Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

6th International Conference
on Electrical Engineering
ICEENG 2008

Multiagent System Case study in Network Routing

By

Gen. Prof. Dr. Ismail A. Ghaffar* Dr. M.Abo Rizka ** Maj. Mohamed. A. ElShafie*

Abstract:

Designing a smart routing is a problem running in a dynamic changing environment which can be manipulated by the mobile intelligent agent, it is an autonomous system that inhabits dynamic environments, senses the environmental changes and acts to these changes to realize its goals.

This paper provides a proposed a Routing Software Development Kit (RSDK) which is a multi-agent simulating environment that simulates the distance vector routing algorithm, but in an agent based suite, taking into consideration some main factors such that route discovery, convergence time and instantaneous network load.

The proposed agent based algorithm draws upon some basic biologically inspired principles called ANTS which facilitate coordination among mobile agents that implement the routing tasks; simulated pheromones are used to control the movements of agents and minimizing the resulting agent load on the network.

Keywords:

Computer Networking - Routing protocols - Network Management - Intelligent Agent

---

* Egyptian Armed Forces
** Arab Academy for Science and Technology

1.Introduction:

Network management protocols like traffic control and routing gain a great important aspect nowadays .Network nowadays are: heterogeneous, dynamically changing, inherently decentralized and increasing in size and complexity [1][2]. To cope with modern network characteristics a smart routing design is a must. Current routing protocols face some constrains and drawbacks like limited size networks, heavy load, slow convergence time,….etc. So designing a routing algorithm based on mobile agent can improve performance and decrease these drawbacks .

The structure of this paper is as follows: section 2 is a survey on agent technology, section 3 is a survey on network routing, section 4 about agent infrastructure , section 5 describes out agent modeling, section 6 simulation and analysis of the proposed agent, section 7 about the conclusion and future work.

2. Agent Technology:

Agent technology is a very active field of distributed artificial intelligence (DAI) research in the recent years. Agents with their intrinsic features gain a great popularity in many applications area.
intelligent agents can be defined as software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and so employ some knowledge or representation of the user's goal or desires (IBM) [3], also can be defined as a hardware or (more usually) software-based computer system that enjoys the following properties:

- Temporal continuity
- Personality
- Autonomy
- Goal-oriented
- Communication ability
- Flexible
- Collaborative
- Reactive
- Capacity for reasoning
- Trustworthiness
- Mobility
- Adaptability

The characteristics of agents can be grouped into two large categories of internal and external properties; Internal properties: are those that form the internal being of an agent, the properties that determine the actions within the agent, internal properties include the ability to learn, reactivity, autonomy, and goal-orientedness. The external properties: include all those characteristics that affect the interaction of several agents or human-agent communication [4].

Intelligent agent technology is suitable for many dynamic environments. It is used in the absence of the expert, also may be designed to achieve a specific goal and learn new abilities by continuously monitoring and exploring that environment.

Intelligent agents continuously perform three functions [5]:

a. Perception of dynamic conditions in the environment.
b. Action to affect conditions in the environment.
c. Reasoning to interpret perceptions, solve problems, draw inferences, and determine actions.

Agents may be classified according to [6][7]:

- Mobility: as static or mobile
- Presence of a symbolic reasoning model: as deliberative or reactive; Deliberative agents possess an internal symbolic, reasoning model and they engage in planning and negotiation in order to achieve coordination with other agents. Reactive agents on the contrary do not have any internal, symbolic models of their environment, and they act using a stimulus/response type of behavior by responding to the present state of the environment in which they are embedded.
- Exhibition of ideal and primary attributes: such as autonomy, cooperation, and learning. From these characteristics, agents may be classified into four types as in the following figure

