Traffic optimization in Fog based systems using CPEP Routing Algorithm

MSc Research Project
MSc in Cloud Computing

Suhail Ahuja
Student ID:20191065

School of Computing
National College of Ireland

Supervisor: Dr. Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Suhail Ahuja		
Student ID:	x20191065		
Programme:	MSc in Cloud Computing	**Year:**	2022
Module:	MSc Research Project		
Supervisor:	Dr. Rashid Mijumbi		
Submission Due Date:	15/12/2022		
Project title:	Traffic optimization in Fog based systems using CPEP Routing Algorithm		
Word Count:	7088	**Page Count**	23

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: Suhail Ahuja

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):	

Traffic optimization in Fog based systems using CPEP Routing Algorithm

Suhail Ahuja
x20191065

Abstract - As the count of Internet of Things (IoT)-based application grows, chances of the network congestion also increase. Fog Computing helps to overcome the challenges associated with processing a huge volume of sensor data. To overcome the congestion, the data produced by the IOT sensors/applications is required to be routed in an appropriate and optimum manner, but doing this is a difficult task. In fog computing, the data is kept locally on the fog nodes instead of being sent to the cloud servers, as using the cloud servers increases latency time. By using fog computing (FC) and customized routing protocol (CPEP), this study aims to increase the data transfer (throughput) and reduce the congestion and latency on the fog network. To prevent congestion, efficient strategies should be used to transfer the data to the target node with the least amount of delay and overhead possible. Our method, which consists of four elements, is presented below. In order to ensure that only one type of data is provided to each server, we first made use of content awareness to classify the requests according to their data categories. Following that, a priority-based technique is utilized to classify the requests depending on their importance levels. The processing time is shortened by 70.3615% and also the throughput is increased by 374.485%. The ideal path is then determined taking into account the energy level of nodes.
Additionally, a route awareness technique is employed in the last section to further minimize response time, as a result, this facilitates the quick routing of data within IOT application services network.

1. Introduction

- By relocating the processing, storage, and networking operations to the network's edge, fog computing (FC), expands the potential of cloud computing. It is an additional technology that facilitates location-aware and latency-sensitive applications by delivering its applications & services closest to the end users. In order to decrease the load on the cloud server, fog servers, which are situated between the cloud server & end users, preprocess data before sending it to the server.

The expansion of the Internet of Things (IoT) has greatly increased the number of devices that are connected to the internet and as a result, the volume of data that moves across the Internet has also increased to a great extent. In the central cloud architecture, this data might overload the network, possibly resulting in latency- and bandwidth-related problems. Using Fog Computing the calculations of the IoT data may also be conducted locally and near to the data sources, this may prevent the build-up of data over one central place thereby minimizing connection latency, and therefore optimizes the bandwidth consumption.

By 2025, 41.6 billion IoT devices are expected to be in operation, with the potential to produce up to 79.4 ZB of data, as per analysis by International Data Corporation (IDC). As per Rydning et al. (2018), The Global Datasphere is expected to grow to a size of 175 ZB by 2025, according to IDC's forecast. According to the report, by 2025, every connected person will interact with the data in approximately every 18 seconds.

The conventional IoT network seems unable to effectively manage the enormous number of IOT systems that are being connected to the Internet daily. Most Internet of Things (IOT) applications rely on moving task computation & storage to far-off cloud computing (CC) data centres and the servers in order to increase the processing speed, reduce the cost, and to save the energy. Unfortunately, such remote servers are physically positioned far from the IOT devices and the end user, causing significant latency & congestion on communication channels. As more devices are connected to the network, topological changes could take place as data is transferred from IoT devices to the related fog servers. Additionally, the use of centrally managed provider or cloud centric cannot provide enough connection and computing close to network's edges, and as a result, becomes ineffective or unproductive in highly scattered situations. The massive increase in cloud-based data centre traffic poses several difficulties for successfully using cloud services, including intense communication, unanticipated delays, and insufficient support for end-user connection. Because of this, Cisco created the advanced computing concept "fog computing" to solve these problems. Incorporating various computing resources, fog services are comparable to a lighter-weight version of cloud services. Instead of replacing cloud computing, it might be seen as an expansion or enhancement of it. It aims to reduce network congestion problems, eliminate the latencies, and better support the flexibility or mobility of the end device. Additionally, fog computing contributes to improving bandwidth consumption since computation for data gathered from the IOT may be done close to the data source. But it may be challenging to handle the variety of data that wireless IoT devices create. So, for the same purpose to reduce the latency and to increase the throughput, a new routing algorithm is proposed.

1.1 Research Question

"Can CPEP Routing Algorithm optimize the routing efficiency?"

2. Related Work

Our research concentrates on content-based, priority-enabled, context-aware transmission over the optimal route with the best energy for FC-based IoT networks. In fog computing, efficient allocation of resources, traffic control, and QoS maintenance are essential. In order to effectively manage the traffic & routing among the fog network while reducing latency and improving overall transmission delay, a variety of tactics and research techniques were discovered traditionally and are covered in this part. The importance of effective data forwarding, traffic & resource management has been shown in the literature review. Also, as per Alomari et al. (2021), Alternative flow rules and queue structures enhance efficiency by allocating more resources to the data with the highest priority.

We can see below in the past research papers; researchers have already used a variety of unique tactics to optimize the traffic over the fog and edge network. The several strategies used are described here in this section. The subsections will thoroughly examine the methods used in the process of optimizing traffic control in the fog-based environment, including different types of routing algorithms and other procedures.

2.1. Priority and content-based traffic optimization - Prioritizing the flow of network and packets may help to resolve many network performance problems. One of the most well-known and efficient strategies for prioritizing is the use of SDN. Many traffic management issues may be resolved with this strategy if priority & content-based forwarding is used in an intelligent and responsible manner.

To show this, Shinkuma et al. (2020) extremely informative study, provides us with a very substantial source of information for our planned research. Shinkuma et al. used machine learning approaches to optimize real-time data in networks made possible by SDN, and they also put forward a framework for controlling flows in the SDN-based edge cloud system that makes predictions using the data gathered from multiple IoT devices. By classifying the data according to their importance, their system utilizes the predictions provided in real-time extremely accurately. The system was evaluated in a simulated environment with the mobile traffic using a dataset with actual measurements and several network topologies. A real-time predictions-based strategy is utilized to reduce latency and optimize traffic. In this case, edge-cloud servers handle the data processing based on priority.

