Development of a Bi-directed Routing Model for Mobile Agents in Distributed Systems

Oladejo Oluseyi O.
Department of Computer Science and Engineering, Obafemi Awolowo University, Ile- Ife, Nigeria

Akinyemi Bodunde O.
Department of Computer Science and Engineering, Obafemi Awolowo University, Ile- Ife, Nigeria

Amoo Adekemi O.
Department of Computer Science and Engineering, Obafemi Awolowo University, Ile- Ife, Nigeria

ABSTRACT

Routing in a distributed system is a critical issue due to its complexity. Existing routing model (undirected) for mobile agent lacks provable efficient algorithm which consumes response time and bandwidth. This study planned, simulated and accessed a best migration scheme for mobile agents dynamic routing in a distributed computing environment. This was with a perspective to eradicating issues such as congestion, time consumption and packet loss connected with mobile agent framework method of handling distributed systems. The routing model was exhibited as a Bi-directed Chinese Postman Problem (BCPP). Linear Optimization was used to provide all conceivable itineraries and to calculate the optimal path for mobile agents. Backtracking algorithm was engaged to decide the availability of the link or node. Simulation of the model was carried out for a single mobile agent migration. MATLAB Simulink tools were used to simulate the formulated model. The proposed routing model performance was benchmarked with the existing Chinese Postman Problem (CPP) routing scheme using Network bandwidth, Response Time and the Network throughput as performance parameters. The simulation result showed that the proposed model optimizes network bandwidth, minimizes the network latency and throughput, thereby improved the mobility properties of mobile agents in distributed systems. It was concluded that the proposed migration scheme for mobile agents routing in a network reduced congestion, minimized time and expense of resources as compared to CPP.

General Terms

Mobile agents, Distributed System, Routing

Keywords

Chinese Postman Problem, Bi-directed Chinese Postman Problem

1. INTRODUCTION

The word distributed in terms such as distributed system, distributed programming, and distributed algorithm originally referred to computer networks where individual computers were physically distributed within some geographical area [1]. Nowadays, the terms are used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing [2].

Distributed systems consist of accumulation of self-directed, loosely coupled computers, connected through a network and distribution middleware, which endows computers to organise their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility. The use of distributed systems to solve computational problems is referred to as distributed computing. Computer network is very vital to the existence of a distributed system.

A computer network comprises nodes and communication links which implement the distributed network environment. A computer network was defined as an aggregate of two or more autonomous computers that are separated by physical distances and connected together [3]. The growing complexity of the computer networks give rise to faults and performance inefficiencies.

While there is no single definition of a distributed system, the following defining properties are commonly used:

(i) They are several autonomous computational entities, each of which has its own local memory [4].

(ii) The entities communicate with each other by message passing [2].

A distributed system may have a common goal, such as solving a large computational problem [5]. Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users [4]. Other typical properties of distributed systems include the following:

(i) The system has to tolerate failures in individual computers [5].

(ii) The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program [4].

(iii) Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input [5].

Various routing schemes and algorithms had been assumed to transfer data and messages between nodes in a distributed system and to manage the traffic in the network.

One of the efficient routing schemes is the application of mobile agents in managing and optimizing the network [6].

Mobile agents are programs that can migrate from host to host with the intent of fulfilling a task on behalf of a user. Various

researchers have effectively used mobile agent-based routing scheme for network management. Software agents were used by these researchers in finding and maintaining routes. Mobile agents' implementation in a network-based application devolves and regionalizes the network management roles and tactically carries out administrative tasks thereby improving the reliability and quality of service. They also reduce the traffic need for the management. This is achievable by mobile agents' generation on a repeated basis and relegation to precise destination on the network. Mobile agents replace and outperform the Remote Procedure Call (RPC) interaction between the clients and servers. They travel between the various nodes on the network and autonomously carry out administration task.

The summary report for the agent's trip is sent back to the source node, as soon as it has completed its tasks. The network traffic generated by mobile agents is very light, since there are very few communications between the agent and the source node during the process of searching. Therefore, mobile agent is an effective way for improving network performance [7]. One important thing to note is that network must be flexible enough to allow these software agents to communicate with one another. The mobile agent routing assesses different nodes on the network by determining the best route through the usage of different paradigms such as Onion algorithm, Ant algorithm etc.

The duty of a routing scheme and algorithm is to move the messages along the network edges to their final destination. There are two types of routing algorithm: static and dynamic version. The static routing algorithms determine the path which the messages/ packets will take before the actual start of routing (i.e. the preprocessing phase) while the dynamic routing algorithm allows addition and deletion of nodes and/or links in order to model the node/link failure and network growing.

Modern computer and telecommunication networks are able to respond quickly to randomly fluctuating demand and failure by rerouting traffic and by allocating resources [8]. In the past, network application has been developed on centralized client server based network management, such network application faced a lot of problems such as low degree of flexibility, unreliability, lack of scalability, etc., as the network becomes more geographically spread. Currently heterogeneity of networks has gained a significant approval not only in the network management but also in diverse areas of computer science such as software engineering and human interface.