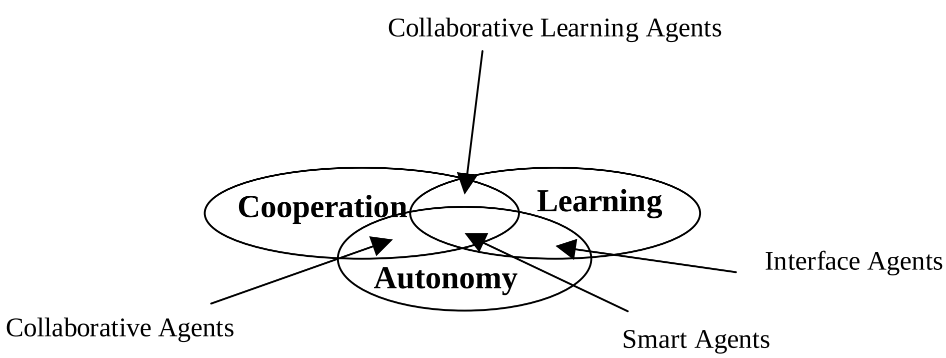

Figure (1): A Part View of an Agent Typology W.R.T Autonomy, Cooperation and Learning

- Roles: as information or Internet agents which manage vast amount of information in the WAN.
- Hybrid philosophies: which combine two or more approaches in a single agent.
- Secondary attributes, such as versatility, benevolence, veracity, trustworthiness, temporal continuity, ability to fail gracefully, and mentalistic and emotional qualities.

Finally; using mobile agents to achieve routing task can offer networks flexibility, scalability in size, multiple network management strategies .

3.Network Routing:

routing is defined as the act of moving information across an internetwork from a source network to a destination one which may not be similar. It occurs at layer 3 (Network Layer) of the OSI reference model also it may be direct or indirect. Each router keeps the set of mappings between destination networks addresses and routes to next routers. These are stored in a table called routing table [4].

Routing algorithms involve two main functions; determination of the optimal routing paths and transportation of data packets through an internetwork (switching). Routing algorithms can be differentiated based on the goals of designing the algorithm ,the different types of link metrics used and the impact of each algorithm on network and router resources [8].

Designing a routing algorithm should achieve some goals. It should provide *optimality and fairness* which refers to the capability of the routing algorithm to select the best route with fairness between all nodes. The algorithm should be *simple* and provide its functions efficiently with a minimum number of software and utilization overhead (*low overhead*. The algorithm should perform its task correctly in the face of unusual or unforeseen circumstances (*robustness and stability*). It should have a *rapid convergence time* in distributing routing update messages that permeate networks, stimulating recalculations of optimal routes and causing all routers to agree on these routes. Routing algorithms should quickly and accurately adapt to a variety of network circumstances (*Flexibility*) [8][9] .

There are different routing metrics which can be used like path length ,reliability of the link ,delay of transmitting data, bandwidth of the link, load, communication cost [9].

Routing algorithm' types can be classified to nonadaptive routing algorithms in which The choice of the route is computed in advance, off line, and downloaded to the routers when the network is booted (static routing) and adaptive routing algorithms in which changes its routing decisions to reflect the changes in the topology, and usually the traffic as well [9].

Distance vector routing algorithm is one of the adaptive algorithms. It is simple ,widely used, has low overhead computation and low memory usage but it suffers from some drawbacks like counting to infinity problem, split horizon, poison reverse and triggered updates [8][9].

The most common routing protocol in the internet is RIP in which hop count is used as a metric and its maximum hop count is 15,there is a broadcasted every 30 Sec or when network topology changes and the network is unreachable if not readvertised in 180 Sec [8][9].

So, as a conclusion ; the development of a good routing algorithm should takes into consideration two main factors : first one is the convergence time (time until all routing tables are built correctly) , the second one is the network load (load on network due to building routing table process until convergence time).

4. Agent Infrastructure:

The abstract design of agents means the set of structural components in which perception, reasoning and action occur. Intelligent agent generally consists of three main layers namely [10]:

1. Incremental concept formation layer (Knowledge base).
2. knowledge source layer (intermediate concepts)
3. inference engine layer

The following figure illustrates the architecture of the intelligent agent system

