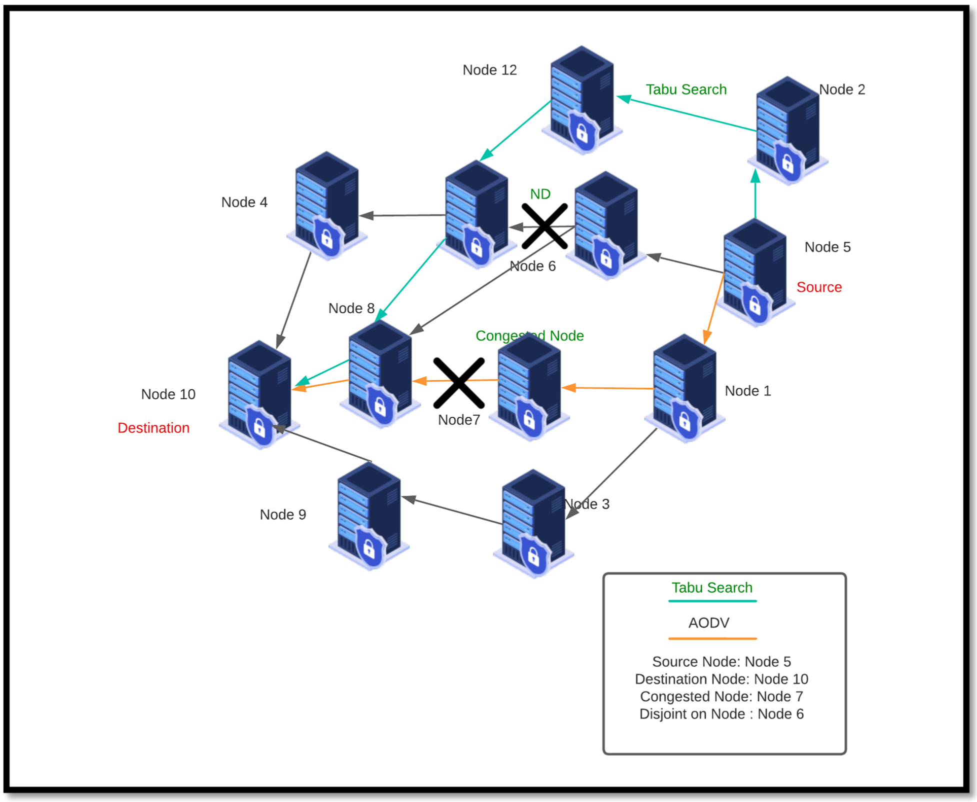

Figure 2: Architecture of proposed Tabu search Algorithm

Step 1: There are n nodes in a network, including a source node and a destination node.
Step 2: Congestion is caused when a source node sends a large number of packets over a route to a destination node.

Step 3: In the event of congestion, we must find an alternate route to the destination node.

Step 4: When congestion occurs in an existing algorithm, transfer the high-energy node to the low-energy node in comparison to the existing method. To locate an alternate path, we propose a tabu search technique.

Step 5: We create a fitness function with three parameters for each node to discover an alternative path. The first one is PDR; a high PDR indicates that congestion is at a minimum at that node. Next is the number of specific neighbour nodes. If one node is located in a dense environment (with a large number of neighbours), communicating with that node might cause congestion.

Step 6: To determine an alternate path, we calculate the fitness value of each node. After that, the unfit solution will be removed where the route is already congested. In the next step, we will apply the tabu search technique to construct a random path and determine its fitness function. If a new fitness function is better than an existing solution, we will consider it.

Step 7: In the output of tabu search, we will get the least congested or congestion-free route to send data from the source to the destination.

A proposed network topology is built using the framework's architecture to determine the best route between the transmitter and the receiver 2. To select the optimum route, we will first use the AODV protocol. Data transmission on that node begins when route establishment is complete. Using Network Animator, each topology is simulated. The creation of a NAM trace file is the initial step in using NAM. Along with packet trace details, the NAM trace file should also include topological information about nodes, connections, buffers, node connectivity, etc.

3.2.1 Ad-hoc On-demand Distance Vector Routing Protocol

AODV is combination of on-demand and distance vectors, which is hop-to-hop routing methodology. A node sends a ROUTE REQUEST when it learns the path to a particular receiver node. The route request is then sent to route discovery, which additionally build a reversed route to the destination for itself. The query produces a new Response with the hop count necessary to reach the target when it finds a node having a route. A route from source to the destination is established by all nodes that take part in sending this response back to the source node. In contrast to source routing, this route is built hop-by-hop from each node from sender to the receiver.