The research project suggested by Diro et al. (2018) takes a genuinely unique method and delivers priority-based service quality in networks incorporated with SDN, In this work, many flow channels for IoT data are designed, each with a different degree of priority. In this a flowchart is determined using the prioritization of data and using that flowchart data is sent. Priority data is sent right away, whereas non-priority data is kept in a separate queue. In terms of flow space utilization and throughput, their suggested approach beats the standard OpenFlow system. They provide a plan for future SDN/OpenFlow improvements in networks that are delay-sensitive and where QoS is important, and along with this a deeper knowledge of software defined networking / flow space restrictions. This paradigm relies on the transmission of urgent (prioritized) packets to increase throughput and reduce loss rate.

In situation of emergency to make the optimized utilization of the available network resources, Lu et al.[9] proposed a most suited model which uses SDN. The main objective of this model was to provide the quality of service in times of emergency. This model basically divides the network into several flow pathways in accordance with the importance and fundamental needs of the tasks.

2.2 Using context awareness to improve traffic –

In this, we make use of the data context to modify the routing paths and transport the data directly to the saved compute nodes there, hence lowering the overall route's node count and also the number of iterations to find the best path for the transmission.

As part of fog computing systems, many techniques have been proposed that allows the compute nodes to either receive and process IoT data or broadcast it to other compute nodes. A context-aware routing method decreases communication time and boosts bandwidth use in comparison to a regular routing approach.

For context-aware data transfer, Wiener et al. (2019) presented a conceptual model for fog computing systems. As the context for IoT data changes, the authors of this study advise adjusting the applications accordingly.

Zhao et al. (2014) also proposed a routing protocol which relies on the context awareness and In order to create routing decision, it first utilizes a variety of the context information like the quality of the link, energy that a node possesses. Second, it permits all suitable nodes to take part in packet forwarding. Third, it includes relative node mobility to boost performance even further.

2.3 Determining the most efficient path to reduce traffic-

Routing time delay, transmission overhead, and throughput are the most important factors to be considered while routing IoT data. For routing of the traffic in IOT systems, there are several methods already available. Since the sensor data produced by IoT devices is highly sensitive to the delays, & is required to be routed in a quick and intelligent manner. The routing techniques play a crucial role in routing optimization or controlling the IoT traffic.

2.3.1 Routing protocols for the Internet of Things –

To reduce the overall energy, Chelloug et al. (2015) devised an approach that develops a topology that can be used centrally in order to provide the route to the analytical data gathered by numerous sensor devices to the destination nodes linked to the Internet. The main goal was minimizing the energy being consumed.

A significant protocol published by da Silva Fre et al. (2015) focuses on the traffic reduction by the aggregation of the data according to the content with routing the traffic through highly secured lines by providing the information about connection quality. The primary objective is to route the variety of data via well selected communication routes to nodes that have the capability of processing and retrieving accurate data. This lessens the volume of redundant traffic communication. Additionally, it lessens the frequency of transmission of the packets to and from the IOT sensor devices.

Christy et al.(2017) also proposed an energy efficient routing method for the wireless networks. In this, nodes periodically use the energy and also energy conservation is a main and crucial consideration while building this routing protocol. In this, the nodes that are not involved in data transmission are deactivated, or placed into sleep mode.

Tariq et al (2009) presented a routing scheme for the wireless network with sensors. In this routing scheme basically to make the routing energy efficient interaction between the MAC layers & routing is created and also in this rather than searching the node in the whole network the algorithm selects the best next node in neighbour of the current node by making utilizing the radio information at the MAC layer.

A - Protocols that utilize Node location - The protocols that examine the node's location rather than the routing tables often examine the node's location to identify the appropriate routing path. This helps to restrict the transmission range of the nodes, and it then lengthens the life of the network.

By improving the energy-detecting technique, Ukani and Thacker (2015) provided an enormously important study for the transmitting multimedia data among wireless networks that contains sensors. The proposed protocol seems to be faster when there exists any void in the network. The author placed a strong emphasis on QoS in order to collect information about the neighbour hops that suited the requirements of network context like audio and video traffic. The simulation's findings demonstrated that the protocol could choose a routing path more skilfully than greedy forwarding.

In this paper, author mentioned that the protocol was tested in numerous topologies and with voids in the network. This study demonstrates an increase in the proportion of delivered packages. However, other elements must be considered during the simulation in order to get valid results.

Also, to minimize the end-to-end latency in the data packet transmission, Karim and Nasser (2012) proposed a routing protocol which uses the location parameters to efficiently route the data packets on the wireless sensor network. In this protocol, the author also used an excellent fault tolerance mechanism which can easily detect the faults in the network even while routing the data packets.

B - Protocols that utilize data - Routing strategies based on the data are generally considered for problems with data fusion. In this, the nodes collaboration is created with one another to boost the quality of the transmission and reduce network energy consumption.

A routing technique was described by Samaras and Triantari (2016) that enables sink and the source nodes in the mesh topology network to connect and communicate. This routing protocol also enables the intermediary nodes to collect the data and send it to a sink node.

The previous routing protocols were not analyzing the parameters like the type of the service, priority, energy. Also, were not having the sequential flow and backup systems for facilitating the data packets. Because of this reason, they were having less throughput and high overall latency. Also, the drop of packets was more. However, the proposed routing algorithm is capable to overcome the previous issues and also increase the packet delivery rate and the count. Also, the overall latency is decreased to the minimum and congestion is improved.