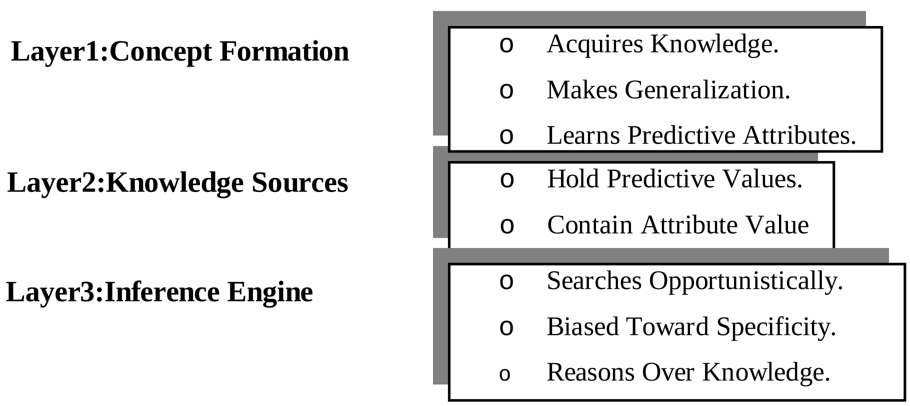

Figure (2): Intelligent Agent Architecture

There are different methods exist to build an intelligent agent systems from these methods. These agents may be rule-based semi-autonomous agents consisting of user programmable rules for carrying out user defined tasks [10].

Agent can use different sources for learning and building its knowledge base from observing user actions in the environment, or by receiving user feedback or being trained by the user with specific examples, or from asking for advice from agents that assist other users with the same task [10].

Agents used in implementing the RSDK system should posses the following characteristics to be suitable for working in the routing field as a dynamic environment :

1. Agents encapsulate a thread of execution along with a bundle of code and data. Each agent runs independently of all others; self contained, and reserve all of its state when moves from one network node to another.
2. Any agent can move easily across the network, so the agent can call the underlying infrastructure to move itself to a neighbor node.
3. Any agent must be small in size. Because there is some cost associated with hosting and transporting an agent.
4. An agent is able to cooperate with other agents in order to perform complex or dynamic tasks. Agents may read from and write to a shared block of memory on each node, and can use this facility both to coordinate with other agents executing on that node, and to leave information behind for subsequent visitors.

5. Agent Modeling:

The proposed mobile agent will be composed of two main parts :

1. Data segment of the mobile agent consists of (4 fields):
   - Agent_ID Field
     This field is a unique ID value for each agent in the field of multiagent system.
   - From_Node Field
     This field is the address of the node from which the agent migrates
   - To_Node Field
     This field is the address of the node to which the agent will migrate.
   - Routing_Information_Carried
     This field is the routing table entries or part of it, carried with the agent during migration from node to another.
2. Code segment of the mobile agent

This segment will contain the functions which access its data segment, perform migration, routing tasks (adding new entries, updating, deleting) and so on.

Designing of the proposed routing agent module should has some requirements which are *route discovery*; that is the process by which every node in a network can obtain a route for every other node in that network, *convergence time;* which is the time at which every node has an optimal shortest path to every other node in the network, *instantaneous network load*; which is the load of the algorithm at an instance and the capability of dealing with failures.

Firstly; The route discovery process occurs as follows:

1. Agent at its local node, carry all routing table

2. Select a random neighbor to travel to.
3. At the arrival router (node ny), the mobile agent updates routing table of router (ny) based on the equation:
   D(y,j)= min (D(y,j) , [d(y,x) + D(x,j)])
   For all nj in routing table carried Where :
   D(y,j) cost of link from node ny to node nj on node ny
   d(y,x) cost of link from node ny to node nx carried with agent
   D(x,j) cost of link from node nx to node nj carried with agent

Secondly; the adopted process of the route discovery is based on the use of random selection which satisfy fairness and simplicity but certain areas of the network may remain unvisited for long periods which may lead to high convergence time. So to achieve low convergence time, the agent needs a depth-first search of the network, based on network information carried by predecessor agents.

To implement information based depth first search Ant Based Control Algorithms should be used. These Algorithms take inspiration from biological ant's behavior in finding shortest path. Biological ants use pheromone trails to follow the path of the predecessor ant. Stemming from that, the proposed agent will use artificial pheromone to repel successor agents as shown below [11].

