AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL BEE COLONY ALGORITHM ON NETWORK ROUTING

A PROJECT REPORT

Submitted by

AISHWARYA B (1BM14CS012)
ANITHA D M (1BM14CS018)
ASHIKA M (1BM14CS023)
HARSHITHA N (1BM14CS043)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

Dr. Kavitha Sooda
Associate Professor, BMSCE

BMS COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Department of Computer Science and Engineering
B M S College of Engineering, Bull Temple Road, Bangalore 560019

CERTIFICATE FROM GUIDE

Certified that this project work **"AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL BEE COLONY ALGORITHM ON NETWORK ROUTING"** is the true work**.** This project work was carried out by AISHWARYA B (1BM14CS012), ANITHA D M (1BM14CS018), ASHIKA M (1BM14CS023), HARSHITHA N (1BM14CS043) under my supervision. This is to certify that I have gone through the complete project report. The students have incorporated all the correction and suggestions made by me.

GUIDE

Kavitha Sooda
Associate Professor,
Dept. of CSE,
BMS College of Engineering,
Bangalore-560019

Date:

BMS COLLEGE OF ENGINEERING BANGALORE -560019

CERTIFICATE FROM THE GUIDE

Department: Computer Science and Engineering

Candidates Degree Registered for: B.E.

Candidate details:

SL. NO.	Student Name	USN	Student's Signature
1	AISHWARYA B	1BM14CS012	
2	ANITHA D M	1BM14CS018	
3	ASHIKA M	1BM14CS023	
4	HARSHITHA N	1BM14CS043	

Certified that these candidates are students belong to Computer Science and Engineering of BMS College of engineering. They have carried out the project work of titled "**AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL BEE COLONY ALGORITHM ON NETWORK ROUTING** "as final year (7 & 8 Semester) dissertation project. It is in partial fulfillment for completing the requirement for the award of B.E. by VTU. The works is original and duly certify the same.

Guide Name Signature

HOD's Signature

Date:

ACKNOWLEDGEMENTS

We are extremely grateful to our principal, Dr. Mallikarjuna Babu K and our head of department, Dr. B G Prasad for giving us this opportunity to work on projects that has been implemented.

Sincere regards to our guide, Dr. Kavitha Sooda for guiding and motivating us to work efficiently and also to get a clear understanding of the survey that we did. We were guided in the right direction to come up with new ideas and go ahead with the feasible and innovative one.

We are also thankful to our team mates whose coordinated efforts helped us in completing this project. Our classmates have also been supportive and helpful in successful completion of our project work.

ABSTRACT

The aim of this project is to highlight and find the most optimal path in network routing, by using one of the Swarm Intelligence algorithms - Artificial Bee Colony (ABC) Algorithm. There are a number of hindrances in routing that leads to delay of transmission of data or transmission of incorrect data. Some of the common reasons for this are congestion in the network, loss of data packets, routing of packets along paths which are not considered to be optimal.

This project discusses the improvisation required for the existing ABC algorithm to obtain best results for network routing. The ABC algorithm utilizes the foraging nature of bees to find optimal routing paths. The proposed algorithm discusses ways to overcome the defects of native ABC algorithm by implementing a prior Agent based learning and Information Learning on the native ABC algorithm. The technique of Information learning: Multi Population Strategy with a prior learning on the QoS metrics are applied to the paths in the network to get the most optimized results. The result clearly depicts that this improvised ABC algorithm is more efficient and performs better than the existing ABC algorithm. This would be our approach to solve the problems in the domain of network routing.

TABLE OF CONTENTS

List of Figures

Chapter 1

Introduction

1.1 Overview:

Routing is the process of selecting paths in a given network along which to send the traffic. It is one of the important aspects of network communication which plays a major role in affecting the performance of any network, because other parameters of the network like reliability, congestion and throughput depend directly on it. With the enormous amount of data flow occurring on the Internet in addition to the conventional routing mechanisms, there are several scenarios for increased bottlenecks in the routing paths [1]. A network bottleneck results in slow communication speeds and limits user efficiency and productivity on a network. In order to improve the network routing there is a need for implementation of optimization techniques for the existing paths. The main challenges faced in obtaining an efficient routing path are various factors like increased operational costs, vessel utilization and reducing transport time have to be considered. Since many engineering and scientific fields involve these optimization issues, researchers pay greater attention to optimization techniques such as Swarm Intelligence.

Swarm intelligence (SI) is defined as the collective behavior of decentralized and self-organized swarms [2]. Some of the well known examples for these swarms are fish schools, bird flocks and the colony of social insects such as ants, bees and termites. The Artificial Bee Colony (ABC) algorithm is a swarm based meta-heuristic algorithm for optimizing numerical problems inspired by the intelligent foraging behavior of honey bees. The comparison results demonstrated that ABC is better than or at least comparable to the other EAs, such as GA, differential evolution (DE), and PSO [3].

Due to its simplest nature and efficiency, ABC is preferred over EA, GA, DE and PSO. Exploitation and Exploration are the two main characteristics of ABC. One possible limitation with the normal ABC algorithm is that for a given bee colony, the colony always tends to maximize exploitation and restricting exploration. Since exploration is also one of the important features of ABC algorithm, there exists a need to improvise the existing ABC algorithm to balance between Exploration and Exploitation processes to obtain better performance [4]. There are several learning mechanisms that can be used to improvise the results of ABC algorithm, out of which this paper focuses on Information Learning technique. Basically the whole population is been divided into several sub populations so that it can assign different individuals to different sub regions. So the whole population is divided into different groups or different sub regions and hence every neighbour's information can be improvised [5].Due to this independency there may be chances of inappropriate and inefficient solution to the given problem. Thus this can be solved by hence implementing multi population technique and two search mechanisms.

1.2 Motivation:

Artificial bee colony is a optimization technique which simulates the intelligent foraging behavior of honey bee swarms. It is swarm based meta-heuristic algorithm. In order to improve the network routing there is a need for implementation of optimization techniques for the existing paths. The main challenges faced in obtaining an efficient routing path are various factors like increased operational costs, vessel utilization and reducing transport time have to be considered. Since many engineering and scientific fields involve these optimization issues, there is a need to improvise these optimization techniques. Artificial bee colony algorithm has drawbacks. It is good in exploration and bad in exploitation. In order to overcome this problem information learning was used on ABC but it was not enough efficient. Hence artificial bee colony algorithm is further enhanced by using prior learning mechanism.

1.3 Objective:

The objective of the project is to find the shortest and the most efficient optimal path in networking by improvising the artificial bee colony algorithm based on information learning. The improvisation is done by using prior learning mechanism. Two techniques of Information learning like Multipopulation strategy and Two Search Mechanisms with a prior learning on QoS metric paths are applied to get the most optimized results.

1.4 Scope:

Artificial bee colony algorithm based on information learning is improved by prior learning mechanism to find the optimal path. Prior learning is done before applying information learning to the network. This prior learning includes selecting all possible paths based on the QoS constraints set by the user. The QoS metrics used for this system are bandwidth and delay. This will help to find the optimal path more efficiently in network routing.

1.5 Existing System:

To overcome the drawbacks as mentioned above, many improvisation techniques were applied to the original ABC algorithm. One of the techniques used was Information Learning mechanism on the original ABC algorithm. Two main modifications were Multi Population Strategy and Two Search Mechanism [5].

- Multipopulation Strategy

Initially the given population is divided into a number of sub populations, where each of the sub population is been assigned to a different sub region. Each of the sub population will search for that given sub region and later through Two Search Mechanisms information will be exchanged between the sub populations [5]. A random size is selected from the set of population size and the results are calculated, if the obtained result is not optimizing the solution in terms of global fitness factor, then the size of the population is varied accordingly to optimize the solution.

- Two Search Mechanism

The Two Search Mechanism basically involves the Lbest Search equation and the Gbest Search equation. Both of these search equations are applied to each of the sub population to obtain an optimal solution through information exchange between the sub populations. Lbest search equation is applied onto the Employer Bee phase to obtain a Lbest candidate for that given population [5]. Similarly Lbest candidates are found for each of the sub population in the given whole population. Based on the fitness factor, Lbest candidates are obtained. This hence optimizes the exploration process. Similarly the Gbest Search equation is applied onto the Onlooker Bee phase to improve the exploitation process. The Gbest candidate thus determines how the information has to be carried between the populations through a Transmission Vector to maximize the output [5]. The global fitness factor is the main determining criteria as to who is the Gbest candidate for the given population. Higher the global fitness factor higher the optimum results. Thus by these techniques mentioned above Information Learning is applied onto the existing ABC algorithm to get optimized results [5].

1.6 Proposed System:

From the above seen Literature Review and ABC related works, it was evident that there is a need for an improvisation for the existing algorithm. When ABC algorithm with Information Learning (ILABC) is implemented on network routing to obtain optimized results, it will be inefficient to apply it on all paths becomes it is time consuming and also wastage of resources allotted. Therefore the proposed system implements a prior learning before Information Learning to improve efficiency. This prior learning includes selecting all possible paths based on the QoS constraints set by the user. The QoS metrics used for this system are bandwidth and delay. Higher bandwidth and lesser delay is preferred for optimized results. Thus this proposed system improves efficiency according to the user requirements.

Chapter 2

Literature Survey

2.1 Network Routing

Network Routing is the process of moving data packets from the source to the destination. A mobile ad hoc network (MANET) is a continuously self-configuring network of wirelessly connected mobile devices [6].Routing in mobile ad hoc networks is broadly classified as Reactive (on Demand) and Proactive (Table Driven) Protocols. Reactive protocols set up routes on demand. In the Ad-Hoc On-Demand Distance Vector (AODV) reactive routing protocol, topology information is transmitted by nodes on-demand to only those nodes to which it wants to establish a route. Proactive protocols like the Destination Sequence Distance Vector or DSDV maintains a list of destinations and their routes by periodically sending routing tables throughout the network. However, both the protocols are not highly scalable since they result in traffic overhead due to increased mobility. This drawback can be overcome by Improved Artificial Bee Colony Algorithm as described in the following sections.

2.2 Learning Mechanisms

Learning is a process of modifying the existing or acquiring new knowledge and behaviors. The ability of learning is acquired by all living creatures including human beings and some machines. There are many types of learning mechanisms out of which they are broadly classified into Supervised and Unsupervised Learning. From their names only, it is evident that Supervised learning is a task of inferring outputs from the supervised trained data [7]. This supervised trained data is under the supervision of supervisor or trainer. Using this trained data set learning can be easily implemented to any test data to obtain an optimal solution. On the other hand, under Unsupervised learning algorithm, results are inferred from datasets without labeled responses or classes [7].

2.3 Bio Inspired Algorithms

Optimization plays a significant role in solving complex mathematical problems in engineering. Optimization algorithms can be categorized as deterministic and stochastic in nature. The former method requires an enormous computational effort which tends to fail as the size of the problem increases, which is a motivation to employ bio inspired algorithms [5]. These algorithms are a motivational approach for employing computationally efficient alternative models for deterministic approach. The significant beauty in nature inspired algorithms is that it receives its sole inspiration from nature, they describe and resolve complex relationships from intrinsically very simple initial conditions and rules with little or no knowledge of the search space.

Bio inspired computing is efficient in solving problems of almost all areas, including computer networks, data mining, power systems, image processing, security, control systems, robotics and many more. [8]

2.4 Original ABC

Artificial Bee Colony Optimization is inspired by the behavior of bees in nature [9]. This solution search process is a step by step iterative process. It has gained an enormous attention because of its ease of implementation and has been successful in solving many complex optimization problems. Good Balance between exploration and exploitation is very much necessary for good performances on problem optimization. The colony of artificial bees contains three groups of bees: onlookers, employed bees and scouts. A bee which is waiting in the dance area, making decision to choose a food source is called Onlooker bee. The one which is going to the food source visited by it before is called an employed bee and the one that carries out random search for new food sources is called a Scout. The food source location corresponds to a possible solution to the optimization problem and the nectar amount corresponds to the quality of the solution. For every food source, there is only one employed bee [2]. An employed bee makes modification on the position in her memory depending on the local information and tests the nectar amount of the new source. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout. If the nectar amount of the new food source is higher than the previous one, then the employed bee memorizes the new position and forgets the old one [10]. After completing the search process, they share the nectar information of the food sources and their position information with the onlooker bees in the dance area [10]. The search process of ABC has three major steps:

I) Employed bees are sent to a food source and their nectar quality is estimated

II) food sources are selected by Onlooker bees based on information collected from employed bees and estimate their nectar quality

III) scout bees are determined and are employed on possible food sources for exploitation.

The position of the food sources is selected by the bees at the initial stage and their nectar qualities are measured. The employed bees then communicate the nectar information of the sources with the onlooker bees waiting at the dance area within the hive.

The main steps of the algorithm are as below:

```
1: Initialize Population
2: **repeat**
3: Place the employed bees on their food sources
4: Place the onlooker bees on the food sources depending on their nectar amounts
5: Send the scouts to the search area for discovering new food sources
6: Memorize the best food source found so far
7: **until** requirements are met. [3]
```

Drawbacks of Original ABC

- As from the above working of the Artificial Bee Colony algorithm it is evident that the algorithm provides efficient results in exploration processes at the cost of exploitation process.
- For a given bee colony, best results are obtained only for exploration of food source and exploitation process is not that efficient.
- It is observed to face certain challenging problems such as slow convergence speed compared to other representative population-based algorithms and also being trapped into the local optimal solution [11].

So there exists a need for an improvisation for this existing system which is the purpose of this paper [4]. An improvised ABC is put forward where searching is done by bees only around the best solution of the previous iteration for better exploitation. Under the Information Learning ABC, two strategies are incorporated i.e. multi population technique and two search mechanism.