Disadvantage: The need for addressing problems, traffic partitioning along each route, and packet sequencing at the destination restrict the use of more than one transmission channel in AODV.

3.2.2 Tabu search algorithm

A popular meta-heuristic for improving model parameters is tabu search. The fundamental principle of tabu search is that by enhancing local search with short-term memory, it becomes more capable of finding the best alternatives. It is an incremental search that works like a hill-climbing algorithm, starting with a feasible solution and working toward finding the best one. The algorithm keeps track of previous local optimal locations in order to quickly and effectively reach the near-global optimum. The objective is to prevent the occurrence of the same local optimum many times during the procedure. The retracing method begins by returning to a local optimum in the tabu list and then looking for a new optimal in various directions. When the backtracking criteria (BC) is met, backtracking is executed. In this study, we discuss how to utilize tabu search to choose the optimal route to reduce congestion.

1. **Short-term memory:** This prevents the search algorithm from revisiting alternatives that have previously been examined and is based on the recency of their occurrence. In order to customize and enhance a search, it may also be used to return to effective parts of the search. This is executed via the Tabu List and is often known as "intensification."

2. **Intermediate term:** Search biasing rules for intensification favor promising search space regions.

3. **Long-term memory:** By avoiding previously investigated areas, long-term memory, which is derived from the rate of occurrence, is utilized to expand the search and investigate previously unexplored parts of the search space. This is also referred to as "diversity" and is performed via frequency memory.(Glover & Martí 2018)

Our proposed congestion control scheme consists of three phases:

A.Make parameter combinations: We will combine three parameters to verify the fitness value using Tabu search. PDR, the number of neighbor nodes, and bandwidth are the parameters.

B.Compare combinations: Next, these algorithms must properly compare the performance of various combinations. It will be compared with the present solution, as well as short- and long-term memory.

C.Select best combination: Finally, meta-heuristic algorithms choose the best combination based on fitness value. The most significant aspect that determines the termination is the criteria for selecting the optimal path, which is a better fitness value.

Advantage Tabu Search:

- Can choose non-improving strategies to avoid local optimal situations.
- The use of the Tabu List can stop loops and the use of unfit solutions.
- It is possible to apply both sequential and distinct solutions.

3.2.3 Pseudo-code for the proposed Algorithm

The suggested approach attempts to alleviate network congestion in edge computing by focusing on maintaining throughput and QoS metrics. The proposed algorithm fig3 employs the Tabu search method to improve network efficiency and minimize congestion as well as energy consumption, as mentioned. The program initially collects several parameters needed by the Tabu search technique. Additionally, it begins data transmission and determines the PDR for each node. If PDR is greater than the threshold, data transmission takes place. Furthermore, if the PDR is less than the threshold, the unfit solution is excluded, and an alternative direction is called using the Tabu search method. In the alternate route function, we consider nodes that are disjoint for path in networks and remove the node from the initial solution. At the end of the alternate route, the function will get an output of all fitness values (PDR, number of nodes, and bandwidth). In the next loop, we will apply Tabu search for route optimization. We have set iteration; it will generate a solution using a random walk from source to destination. The new solution will be considered the best if its fitness value is the best. And data transmission will happen. Otherwise, it will generate new solutions until it reaches the maximum number of iterations. Finally, network congestion has decreased, and performance measures such as throughput, energy efficiency, and energy consumption have been recorded.

```
1. Set Source node
2. Set Destination node
3. Searching Protocol= Tabu Search (T)
4. P= Path in networks
5. Q= Node in each  Path
6. Perform Computing path fitness of each path by calling Compute path fitness ();
Let Equation 1: PDR = (Packet Receive) / (Packet received + packet drops)
Let Equation 2: fitness[y,z] = (alpha * para1[y,z]) + (beta * para2[y,z]) + (gamma * para3[y,z])

Compute path fitness :
For i= 1:P
    For j= 1:Q
        Compute PDR at each node (j) using equation 1
        If PDR at node ij < 0.80
            mark node as congested call Alternate route () ;
        Else
            keep on transmitting data

        End if
    End for
End for

Alternate route ():
    For i= 1:P
        remove congested paths from initital set of solution
        For j=1:Q
            Compute fintess of each node using equation 2
        End For
        compute fitness of each path
    End For


    For i=1 : T //Tabu search
        generate random solution using random walk between source node and destination node
        get it's fitness values
        If fitness new solution > Fitness old solution
            Mark Solution= Best
            start data transmission
        End if
    End for
```

Figure 3: Pseudo-code of proposed Algorithm

4 Design Specification

The Tabu search algorithm is used to adjust the channel capacity and create a dense network to mitigate congestion. The appropriate channel capacity and a dense network for the data transmission are established based on the best fitness solution offered by Tabu Search. A flowchart depicts the components of the proposed congestion control mechanism.