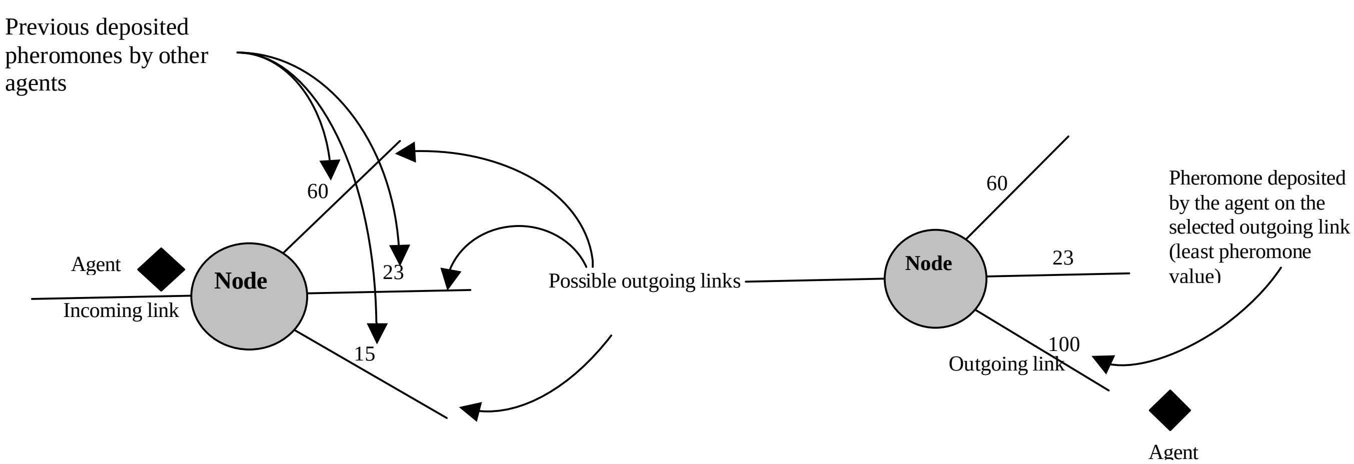

Figure (3): Agent's arrival at a node and selecting an output link with the smallest pheromone value

Thirdly; the adopted process of the route discovery is based on carrying all routing table entries with the agent to the neighbor node which leads to excessive overhead. So to minimize the instantaneous network load of the proposed algorithm the agent needs a flag to identify routing table entries which are: new entries and updated entries that have not been yet transferred to a particular neighbor.

The proposed algorithm uses flags to identify new or updated entries by assigning to each new or updated entry a flag with value 1 to all neighbors; next the agent selects entries with flag value 1 for a selected neighbor. Changes these values to 0 as shown

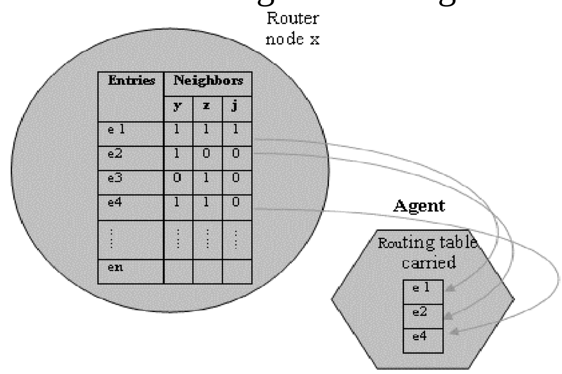

Figure (4): Agent's migrating agent from node x to node y and selecting entries with flag value 1

At Last ; the next scenario shows how the proposed algorithm deals with link or node failure