Managing the traffic in a distributed network environment generates a big problem in controlling and scheduling, since the network management activities for these heterogeneous networks involve gathering and analyzing massive amount of data from the network [9]. Several researchers have worked on how to develop routing protocols and schemes that can efficiently find routes between two communication nodes using different approaches in searching for a new shortest route or modifying the existing route. The intelligence of mobile agents has been applied to proactively make dynamic decision such as detecting link/path failure, finding the next and shortest route to the destination, and then optimizing the travel plan, as the agents travel around the network. Hence, in this work, attempt was made to develop an enhanced migration scheme for mobile agent dynamic routing in a distributed computing environment with a view to eliminate

problem such as congestion, time consumption and packet loss associated with mobile agent system approach of managing the network.

2. RELATED WORKS ON MOBILE AGENT ROUTING

There has been quite a good amount of work in mobile agent routing regarding the migration schemes. In [10], a mathematical model for the performance analysis of the mobile agent system using parameters derived from their Mole agent system was proposed. They used this model to study an abstract application with a single mobile agent that may visit several servers and also validated the model by using its results in a small experimental application.

In [7], the mobile agents' performance for network management based on their behavioral analysis was examined. An overall agent based routing model was put forward by concentrating on the behaviour analysis of the mobile agents used in the network routing. This was conducted based on the response of mobile agents to system failure i.e. mobile agents with Weak Reaction Capability (MWRC) and mobile agents with Strong Reaction Capability (MSRC).The probability of success that an agent will find the destination was examined as well as the population distribution of the mobile agents (i.e. the number of the mobile agents) for the two cases by developing a mathematical model. It was confirmed that the probability of success and the number of agents for MSRC are more than those of MWRC. Accordingly, it was stated that a little number of mobile agents and a good probability of finding the destination were attained by choosing an optimal number of mobile agents created for each request and giving them optimal life-span limit. The structure presented in [7] was implanted into the mobile agents, thereby diminishing the computation overhead. However, only behavioural analysis of the mobile agents cannot decide their performance in network management. The network topology might not be known beforehand.

Determination or discovery of network topology is imperative due to the rapid changes in the network. Lack of scalability is therefore inevitable. The scheme lacks reusability of the routing agents since they are given optimal life-span limit. The scheme is also unreliable since it is based on response of mobile agents to a system failure.

Likewise, an anonymous networking approach based on mobile agents was developed by [1]. It was presented by the authors, a first- generation prototype of an ad-hoc onion routing network which ensures data protection against traffic analysis for mobile and/or multi agent systems. Onion routers are referred to as computer programs that perform routing for the application-layer of the network. Anonymous connections and communication is usually allowed between the application mobile agents. It was developed on the platform of JADE multi agent which runs as the onion agents migrate across the network to different network nodes, so as to maximize the number of agents on the network, making more effective the anonymity communication. The major advantage of this routing approach is that it provides a private communication services.

In [11], the authors put forward a refreshingly new routing scheme for mobile agents in mobile ad hoc networks with a view to overcoming the flaws and defects of on-demand routing protocols in networks. In [11], both the strength of the

on-demand routing of Ad-hoc on demand Distance Vector (AODV) routing protocol and the distributed topology discovery mechanism of Ant based algorithms was combined together. It was revealed that the hybrid protocol of Ant-AODV is able to bring about a reduced end–to-end delay and high connectivity as compared to the individual AODV and Ant based algorithms. It was clearly seen in the result that there was huge reduction in latency as well as in the number of the route discovery of the mobile agents by the application of Ant-AODV hybrid routing protocols. In spite of this, routing delay was still inherent; because of the major flaw of Ant algorithms that provides topology information to nodes. The efficient performance of Ant algorithm is not guaranteed in an instance of dynamic topology and when the lifetime of mobile agent is small.

Furthermore, a relative study was carried out by [12] on Mobile Agent Based Routing scheme (MAR) and Routing Information Protocol (RIP) scheme. In this study, the usefulness and the disadvantages of some best effort schemes using Dijkstra (link state routing) or Bellman-ford (distance vector routing) algorithms were presented. A best effort mobile agent-based routing scheme which employed the routing procedure (next hop routing) that is similar to RIP was implemented in MAR, each node in a network sends routing mobile agents to all its neighbour node destination to discovered the routes, the mobile agents track a path towards the specified destination through a set of intermediate nodes by updating the routing information at each of the node. Simulation was carried out in RIP and MAR using some parameter like delay. The result showed that MAR scheme has better performance than RIP scheme. MAR scheme displayed less communication overheads and less convergence time. It equally showed great scalability, flexibility, and adaptability. A major drawback of this scheme is its failure in making provision for wandering agents that could congest the network traffic. Computational overhead is another disadvantage of this scheme because of the increase in the number of mobile agent generated on each node in the network.

Also in [13], an optimal migration scheme for mobile agent routing in a data communication network was presented. The routing model was presented as a Chinese Postman Problem (CPP). Combinatorial Optimization was applied to provide all possible itineraries and to compute an optimal path for mobile agents. Look-ahead algorithm was used to determine the link or node availability. The results showed that the proposed model minimizes the network latency and the throughput and thereby improves the mobility properties of mobile agents in a data communication network. Combinatorial algorithm used is not likely to give optimal solution because the time complexity is also exponential. It also belongs to the class of NP-hard problems that is computationally impractical.