3. Research Methodology - An overview of the methods, procedures, and algorithm that have been used to carry out the research are –

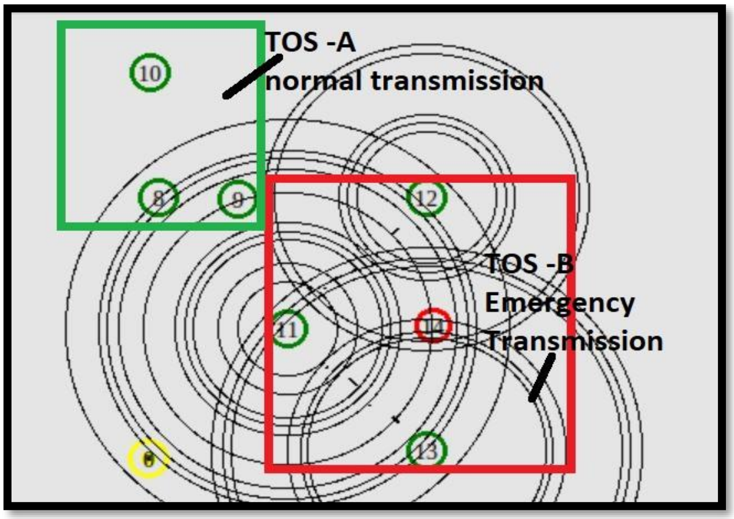

Fig.1-Topology

3.1 Topology - There are two sink nodes & 8 wireless nodes in our proposed network architecture. The network is made up of independent nodes and the network connections connecting these nodes. To simulate the implementation, Network Simulator 2 has been used to develop the wireless topology. For the simulation, we set up an Oracle VirtualBox and then created a virtual machine. On that virtual machine, we installed Ubuntu 18.04 version.

After that, to configure the NS2 on the Ubuntu we used several commands like – We first update the apt package in the Ubuntu and then installed NS2 from the apt package. Now after installing the NS2, we have installed the NAM which stands for Network Animator. This is basically used in the implementation to visualize the traces of the data packet as well as the traces of simulation of the network. Using this NAM we achieved the actual topology layout, animations at the packet level as well as using this we are able to examine the results. After making/ setting up the configuration part of NS2 on Ubuntu.

We divided the whole protocol into 4 parts, and so we are able to create and implement the CPEP routing protocol. There are two groups of the nodes specifically – one with the nodes which are emergency nodes means the data packet transmitted needs to be serviced first with the highest priority and in the minimum possible time and the other – a group of nodes which have normal or less priority. The device layer has the wireless sensors and also these sensor devices have a RFID tag and simultaneously the fog layer has the fog devices or the servers which reads and processes the data. The source devices are represented by the nodes -8,9,12 and the servers are represented by 10 and 13 nodes. Firstly, the data is categorized and a particular type of data is processed by a particular server. Then priority is analyzed and the nodes with high priority are allowed to send their data first so that their packets can be served before the packets with the less priority. After this the energy of the nodes is analyzed and also for the better visualization of the energy levels three colors of the nodes have been used – Red, Yellow and green. These nodes are initialized with some starting energy. The node with the highest energy will have the green color and as the energy of the node gets used, then the color of the node changes to Yellow and then in the end when the energy is decreased below a certain low level or to the minimum lowest levels then these nodes are turned to red color. This energy analyzation of the nodes is done so that the data packet can be transmitted through the nodes that possesses higher energy levels. In the last, the path is saved, which means the protocol tries to follow the same path to avoid the latencies and get the data packet facilitated quickly. In comparison to traditional routing techniques, this topology improves traffic congestion and also minimizes total routing latency in the IoT sensor networks with the emergency and non - emergency nodes.

3.2 CPEP Algorithm – Here in this section the proposed algorithm is explained in detail and for the same purpose is categorized into smaller parts to showcase the clear and easy understanding. Also, these parts are used to create the whole routing algorithm.

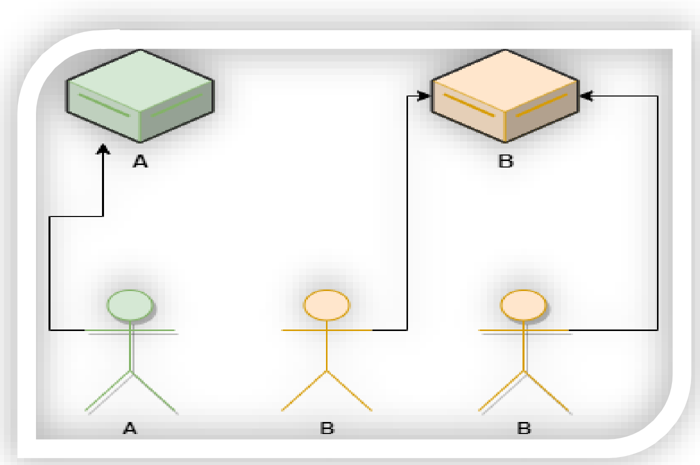

Figure 2 – Content Type Awareness (Part 1)

Part 1- Here as can be seen in the above image the whole transmission is categorized into smaller sub parts on the basis of the type of service. If the request or the data packet is of A or non - emergency type then it will be serviced by A type of server and in the same way if it is of B or is having some emergency data type then it will go to the B type server. Doing this will decrease the congestion on the network and also will enable the emergency packets to be facilitated first without any delay or the need to wait in queue.

In the above image, B is marked red to represent the emergency type of service and A in green to represent the non-emergency data type requests. So, with the help of this concept, the emergency data type can be served quickly on the special dedicated server.

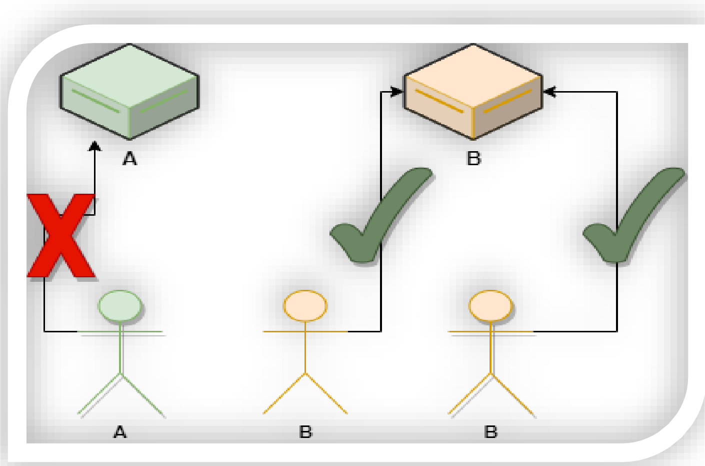

Figure 3. Prioritization (Part 2)

Part 2 – Priority – In this part we have divided the whole transmission on the basis of the priority and implemented the sequential transmission of the data packets to further enhance the throughput, minimize the packet loss rate, to reduce the latency to the lowest possible, and also it helps to avoid any chance of transmission/service failure. In the above image we can see that while the emergency nodes are transmitting the data then the other group will stop the transmission for that moment or will slow down and will be kept as the backup just in case the B server fails then A will be used and the B data packets will be serviced quickly by the A server. So, in this way by implementing the concept of priority queue, sequential transmission and backup. The transmission over the network becomes highly optimized and efficient. In this way, the loss of the data packet rate is highly reduced.