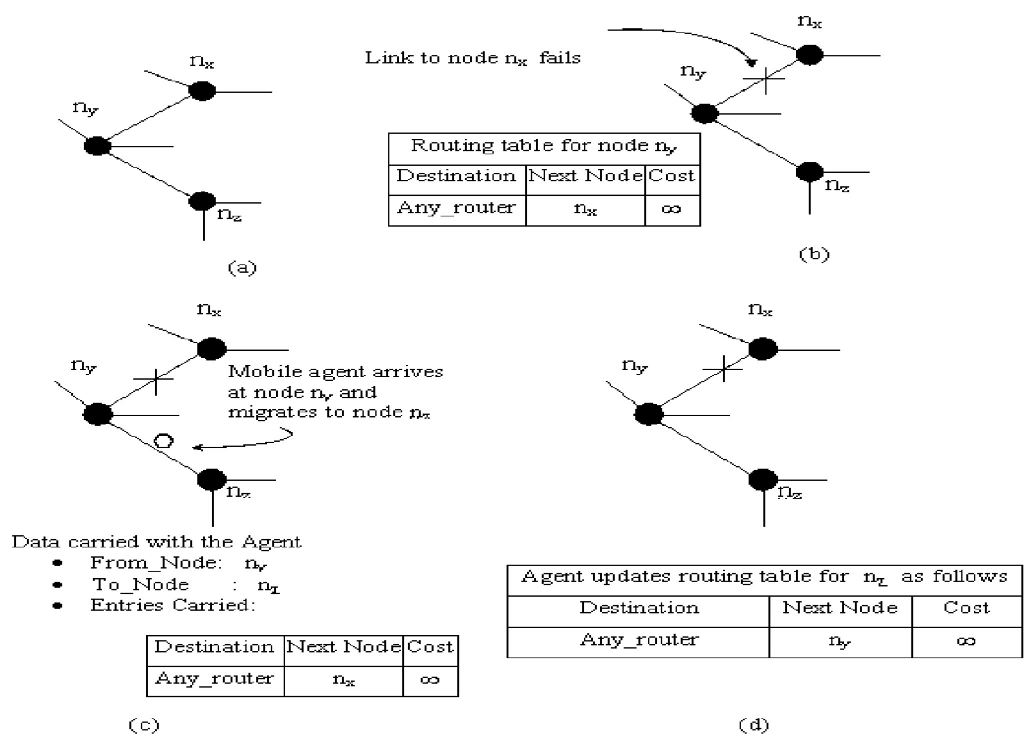

Figure (5): dealing with changes in network topology
(a) The subnetwork (b) the subnetwork if a link fails
(c) A mobile agent traveling (d) mobile agent arriving with entries (∞)

The proposed agent based algorithm will be composed of two cooperative agents:
1. Mobile agents: which traverse the network and perform routing tasks.
2. Stationary agents: This acts as an interface over the router between mobile agents and routing table.

The cooperation between mobile agents and stationary agents by means of message sending and receiving and the cooperation between individual mobile agents is achieved by pheromones deposited at the stationary agents

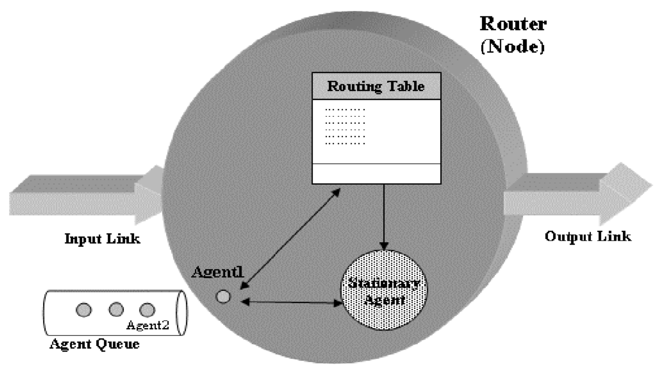

Figure (6): The Proposed two Cooperative agents

The proposed algorithm is:

1. At each node (Router) a stationary agent (Node_Agent) is created , the Node_Agent takes a map of its local routing table
2. At the first creation of the Mobile_Agent:
   - It determines its router's home address and assigns it to its "*From_Node*" field.
   - It sends a "Creation "message to its local Node_Agent.
3. When Node_Agent receives that "Creation" message:
   - It determines the address of next router among its neighbors, to be traveled to by the Mobile_Agent based on random selection.
   - It determines routing table entries from its routing table map, which have flags (false) for the link of the selected next router, and assign these entries to its "Data" vector field.
   - It changes the flag of these entries to "*true*" value.
   - It assigns a pheromone value (decaying exponentially with time) to the selected outgoing link (to the selected next router).
   - Sends "Go" message to the Mobile_Agent associated with:
     a) Next router to travel to.
     b) "Data" Vector that will be carried by the Mobile_Agent during traveling to the next router.
4. When the Mobile_Agent receives that "Go" message:

- It detects next router from that message and assign it to its “To_Node” field.
- It detects “Data” vector and assign it to its “Routing_Information_Carried” field.
- It will move itself to next router.