In this research, attempt was made to develop a bi-directed routing model in distributed systems to improve migration scheme for mobile agents. This will help in addressing the problem of traffic deadlock frequently faced by the mobile agents in a network and the update overhead problems of the routes due to dynamism and intermittent failure of the network.

3. DESCRIPTION OF THE MODEL

The proposed model of the migration scheme for mobile agents in a distributed system network is shown in Figure 1. The instantiation of the network by the administrator is the first stage of the model segment. The essence of this stage is to state explicitly the source node as well as the destination node of the mobile agents. The second stage is the precise route detection stage. This stage is very necessary, since mobile agents from time to time require travelling to numerous hosts to carry out their duties; the mobile agent, without fail, will need to know all the possible itineraries that are available for migration. The itinerary is then best determined at the time the agent is being instantiated. The third stage is the inherent route detection stage. Mobile agent uses this stage to know the best path in terms of minimal delay, minimal cost and minimal resource consumption during the routing process. Migration procedure was addressed at the fourth segment while the fifth segment explains the response of the mobile agent in an event of unexpected network failure.

3.1 Mobile Agent Route Planning

Network optimization is a problem that lies in the middle of the great divide that separates the two major types of optimization problems, continuous and discrete. The migration plan for mobile agent routing in a network is a network optimization problem. Network optimization problems such as shortest path, assignment, max-flow, transportation, and transshipment, are undoubtedly the most common optimization problems in practice. Optimization is the process of distributing limited resources to the best possible effect. In network optimization problems, it is required to optimize a certain objective function (minimize cost or maximize output) subject to some constraints. It requires finding the optimum solutions out of all feasible solution that satisfies the constraint equations. All network optimization problem of interest are linear optimization problems in which both the objective function and the constraint equations are linear.

In this research, the network optimization technique is required to minimize the consumption of network resources such as bandwidth subject to capacity and demand constraints. Efficient routing of mobile agent in a network is modeled as an Arc routing problem being one of the well-known problems in linear optimization. The main objective of an arc routing problem is to minimize the sum of the cost of traversing on required arcs. Generally speaking, one seeks the cycle that covers all arcs in required arcs R with a minimum associated cost. In order to formulate the network optimization problem for mobile agent routing in a distributed system network, Bi-directed Chinese Postman Problem (BCPP) approach was used.

3.2 Formulation of Bi-Directed Chinese Postman Problem

A bi-directed graph $G = (V, E)$ is a pair $G = (V,E)$, where $V = \{1, . . . , d\}$ is a set of nodes, and E is a set of edges defined by two-element subsets of V . Two nodes u and v are *adjacent* or neighbours, if u,v is an edge of G and in this case the edge is drawn as bi-directed, $u \longleftrightarrow v$. Two edges are adjacent if they have an end node in common. A *path* from a node u to a node v is a sequence of adjacent edges connecting u and v for which the corresponding sequence of nodes contains no repetitions. The nodes u and v are called the *endpoints* of the path and all the other nodes are called the *inner nodes*. Bi-directed graphs are essentially undirected graphs with edges represented by bidirected arrows instead of full lines. A bidirected graph is *connected* if its underlying undirected graph is connected. A bidirected graph $G = (V, E)$ has an associated *incidence matrix* $I^G : V \times E \rightarrow \{-2, -1, 0, 1, 2\}$. If

the endpoints are distinct, the edge is called a *link*; otherwise, it is a *loop*. Additionally, two orientations are assigned to each
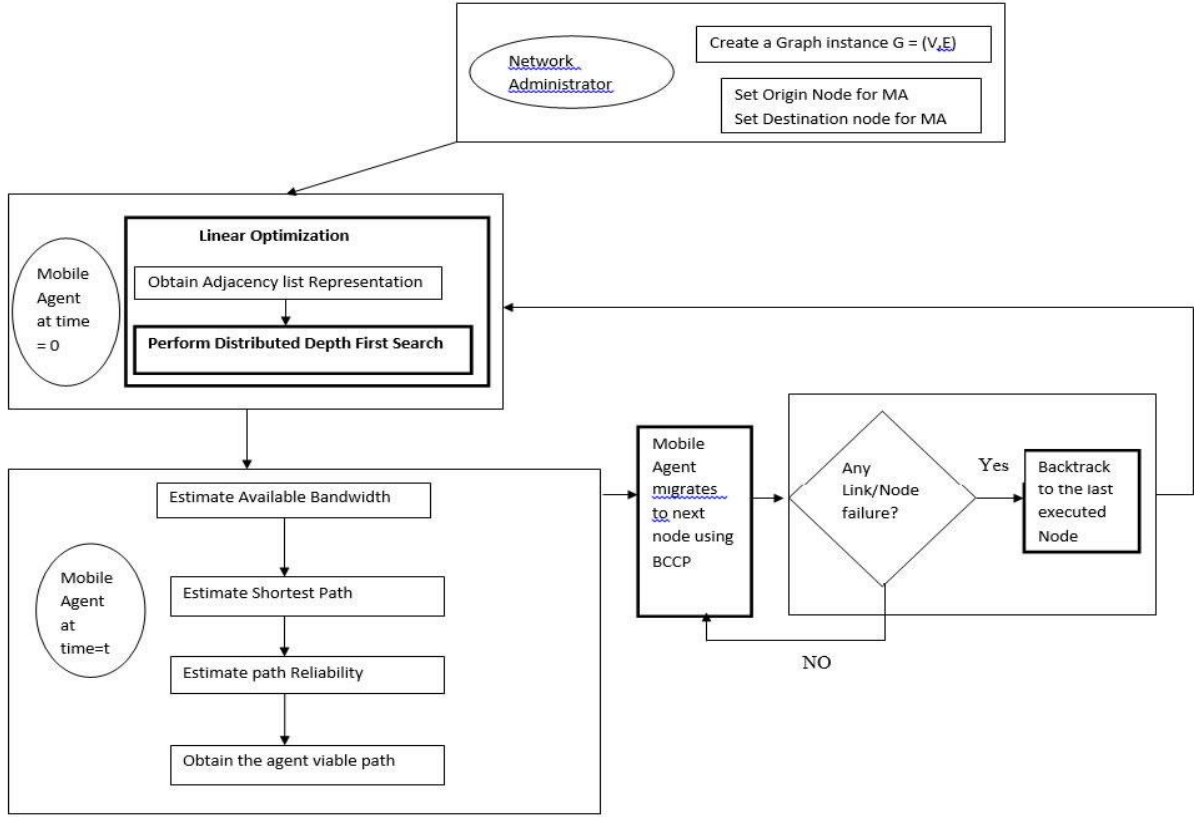