2.5 Multi-population technique

Multi population technique improves the performance of the existing ABC algorithm by using the benefits of cooperation as a social behavior. Here, the population is divided into sub populations and the information between the sub populations is shared to obtain the cooperation. Partial vectors are obtained by partitioning the solution vector into two or more smaller vectors and each of it is optimized using separate population [12]. Partial solutions are assembled to obtain a complete solution and thus cooperation is obtained. Instead of focusing on a specific area, the solutions are distributed over the search space [12]. Hence high-quality solutions can be generated. This improvised ABC also employs three types of bees i.e. (employed bees, onlooker bees and scout bees), similar to existing system of ABC. In initialization phase, it generates initial populations which are randomly distributed. After initialization, evaluation of the populations is done and fitness of their individuals is determined. In the next phase, based on the local information each population evolves. As stated, initially the food sources are randomly selected by the bees and their nectar quantities are determined. At the beginning of each cycle, employed bees of the population share the nectar information of the food sources with the onlooker bees which are waiting in the dance area. After this, the employed bees go to the food locations which were visited by them before and then selects a new food location in the neighborhood and calculates the nectar quantity. Based on the information provided by the employed bees, the onlooker bees select a food source depending on the probability related to that food source. As the nectar quantity increases, the probability associated with the food source also increases [12]. After it arrives at the promising area, it selects a new food source based on the information stored in its memory. Then the nectar quantities of the food locations are determined and scout bees are sent to possible new food locations. A food source is abandoned in case it cannot be improved after a certain number of iterations (called limit, which is evaluated manually). Depending on the nectar quantities, the scout bees randomly produce a new position. The new food sources found by the scout bees replace the abandoned food sources. Then the best solutions obtained by the cooperative populations are shared to obtain cooperation. This involves two main steps; each population is evaluated and the best solution obtained by its individuals is determined. In the next step, these solutions are added to a list and the best solution among them is determined. Then the food positions in each population is updated. If the newly found food source has better nectar, then the candidate food source is updated. These steps are repeated through a number of iterations (limit) and returns the best solution found by cooperative populations and terminates the algorithm.

Chapter 3:

Software and Hardware Requirement Specification

3.1 Functional Requirements

The project consists of Java programs, Javascript documents, and HTML documents. These programs will illustrate physical concepts and will be available on the world wide web. The premise of the project is to have a gedanken ("thought experiment") on a computer. Instead of imagining objects moving, accelerating, colliding, and bouncing, the computer simulates and animates these objects. The goals of the project are to:

- Facilitate the understanding of physics through virtual tutoring, experiments, and demonstrations
- Allow students to do experiments which would be impossible, difficult, costly, or dangerous in real life
- Give physics teachers material which is freely available on the web to supplement their curricula
- Create a set of basic objects which allows new Java programs to be written "on the fly", according to the principles of "Just In Time Teaching" (JITT).

3.1.1 Java

- **Java programs can run in a browser**. With the help of the Java Plug-In, most users can run complicated programs from their home, without having to buy any software. The programs can also be accessed in the middle of a lecture provided the lecturer has a computer with web access.
- **Java is cross-platform.** As long as the operating system in question has a current java runtime environment (jre), that OS can run the Java program. Java programs can be run in Windows, Mac, Linux, Unix, and Solaris.
- **Java is Object Oriented.** Object oriented languages allow the programmer to build a program using blocks of code which are called objects. Objects encapsulate functions and data. By "encapsulate" I mean that a programmer does not need to know the

exact mechanics of the object in order to program with them, but instead gets all required data from built in behaviors of that object. It is then easy for even the most novice programmer to create a complicated program simply by using built in behaviors.

3.1.2 Java Terminology

- **Applet**: a java program that runs on the web in a browser
- **Java:** a high level programming language that is cross-platform, object oriented, and can be run in a browser, on a server, or as a stand-alone application. It was created by Sun Microsystems and first released in 1995.
- **Java plug-in** : a java interpreter which allows users with older browsers to run new versions of java. In addition, it also ensures that the programmer and the user are using the same interpreter so that less bugs are caused.
- **Java Runtime Environment (jre)**: the interpreter loaded either by the Java plug-in, or contained in the browser, or contained in the operating system.
- **Java Development Kit (jdk)**: a kit from Sun which includes a compiler, a class loader, an applet viewer, and demos.
- **Java Virtual Machine (jvm)**: a term for the entire software package that runs the java program, including the java runtime environment
- **Methods:** functions contained in a Java object. These are the available behaviors of the object.

3.1.3 Types of Java Applications

There are mainly 4 type of applications that can be created using java programming:

- **Standalone Application**

It is also known as desktop application or window-based application. An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

- **Web Application**

An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, jsf etc. technologies are used for creating web applications in java.

- **Enterprise Application**

An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

- **Mobile Application**

An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

3.1.4 Java Architecture

- **Compilation and interpretation in Java**

Java combines both the approaches of compilation and interpretation. First, java compiler compiles the source code into bytecode. At the run time, Java Virtual Machine (JVM) interprets this bytecode and generates machine code which will be directly executed by the machine in which java program runs. So java is both compiled and interpreted language.

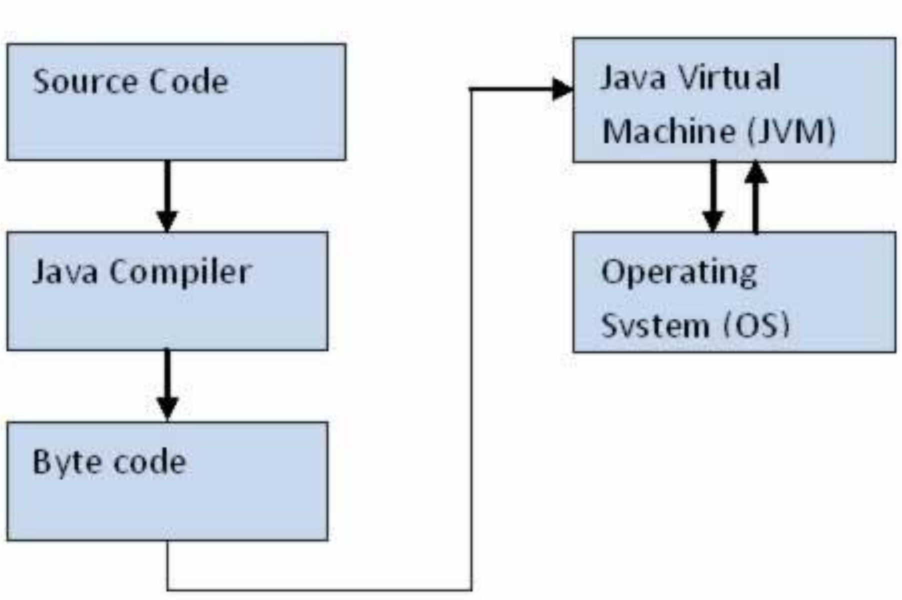


Figure 3.1: Java Architecture

- **Java Virtual Machine (JVM)**

JVM is a component which provides an environment for running Java programs. JVM interprets the byte code into machine code which will be executed the machine in which the Java program runs.

- **Why Java is Platform Independent?**

Platform independence is one of the main advantages of Java. In another words, java is portable because the same java program can be executed in multiple platforms without making any changes in the source code. You just need to write the java code for one platform and the same program will run in any platforms. But how does Java make this possible?

As we discussed early, first the Java code is compiled by the Java compiler and generates the bytecode. This bytecode will be stored in class files**.** Java Virtual Machine (JVM) is unique for each platform. Though JVM is unique for each platform, all interpret the same bytecode and convert it into machine code required for its own platform and this machine code will be directly executed by the machine in which java program runs. This makes Java platform independent and portable.

Let’s make it more clear with the help of the following diagram. Here the same compiled Java bytecode is interpreted by two different JVMS to make it run in Windows and Linux platforms.

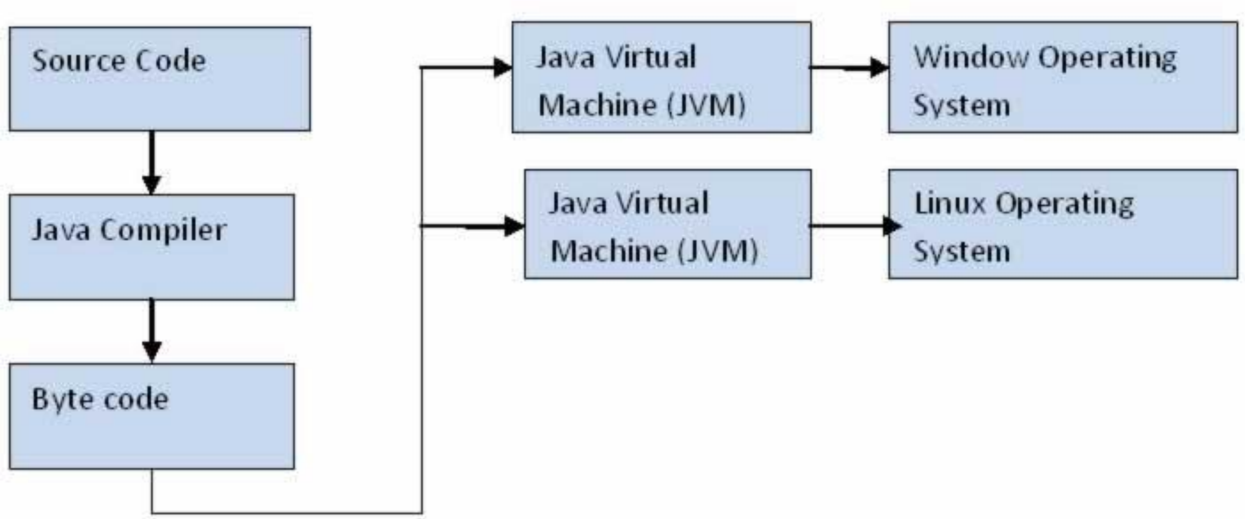


Figure 3.2: Platform Independent Java

- **Java Runtime Environment (JRE) and Java Architecture in Detail**

Java Runtime Environment contains JVM, class libraries and other supporting components.

As you know the Java source code is compiled into bytecode by Java compiler. This bytecode will be stored in class files. During runtime, this bytecode will be loaded, verified and   JVM interprets the bytecode into machine code which will be executed in the machine in which the Java program runs.

A Java Runtime Environment performs the following main tasks respectively.

1. Loads the class

This is done by the class loader

2. Verifies the bytecode

This is done by bytecode verifier.

3. Interprets the bytecode

This is done by the JVM

These tasks are described in detail in the subsequent sessions.A detailed Java architecture can be drawn as given below.

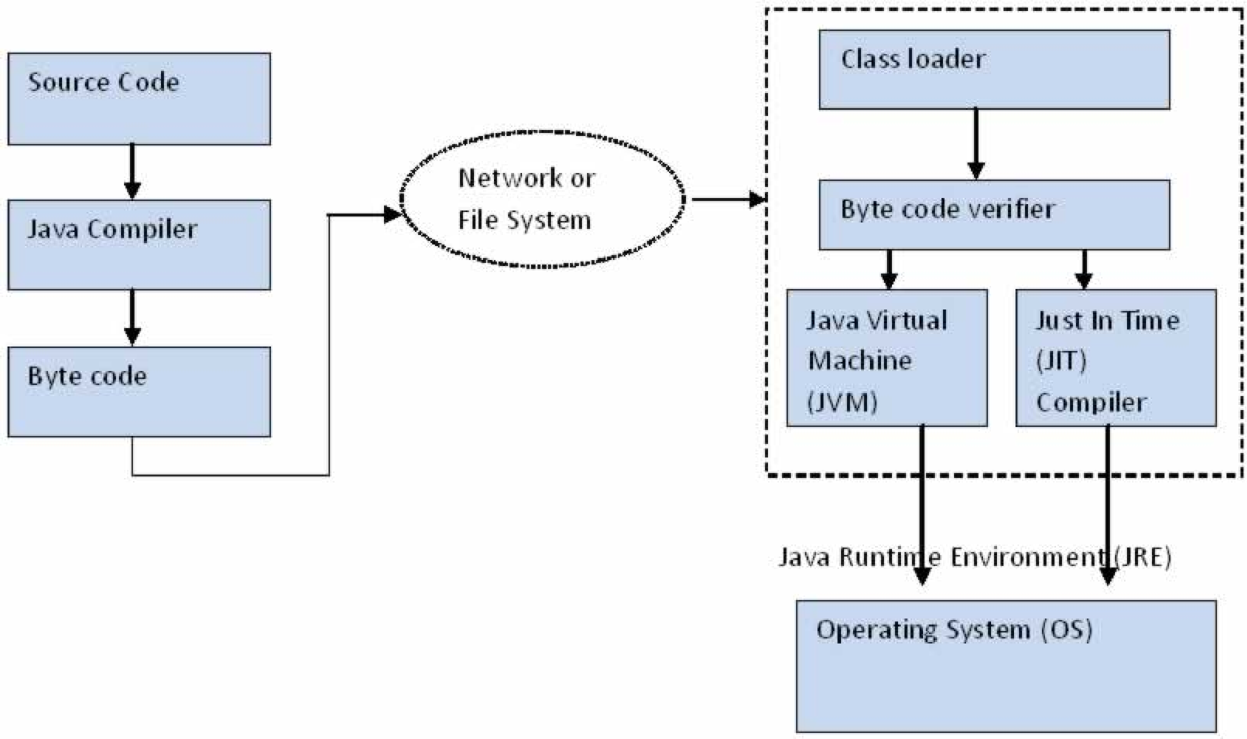


Figure 3.3: Java Architecture in Detail

a) Class loader

Class loader loads all the class files required to execute the program. Class loader makes the program secure by separating the namespace for the classes obtained through the network from the classes available locally. Once the bytecode is loaded successfully, then next step is bytecode verification by bytecode verifier.

b) Byte code verifier

The bytecode verifier verifies the byte code to see if any security problems are there in the code. It checks the byte code and ensures the followings.