5. On arrival of each Mobile_Agent at a router it will:
- send a “Arrived” message associated with :
    a) Its “ From_Node ” field.
    b) Its local “Agent_ID”.
- Wait for a message.

6. When the Node_Agent receives that “Arrived” message :
- it checks its “engaged” state

    If it is false

    a) It sends a “Begin” message to that Mobile_Agent.
    b) It changes its “Engaged” state to “true” value.

    Else if it is true

    It puts the “Agent_ID” of that “Mobile_Agent” to its local “Agent_Queue” field.

7. When the Mobile_Agent receives that “Begin” Message;

It will begin updating the routing table of its local router based on

$$D(y,j)= \min (D(y,j) , [d(y,x) + D(x,j)])$$

For all n_j in routing table carried

Where :

$D(y,j)$ cost of link from node n_y to node n_j

$d(y,x)$ cost of link from node n_y to node n_x

$D(x,j)$ cost of link from node n_x to node n_j

Hint: suppose agent migrates from node n_x to node n_y

Determine its neighbor cost (cost for link from previous router to that router).

- For each entry in its “Routing_Information_Carried” field (entry1) it searches for an entry in the routing table of that local router (entry2) which has the same “Destination” field as the “Destination” field for the entry (entry1).

  If it is found

  If cost for (entry2) > neighbor_cost + cost for (entry1)

  - Update entry2 as follows:
    - “Destination” field for entry2 is the same
    - “Next_Node” field for entry2 = “From_Node” field of that Mobile_Agent.
    - “Cost” field for entry2 = neighbor_cost + cost for entry1.
  - Add this updated entry to a temporary "Changed_Entries” vector in the Mobile_Agent.

Else if cost for (entry2) < neighbor_cost + cost for (entry1)

- Leave entry2 without changing.

If it is not found

i. Create new entry equal to entry1

ii. Append this new entry to the local routing table of the router

iii. Add this entry to a temporary "New_Entries" vector in the Mobile_Agent

- Upon finishing updating local routing table :

It sends a "Finish" Message to the Node_Agent associated with:

i. "Changed_Entries".

ii. "New_Entries".

- Wait for message.

8. When Node_Agent receives that "Finish" Message :
    - It changes its "Engaged" state to "false" value.
    - It determines if it's local Agent_Queue is empty or not:

      If it is not empty

      i. Dequeue an agent's ID and sends a "Begin" Message to the Mobile_Agent of that ID.

      ii. Changes its "Engaged" state to true value.
    - Detects the "New_Entries" and associated with "Finish" Message , and adds each entry of these entries with a flag value "false" to each one its connected outgoing link in its "Local_Routing_Table_Map" vector.
    - Detects the "Updated_Entries" associated with "Finish" Message, update its entries in its "Local_Routing_Table_Map" for each entry in the "Updated_Entries" to each connected outgoing link with a flag value 'false'.
    - It determines the "Next_Node" to be traveled to by the "Mobile_Agent" as follows:

      i. If the number of outgoing link=1; it will be selected.

      ii. If the number of outgoing links>1, it will select the link with the least pheromone value.

      iii. It assigns a pheromone value (decaying exponentially with time) to that selected link.
    - Looks up in its "Local_Routing_Table_Map" for that selected outgoing link, select entries with flag "false" and attaches these entries to its "Data" vector.
    - changes flags for selected entries to that link to "true" value.
    - It sends a "Go" message associated with "Next_Node " field and "Data" vector field to the Mobile_Agent.
9. Repeat again from step (4).

6. Simulation and analysis of the proposed agent:

Java is an object-oriented language, has a security model and platform independent. All of these make it the perfect environment in which agent-based tools can be developed .

Because agent is dispatching in the form of bit stream, Agent's code segment consumes significant amounts of resources. If some codes of the agents are supplied firstly as pre-loaded software modules (*System Classes*) at each node, which will result in smaller light weight agents [12][13].