Figure 1: Description of the Model

of the edges, one with respect to each of its endpoints. There are two kinds of orientations—positive and negative—and thus it is said an edge is *positive-incident* or *negative-incident* to an endpoint. If an edge e is not incident to a vertex x, then $I(x, e) = 0$. For a link e and a vertex x, $I(x, e) = +1$ if e is positive-incident to x, and $I(x, e) = -1$ if e is negative-incident to x. For a loop e and a vertex x, $I(x, e)$ has the value of $+2$ if e is twice positive-incident to x, -2 if e is twice negative-incident to x, and 0 otherwise. The *in-degree* of a vertex x in graph G is defined as

$$deg_G^-(x) = - \sum_{\{e \in E \,|\, I(x,e)<0\}} I(x, e) \qquad (1)$$

Similarly, the *out-degree* is defined as

$$deg_G^+(x) = \sum_{\{e \in E \,|\, I(x,e)<0\}} I(x, e) \qquad (2)$$

Let

$$bal^G(x) = deg_G^+(x) - deg_G^-(x) = \sum I(x, e) \qquad (3)$$

Given a weighted bi-directed graph G, CPP is to find a minimum weight circuit of G, or report that one does not exist. CPP is polynomially time solvable on both undirected and directed graphs. For undirected graphs, CPP is reducible to minimum cost perfect matching. For directed graphs, it is reducible to minimum cost network flow. It becomes NP-hard on mixed graphs, which are graphs with both directed and undirected edges.

To find a Chinese postman circuit, the case G is Eulerian is first considered. Since an Eulerian circuit of G is has the smallest possible weight of any circuit, it is also a Chinese postman circuit. In the general case, however, when G is not Eulerian, our approach is to make the graph Eulerian by duplicating some of the edges, and then using a standard algorithm to find an Eulerian circuit. The general formulation of the Bi-directed Chinese Postman Problem as described in the literatures is presented as follows

$$\text{Min } Y = \sum w(e)f(e) \qquad (4)$$

Subject to:

$$\sum_e I^G(x,e)f(e) + bal^G(x) = 0 \text{ for each vertex } x \qquad (5)$$

$$f(e) \ge 0 \qquad \text{for each edge } e \qquad (6)$$

where:

w(e)　=　the weight function of *G*

f(e)　=　the number of additional copies of edge *e*

I^G(x,e)　=　Incidence flow balance

bal^G(x)　=　*balance* at each vertex, x

The objective function which is the minimization of the distance required to travel through the network and cover every edge at least once is described in expression (4), while continuity of flow requirement for the network is described by expression (5); the number of edges traveled going into any

node must equal the number going out, while the non-negativity restriction is described in expression (6)

This Bi-directed Chinese Postman model concentrates exclusively on the shortest path. The recurrent theme in routing problems is to minimize the total distance traveled in describing an edge covering tour i.e. in traversing every edge in the network at least once.

In this research, Bi-directed Chinese postman model shifted attention to the amount of bandwidth usage in choosing the best path for the mobile agent.

3.3 Computation of All Possible Paths in a Network

Mobile agents often needs to travel to multiple hosts to perform their tasks, however, it is important to determine the itinerary at the time the agent is designed or instantiated, because the topology of the network might not be known. Route determination will specify the order in which the mobile agent visits the hosts. Mobile agents' itinerary can be determined either statically or dynamically. That is, it can be calculated either before the agent is dispatched or while the agent is migrating. Dynamic itinerary planning is more flexible, and can adapt to environmental changing in real time.

A distributed system needs to carry traffic where the network has different links of different capacity (bandwidth); the traffic may be routed via different paths to the destination. Broadly speaking, for communication to occur, there must be a source and a link. Given a source node, a mobile agent is required to deliver a packet to another node on the network. There are several links connecting the source end and the receiver end.