The flowchart outlines (as shown in fig.4) the proposed methodology's procedure. First, we will deploy a number of nodes in the network and define the source and destination nodes. The route database first originally updated by transferring packets, and when the sender node has data to transfer to the receiver node, it will establish various paths by transmitting packets (Request and Reply packet). The AODV algorithm will be used to choose this route. Once the optimal path has been identified, data transmission will commence. Data transmission across each path cannot be performed at random since the nodes may be congested. First, we will examine the Packet Delivery Rate on each node; if it is less than 80%, this indicates that congestion may occurred on that node. To avoid routes with congested nodes, we will compute a fitness value and other solution for each node, which will be taken as input to Tabu search optimization. The fitness value will be determined by three factors:

1. **Packet delivery rate:**If a knot has a greater delivery rate and drops fewer packets, the node will be less congested.

2. **Number of neighbours:** The number of neighbours is the second parameter. If a node has more neighbours (dense network) within a 250-meter radius, it is more likely to receive data from more neighbours, causing congestion.

3. **Bandwidth of the node:**The node with the most available bandwidth is likely to be less crowded.

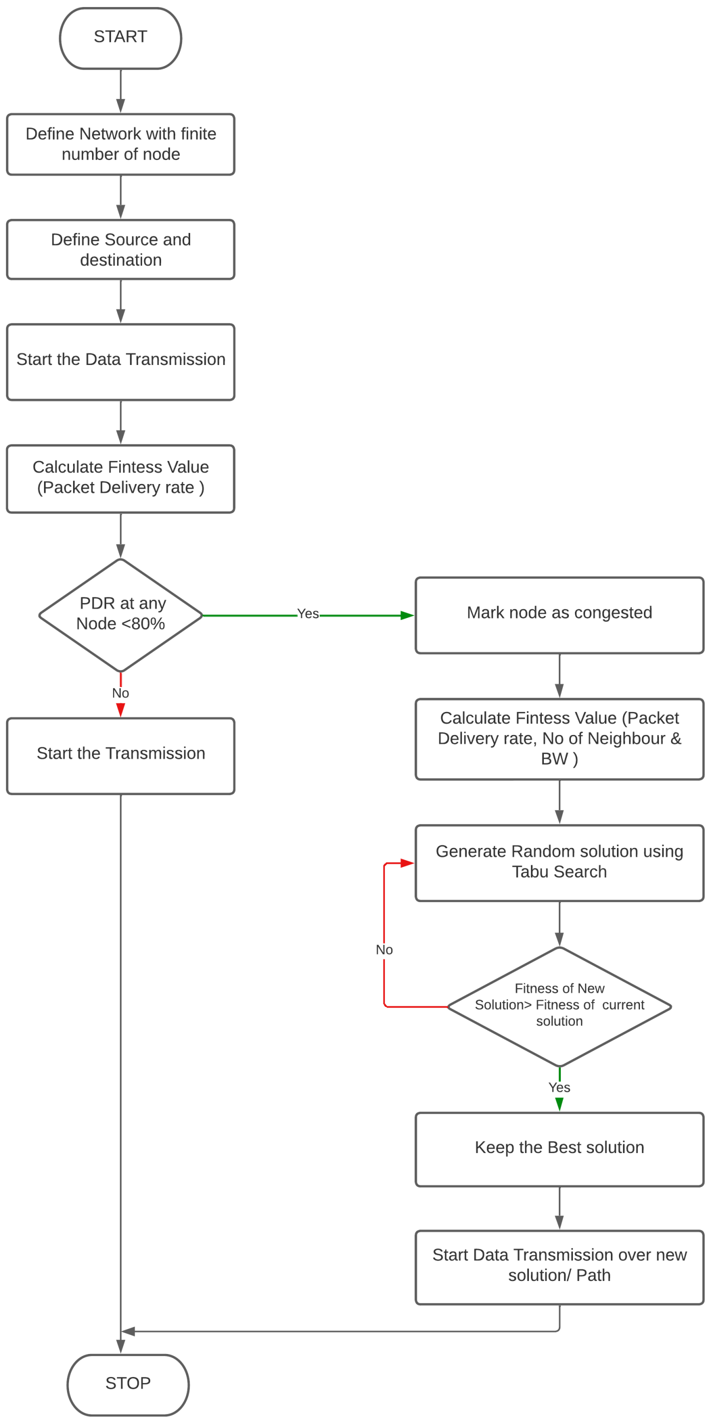

Figure 4: Flowchart of proposed Tabu search Algorithm

Once we compute the fitness value of the nodes, the corresponding fitness value of the paths will be removed. After this Tabu search optimization will be applied, which will choose the path having nodes with higher fitness values with respect to intermediate, future and short-term memory. Over the chosen optimal path, the data transmission will be done. Continuing the Data transmission process in case of PDR is greater than 80% means no congestion.