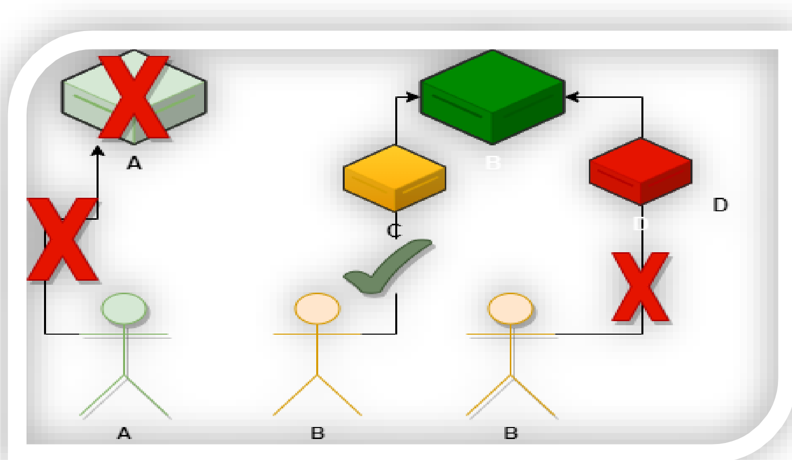

Figure 4. Energy Awareness (Part 3)

Part 3 – Energy Level Detection - After the detection of the type of service and priority, then comes the energy detection and selection of the best path. Here, the energy associated with the nodes is analyzed. This is done to transmit the data packets through the best optimal path via nodes which have higher levels of residual energy. In the above image this is depicted by using the green, yellow and red color nodes. The nodes with the highest energy level are in the green color and the nodes with the energy level between highest and the lowest are in the yellow color and the node with the least energy are represented by the red color. Initially all the nodes possess in the image we can see that since the node D energy is deprecated to the lowest levels then the route from B – C- B is opted. This strategy helps to transmit the packets thorough the best path with the nodes with the higher energy levels. Also, this helps to increase the throughput and reduce the packet drop. This also helps to reduce the overall latency.

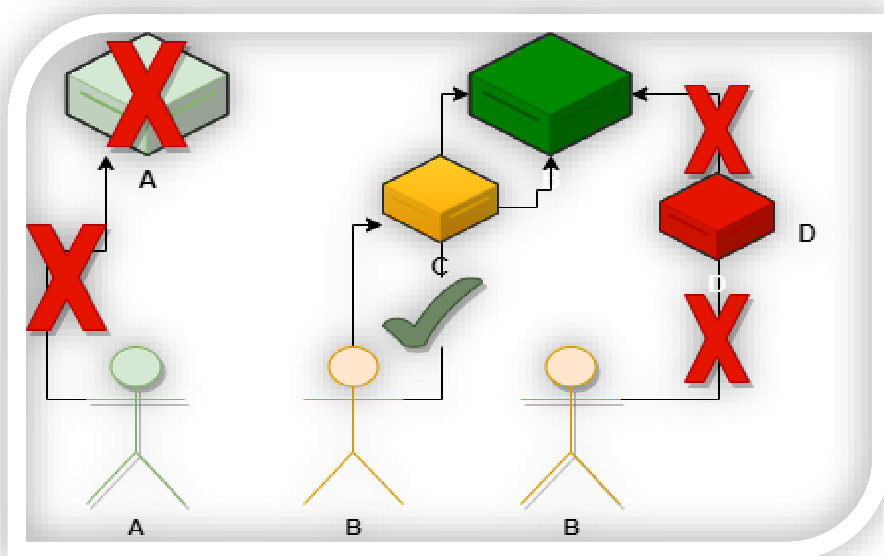

Figure 5. Context saving and awareness

Part 4 – Context or Path saving and awareness - This is the last step of our routing algorithm and in this the paths used in the transmission of the data packets are saved and among all that saved paths, the best path is opted. This helps to save the time to analyse the nodes and the paths again and again. This lowers the transmission delay and increases the throughput. So these were the 4 parts of our proposed routing protocol. Below is the pseudocode –

Algorithm –

```
Input:    Data packet p, packet time out ot, ToS(Type of the Service) , Network graph G,
          source S, energy e and the destination D.
Output:   Efficient management of the traffic
          Initialize the routes vector RV.
          Initialize the route R.
          Initialize the network vector NV.
          Initialize the nodes A, B, C
          Initialize the servers AS, BS, CS (general TOS server)
          While ot =/ 0
                   If TOS =A then
                             D =modify destination to A TOS server
                   elseif TOS=B then
                             D=modify destination to B TOS server
                   elseif TOS=C then
                             D=modify destination to C TOS server
                    else
                              D=modify destination to C TOS server
           for each node in NV do
                  Check the priority
                            If current node has highest priority data
                                      Start current node
                                      Stop other nodes
                            endif
            end for
          for each route a in RV do
          |                 If current route in BR
          |                         Use current route
          |                 endif
          |        Check the energy of each node in NV
          |
          |                if the current route has nodes with better energy than
          |                 previous route
          |                 then
          |                         Track the best possible route.
          |                          Use the best possible route.
          |                          Add best possible route to BR
          |                 endif
          end for
          =0
```

3.5 Pseudocode Explanation- This starts with the initialization of the network vector, nodes, server nodes, route and the route vector. In the first part of the algorithm, we are identifying the type of service and categorizing the whole transmission of the data packets on the basis of type of the service. For example, if there is A type of data then that means the TOS is A, so now its destination node will be the TOS server A. So, in the same way the whole transmission will take place. Now after that we have applied the for loop or iteration on the node vector (NV) to check for the priority. Now here if a node or group of nodes is having more priority, then that node will start transmission and the other nodes will stop or slow down so that the priority data could be facilitated first without any delay. Now after that we are checking the energy left or associated with each node and doing this the best path or route can be detected with the nodes which have significant level of energy left with them. In the end this path will be stored in the best route vector BR to implement the context awareness. Doing this will further reduce the latency and will help to minimize the congestion over the fog networks. The main goal is to get the best possible path from all the available paths in the network vector to route data packets through the path with the maximum energy and to reduce the latency or the response time.