The concept of multipath routing is employed to give the source node a choice at any given time of multiple paths to a particular destination by taking advantage of the bandwidth constraints and connectivity of the underlying network. Application of multipath routing improves the performance of the mobile routing in a data communication network by providing the assurance that a message generated at one place in the network can actually be routed to the intended destination.

Multipath routing ensures that when the primary (selected) path fails, an alternate one will be used to salvage the packet. In this case, the multiple paths are not used simultaneously. The traffic takes one of the multiple paths at a time. Other paths are kept as backup in case the used one is broken. When the selected path is broken, a new path discovery procedure is initiated, unlike in a single path routing case, when a node fails to transmit a packet to its next hop, a route error message will be sent back to the source indicating the breakage of the path. With multiple alternate paths available, nodes can actively salvage the packet by sending it to an alternate path; a route error will occur only when all the available paths fail. Therefore, by keeping multiple paths to a destination, the occurrence of route error is reduced.

Mobile agent routing is an ordering problem i.e. it must be prescribed in a sequential manner, in which one step has to be completed before the commencement of another. In lieu of this, the mobile agent will need to know all the possible itineraries that are available for migration. Therefore, the

selection of paths was guided in this research by minimizing network resources usage as optimization criteria.

3.4 Error detection and recovery

Error detection and recovery is also a characteristic that a mobile agent must possess. In a dynamic network, where node or links can be added or removed spontaneously, mobile agent must be able to detect the link/ node failure and re-route the network. Any routing algorithm that requires routers to know about every single destination in a network becomes infeasible as the network grows. The storage requirements, communication and update overhead will be costly.

In this research, Backtracking algorithm was used to determine the link or node availability. The mobile agent will perform a forward checking, to detect a failure, if it exist, the mobile agent will backtrack to the last node it just executed, if not, it launches it operations. It ensures that when the primary (selected) path fails, a new path discovery procedure is initiated, and alternate one will be used to salvage the packet.

This feature makes the mobile agent to be fault- tolerant. Backtracking is a form of recursion for every perform connectivity.

3.5 Migration Scheme for Mobile Agent Routing

The migration scheme model can then be formulated as follows:

$$\text{Min } Y = \sum w(e)f(e)B_{ij}P_{ij} \qquad (7)$$

Subject to:

$$\sum_e I^G(x,e) \, f(e) + bal^G(x) = 0 \qquad \text{for each vertex x} \quad (8)$$

$$f(e) \geq 0 \qquad \text{for each edge e} \qquad (9)$$

where:

$w(e) = $ the weight function of G

$f(e) = $ the number of additional copies of edge e

$I^G(x,e) = $ Incidence flow balance

$bal^G(x) = $ *balance* at each vertex, x

$B_{ij} = $ the available bandwidth between node i and node j

P_{ij}, i,j = 1,2,…,n is the zero-one variable that determines whether the connection between node i and node j exists (Value 1) or not (value 2)

Expression (7) is the objective function which is the minimization of the distance required to travel through the network and cover every edge at least once.

Expression (8) expressed the continuity of flow requirement for the network; the number of edges traveled going into any node must equal the number going out.

Expression (9) is the non-negativity restriction.

4. RESULTS AND DISCUSSION

The Model simulation is carried out for a single mobile agent migration on a distributed system network. Simulink tools were used to simulate the model formulated. The topology construction of the simulation is shown in Figure 2.
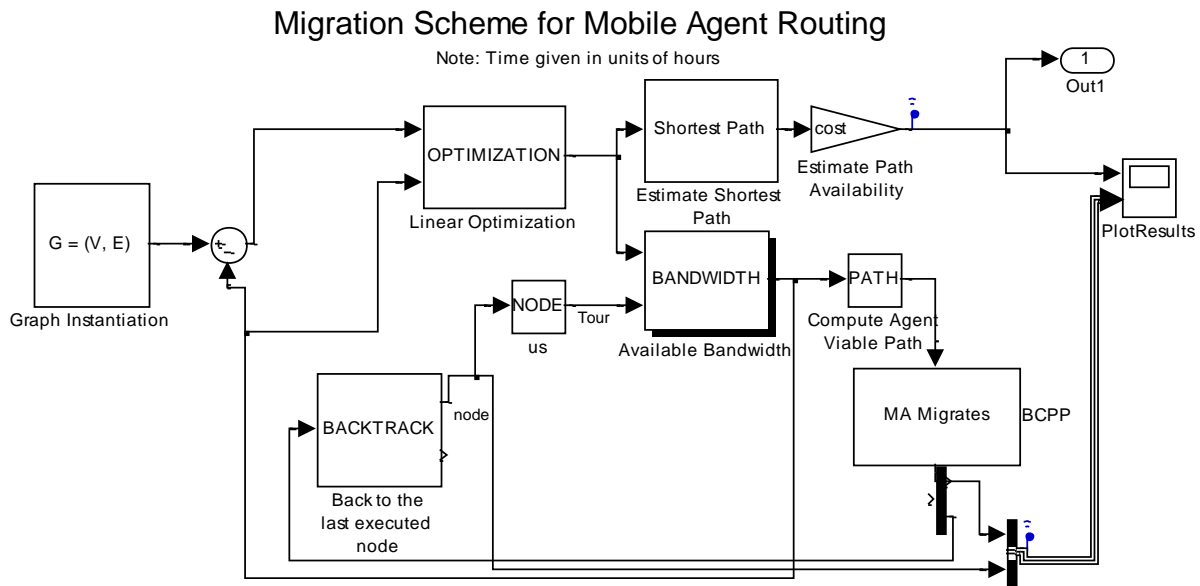
Figure 2: Simulation Topology

4.1 The graph Output

Simulink includes an extremely rich environment allowing users at several levels to customize the kinds of information that can be extracted from simulations. One major way of displaying simulation results is through the use of Scope block. The scope generates and displays the graph output of any simulation result. The scope is an important block of the Simulink library browser which belongs to the Sink category.