5 Implementation

This section describes how to implement the Tabu search and the AODV algorithm to reduce network congestion in edge computing. These tests are carried out with the help of the well-known network simulator 2 (NS2). We used the NAM GUI, which supports Tcl and C++, to build the network topology. The NAM uses the proposed algorithm to pick the best path, and network traffic is monitored. This section describes the implementation paradigm as well as the tools used to create the implementation prototype. Also, we used awk language as input as Ns2.

5.1 AWK Script

Awk is a general-purpose programming language. Text-based data may be processed with AWK in both files and data streams. The writers' last names, Alfred Aho, Peter Weinberger, and Brian Kernighan are the source of the acronym AWK.(Vishwakarma n.d.)

1.The majority of AWK implementations employ advanced regular expressions by default.

2.AWK scans the input file and performs the command(s) provided in action when it encounters a line that fits the pattern.

3. The trace file is interpreted using the awk or Perl scripting languages, which extract data from the trace file trace. tr.

4. Create the awk script, which is equivalent to C, and save it as *****.awk.

After running the awk file on the network, the user can record the outcomes for various network scenarios, and the output is subsequently translated to a graphical representation using XGRAPH.

5.2 Network Simulator 2

NS is best used to simulate local and large area networks. NS2 stands for Network Simulator Version 2. This event-driven, open-source simulator was created as a result of a focus on computer communication network research. The plumbing packages are combined with an incident generator and packages for network object element duties carried out in the back-end via the OTCL (Object-oriented TCL), a scripting language used by NS2. It is used to manage and monitor the routing algorithm's behaviour in real-time. The method for putting the simulation into action is fragmented into several parts, which are explained below:

1. **Topology:** NS 2 contains a framework of containers and helpers that streamlines the process of creating basic facilities and defining their interdependencies.

2. **The creation of models:** In most cases, helpers are used to add models to simulations (such as IPV4, UDP, IPv4,connections,and point-to-point devices).

3. **Configuration of nodes and links:** Models typically establish their default configuration using the attribute system, such as the size of packets transmitted by an application or the MTU of a point-to-point link.

4. **Implementation:** Simulation facilities produce incidents, and the user-requested data is logged.

5. **Performance evaluation:** When the simulation is complete, and the data is provided as an event trace with a time stamp.(Chung & Claypool 2019)

The Fig.5 outlines the main steps for running a network simulation with NS-2. Trace files and other output files need to be analyzed in order to get useful findings. This initiates the event scheduler's initialization. Secondly, by utilizing the plumbing modules and the network objects in the package, the user helps configure the network architecture, providing the program agents with the knowledge they need to start and terminate packet forwarding. The NAM window has been displayed to demonstrate the simulation options following the execution of the ns command.. On a virtual computer made in VMware and running the Ubuntu 16 operating system, the simulator has been set up. The topology of the network primarily consists of 100 wireless nodes, with one node serving as the source and another as the destination.

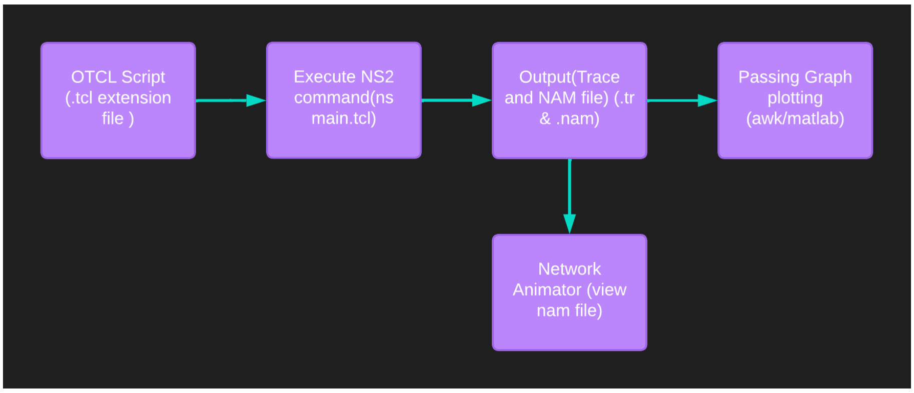

Figure 5: NS2 Simulator workflow

5.3 NAM package

NAM is Network animator. output of a NAM file NAM is a Tcl/TK based visualization application for examining both real-world and network simulation event traces. NAM was among the first programs to offer basic purpose, packet-level, and connectivity visualization, using input from network simulators (like ns) or real systems. A trace file must be created before using NAM. NS often produces this trace file. The trace file may be animated using NAM once it has been created.