4.Design Specification - The design process for our project is briefly described in this section, along with a list of the tasks being executed and tools being used.

The source layer consists of the IOT sensors that create the signals or the traffic to be transmitted. The sensors are categorized on the basis of the type of the service they require, like in the red color block nodes generate A type signals and green color block nodes generate B type of signals. So here since the first part of the whole research is the categorization on basis of the type of service, so we have categorized and divided them in particular groups and also, they have separately dedicated servers dedicated to the type of the request to be facilitated. In the Cloud architecture the main issue was the latency because in that architecture the request gets serviced on the cloud servers while in our proposed architecture the services are fulfilled in the fog layer which reduced the overall latency to the great extent and helps to improve the overall throughput. The algorithm is executed on the nodes to fetch the energy levels to select the best optimal path with the highest energy. The nodes with the high energy are set to be in the green color and their color gradually changes to yellow and red when their energy gets used up. So, these tasks are executed in sequence in our proposed algorithm for the efficient routing of the data packets.

5. Implementation

This section describes the implementation and also the resources used to create the implementation simulation.

5.1 Network Simulation - The approach to study or do research on the computer network is referred to as the network simulation. In this software application, the relationships between the different network components, such as connections among N-switched, routers and nodes are analyzed to determine how well a network performs. In an analysis lab, the performance of the network, various applications, services and supports can be evaluated. To assess how the proposed network or the protocols will function under different circumstances, other environmental factors could also be modified in a controlled manner using this.

5.2 Network Simulator - A network simulator is a software application that forecasts how well a computer network will run. In the simulator a computer network could be simulated by using connections or the links, devices & applications and network performance can be recorded. These are used when communication networks become too complex for fixed analytical methods to provide a clear insight of system performance

Different Network Simulations - There are both free sources and paid network simulators and network simulation tools-

Ns3, Netkit, Marionnet, JSIM (Java-based Simulation), OPNET, QualNet. For our research purpose we have used Network Simulator – version 2 (NS-2).

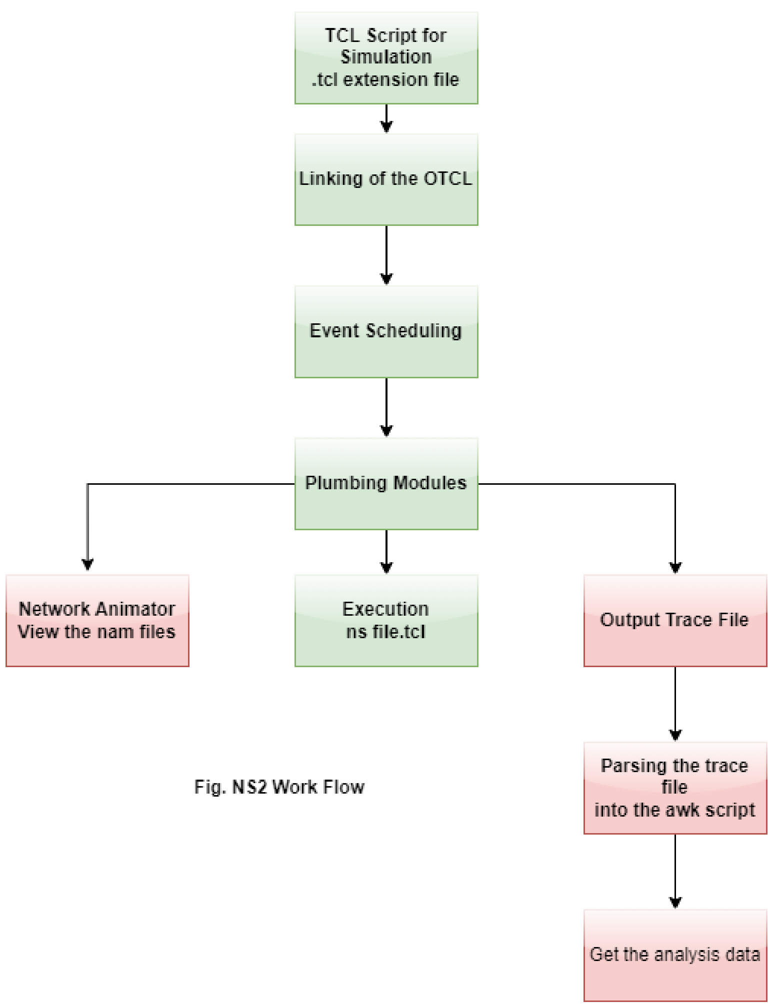

Fig. NS2 Work Flow

NS2- Network Simulator Version 2 is referred to as NS2. This open-source, event-driven simulator used generally for the computer - communication network research. It is a simulation tool for examining or analyzing the continually changing nature of communication networks. It provides a highly flexible framework for wireless & wired simulation that supports a range of network elements, protocols, routing types and traffic patterns. A handful of the network protocols supported by NS2 are TCP, UDP, DHCP and HTTP. These protocols can also be modelled using this tool. OTcl & C++ were used to develop the code for NS2. OTcl uses techniques from object-oriented programming. This was developed at MIT as object-oriented addition to the Tcl language (Tool - command language)

The user runs the OTCL instructions or scripts with the ns command during the NS2 simulation process. This starts the event - scheduler initialization, & by utilizing plumbing modules and also the network objects. It also helps to set up the network topology and notifies the application agents when to start and stop transmitting the packets. Additionally, this makes it simple to create a range of network traffic patterns, such as variable bit rate (VBR), available bit rate (ABR) & constant bit rate (CBR). So, Network Simulator 2 (NS2) provides thorough assistance for simulating numerous protocols across wired and wireless networks.