The discrete output against time can also be viewed in MATLAB by connecting the Outport block of the Sink category to the link that connects to the Scope block. The Outport provides an output port for a subsystem or model. The 'Output when disabled' and 'Initial output' parameters only apply to conditionally executed subsystems. When a conditionally executed subsystem is disabled, the output is either held at its last value or set to the 'Initial output'

4.2 Assessment Measures used

The network performance of the simulation program that was developed for the proposed scheme i.e. Bi-directed Chinese Postman Problem (BCPP) scheme was compared with the performance of one of the existing scheme i.e. the Undirected Chinese Postman Problem or simply put Chinese Postman Problem (CPP) scheme. For the purpose of performance comparison, the following performance metrics were used:

(a) Network Bandwidth against the number of nodes on network using the CPP scheme and BCCP scheme

(b) Response Time against the number of nodes on network using the CPP scheme and BCPP scheme

(c) Network throughput rate against the number of nodes on network using the CPP scheme and BCPP scheme

4.2.1 Network bandwidth

The network bandwidth refers to the amount of data that passes through a network connection over time. It is the information carrying capacity of a link, connection or the network. The greater the capacity of the connection of a network, the better is the performance of the network. Network bandwidth is directly proportional to the size of packet transmitted per unit of time.

Figure 3 shows the degree of bandwidth utilization in the two schemes under consideration. The simulation result shows that the network bandwidth usage increases as the number of nodes increases on the network. Table 1 illustrates that the increase in CPP scheme is significantly higher than the BCPP scheme. The network bandwidth consumption by a mobile agent that utilizes CPP scheme to route a network of 600 nodes is 1900Bytes while a mobile agent that utilizes BCPP scheme to route a network of the same size only consumes 1813Bytes. This means that the performance comparison of the network bandwidth consumption of BCPP scheme gives a decrease of 4.6% over CPP scheme.

It is then demonstrable in this simulation that the BCPP scheme has a better performance in optimizing the network bandwidth. This implies that, by using the BCPP scheme, the network traffic of a Distributed System can be controlled at a minimal expense of the network bandwidth.

4.2.2 Response time

Response time refers to the time taken to transfer data from its source node to its destination node. The response time of a network is directly proportional to the capacity of the network. The response time is used to determine the latency of the network. The lower the response time, the lower the network latency and vice versa. The smaller the delay or latency, the better is the performance of the network. Figure 4 shows the degree of response time in the two schemes under
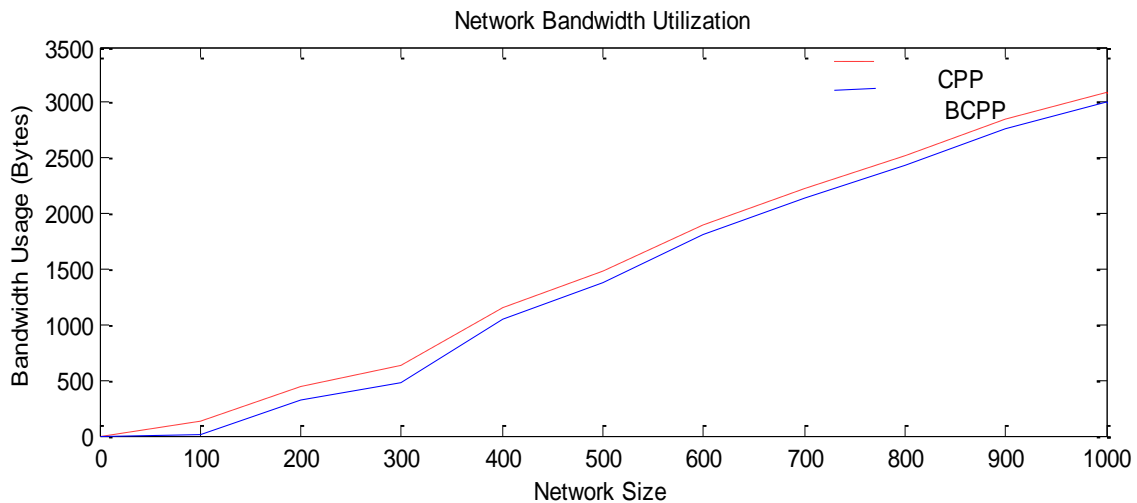
Figure 3: Plot of network bandwidth utilization of the two schemes