1. The code follows JVM specifications.

2. There is no unauthorized access to memory.

3. The code does not cause any stack overflows.

4. There are no illegal data conversions in the code such as float to object references.

Once this code is verified and proven that there is no security issues with the code, JVM will convert the byte code into machine code which will be directly executed by the machine in which the Java program runs.

c) Just in Time Compiler

You might have noticed the component "Just in Time" (JIT) compiler in Figure 3. This is a component which helps the program execution to happen faster. How? Let's see in detail.

As we discussed earlier when the Java program is executed, the byte code is interpreted by JVM. But this interpretation is a slower process. To overcome this difficulty, JRE include the component JIT compiler. JIT makes the execution faster.

If the JIT Compiler library exists, when a particular bytecode is executed first time, JIT complier compiles it into native machine code which can be directly executed by the machine in which the Java program runs. Once the byte code is recompiled by JIT compiler, the execution time needed will be much lesser. This compilation happens when the byte code is about to be executed and hence the name “Just in Time”.

Once the bytecode is compiled into that particular machine code, it is cached by the JIT compiler and will be reused for the future needs. Hence the main performance improvement by using JIT compiler can be seen when the same code is executed again and again because JIT make use of the machine code which is cached and stored.

- **Why Java is Secure?**

As you have noticed in the prior session “Java Runtime Environment (JRE) and Java Architecture in Detail”, the byte code is inspected carefully before execution by Java Runtime Environment (JRE). This is mainly done by the “Class loader” and “Byte code verifier”. Hence a high level of security is achieved.

- **Garbage Collection**

Garbage collection is a process by which Java achieves better memory management. As you know, in object oriented programming, objects communicate to each other by passing messages. (If you are not clear about the concepts of objects, please read the prior chapter before continuing in this session).

Whenever an object is created, there will be some memory allocated for this object. This memory will remain as allocated until there are some references to this object. When there is no reference to this object, Java will assume that this object is not used anymore. When garbage collection process happens, these objects will be destroyed and memory will be reclaimed.

Garbage collection happens automatically. There is no way that you can force garbage collection to happen. There are two methods “System.gc ()” and “Runtime.gc ()” through which you can make request for garbage collation. But calling these methods also will not force garbage collection to happen and you cannot make sure when this garbage collection will happen.

In the next chapter, we will create, compile and run our first Java program and will understand our program clearly.

3.2 Non-Functional Requirements

Performance

The performance requirement is usually measured in terms of response time for a given screen transaction per user. In addition to response time, performance can also be measured in transaction throughput, which is the number of transactions in a given time period, usually one second. For example, you could have a performance measurement that could be no more than three seconds for each screen form or a transaction throughput of one hundred transactions in one second. Regardless of the measurement, you need to create an architecture that allows the designers and developers to complete the system without considering the performance measurement.

Scalability

Scalability is the ability to support the required quality of service as the system load increases without changing the system. A system can be considered scalable if, as the load increases, the system still responds within the acceptable limits. It might be that you have a performance measurement of a response time between two and five seconds. If the system load increases and the system can maintain the performance quality of service of less than a five second response time, then your system is scalable. To understand scalability, you must first understand the capacity of a system, which is defined as the maximum number of processes or users a system can handle and still maintain the quality of service. If a system is running at capacity and can no longer respond within an acceptable time frame, then it has reached its maximum scalability. To scale a system that has met capacity, you must add additional hardware. This additional hardware can be added vertically or horizontally.

Vertical scaling involves adding additional processors, memory, or disks to the current machine(s). Horizontal scaling involves adding more machines to the environment, thus increasing the overall system capacity. The architecture you create must be able to handle the vertical or horizontal scaling of the hardware. Vertical scaling of a software architecture is easier than the horizontal scaling. Why? Adding more processors or memory typically does not have an impact on your architecture, but having your architecture run on multiple machines and still appear to be one system is more difficult.

Reliability

Reliability ensures the integrity and consistency of the application and all its transactions. As the load increases on your system, your system must continue to process requests and handle transactions as accurately as it did before the load increased. Reliability can have a negative impact on scalability. If the system cannot maintain the reliability as the load increases, then the system is really not scalable. So, for a system to truly scale it must be reliable.

Availability

Availability ensures that a service/resource is always accessible. Reliability can contribute to availability, but availability can be achieved even if components fail. By setting up an environment of redundant components and failover, an individual component can fail and have a negative impact on reliability, but the service is still available due to the redundancy.

Extensibility

Extensibility is the ability to add additional functionality or modify existing functionality without impacting existing system functionality. You cannot measure extensibility when the system is deployed, but it shows up the first time you must extend the functionality of the system. You should consider the following when you create the architecture and design to help ensure extensibility: low coupling, interfaces, and encapsulation.

Maintainability

Maintainability is the ability to correct flaws in the existing functionality without impacting other components of the system. This is another of those systemic qualities that you cannot measure at the time of deployment. When creating an architecture and design, you should consider the following to enhance the maintainability of a system: low coupling, modularity, and documentation.

Manageability

Manageability is the ability to manage the system to ensure the continued health of a system with respect to scalability, reliability, availability, performance, and security. Manageability deals with system monitoring of the QoS requirements and the ability to change the system configuration to improve the QoS dynamically without changing the system. Your architecture must have the ability to monitor the system and allow for dynamic system configuration.

Security

Security is the ability to ensure that the system cannot be compromised. Security is by far the most difficult systemic quality to address. Security includes not only issues of confidentiality and integrity, but also relates to Denial-of-Service (DoS) attacks that impact availability. Creating an architecture that is separated into functional components makes it easier to secure the system because you can build security zones around the components. If a component is compromised, then it is easier to contain the security violation to that component.

3.3 Hardware Requirements

- Minimal hardware requirements would be needed for our project. A memory space of about 1 GB with free disk space of 1 GB is recommended.
- A system with a processor speed of 1.5 GHz would be sufficient.

3.4 Software Requirements

- Java: Java version 1.6 is recommended for implementing the algorithm.
- Net Beans (IDE): The java code will be executed on the given netbeans platform(6.9.1)

3.5 Cost Estimates

The cost estimate for this project is very basic or minimal as no new hardware components are used and Net Beans is free, open sourced.

Chapter 4

Design

4.1 High Level Design:

4.1.1 System Architecture:

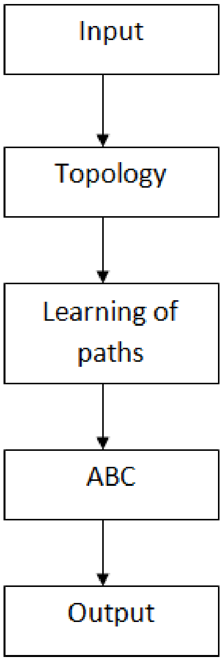


Figure 4.1: Flowchart of the working of the proposed system.

4.1.2 Abstract specification of Sub-systems

The system will have the following subsystems with the respective functions:

- Input: The input to our system will be the source from which data has to be sent, and the destination to which it has to send the data. This is specified by the user and can change for every execution.
- Topology: The topology that is considered here is a homogeneous network topology using wireless adhoc standard. This protocol uses a dual band wireless network technology. The network comprises of interconnected routers and nodes. Any good link can be utilized only to 80% of its capacity, i.e., 360 Mbps.
- Learning of paths: There are two phases of learning involved – Agent based learning and Information learning on ABC Algorithm. In agent based algorithm, we consider two user specific metrics namely bandwidth and delay. The paths are filtered in a manner so as to select only those which use less bandwidth and give less delay. From all the possible paths, this process filters it down to fewer paths between the source and the destination. The information learning on ABC is applied in the next stage.
- Artificial Bee Colony Algorithm: In this stage, the information learning happens. The whole network is divided into subpopulations where each subpopulation discover the neighboring nodes based on the regions that have been formed. The path selected from the learning of the path mentioned above, the algorithm runs to transfer data from source to destination .
- Output: The output is the most optimized path discovered between the source and the destination. This path is the best route that is chosen from among the paths filtered in the previous stage of information learning.