5.3 Proposed Routing Protocol Algorithm – Our algorithm is capable of providing the most appropriate or the best path as compared to all the paths available in the network for the data packets to reach from source to the destination and thus helps in the preventing the congestion over the network by minimizing the overall delay among the data packets transmission. Our algorithm is categorized into 4 sub parts which are as following –

i) TOS or Content check - Here TOS stands for the type of service according to which data packets, prior to being delivered to the network, are checked for the data type. In simple words, this means that a particular type of data will be facilitated by a particular type of servers and there will be specialized servers for the emergency data packets to be served quickly without getting being delayed. In the traditional structures of the network, the devices used to have/communicate using a few bits and there were very fewer data packets over the network but now since it has multiplied n times. This content awareness approach helps by reducing network congestion and by enabling quick transmission and execution of the data packets.

ii) Priority Check -Here the main goal is to assess the priority and make sure that crucial data is delivered promptly, considering the standards for data transmission rate and latency. Data categories that are greater than the specified threshold value or that are only important are considered as the priority data. These significant data packets are placed in a distinct queue and delivered over the network with a higher priority and more energy. In an emergency, this enables quick choices to be taken on the particular dedicated main server.

iii) Energy Check- In this part of the algorithm, basically we checked the energy of the nodes. The nodes with high energy will be denoted in the green color initially and their color changes with the changing energy levels. Node turns to yellow when the energy level is between the high and the low, and in the end turns to red if their energy level reaches the minimum or low level. All the nodes have their initial energy and as they start transmitting the data packets their energy starts decreasing as during the transmission of the data packets the energy get used and so after a certain interval of the time the nodes energy gets reduced to a minimal value. The main goal is to check and select the path in which the nodes possess the highest energy value. For Example- If a data packet has to reach from source A to Destination D and the network has two intermediate nodes B and C. So, in this case the path with the shortest distance and which has the highest energy nodes will be selected for the data packet's transmission.

iv)Context Awareness- The path for the transmission will be saved and will be known means the transmission of the data packets will try to select the same saved path been used previously. Using this the overall latency is minimized, and the IoT network congestion is also reduced.

6. Evaluation

Multiple experiments have been performed - first to analyze the outcomes of the pre-existing algorithm and then the results of the proposed algorithm were recorded to compare the proposed routing protocol with the pre-existing routing protocol. Here, this section describes the outcomes and analysis of the tests carried out utilizing NS2 in the both proposed and pre-existing routing protocols.

6.1 Analysis of pre-existing routing –

AODV protocol: - Firstly, the experiments were performed on the AODV to record the results and compare with the proposed routing algorithm. For performing this experiment, a total of 5 cycles or rounds were performed, initially the count of the nodes was taken as 10 and then in each of the subsequent iteration the number of the nodes was increased by 10. Means in the first iteration the count was 10 then in next iteration the count of the nodes was 20 and then so on till 50. Using the simulation trace file, the experiment's results were recorded for each of the iterations and then for the evaluation of the data, results related to the performance measures like the throughput and the delay time were gathered using the awk scripts.

6.1.1 AODV- Average end-to-end delay - The time difference between when the packets were transmitted by the sender and when they were received by the destination node is used to determine the average end-to-end latency.

Below is the result of each of the iteration in which initially the number of nodes was kept as 10, and then it was increased to 10 for each of the subsequent iterations. The delay time was calculated using an awk script.

AODV Routing Algorithm		Nodes				
		10	20	30	40	50
	Delay Time(m sec)	130.8	207.323	119.481	249.649	132.198

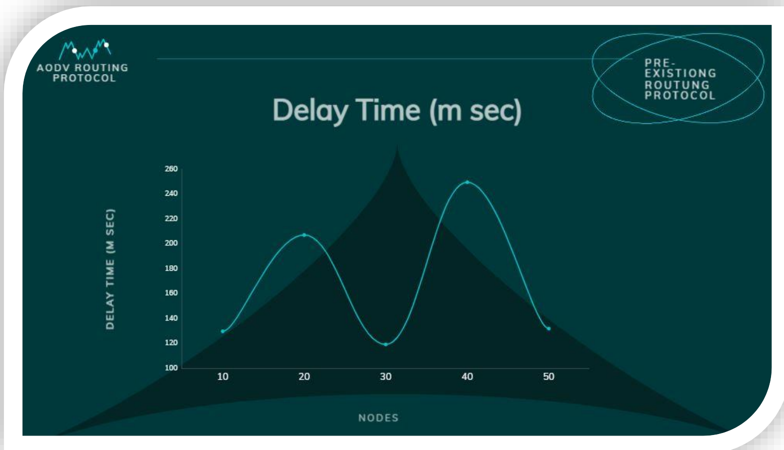

Figure-7 Average end-to-end Delay

6.1.2 Throughput – Throughput indicates the volume of data being transmitted from the source in a given moment of time. The rate at which packets are transferred from source to a particular destination is used to determine throughput, and it is represented in kb/s.

AODV Routing Algorithm		Nodes				
		10	20	30	40	50
	Throughput(kbps)	1289.72	1013.11	1290.1	963	954.24

Now to create the graph for AODV throughput we have plotted the nodes count on the X axis and Throughput in kbps on the Y axis.