The following figure represents a snap shot screen for the package running a simulated network consisting of 14 nodes (routers) , connected together as shown with associated cost links.

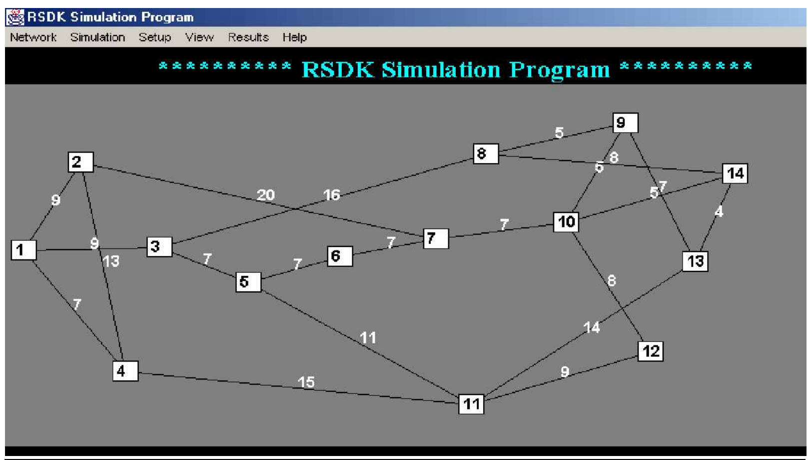

Figure (7): The Test Netwok

The Analysis of path-cost convergence and route discovery shows that a number of agents increase the convergence time decrease but there is an associated overhead [4].

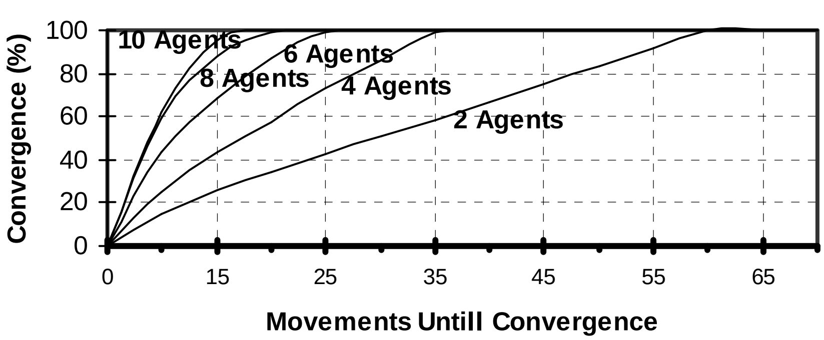

Figure (8): Convergence time with different agent populations

Assuming that agent header consists of a 4 byte; each routing table entry occupies 20 bytes and the average protocol overhead =(x*20+n*4)/ t.

x: cumulative number of entries till convergence

n: number of mobile agents

t: convergence time

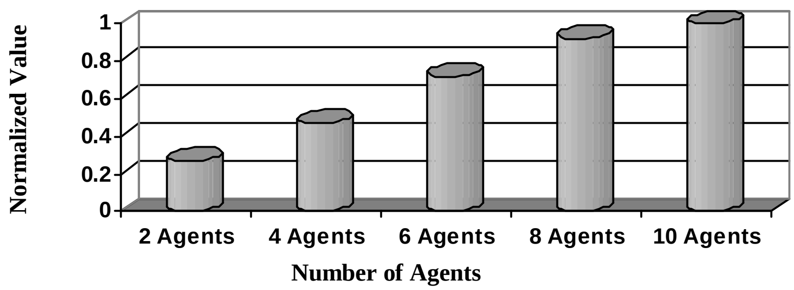

Figure (9): Routing Overhead until Convergence for different agent population

The previous two graphs show that 4 agents would result in an optimal trade-off between convergence time and overhead [4].

Figure (10): Comparison of Routing Overhead with Path-Cost Convergence

7 Conclusion and Future Work:

Our proposed routing agents have the authority to control their actions and strategies *(autonomy).* They traverse the network collecting specific information (*mobility),* make several decisions to adapt their behavior *(rule based reactive agents),* and/or their change in the existing environment affecting their future actions *(proactive agents).* Finally; they coordinate their actions though pheromones and flags *(Socialability)* towards a directed global goal.