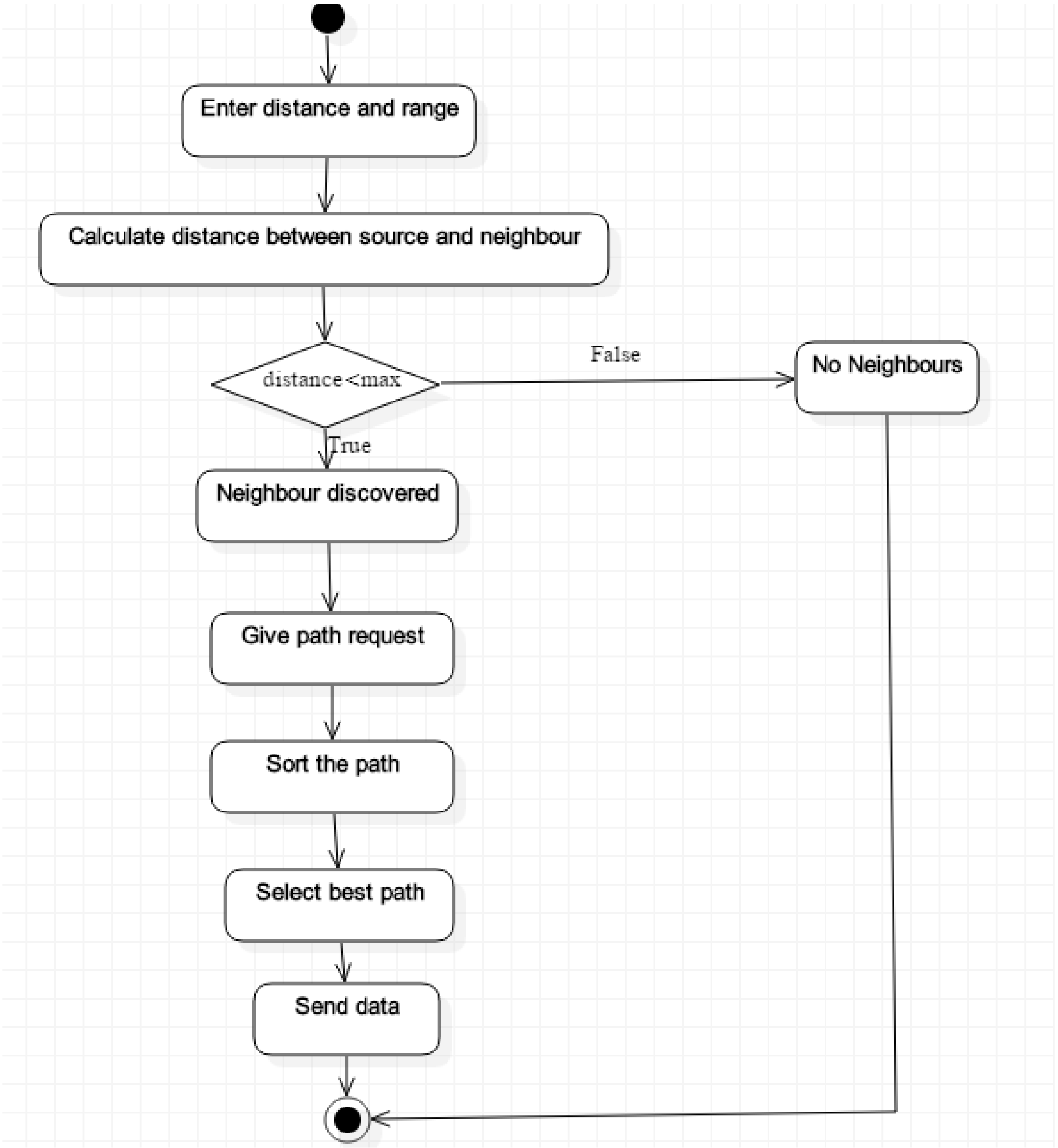


Figure 4.2: Flowchart of the modules

4.1.3 Use-Case Diagram

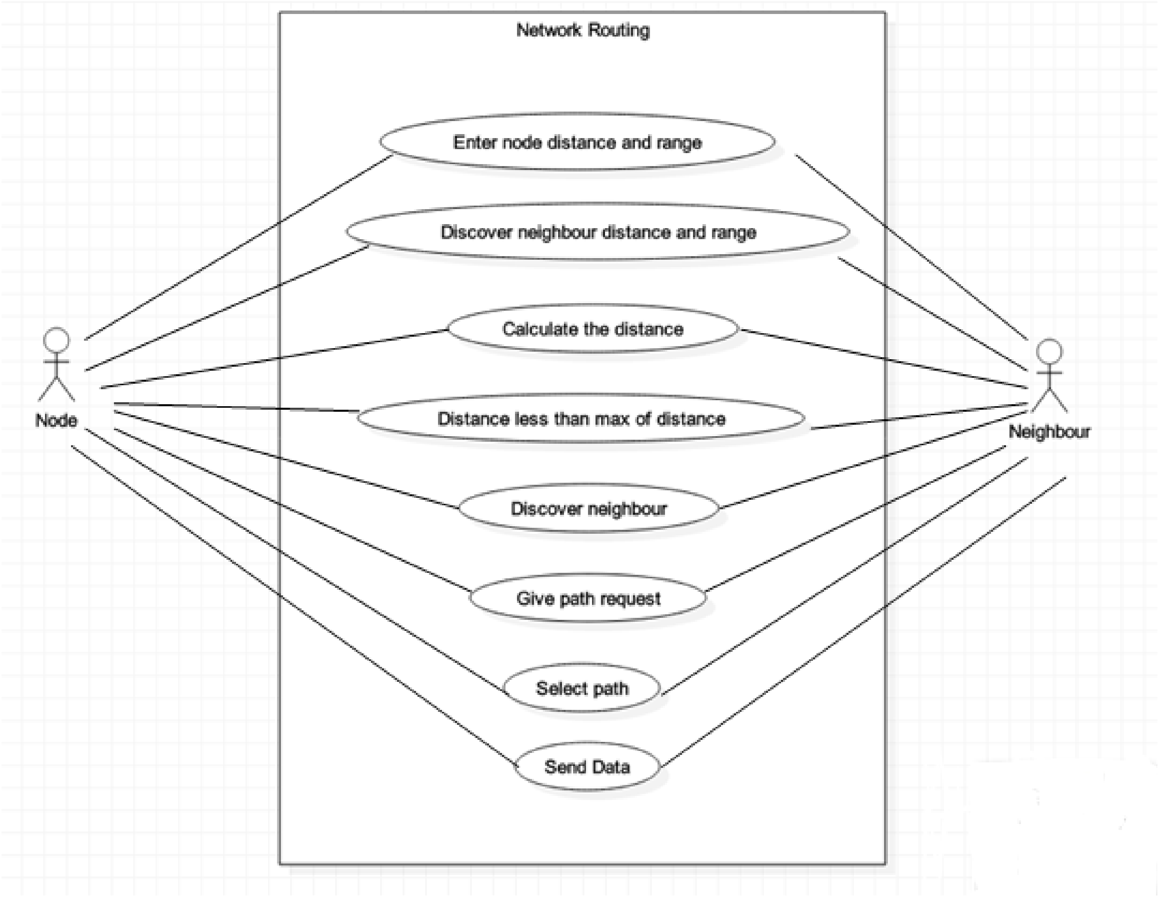


Figure 4.3: Use Case diagram of the system

4.2 Detailed Design

4.2.1 Sequence Diagram

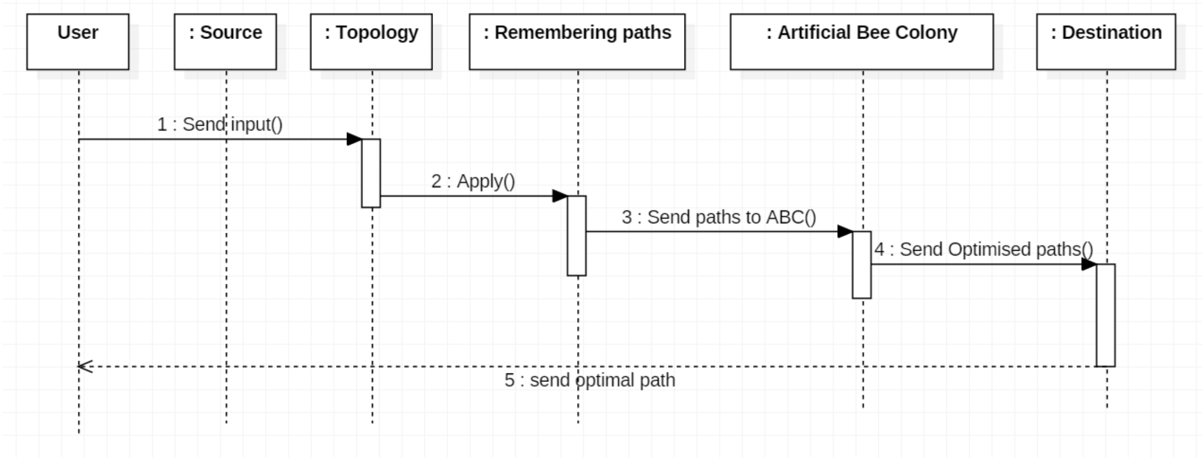


Figure 4.4: Sequence diagram of the working of the proposed system.

Chapter 5:

Implementation

5.1 Overview of technologies Used

- ### Java JDK 1.6

  The **Java Development Kit** (**JDK**) is an implementation of either one of the Java Platform, Standard Edition, Java Platform or Enterprise Edition platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows. The JDK includes a private JVM and a few other resources to finish the development of a Java Application. The JDK has as its primary components a collection of programming tools, including:

  - appletviewer – this tool can be used to run and debug Java applets without a web browser
  - java – the loader for Java applications. This tool is an interpreter and can interpret the class files generated by the javac compiler.
  - javac – the Java compiler, which converts source code into Java bytecode
  - javadoc – the documentation generator, which automatically generates documentation from source code comments
  - jar – the archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.
  - javafxpackager – tool to package and sign JavaFX applications
  - jdb – the debugger

- ### NetBeans 6.9.1

  NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called modules. NetBeans IDE 6.0 introduced support for developing IDE modules and rich client applications based on the NetBeans platform. NetBeans IDE 6.9, released in June 2010, added support for OSGi, Spring Framework 3.0, Java

EE dependency injection (JSR-299), Zend Framework for PHP, and easier code navigation (such as "Is Overridden/Implemented" annotations), formatting, hints, and refactoring across several languages. Users can choose to download NetBeans IDE bundles tailored to specific development needs. The bundle that we have used for our project includes the following :

- Netbeans Platform SDK
- Java SE
- Java FX

5.2 Implementation Details of Modules

5.2.1 Node Initialisation Module

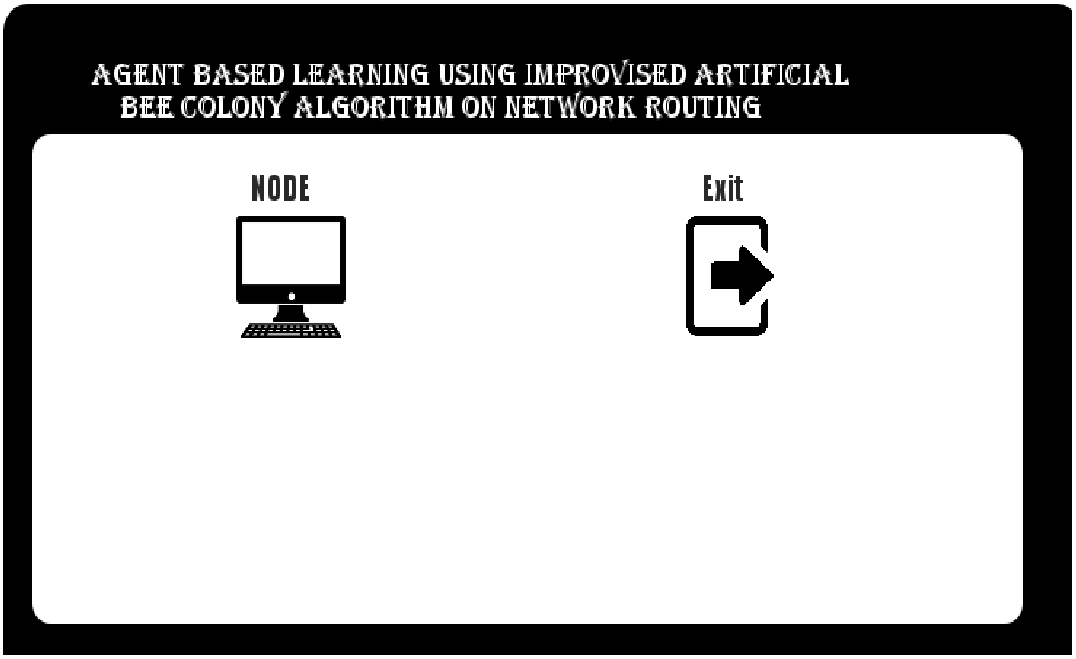
This is the most important and the most vital module of the project. It contains initialisation of nodes. Since nodes are the vital part in any simulation when it comes to networking, this module plays a very important role in node initialisation. There may be ‘n’ number of nodes created by the user as per the requirement. Each node has node specific details stored in it. The node details that are being stored are node name which is unique to each node, port number which is the system port number through which the corresponding communicates with other nodes, sname which is the system name in which the nodes are been created, distance parameter and range parameter together which determines as to which region the node belongs to in the region based learning, bandwidth and delay QoS metrics value based on which initial learning of paths take place depending on the user set threshold values , destination node indicating as to which node in the region the data has to be sent, Neighbour node details containing all the possible nodes the current node can identify and connect with in that network, Receive message tab to display any message that has been received, File name indicating which file has to be sent from the source to destination.

Each of the node also contain three buttons like

a) Browse - This button is used to browse for the data or file that has been stored in the local system.

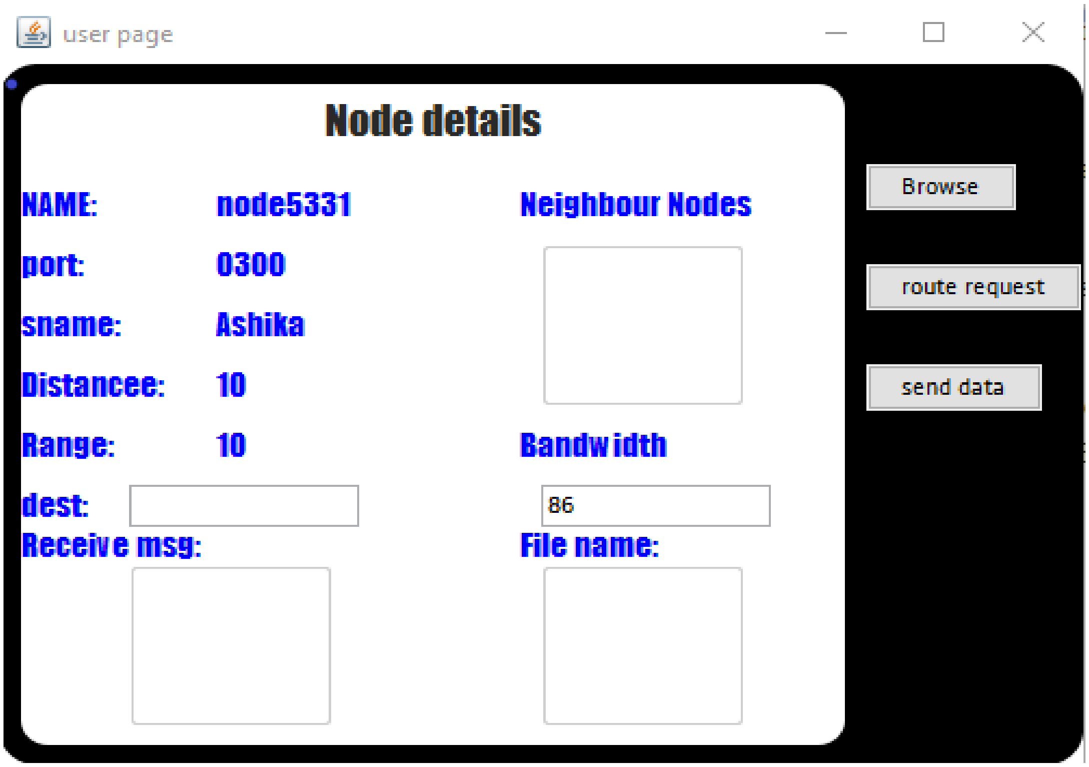
b) Route Request - This button is used to find all possible paths from source to destination based on the underlying algorithm.

c) Send Data - Once the best path is chosen by the user based on the prior learning, this button is used to send data or file selected using the Browse button mentioned above from source to destination.



AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL
BEE COLONY ALGORITHM ON NETWORK ROUTING

Figure 5.1: GUI of the system



user page

Node details

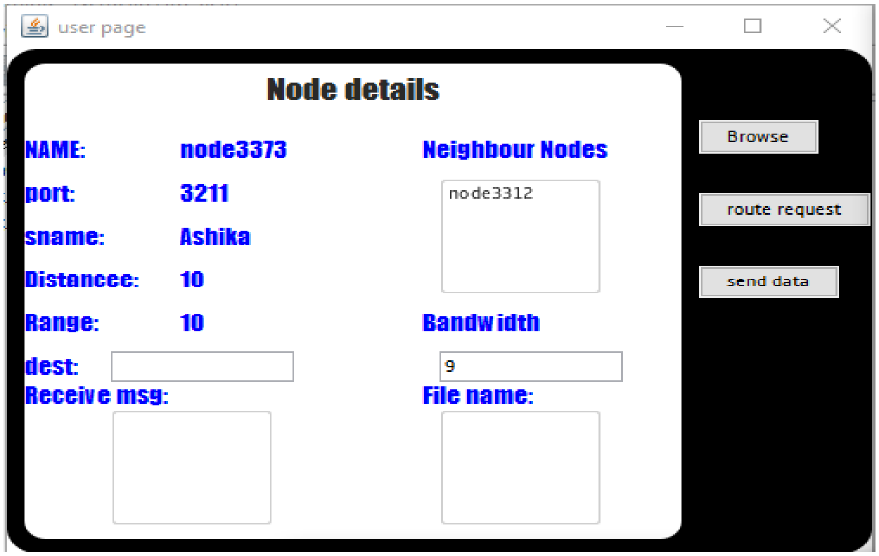
NAME:	node5331	Neighbour Nodes
port:	0300	
sname:	Ashika	
Distancee:	10	
Range:	10	Bandwidth
dest:		86
Receive msg:		File name:

Figure 5.2:Node Initialisation module

5.2.2 Neighbour Discovery Module

This is the second module of the implementation which involves neighbour discovery. Based on the distance and range parameter as specified during each of the node during node initialisation, the nodes will start discovering its neighbouring nodes. The neighbour discovery is carried out based on Euclidian distance formula. Once the details is been obtained from the node, all the details is been broadcasted using Multicast IP protocol in this module to notify all nodes about their neighbours. MulticastSocket and Inet address are used to communicate with all nodes present in that region respective to that node. The region for every node is calculated based on the Euclidian formula where the inputs for this case are distance and range parameter in each node.