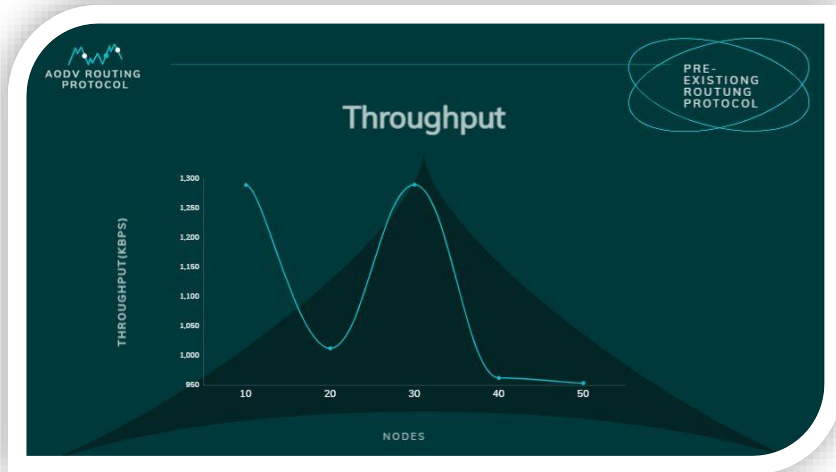

Figure-8 Throughput

6.2 Analysis of proposed routing – CPEP protocol - Now, second experiment uses the proposed (CPEP) algorithm to build a wireless network topology to evaluate the performance parameters while using the proposed routing technique. The experiment used 50 wireless nodes in total, with a total of 5 repetitions, where the number of nodes in each iteration was raised by 10 each time. After the simulation the results were recorded in the trace file and then for the evaluation purpose awk scripts were used. Separate scripts were used for the evaluation of the end -to - end delay and for the calculation of the throughout.

6.2.1 CPEP- Average end-to-end delay - 10 nodes were used in the start, and that number was subsequently raised to 10 for each succeeding iteration. The results gathered are as following –

Proposed(CPEP) Routing Algorithm		Nodes				
		10	20	30	40	50
	Delay Time(m sec)	38.538	42.528	33.696	41.064	39.002

Now, in order to build the graph for the CPEP(proposed) routing protocol end-to-end delay, the number of nodes was plotted on the X axis and the delay time in milliseconds was plotted on the Y axis.

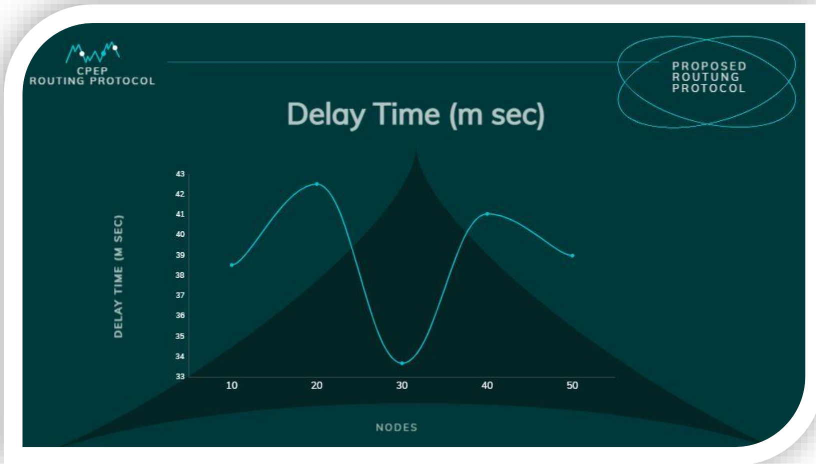

Figure -9 Average end-to-end Delay

6.2.2 Throughput - The proposed algorithm's throughput is observed in kbps. Here as well the nodes are increased by 10 after each iteration.

Proposed(CPEP) Routing Algorithm		Nodes				
		10	20	30	40	50
	Throughput(kbps)	6119.53	6136.46	6127.1	6149.42	6127.8

Now, in order to generate the CPEP(proposed) routing throughput graph, we have plotted the number of nodes on the X axis and the throughput in kbps on the Y axis.

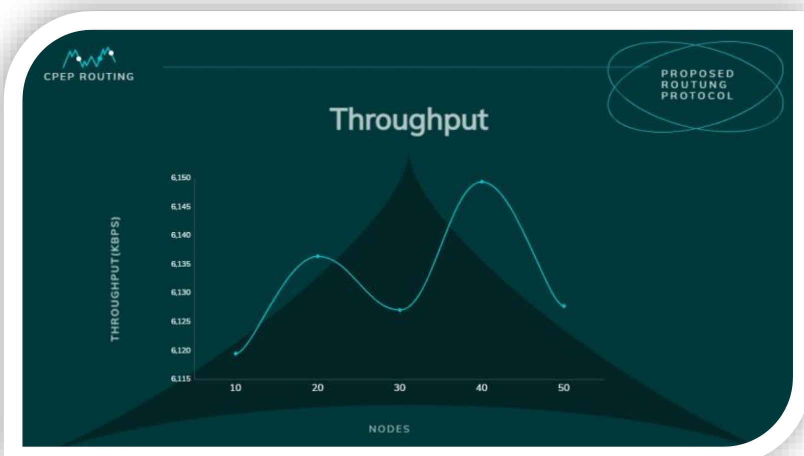

Figure-10 Throughput

6.3. Comparison of CPEP protocol with pre-existing protocol –

6.3.1 Comparison of Average end-to-end delay – For the comparison purpose, outcomes of average end - to - end delay for both of the routing protocols was gathered and was plotted on the graph. Also, from the graph, it is clearly visible that the delay in AODV is much more than the proposed CPEP routing protocol.

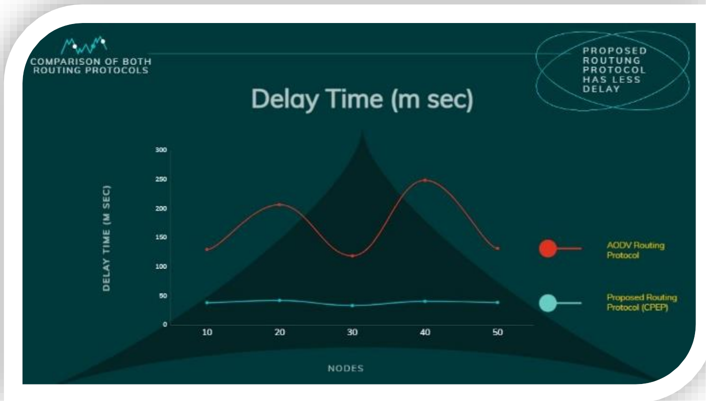

Figure-11 Comparison of Average end-to-end Delay

6.3.2 Comparison of Throughput – The outcomes of the throughput for both the routing algorithms were recorded and represented in the below graph. From the graph, it is clearly visible that, the throughput of the proposed CPEP routing protocol is more than the pre-existing protocol.