5.4 Creating topology for proposed research in NS2

In the proposed research, a topology of 100 nodes is designed, and at the start, the source (sender) and destination (receiver) are declared. In our topology, node o is the source, and node 99 is the destination. Multiple paths are used by Node 0 to begin transmission toward Node 99, and the AODV algorithm is used to choose the best path. Node in Fig6 with CYAN colour represents best path. Using the NS2 simulator and NAM GUI, the proposed study is fully simulated.

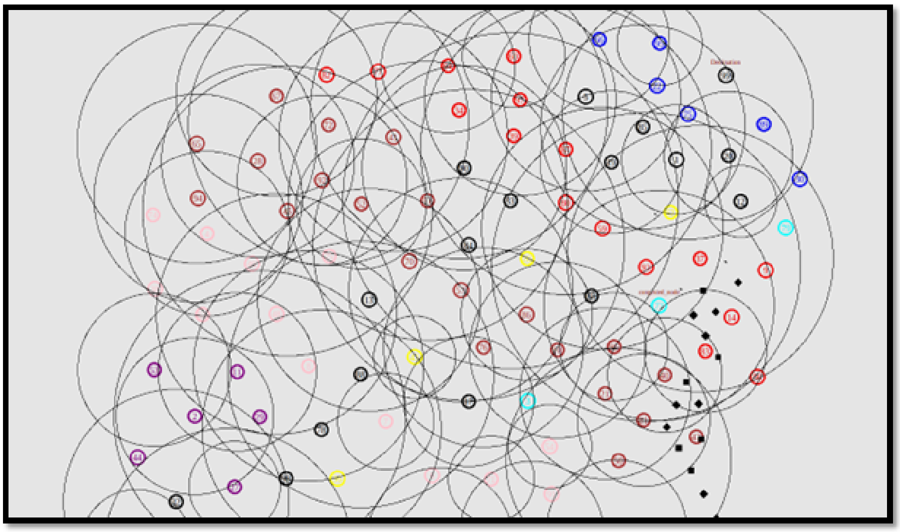

Figure 6: Topology of proposed research in NS2

5.5 Evaluating fintess with PDR parameter

The goal of the study is to mitigate congestion; hence, the first stage of the proposed approach is to identify congestion using the metric packet delivery rate (PDR). PDR can be determined using the formula shown in fig7. This is stored in an "awk file," and this "awk file" is called in "main.tcl" to compute the PDR. "check_congestion.tcl" will be used to create the annotation for this file. The simulation's output will be printed in text format on the "congested_node" file. If the PDR number is less than 80%, the node is declared congested.

```
for(z=2;z<7;z++)
{

pks_sent=0
pks_rec=0
pks_drp=0

for(x=0;x<i;x++)
{
if(p[0,z]==src_sent[x] && tim[x]>=28.9 && status[x]=="+" && agent[x]=="AGT")
{
pks_sent=pks_sent+1
}

if(p[0,z]==dst_rec[x] && tim[x]>=28.9  && status[x]=="r" && agent[x]=="AGT")
{
pks_rec=pks_rec+1
}

if(p[0,z]==src_sent[x] && tim[x]>=28.9  && status[x]=="d" && agent[x]=="IFQ")
{
pks_drp=pks_drp+1
}
}


pdr = (pks_rec) / (pks_rec + pks_drp)

print p[0,z]" " pdr > "pdr_node"

if(pdr < 0.8)
{
print ("$ns_ at "tm" \"$ns_ trace-annotate \\\"Pdr of node "p[0,z]" is found to be "pdr" wh
print "$ns_ at "tm" \"$node_("p[0,z]") label congested_node\"" > "check_congestion.tcl"
print p[0,z] > "congested_node"
tm=tm+0.1
}
```

Figure 7: Best path function

5.6 Checking for the best solution using Tabu search

To specify the proposed Tabu search algorithm, we developed the "tabu search.awk" file, which is shown in the Fig.8. This function provides a brief overview of how to determine the fitness value of the existing and new solutions. If fitness_new is superior to current_fitness, then the new solution is the best_solution (alternate path). This algorithm's result is printed on "alternate_solution."

```
current_solution = stm[s-1]

new_solution = int(rand()*i)

flag=0

for(y=0;y<s;y++)
{
if(new_solution == stm[s])
{
flag=1
}
}

if(flag==0)
{

fitness_new = p[new_solution,8]
fitness_current = p[current_solution,8]

print fitness_new" "fitness_current > "test"

if( fitness_new >  fitness_current)
{
best_solution = new_solution
best_fitness = fitness_new

stm[s] = new_solution
s++
}
}

}

print best_fitness" "best_solution > "alternate_solution"
```