Thus this module is analogous to Employee Bee phase in the Artificial Bee Colony algorithm, where the neighbour discovery happens in a similar way of honey bees foraging for the food source. The Employee Bee phase in Artificial Bee Colony algorithm is sent to the food source and their nectar quality is estimated. Similarly in this phase al the initialised nodes starts discovery it possible neighbour through Multicast IP protocol and all the possible neighbours to which a node can establish a connection is been clearly displayed in the Neighbour node tab for each node. The Information learning technique which contains the Multi Population mechanism is been thus implemented in this module by discovering other nodes based on the different regions.



Node details

NAME: node3373

port: 3211

sname: Ashika

Distancee: 10

Range: 10

dest:

Receive msg:

Neighbour Nodes

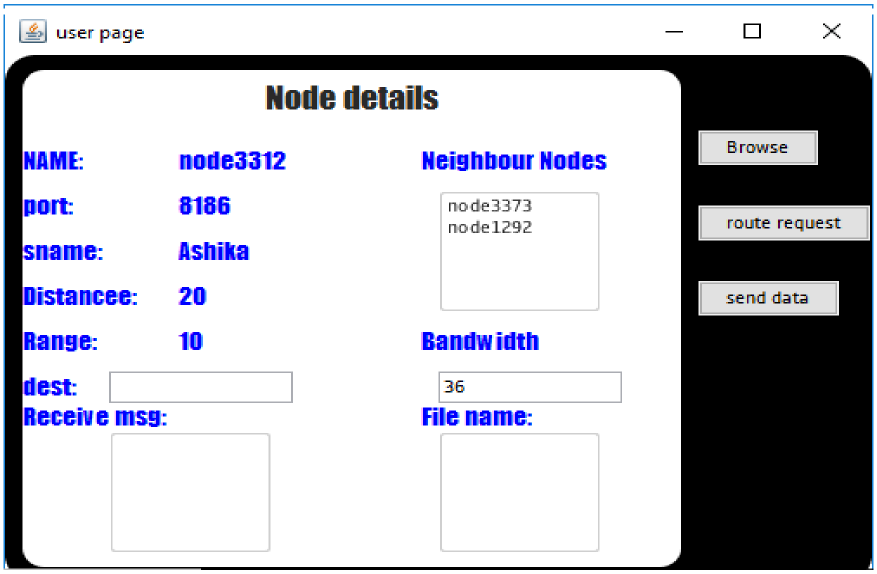
node3312

Bandwidth

9

File name:

Figure 5.3(a): Neighbour Discovery module



Node details

NAME: node3312

port: 8186

sname: Ashika

Distancee: 20

Range: 10

dest:

Receive msg:

Neighbour Nodes

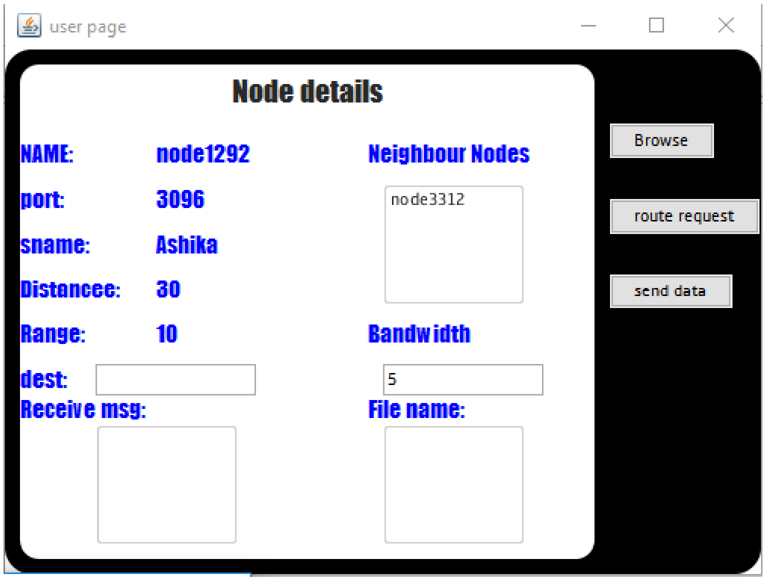
node3373
node1292

Bandwidth

36

File name:

Figure 5.3(b): Neighbour Discovery module



NAME: node1292

port: 3096

sname: Ashika

Distancee: 30

Range: 10

dest:

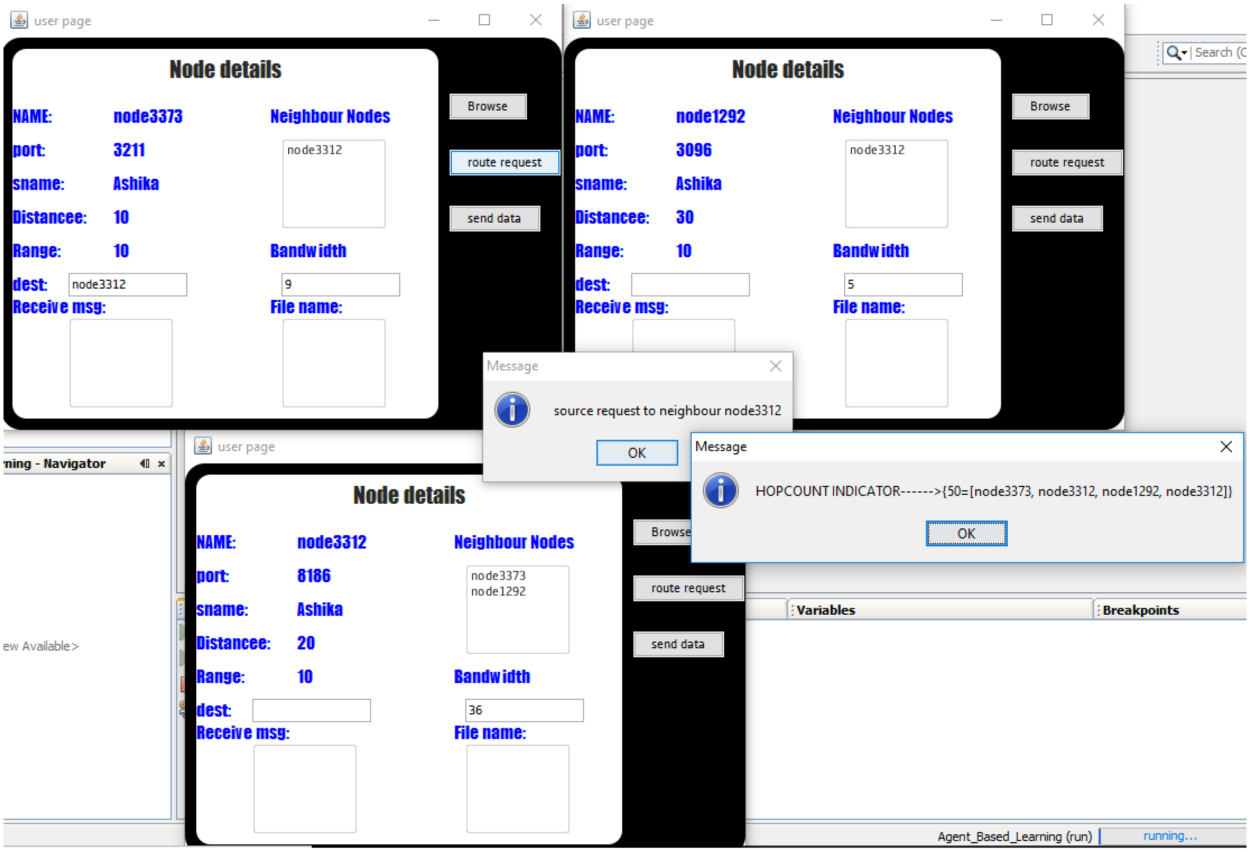
Neighbour Nodes: node3312

Bandwidth: 5

Figure 5.3(c): Neighbour Discovery module

5.2.3 Path Discovery Module

This module involves in the main functioning of the project where all possible paths between source to destination is formed. The source starts finding all possible paths from source to destination through all intermediate nodes also. Once all the possible paths are being formed from source to destination, all the paths which are possible from source to destination information is being sent back to the source so that it can be used later from the source to select as to which path it can take to transfer the data. This module also plays a very important role in the project because this is the one which finds all the possible paths from source to destination based on the underlying algorithm. This module is analogous to Onlooker Bee phase and Scout Bee phase in the native Artificial Bee Colony algorithm. According to the native Artificial Bee Colony algorithm, in the Onlooker bee phase, the onlooker bees select those food sources that are been identified by the Employee bees and estimate their nectar quantity. Similarly the Scout Bee phase in the native Artificial Bee Colony algorithm will come into picture when there is no food source detected and to randomly search for food sources. Taking this analogy in this project, when the destination node is a direct neighbour of the source node, then Onlooker phase is implied, and if the source and destination has any intermediate nodes, then Scout bee phase come into the picture. Thus the Onlooker Bee phase and Scout Bee phase is been implemented in this module.



user page

Node details

NAME: node3373
port: 3211
sname: Ashika
Distancee: 10
Range: 10
dest: node3312
Receive msg:
Neighbour Nodes: node3312
Bandwidth: 9
File name:

user page

Node details

NAME: node1292
port: 3096
sname: Ashika
Distancee: 30
Range: 10
dest:
Receive msg:
Neighbour Nodes: node3312
Bandwidth: 5
File name:

user page

Node details

NAME: node3312
port: 8186
sname: Ashika
Distancee: 20
Range: 10
dest:
Receive msg:
Neighbour Nodes: node3373
node1292
Bandwidth: 36
File name:

Message

source request to neighbour node3312

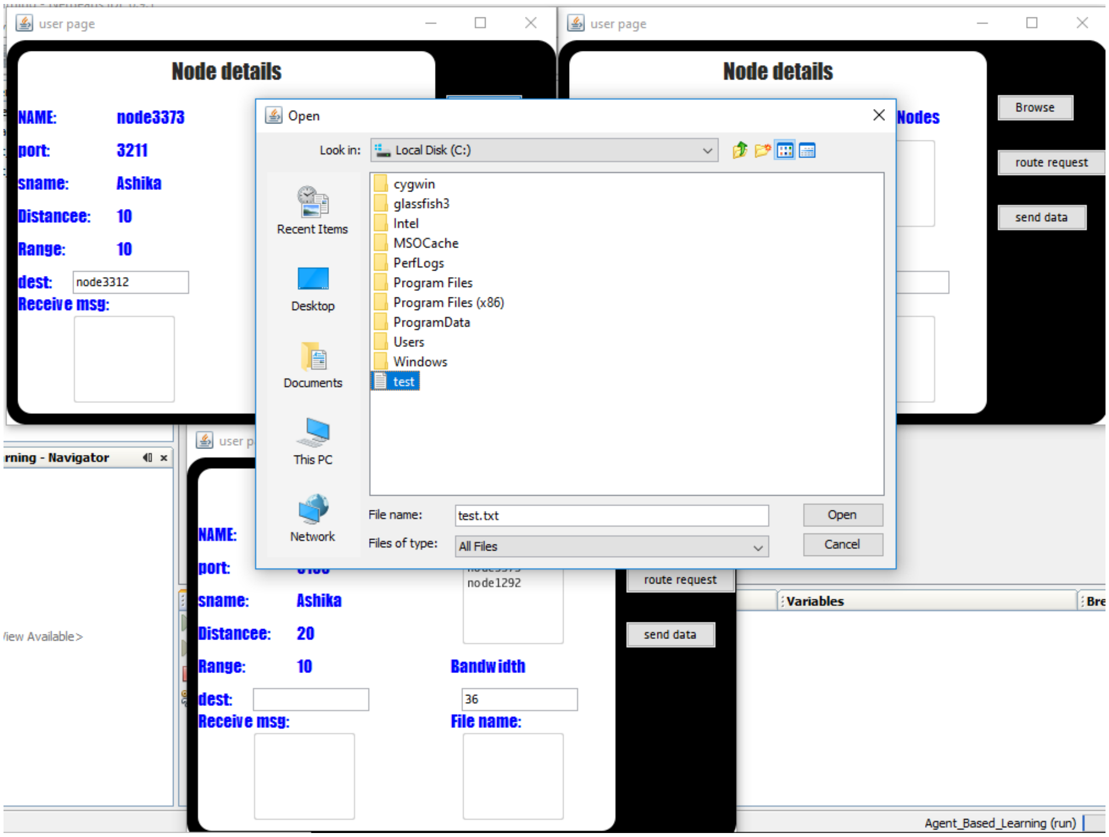
Message

HOPCOUNT INDICATOR------>{50=[node3373, node3312, node1292, node3312]}

Figure 5.4: Route Request module

5.2.4 Data Transfer Module

This is the final module of the project which involves the data transfer from the source to destination node. The source and the destination are input by the user. The destination node number is entered in the source node dialog box. The user then selects a file to be transferred to the destination and then selects weather the data has to be sent with highest bandwidth or lowest delay. The most optimal route is selected from among all the possible paths. The data is then routed to the destination using the multi socket service. The file can be viewed only by the destination and not any of the intermediate nodes. In case the source and destination are not connected, a pop up is displayed. The user is prompted to enter another connected destination and then routing takes place.



user page

Node details

NAME: node3373
port: 3211
sname: Ashika
Distancee: 10
Range: 10
dest: node3312
Receive msg:

Open

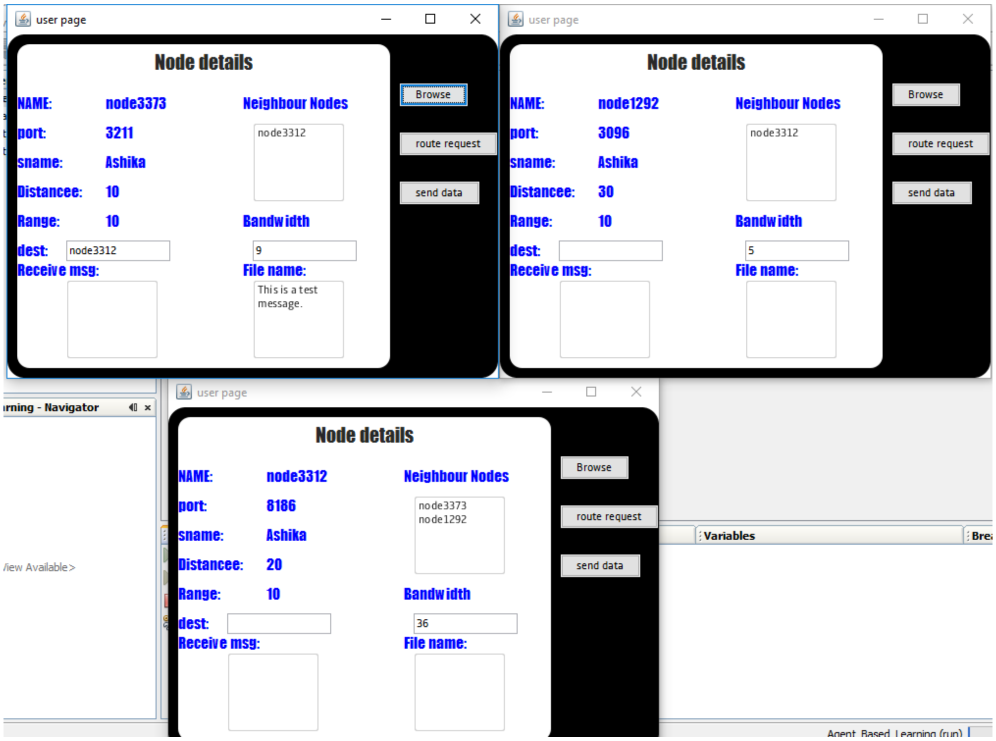
Look in: Local Disk (C:)

cygwin
glassfish3
Intel
MSOCache
PerfLogs
Program Files
Program Files (x86)
ProgramData
Users
Windows
test

File name: test.txt
Files of type: All Files

NAME:
port:
sname: Ashika
Distancee: 20
Range: 10
Bandwidth 36
dest:
Receive msg:
File name:
node1292

Figure 5.5: Browsing File to send data



user page

Node details

NAME: node3373

port: 3211

sname: Ashika

Distancee: 10

Range: 10

dest: node3312

Receive msg:

Neighbour Nodes: node3312

Bandwidth: 9

File name: This is a test message.

user page

Node details

NAME: node1292

port: 3096

sname: Ashika

Distancee: 30

Range: 10

dest:

Receive msg:

Neighbour Nodes: node3312

Bandwidth: 5

File name:

user page

Node details

NAME: node3312

port: 8186

sname: Ashika

Distancee: 20

Range: 10

dest:

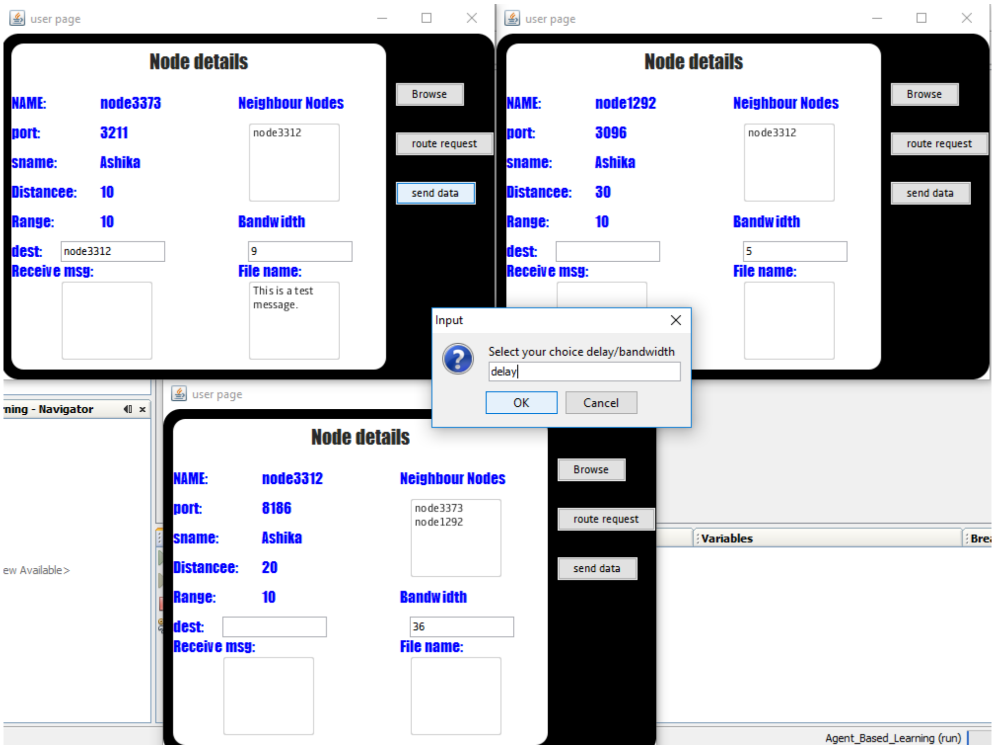
Receive msg:

Neighbour Nodes: node3373 node1292

Bandwidth: 36

File name:

Figure 5.6: Uploading File to send data



user page

Node details

NAME: node3373

port: 3211

sname: Ashika

Distancee: 10

Range: 10

dest: node3312

Receive msg:

Neighbour Nodes: node3312

Bandwidth: 9

File name: This is a test message.

user page

Node details

NAME: node1292

port: 3096

sname: Ashika

Distancee: 30

Range: 10

dest:

Receive msg:

Neighbour Nodes: node3312

Bandwidth: 5

File name:

Input

Select your choice delay/bandwidth

delay

user page

Node details

NAME: node3312

port: 8186

sname: Ashika

Distancee: 20

Range: 10

dest:

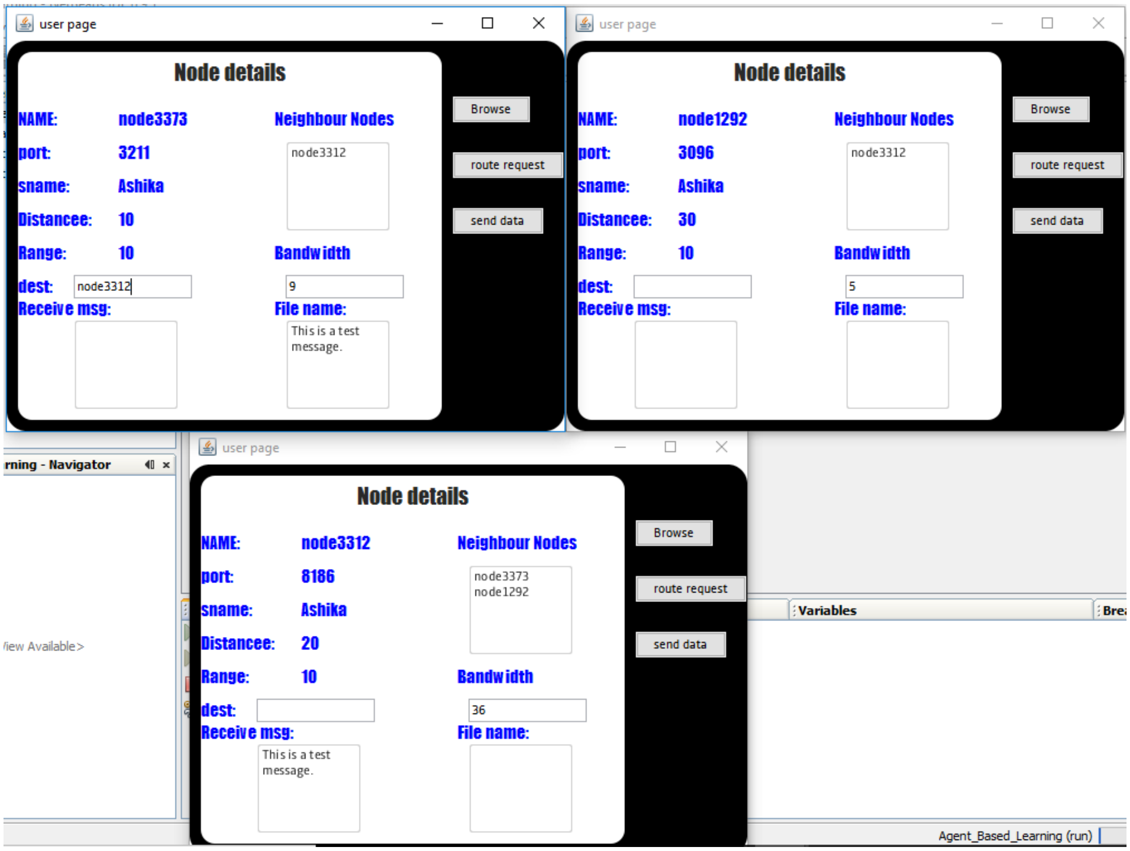
Receive msg:

Neighbour Nodes: node3373 node1292

Bandwidth: 36

File name:

Figure 5.7: User selecting bandwidth or delay to send data



Node details

NAME: node3373

port: 3211

sname: Ashika

Distancee: 10

Range: 10

dest: node3312

Receive msg:

Neighbour Nodes: node3312

Bandwidth: 9

File name: This is a test message.

Node details

NAME: node1292

port: 3096

sname: Ashika

Distancee: 30

Range: 10

dest:

Receive msg:

Neighbour Nodes: node3312

Bandwidth: 5

File name:

Node details

NAME: node3312

port: 8186

sname: Ashika

Distancee: 20

Range: 10

dest:

Receive msg: This is a test message.

Neighbour Nodes: node3373
node1292

Bandwidth: 36

File name:

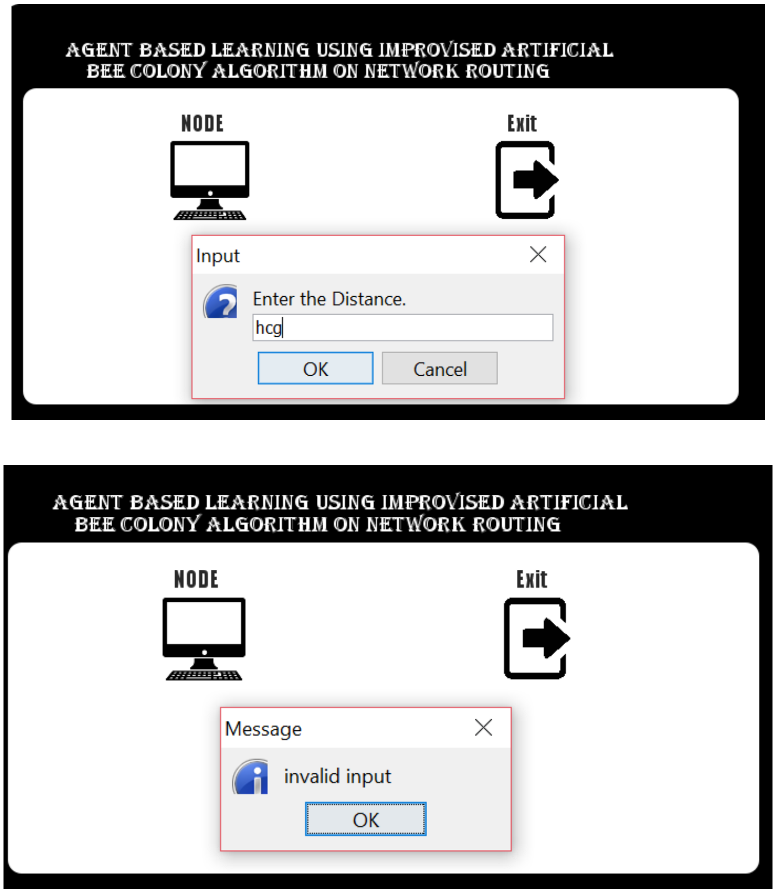
Figure 5.8: Data Transfer Module

Chapter 6:

Testing and Experimental Analysis

6.1 System Testing

Case 1:



AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL BEE COLONY ALGORITHM ON NETWORK ROUTING

Input

Enter the Distance.

hcg

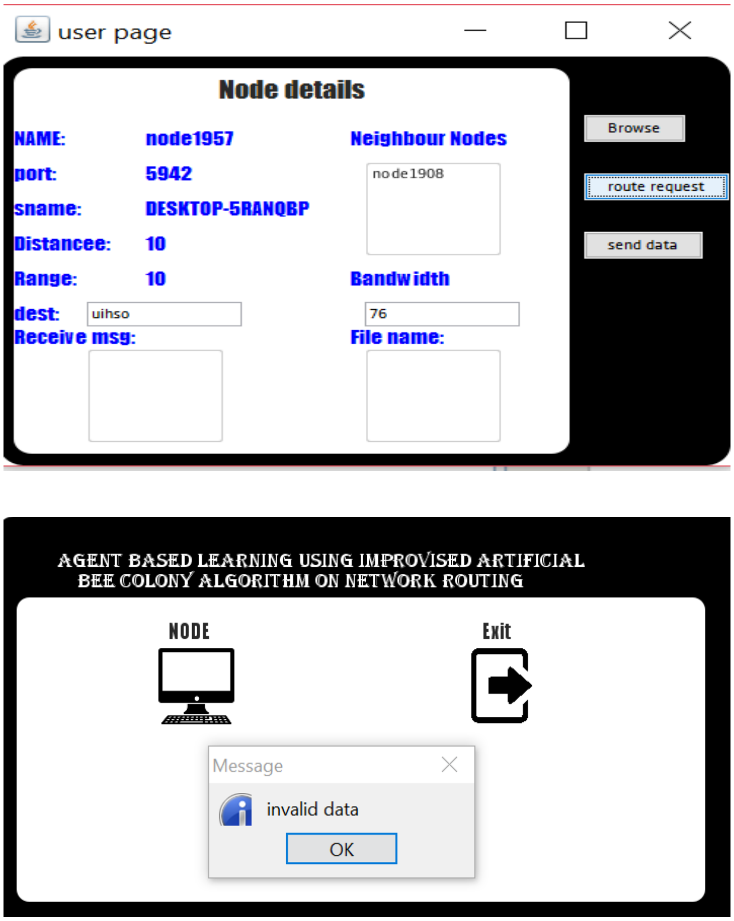
Message

invalid input

Figure 6.1: TestCase1

The input from the user for both distance and range are validated to be integers. In case of any other input, an alert is displayed.