Table 1: Network Bandwidth Utilization of the Two Schemes

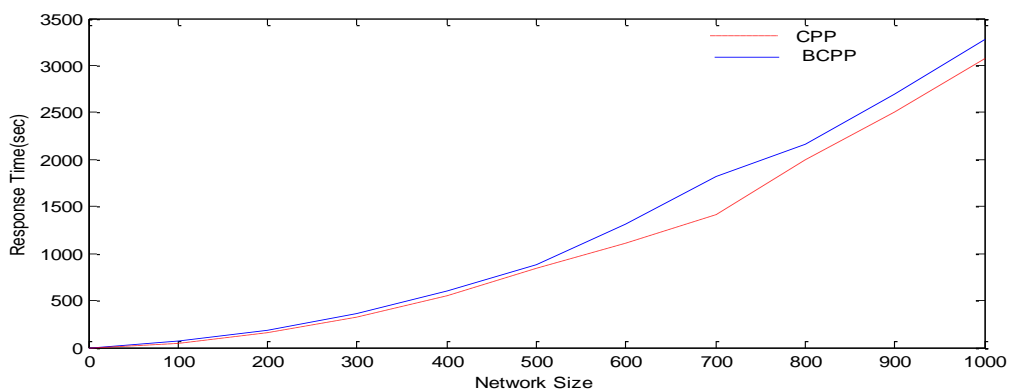| NETWORK SIZE	CPP SCHEME (Bytes)	BCPP SCHEME (Bytes)
0	0	0
100	130	7
200	440	316
300	630	480
400	1160	1053
500	1490	1385
600	1900	1813
700	2220	2134
800	2520	2430
900	2850	2763
1000	3100	2999

Figure 4: Plot of response time of the two schemes

consideration. The response time against the number of nodes of the two schemes were compared. The network response time increases as the number of nodes increases on the network, the increase in CPP scheme is slightly lower than the BCPP scheme. The simulation data in Table 2 shows that the mobile agent that utilizes CPP routing scheme uses less time to reach its destination. A mobile agent that utilizes BCCP scheme spend 1312.8seconds to route a network of 600 nodes while a mobile agent that utilizes CPP scheme to route a network of the same size only spend 1112.8seconds. This means that the performance comparison of the response time on of CPP scheme gives a

decrease of 15.2% over BCPP scheme.

It is obvious in this simulation that there is a reduction in the network delay in CPP scheme than BCPP scheme, thus CPP scheme have a better performance in increasing the effectiveness of the network in terms of reduced delay

4.2.3 Network throughput

During the transfer of data across the network, there is bound to be network failure which will affect the network throughput i.e. the amount of data to be transmitted. Retransmitting packets causes the throughput of the network to decrease. Packet loss

refers to the loss which occurs when one or more packets travelling across a network connection fail to reach their destination. The higher the network throughput, the lower the packet loss, and the better is the performance of the network.

The rates of system throughput of the two schemes were compared. The simulation result shown in Figure 5 reveals that the Network throughput rate decreases as the number of nodes increases on the network, there is a substantial increase in BCCP scheme than the CPP scheme. This means that the BCPP scheme transmits more data than the CPP scheme. According to Table 3, there is a significant increase in the throughput of the network. A mobile agent that utilizes CPP scheme transfers only 0.81kb data over a network of 600 nodes while a mobile agent that utilizes BCPP scheme transfers 0.85kb data over network of the same size. This means that the performance comparison of the drop of the network throughput of CPP scheme gives a decrease of 4.7% over BCPP scheme. It is obvious in this simulation that there is a significant reduction in the rate of packet loss in BCPP scheme than CPP scheme, thus BCPP scheme has a better performance in increasing the effectiveness of the network in terms of reduced packet loss. Thus, in spite of retransmission of data by mobile agent due to intermittent network failure, BCPP scheme still delivers reliably well in Distributed System.

5. SUMMARY AND CONCLUSION

In this work, an improved migration scheme based on Bi-directed Chinese Postman Problem (BCPP) applied to a mobile agent routing in a Distributed Systems was presented. The proposed scheme will choose an itinerary with optimal dependability and cut down the network traffic and eventually ameliorates the overall network performance of mobile agents. Model simulation is carried out for a single mobile agent migration on a distributed system network. The model formulated was simulated using Simulink tools in MATLAB. The working performance of the BCPP routing model was compared with Chinese Postman Problem (CPP) routing scheme using three evaluation metrics. They are: the Network bandwidth, Response Time and the Network Throughput. The BCPP model is effective to procure an enhanced potency in terms of network bandwidth and reduced delay and a minimal packet loss.

6. CONCLUSION

The simulation result of the BCPP scheme on a network of 600 nodes gave a decrease of 4.6% of network bandwidth utilization, decrease of 15.2% of response time and decrease of 4.7% network throughput over the CPP scheme. The performance evaluation result demonstrates that there is improvement in the mobility properties of mobile agents in a network using BCPP routing model. It is evident from the study that the BCPP scheme has a better performance in optimizing the network bandwidth, and in increasing the effectiveness of the network in terms of reduced packet loss. Conclusively, the scheme has been able to optimize the network functions by gathering information about the network traffic to deliver the lower latency, highest capacity and relative dependability in spite of the limited bandwidth and recurrent failure of distributed system network. The proposed migration scheme for mobile agents routing in a network can therefore be embraced by the mobile agent software developers and network administrators for more effective network management.