The proposed routing algorithm satisfies the designed goals of the routing algorithm in which the distributed nature mobile agents achieves *flexibility* ,it satisfies *optimality* and *fairness* between all nodes, using flags for selecting entries carried leads to *reducing overhead* ,pheromones deposited at links leads to *rapid convergence* and its way in dealing with failures leads to *robustness and stability*.

It improves the performance of the current RIP because the algorithm allows the scalability for large networks (no restriction on number of routers) and the ability to use any routing metrics or a combination of some of them. It does not need 180 seconds to know that one of its neighbors became invalid. And only new or updated entries are carried with to the neighbors.

Much work remains to be done in several important areas: provisions for security and safety across networks; tools for design, modeling and inspection; methods for analysis of resource tradeoffs, system overhead and protocol strengths and weaknesses; and application-level frameworks that allow new uses of networked devices.

Link cost; in which agents can determine the cost between two neighbors based on delay time (cost will be difference between the departure time from the source node and the arrival time at the destination node).*Dynamic agent population*; that is a self adjustment strategy would be used for each agent to monitor the resources that it may consume, or die off or spawn new copies of itself. *Dealing with Failures;* in which auxiliary agents that are entrusted with the task of propagating the information.

References:

[1] Kaizer A. Amin and Armin R.Mikler , "Agent-based Distance Vector Routing: A resource Efficient and Scalable approach to routing in Large Communication Networks", Department of Computer Science , University of North Texas, 25 March 2002.

[2] Nelson Minar – Kwindla Hultman Karmer – Pattie Maes, " Cooperating Mobile Agents for Dynamic Network Routing" , MIT Media Lab, E15-305 20 Ames St, Cambridge MA 02139, USA,1999, http://www.media.mit.edu/~nelson/research/routes-bookchapter/.

[3] Benjamin N.Grosof, "Building Commercial Agents, IBM Research Report" T.J. Watson Research Center Tokyo, Zurich. RC 20835 (92272) 08, May 97, http://www.research.ibm.com/people/g/grosof/home.html.

[4] Mohamed A. ElShafie, "Smart Routing Using Intelligent Agent ".Master of science, 2004, Egyptian Armed Forces.

[5] Stan Franklin and Art Graesser ," Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents ",Institute for Intelligent Systems,University of Memphis, Proceedings of the Third International orkshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996 http://www.msci.memphis.edu/~franklin/AgentProg.html.

[6] Hyacinth S. Nwana, "Software agent, an Overview", Knowledge Engineering Review Journal, Vol. 11, No 3, pp.1-40, Sept 1996.© Cambridge University Press.

[7] Jeffrey M. Bradshaw , " Software Agents ", 1997, ttp://citeseer.ist.psu.edu/cache/papers/cs/11219/http:zSzzSzwww.ai.univie.ac.atzSzzCz7EpaolozSzlvazSzvu-sa98zSzpdfzSz01-Bradshaw.pdf/bradshaw97introduction.pdf/

[8] Author: Cisco Systems, Inc. "Internetworking Technology Handbook" copyright1992—2003, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/index.htm.

[9] William stalling, "Data and computer communication", prentice hall.

[10] Ayman Abdalla , "Artificial Intelligence Tools in Networking of the Command and Control Systems", PhD in electrical engineering , Military Technical College, Cairo 2000.

[11] Gianni Di Caro and Marco Dorigo . "Ant Net: A mobile Agents Approach to Adaptive Routing". IRIDIA, Universite libre de bruxelles, Belgium. technical Report IRIDA

[12] Author: Java," Agents: Not just for Bond anymore, Learn what agents are and how to create them using IBM's Aglet Workbench", http://www.javaworld.com/javaworld/jw-04-1997/jw-04-agents.html.

[13] Mitsuru Oshima and Guenter Karjoth , "Aglets Specification (1.0)" . Version 0.60.May 20th, 1997. http://www.trl.ibm.com/aglets/spec10.htm