Case 2:



user page

Node details

NAME: node1957

port: 5942

sname: DESKTOP-5RANQBP

Distancee: 10

Range: 10

dest: uihso

Receive msg:

Neighbour Nodes

node1908

Bandwidth

76

File name:

AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL BEE COLONY ALGORITHM ON NETWORK ROUTING

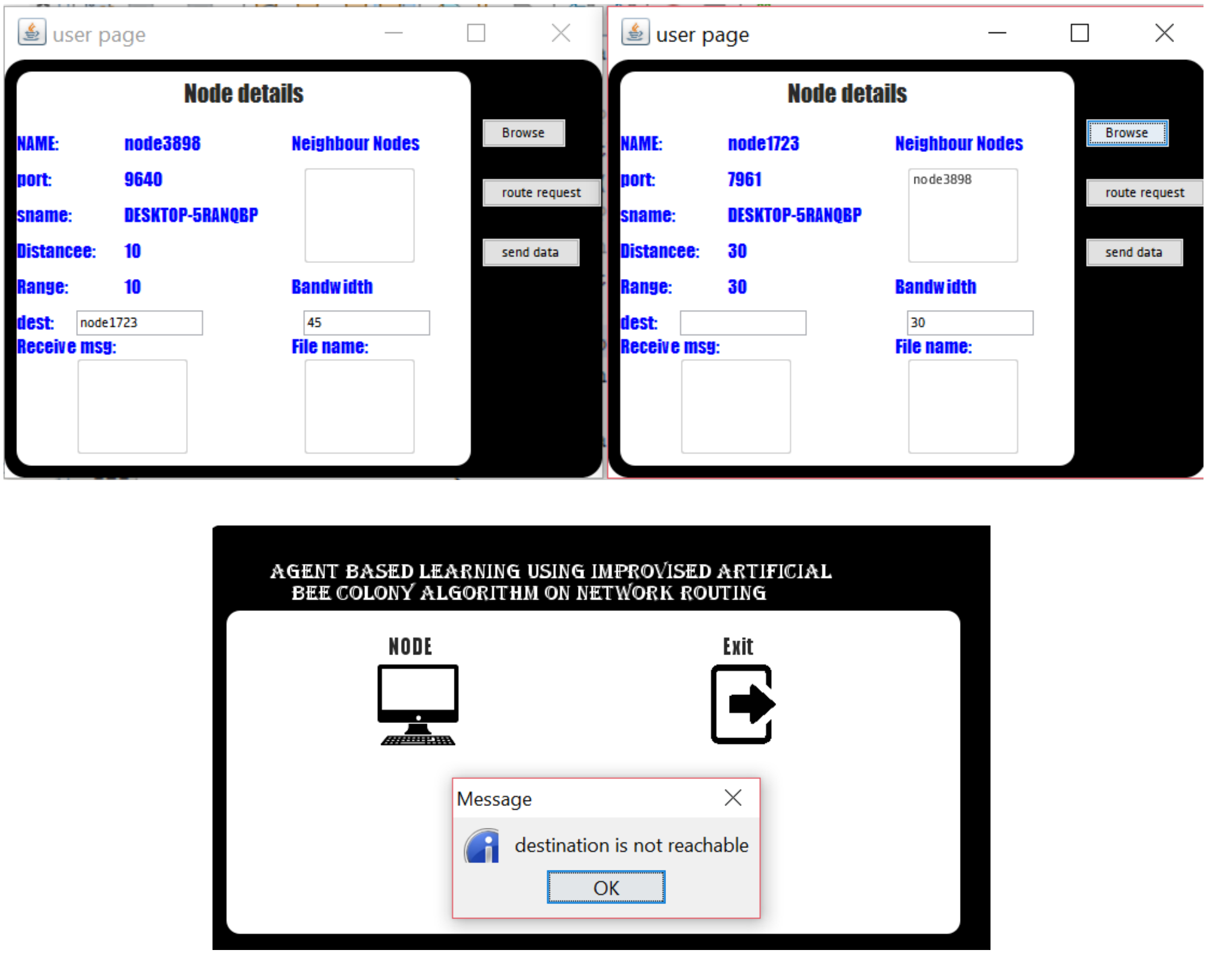
Message

invalid data

Figure 6.2: TestCase2

In this test case, it is checked that the user enters a valid destination node. On failure of this, invalid data is displayed and the user has to re enter the destination.

Case 3:



Node details

NAME: node3898

port: 9640

sname: DESKTOP-5RANQBP

Distancee: 10

Range: 10

dest: node1723

Receive msg:

Neighbour Nodes

Bandwidth 45

File name:

Node details

NAME: node1723

port: 7961

sname: DESKTOP-5RANQBP

Distancee: 30

Range: 30

dest:

Receive msg:

Neighbour Nodes node3898

Bandwidth 30

File name:

AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL BEE COLONY ALGORITHM ON NETWORK ROUTING

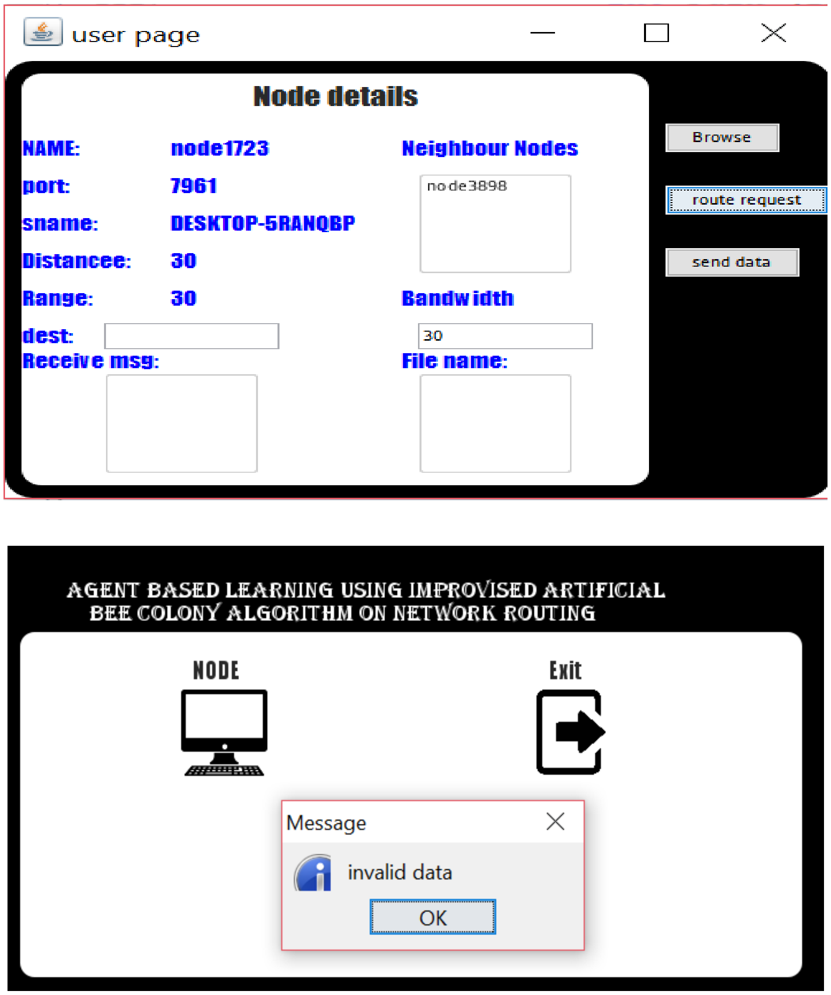
Message

destination is not reachable

Figure 6.3: TestCase3

This test case is used to handle the exception in the situation where no path exists between the source and the destination. In such a case, it is displayed that the destination is unreachable.

Case 4:



user page

Node details

NAME: node1723

port: 7961

sname: DESKTOP-5RANQBP

Distancee: 30

Range: 30

dest:

Receive msg:

Neighbour Nodes

node3898

Bandwidth

30

File name:

AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL
BEE COLONY ALGORITHM ON NETWORK ROUTING

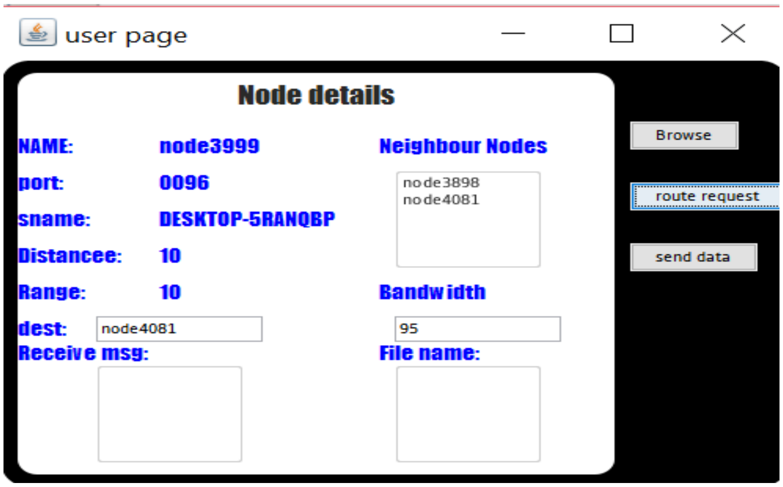
Message

invalid data

Figure 6.4: TestCase4

This test case handles the exception when no destination node is entered, the user tries to request for the path directly. An alert of invalid data is displayed.

Case 5:


user page

Node details

NAME: node3999

port: 0096

sname: DESKTOP-5RANQBP

Distancee: 10

Range: 10

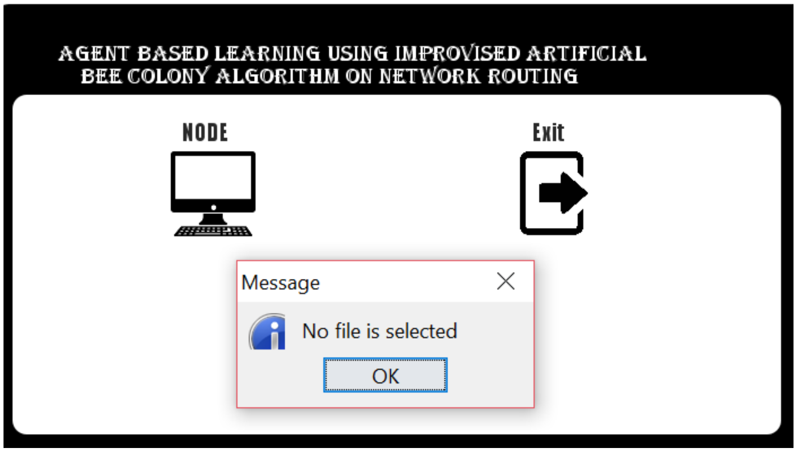
dest: node4081

Receive msg:

Neighbour Nodes: node3898 node4081

Bandwidth: 95

File name:


AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL BEE COLONY ALGORITHM ON NETWORK ROUTING

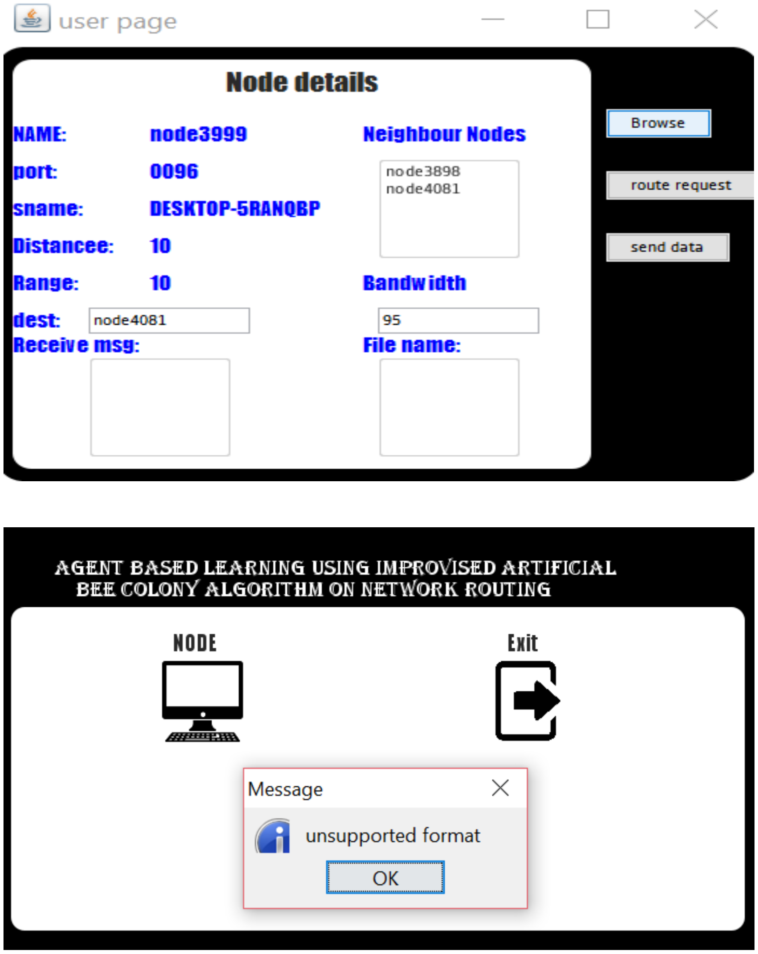
Message

No file is selected

Figure 6.5: TestCase5

This test case prompts the user to select a file when he/she tries to send data without making any prior selection.