Table 2: Response Time for the Two Schemes

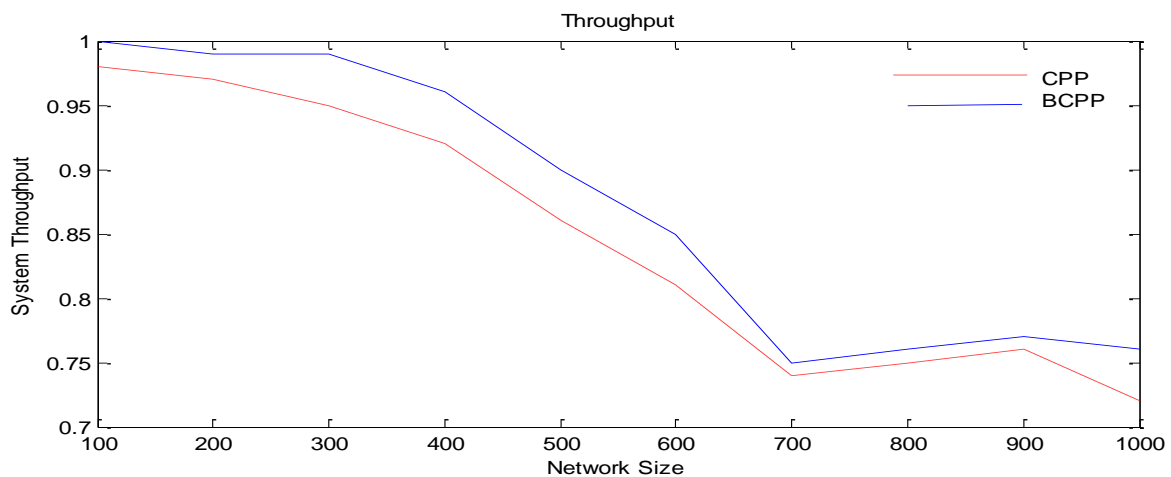| NETWORK SIZE	CPP SCHEME (Seconds)	BCPP SCHEME (Seconds)
0	0	0
100	43.8	71.8
200	163.2	183.2
300	326.5	366.5
400	557.5	597.5
500	846.2	886.2
600	1112.8	1312.8
700	1417.0	1817.0
800	2000.0	2158.0
900	2498.7	2698.7
1000	3076.3	3276.3

Figure 5: Plot of network throughput of the two schemes

Table 3: Network Throughput of the Two Schemes

NETWORK SIZE	CPP SCHEME (Kilobytes)	BCPP SCHEME (Kilobytes)
0	0	0
100	0.98	1.00
200	0.97	0.99
300	0.95	0.99
400	0.92	0.96
500	0.86	0.90
600	0.81	0.85
700	0.74	0.75
800	0.75	0.76
900	0.76	0.77
1000	0.72	0.76

7. REFERENCES

[1] Korba, L., Song, R. and Yee, G. 2002. Anonymous Communication for Mobile Agents. Mobile Agents for Telecommunication Applications, Springer Berlin/ Heidelberg, 2521:171-181.

[2] Andrews G. R. 2000. Foundations of Multithreaded, Parallel, and Distributed Programming. Addison–Wesley, ISBN 0-201-35752-6. 430 -500

[3] Tanenbaum, A.S. 2002. Computer networks. 4th edition, Pearson Education Inc; Prentice Hall PTR Upper Saddle River, New Jersey 07458, 350-380.

[4] Ghosh, S. 2007. Distributed Systems – An Algorithmic Approach. Chapman & Hall/CRC, ISBN 978-1-58488-564-1. 96-130

[5] Peleg, D. 2000. Distributed Computing: A Locality-Sensitive Approach, SIAM, ISBN 0-89871-464-8. 200-300.

[6] González-Valenzuela, S., and Leung, V. 2002. QoS Routing for MPLS Networks Employing Mobile Agents. IEEE Network, 16(3):16-21.

[7] Qu, W., Shen, H., Sum, J. (2005). New Analysis on Mobile Agents Based Network Routing. Applied Soft Computing. 6(1):108-118.

[8] Bandyopadhyay, S., and Krishna P. 1999. Evaluating the performance of mobile agent-based message communication among mobile hosts in large adhoc wireless network. In Proceedings of the Second ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems. 69-73.

[9] Baschieri F., Paolo B., and Antonio C. 2002. Mobile agents for QoS tailoring, control and adaptation over the Internet: The ubiQoS video on demand service. IEEE 2002 Symposium on Applications and the Internet: 109-118.

[10] Strasser, M and Schwehm, M. 2000. A Performance Model for Mobile Agent Systems. In Proceedings of the Autonomous Agents. Vol. 97, 1132-1140

[11] Marwaha, S., Tham, C.K. and Srinivasan, D. 2002. Mobile Agents Based Routing Protocol for Mobile Ad Hoc Networks, In Proceedings of IEEE, Globecom 2002.

[12] Manvi, S.S. and Venkataram, P. 2006. An Agent-Based Best Effort Routing Technique for Load Balancing. Institute of Mathematics and Informatics (Informatica) Lith. Academic Science, 17(3):407-426.

[13] Olofin, B.O, Adagunodo E.R, Aderounmu G.A. 2010. An Optimal Migration Scheme for Mobile Agents Routing in a Data Communication Network. Ife Journal of Technology, Vol. 19(2), 27-32.