Figure 8: Best path solution

6 Evaluation

The next section discusses the simulation and results aspect of the suggested idea. The primary goal here is to mitigate congestion in edge computing with the best solution. There is also a comparison with Tabu search algorithms and EECCA, OLIA, MPTCP(Ma et al. 2022). The suggested algorithm's performance is evaluated using throughput, energy efficiency, energy consumption, etc. The proposed "Tabu search algorithm for congestion control in wireless sensor networks" is tested with 100 nodes.
Fitness Parameter Settings: The packet delivery rate, bandwidth, and neighbour node are analyzed and simulated in 20 iterations. The main measurements of the simulation experiments are as follows:

- **Packet delivery Rate:** The terminology "packet delivery rate" refers to the ratio of packets received to packets received and dropped. This standard illustrates how well the routing protocol works between a source and a destination. The probability of congestion is reduced with a higher data packet delivery rate.

- **Bandwidth:** It is the total of packets received and packets dropped. The most data that can be sent across a given route in a specific amount of time. Higher bandwidth rates are preferable since they reduce the possibility of congestion.

- **Number of neighbour nodes:** This parameter will be calculated using the distance between two nodes. If the distance between two nodes is less than 250 meters, the area is more densely packed with network nodes.

We have set the values of PDR, Number of nodes, and Bandwidth as 40%,20%, and 40% respectively.

The connection becomes temporarily overcrowded due to network congestion. The network performance will be significantly impacted in this situation, and the relevant network metrics will also fluctuate. Link latency rises, data packets are frequently dropped, and network performance significantly decreases when there is network congestion. The predicted value at this period will be lower than the value under typical circumstances. We have set up a wireless network for WSN.

6.1 Results

After running the simulation, NS2 generates the result and stores it in a text file. As the graphical presentation in NS2 is not graphically attractive, we have extracted those results and created bar charts for comparison using Microsoft Excel. The results obtained are evaluated by examining the throughput, energy efficiency and energy consumption performance of the Tabu Search algorithm and the benchmarking mechanism to confirm the effectiveness of the method.

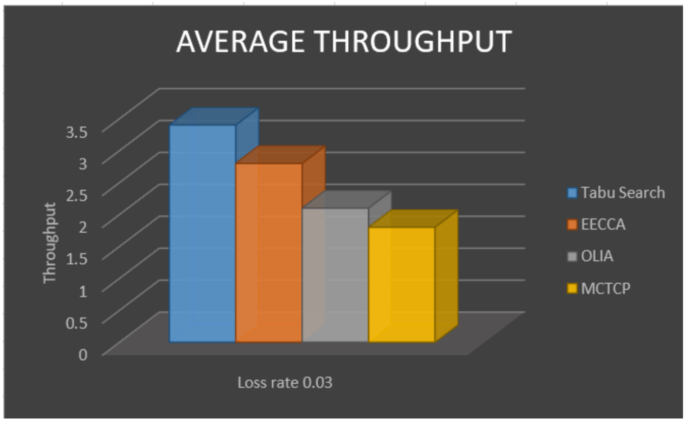

Figure 9: Comparison of average throughput

The throughput performance comparison is shown in Fig.9. The figure shows that Tabu Search has a throughput that is relatively high than EECCA, OLIA, and MPTCP. The expansion of the congestion window and the establishment of the slow start threshold have adaptive qualities due to the algorithm's use of packet delivery rate to monitor and control the state of the network, which raises the PDR of the system.

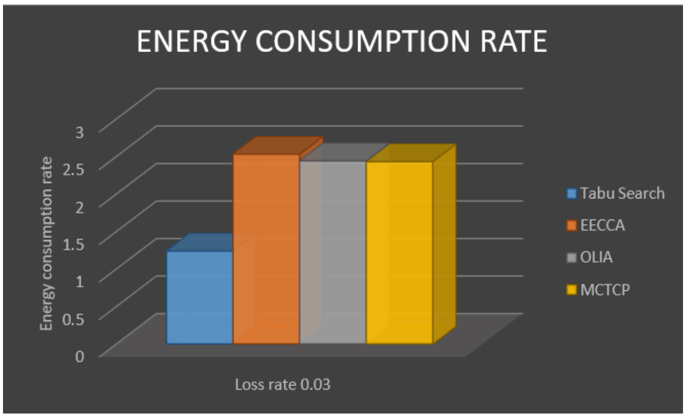

Figure 10: Comparison of energy consumption

In the experiment scenario, Tabu search's average energy consumption rate is compared to that of every other reference protocol in Fig.10. The graph shows that the Tabu search uses less energy than the three benchmark methods. Energy consumption rises sharply as the number of nodes rises, yet it stays relatively low when compared to earlier methods. To avoid the node from undergoing network failure, this energy consumption should be reduced as much as possible. This decrease in energy use would be linked to Tabu search's better throughput gain.