Case 6:


user page
Node details
Browse
route request
send data
AGENT BASED LEARNING USING IMPROVISED ARTIFICIAL BEE COLONY ALGORITHM ON NETWORK ROUTING
NODE
Exit
OK

NAME: node3999

port: 0096

sname: DESKTOP-5RANQBP

Distancee: 10

Range: 10

dest: node4081

Receive msg:

Neighbour Nodes
node3898
node4081

Bandwidth 95

File name:

Message

unsupported format

Figure 6.6: TestCase6

In this network only .html, .java, .jsp, .txt files can be transferred. In case any other type of file is selected by the user, a pop up is displayed which reminds the user that it is not one of the supported formats.

6.2 Performance Analysis

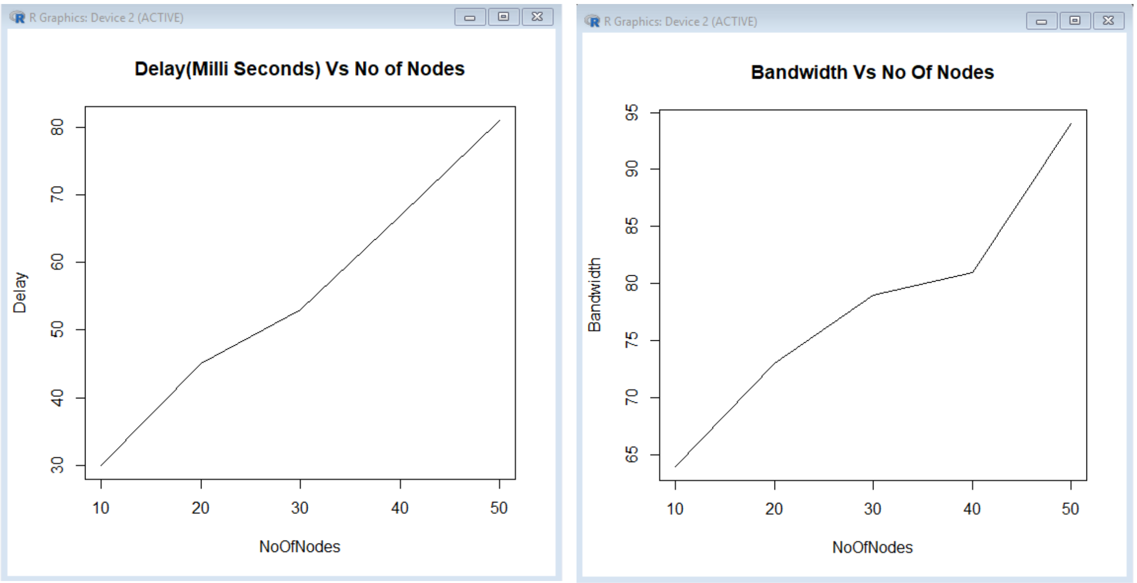


Figure 6.7: Performance analysis with delay and bandwidth respectively

Delay and Bandwidth:

The above line graphs depict the dependency of Delay and Bandwidth on the number of nodes in the routing network.

As we can see, both are directly proportional to number of nodes.

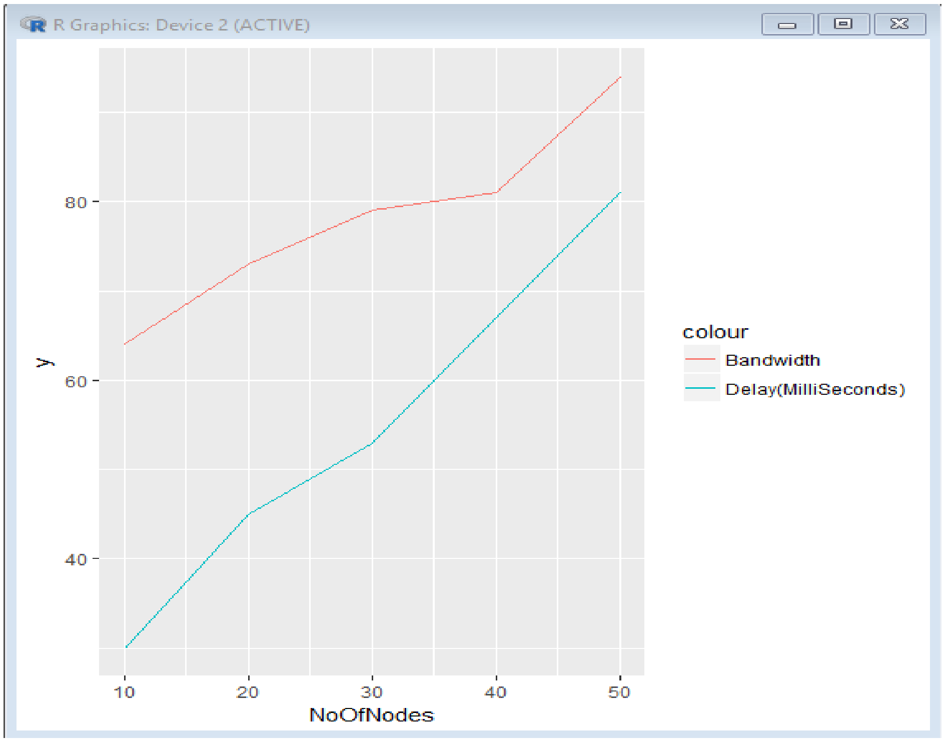


Figure 6.8: Performance analysis when delay and bandwidth are considered on same scale

The above line graph depicts the dependency of both bandwidth and delay on the number of nodes.

From the observation, we can say that both the parameters are directly proportional to the number of nodes. So in selection of an optimal path, higher bandwidth and lesser delay is preferred.

Chapter 7

Conclusion and Future Enhancements

7.1 Conclusion

The original ABC Algorithm performs the exploration process well, but is not so efficient in the exploitation of the resources. This is handled by the Artificial Bee Colony Algorithm with Information Learning. Information Learning on Artificial Bee Colony implements the multipopulation strategy by dividing the population into many subpopulations or regions. The learning within a subpopulation and between subpopulations is based on the QoS metrics of choice. This optimal solution can be obtained by implementing the learning prior to ABC algorithm where all the paths satisfying the threshold value defined by the user based on the QoS constraints are satisfied. This reduces the load on the ABC algorithm as few routes have already been eliminated based on the QoS constraint. So ABC algorithm is then run on fewer nodes to find the optimal paths in a lesser duration and in a more efficient manner.

The implementation of our proposed algorithm, Improvised ABC using agent based learning involves two main concepts: Multipopulation Strategy and QoS metric learning. The multipopulation strategy aids in dividing the network into regions so that local routing can be performed faster. Routing between regions is the next level of search that is executed. This eliminates a global search that may be performed even if the routing needed is local. The QoS metric taken under consideration here is the bandwidth. The total available bandwidth for a path is calculated simply by summation. The path with the maximum available bandwidth is chosen as the preferred path and the data packets are sent along that route. This is how native ABC is improvised by our proposed algorithm.

7.2 Future Enhancements

- The improvised algorithm proposed by us can be simulated using a network simulator for real time simulation.
- In this implementation, we are considering wireless ad hoc networks. The same routing algorithm can be extended to other types of networks with varying topologies.
- In our project we are considering the user specific metrics – bandwidth and delay. Another implementation could be focusing on optimizing the routing taking the network specific metrics – throughput, latency, jitter and power loss.

REFERENCES

[1] T. R. Gopalakrishnan Nair and Kavitha Sooda, “Performance Analysis in Direction Oriented Graded Cognitive Network Using Bayesian Model Approach for Path Determination,” vol. 45, January 2015.

[2] Ms. Neeta Nemade1 and Mr. Dhiraj Rane,” A Review on Bio-Inspired Computing Algorithms and Application, 2016, pp. 12-19.

[3] Dervis Karaboga and Bahriye Akay, “A comparative study of Artificial Bee Colony algorithm,” vol. 214,2009.

[4] Deniz Ustun and Ali Akdagli, “A Study On The Performance Of The Hybrid Optimization Method Based on Artificial Bee Colony And Differential Evolution Algorithms,”2017.

[5] Gao, W., Huang, L., Liu, S. and Dai, C, “Artificial Bee Colony Algorithm Based on Information Learning,” *IEEE Transactions on Cybernetics*,2015.

[6] Yuxia Bai,Yefa Mai and Dr. Nan Wang, “Performance Comparison and Evaluation of the Proactive and Reactive Routing Protocols for MANET,” pp. 187-195,2017.

[7] D. Querlioz, W. S. Zhao, P. Dollfus, J.-O. Klein and O. Bichler, C. Gamrat, “Bioinspired Networks with Nanoscale Memristive Devices that Combine the Unsupervised and Supervised Learning Approaches”,2013.

[8] Binitha S, S Siva Sathya,”A Survey of Bio inspired Optimization Algorithms”,2012.

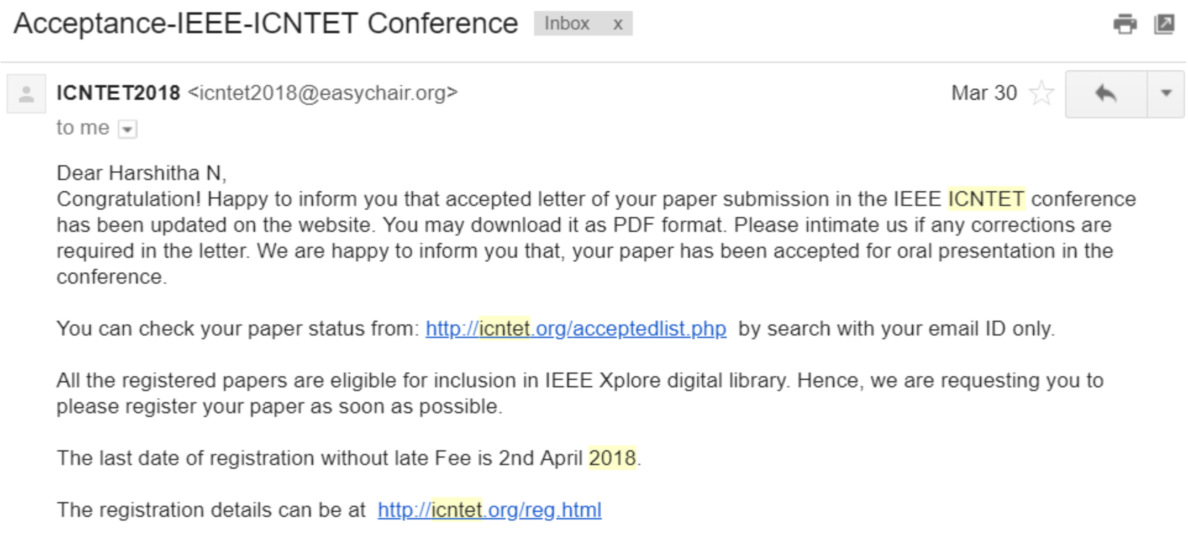
[9] He Zheng, Gang Yang, Junyan Yi, Zheng Yuan, Xiaowei Liu and Meng Shuai,”Simplify the Basic Artificial Bee Colony Algorithm”,2015.

[10] Rashmi A. Mahale, Prof.S.D.Chavan,” A Survey: Evolutionary and Swarm Based Bio-Inspired Optimization Algorithms”,2012.

[11] Weifeng Gao, Sanyang Liu, Lingling Huang, “A global best artificial bee colony algorithm for global optimization”, 2012.

[12] Vahid Zeighami, Reza Akbari, and Koorush Ziarati, “An Efficient Multi Population Artificial Bee Colony”, 2012.

APPENDIX A: Details of Publications



ICNTET2018 <icntet2018@easychair.org>

Mar 30

to me

Dear Harshitha N,
Congratulation! Happy to inform you that accepted letter of your paper submission in the IEEE ICNTET conference has been updated on the website. You may download it as PDF format. Please intimate us if any corrections are required in the letter. We are happy to inform you that, your paper has been accepted for oral presentation in the conference.

You can check your paper status from: http://icntet.org/acceptedlist.php by search with your email ID only.

All the registered papers are eligible for inclusion in IEEE Xplore digital library. Hence, we are requesting you to please register your paper as soon as possible.

The last date of registration without late Fee is 2nd April 2018.

The registration details can be at http://icntet.org/reg.html

Appendix A: Snapshot of acceptance mail from the publication for literature survey

The survey paper supporting this project has been accepted into the International Conference on New Trends in Engineering and Technology 2018 (ICNTET). The conference is to be held in GRT Institute of Engineering and Technology, Chennai on the 7th and 8th of September, 2018. The supporting document is attached above.