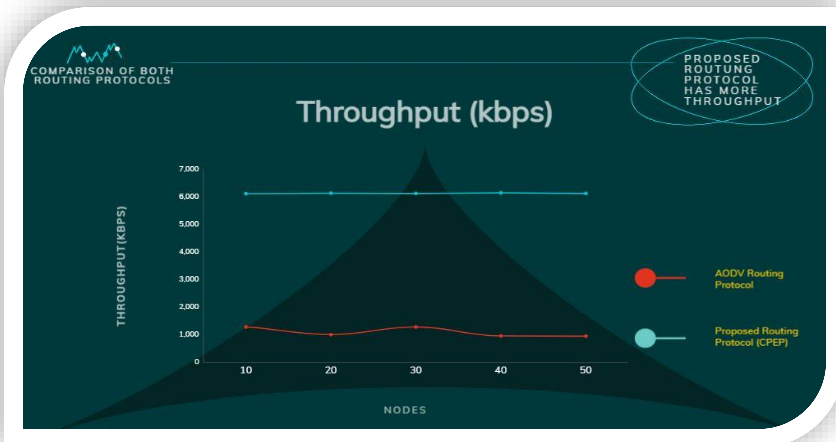

Figure-12 Comparison of Throughput

6.4 Discussion

Here we executed a customized algorithm for the routing of the data packets over the fog network known as CPEP routing algorithm. By creating a unique and wireless topology approach, implemented in the NS2, this research is capable of overcoming the limitations existing in the fog network environment. The algorithm starts with the analysis of the content type and then delivers the data packets or activate the nodes as per the priority. This continues and then to deliver the data packet, proper path with the highest energy is chosen. This path is considered to be the best path for the transmission of the data packets from the source to the destination and then in the end this best path is saved in the best route vector to achieve the concept of context or the path awareness. Our method significantly reduces latency, relieves network congestion, and optimizes the network. Also, by conducting multiple experiments, the proposed approach has been examined. Firstly, we tested the pre-existing traditional routing methods like AODV. Then after that we run various test to examine our proposed approach and again from the outcomes, we got that our customized proposed routing algorithm is capable of providing better results than the pre-existing traditional routing algorithms.

7. Conclusion and Future Work

Through the medium of this research, we tried and fulfilled our primary goal to address important issues such as latency, transmission overhead and also the congestion over the network. This is achieved by our customized routing algorithm which inculcated 4 processes in total like the type of service awareness, priority transmission of the data packets, after that selecting the best path with the highest energy levels and then in the end saving the path to minimize the latency level to the lowest possible. Also, to examine the outcomes, we performed multiple experiments on pre-existing routing algorithms like AODV. Then after that we compared our proposed customized algorithm with the pre-existing algorithm using the awk scripts and got that our proposed algorithm has better results then the pre-existing traditional routing techniques.

In the future this technique could be extended by taking and implementing the approach on a higher level like Using an actual IoT application and on a big infrastructure level and then measuring the latency and the relevant outcomes. Moreover, in future inculcating the cloud in this approach could further help to improve the congestion and also the throughput. Using cloud in our algorithm like after analyzing the type of service and the priority the whole topology could be programmed in such a way that the highly emergency data will be served locally and the other data with less priority could be transmitted to the cloud for the processing. Doing this will help to minimize the overall latency and will also improve the congestion over the fog networks.

Reference

Rydning, D.R.J.G.J., Reinsel, J. and Gantz, J., 2018. The digitization of the world from edge to core. Framingham: International Data Corporation, 16.

Alomari, A., Subramaniam, S.K., Samian, N., Latip, R. and Zukarnain, Z., 2021. Resource management in SDN-based cloud and SDN-based fog computing: taxonomy study. Symmetry, 13(5), p.734

Shinkuma, R., Yamada, Y., Sato, T. and Oki, E., 2020, November. Flow control in SDNEdge-Cloud cooperation system with machine learning. In 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS) (pp. 1304-1309). IEEE.

Diro, A.A., Reda, H.T. and Chilamkurti, N., 2018. Differential flow space allocation scheme in SDN based fog computing for IoT applications. Journal of Ambient Intelligence and Humanized Computing, pp.1-11.

Wiener, P., Zehnder, P. and Riemer, D., 2019, May. Towards context-aware and dynamic management of stream processing pipelines for fog computing. In 2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC) (pp. 1-6). IEEE.

Chelloug, S.A., 2015. Energy-efficient content-based routing in internet of things. Journal of Computer and Communications, 3(12), p.9.

da Silva Fré, G.L., de Carvalho Silva, J., Reis, F.A. and Mendes, L.D.P., 2015, June. Particle swarm optimization implementation for minimal transmission power providing a fullyconnected cluster for the internet of things. In 2015 International Workshop on Telecommunications (IWT) (pp. 1-7). IEEE.

Ukani, V. and Thacker, D., 2015, November. QoS aware geographic routing protocol for multimedia transmission in wireless sensor network. In 2015 5th Nirma University International Conference on Engineering (NUiCONE) (pp. 1-6). IEEE.

Samaras, N.S. and Triantari, F.S., 2016, November. On direct diffusion routing for wireless sensor networks. In 2016 Advances in Wireless and Optical Communications (RTUWO) (pp. 89-94). IEEE.

Nadar, C., Patil, R., Raut, S. and Mhatre, K., 2017, March. Energy efficient optimal opportunistic routing using sleep mode for wireless sensor network. In *2017 international conference on innovations in information, embedded and communication systems (ICIIECS)* (pp. 1-6). IEEE.

Tariq, M., Kim, Y. P., Kim, J. H., Park, Y. J., & Jung, E. H. (2009, February). Energy efficient and reliable routing scheme for wireless sensor networks. In *2009 International Conference on Communication Software and Networks* (pp. 181-185). IEEE.

Karim, L. and Nasser, N. (2012). Reliable location-aware routing protocol for mobile wireless sensor network, IET communications 6(14): 2149–2158.

Zhao, Z., Rosário, D., Braun, T., & Cerqueira, E. (2014, April). Context-aware opportunistic routing in mobile ad-hoc networks incorporating node mobility. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 2138-2143). IEEE.