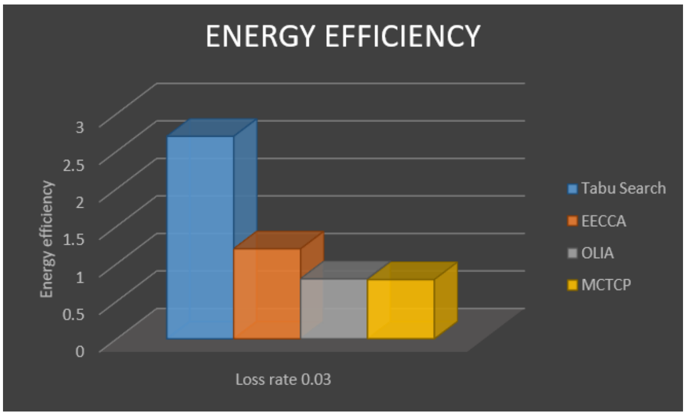

Figure 11: Comparison of energy efficiency

The comparison of the four algorithms' energy efficiency is shown in Fig.11 Energy efficiency measures how efficiently an algorithm can transmit data while consuming the same amount of energy. It is calculated as the throughput-to-energy consumption rate ratio. The chart shows that Tabu Search has a greater energy efficiency than the other reference solution and can transfer more data while using the same amount of energy, which aids in achieving green computing. Optimizing energy efficiency at the source node can be one of the most crucial parts of congestion control in WSNs. The source node uses the most energy at different times while the network is active. The node should have excellent energy efficiency to prevent any network outages.

6.2 Discussion

According to the findings of the previously discussed experiment, the Tabu search algorithm for edge computing is an efficient network congestion control method based on fitness. This approach implements effective node disjoint and the removal of unfit solutions, which increases system throughput while utilizing a minimal amount of energy. In addition to this, the AODV method is also employed for optimization problems. The next step is to find the congested node using PDR as the fitness function. According to the simulation findings, the method not only has a better throughput increase than the reference scheme, but also significantly improved energy efficiency, which solves the edge computing's energy constraint issue. Additionally, results can be compared by employing multiple network types, such as 5G, LTE, and Wi-Fi, which may lead to the discovery of additional parameters that can be compared to the existing approach. Unfortunately, the main issue in this computer era security, was not considered while this research was being planned. As a result, this system may be vulnerable to cyberattacks.

7 Conclusion and Future Work

Wireless sensor networks (WSNs) are now widely used in edge computing. Congestion is a key problem in the WSN that is quite often addressed. When a mote receives more data than it can handle, it must buffer redundant or duplicate data. Congestion happens when the packet is received, which increases the risk of packet loss. Packet losses cause the sensor nodes' communication and energy resources to be consumed, which lowers throughput.

The effectiveness of the presented approach for managing congestion, achieving high throughput, and meeting high QoS standards has been established. Our research demonstrates that while effectively reducing congestion the suggested methodology increases throughput and energy efficiency by reducing energy usage when compared to the existing algorithm. Future work will include the development of a Tabu search method for better message scheduling in control and service channels based on present message priority. However, in our research, we employed a single path congestion control technique. To increase the accuracy of congestion control, we can use a multipath routing mechanism. Because all of these devices are wireless (WSN), they are often vulnerable to various sorts of cyber-attacks; nevertheless, we may apply proactive protocols to control such security breaches. work.

References

Alfakih, T., Hassan, M. M., Gumaei, A., Savaglio, C. & Fortino, G. (2020), 'Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on sarsa', *IEEE Access* **8**, 54074–54084.

Baranwal, G., Kumar, D. & Vidyarthi, D. P. (2022), 'Bara: A blockchain-aided auction-based resource allocation in edge computing enabled industrial internet of things', *Future Generation Computer Systems* .

Chung, J. & Claypool, M. (2019).
URL: *http://nile.wpi.edu/NS/*

Glover, F. & Martí, R. (2018), 'Chapter 4 tabu search - uv'.
URL: *https://www.uv.es/ rmarti/paper/docs/ts2.pdf*

Guo, K., Yang, M., Zhang, Y. & Jia, X. (2019), 'Efficient resource assignment in mobile edge computing: A dynamic congestion-aware offloading approach', *Journal of Network and Computer Applications* **134**, 40–51.

Hull, B., Jamieson, K. & Balakrishnan, H. (2004), Mitigating congestion in wireless sensor networks, *in* 'Proceedings of the 2nd international conference on Embedded networked sensor systems', pp. 134–147.

Ishaq, M., Malik, M. H. & Aydin, M. E. (2018), Managing congestion in vehicular networks using tabu search, *in* 'International Conference on Engineering Applications of Neural Networks', Springer, pp. 118–129.

Jang, K.-W. (2012), 'A tabu search algorithm for routing optimization in mobile ad-hoc networks', *Telecommunication Systems* **51**(2), 177–191.

Kumar, A., Srinivas, P. & Govardhan, A. (2020), 'A multipath packet scheduling approach based on buffer acknowledgement for congestion control', *Procedia Computer Science* **171**, 2137–2146.

Li, S., Wang, Q., Wang, Y., Tan, D. & Li, W. (2019), Delay-aware task congestion control and resource allocation in mobile edge computing, *in* '2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)', IEEE, pp. 1–6.

Lieira, D. D., Quessada, M. S., Cristiani, A. L., Immich, R. & Meneguette, R. I. (2021), Triad: Whale optimization algorithm for 5g-iot resource allocation decision in edge computing, *in* '2021 16th Iberian Conference on Information Systems and Technologies (CISTI)', IEEE, pp. 1–6.

Ma, Y., Tian, H., Liao, X., Zhang, J., Wang, W., Chen, K. & Jin, X. (2022), Multi-objective congestion control, *in* 'Proceedings of the Seventeenth European Conference on Computer Systems', pp. 218–235.

Mahenge, M. P. J., Li, C. & Sanga, C. A. (2022), 'Energy-efficient task offloading strategy in mobile edge computing for resource-intensive mobile applications', *Digital Communications and Networks* .

Qin, J., Xing, Y., Wei, W. & Xue, K. (2020), Edge computing aided congestion control using neuro-dynamic programming in ndn, *in* 'GLOBECOM 2020-2020 IEEE Global Communications Conference', IEEE, pp. 1–6.

Riaz, M. K., Yangyu, F. & Akhtar, I. (2019), Energy aware path selection based efficient aodv for manets, *in* '2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)', IEEE, pp. 1040–1045.

Sharma, S. & Panjeta, M. (2022), Optimization transmit rate-based decentralized congestion control scheme in vehicular ad hoc networks, *in* 'AIP Conference Proceedings', Vol. 2555, AIP Publishing LLC, p. 030006.

Vishwakarma, M. D. D. (n.d.), 'Method of performance evaluation of wanet using ns-2'.

Xiao, X., Zhao, M. & Zhu, Y. (2022), 'Multi-stage resource-aware congestion control algorithm in edge computing environment', *Energy Reports* **8**, 6321–6331.

Xie, R., Jia, X. & Wu, K. (2019), 'Adaptive online decision method for initial congestion window in 5g mobile edge computing using deep reinforcement learning', *IEEE Journal on Selected Areas in Communications* **38**(2), 389–403.

Yang, J., Wang, Y. & Li, Z. (2022), 'Inverse order based optimization method for task offloading and resource allocation in mobile edge computing', *Applied Soft Computing* **116**, 108361.

Zhang, J., Yang, T., Ji, H. & Li, W. (2022), 'Optimal locating method of edge computing device in cyber physical distribution system', *Energy Reports* **8**, 684–694.

Zhang, S., Yi, N. & Ma, Y. (2021), Correlation-based device energy-efficient dynamic multi-task offloading for mobile edge computing, *in* '2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)', IEEE, pp. 1–5.

Zhang, Y., Di, B., Wang, P., Lin, J. & Song, L. (2020), 'Hetmec: Heterogeneous multi-layer mobile edge computing in the 6 g era', *IEEE Transactions on Vehicular Technology* **69**(4), 4388–4400.

Zhang, Y., Zhou, Y., Lu, H. & Fujita, H. (2021), 'Cooperative multi-agent actor–critic control of traffic network flow based on edge computing', *Future Generation Computer Systems* **123**, 128–141.

Zhu, Y., Hu, Y. & Schmeink, A. (2019), Delay minimization offloading for interdependent tasks in energy-aware cooperative mec networks, *in* '2019 IEEE Wireless Communications and Networking Conference (WCNC)', IEEE, pp. 1–6.