Finding the Best Path Routing Using Multi-agent System

Abbas M. Al-Bakry
University of Babylon College of Science Technology
Ahmed S. Hatim
University of Babylon College of Information Dept. computer Science

Abstract

This work presents a new approach to accommodate routing demands in a highly dynamic network where segment status is rapidly changing to find the best path. The router does not need to look at the network as a whole, as it is the case in routing algorithms designed so far. The proposed system consists of several agents (multi-agent) shared by some types of information for the purpose of clarifying the case of the network in general in the way to suggest the best path, which is used by the message from the source to the destination. Each agent activated at each router called Agent Sup-router to acquire local data of the current router this data gathered from the neighboring agents in order to direct path of moving packet from source to destination. In the proposed system, an intelligent notification message exchanged by Agents Sup-routers enable the system to evaluate the current status of the network instantly. This status has been acquired by the proposed distributed Intelligent Agents as an initiation for finding the best path. The proposed system has been implemented using java language and JADE platform.

Keywords
Routing, Multi-agent System, Best Path Routing, JADE

1. Introduction

Computer networks have been growing at an enormous rate ever since the concept was first proposed. Networks such as the Internet have become an integral medium of information transfer, ranging from personal communication to electronic commerce. The basic function of a data network is very simple, delivering data from one network node to another. Data networks can be viewed as huge graph with routers as vertices and transmission lines as edges [Swap05]. Routing is simultaneously the most complicated function of a network and the most important [Cisc89]. Routing can be described as the process of creating a logical connection between nodes in a network so that packets sent by a node can reach their destination. The challenge in developing network routing algorithms is in dealing with the scale and distribution of the physical network. Also, routing algorithms need to cope with events such as node and link failures and recalculate paths whenever such events occur. Finally, routing algorithms need to calculate paths to allow nodes to achieve high network performance [Swap05]. Traditional routing methods do not have enough flexibility to satisfy the new routing demands, like new network services, and mainly the
impressive increase in the amount of users that forces the network administrators to improve throughput. Nowadays, other very studied routing alternatives are based on agents [Benj00]. An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. A robotic agent substitutes cameras and infrared range finders for sensors and various motors for effectors. A software agent has encoded bit strings as its percepts and actions [Stua95]. Multi-agent systems are a relatively new sub-field of computer science. Multi-agent systems are systems composed of multiple interacting computing elements, known as agents. Agents are computer systems with two important capabilities. First, they are at least to some extent capable of autonomous action of deciding for themselves what they need to do. Second, they are capable of interacting with other agents not simply by exchanging data, but the kind of social activity that we all engage in every day of our lives: cooperation, coordination and negotiation [Mich02]. Agents appear to be a promising approach to developing many complex applications. [Mich97]. The rest of the paper is organized as follows. Section 2 describes previous works related to our works. Our proposed system is given in section 3. Section 4 describes the results of the proposed system. Finally, section 5 is our conclusion.

2. Related works

J. Arokia Renjit, et. al.: presented a network routing algorithm which has autonomous adaptability to network traffic conditions has been proposed when routing node has some different paths to a given destination they can evaluate these paths in terms of their latency (delay time), every node works as a distributed autonomous agent for adaptive routing [J.aro08]. Fabio Bellifemine, A. Poggi and G. Rimassa: presented a software describing its intended uses, the main architectural issues are discussed, and the major design decisions are outlined and brief discussion about the current implementation of JADE, its current use to develop applications and to test the interoperability with other agent platforms [Fabi99]. Fabio Bellifemine, et. al.: presented a technological components. The paper presented JADE – a software framework to facilitate the development of agent applications and described its software architecture and its provided functionalities as well as its integration with some technologies [Fabio07]. Fabio Bellifemine, et. al.: presented a software framework (JADE) to make easy the development of multi-agent applications in compliance with the FIPA specifications. JADE can then be considered a middleware that implements an efficient agent platform and supports the development of multi-agent systems [Fabi01]. Nenad S. Kojić, et. al.: Designed an algorithm for packet routing with primary goal to find optimal path through dynamic network topology. This information is used for routing packets into the WMN. too. In this way updated messages are broadcasted all over the network via optimized flooding technique [Nena10]. J. Neto, and Helder Coelho: Described the approach to choose the best path all nodes will react when a network node/arc is inserted/deleted, and eventually producing other reactions on their immediate neighbors. This approach will see each node as an agent capable of dealing with several distinct problems, namely, how to send information to a certain destiny and how to solve all possible and unexpected changes of network topology [João01].
3. The proposed system

The proposed system uses a multi-Agents for finding the best path routing from the source to the destination. Our proposed system can be presented in a conceptual view of each agent describes in figure (1) below.

![Conceptual View of the proposed system](image)

Figure (1): Conceptual View of the proposed system

This proposed system is dedicated to build an intelligent routing techniques. The conceptual view in figure (1) above is presented as a units and the relations between them. The Execution Thread Unit is represented an autonomous agent that perceive the IP configuration setting from User Interface Unit in sequential, the Communication part is responsible for exchange all messages between the agents waiting for processing by Agent behaviors, the Agent Behavior Activator Unit notify the Agent Behavior's Pool to dispatch the appropriate behavior. The Database Connection Unit is used to construct and update the routing table using SQL query.

3.1 Update Routing Table

The routing algorithm is an important part of the system which is used to build a routing table toward generating the best path, and further more the status of the link. Figure (2) clarifies the structure routing table in proposal.

![Routing Table](image)

Figure(2): routing table

We use the MYSQL database management to manage routing tables, we use the Wamp server program that it support the MySQL, each agent uses this table to save the neighbor routers. All operations dealing with the creation and updating of the
database is in the Database Connection Unit (DCU) in figure (1) above. The proposed system relies upon Agent platform can make use of its unique features such as autonomous interaction to the environment and independent reactions through its effectors. The significant feature added by agent to routing protocol is the ability to do much over head processing due to its ability to perceive the environment (i.e., network status in this proposal) and upon conceptualizing the events that occurred within the environment (i.e., Adding router, deleting router, route status, … etc). Routing tables on neighbors routers Agents can be updated When Each router is added in the network, the agent can sends INFORM message to other routers that content the suitable address serial ports, any agent in any router perceive this message, the agent can compare the message content with IP address ports in order to establish and maintain neighbor relationships. The agent can add the adjacent routers to the routing table when receive INFORM message. Figure(3) shows Adding adjacent router Algorithm.

**Adding Adjacent router Algorithm**

**Input**: the INFORM message received (labeled getcontentmsg)  
**Output**: adding adjacent routers to routing table of agents.

**Begin**

1- token1=cutter(getcontentmsg," ; ") 
   set Matchflag=false 
   set WAN IP= all IP addresses of interfaces(ports) 
   for i1=0 to countToken(token1) do 
      begin for 
         if matchFlag=true then break 
         set stra=nextToken of token1 
         token2= cutter (WAN IP," ; ") 
         for i2=0 to countToken(token2) do 
            begin for 
               if matchFlag=true then break 
               set strb=nextToken of token2 
               token4= cutter (strb",".) 
               token3= cutter (stra",".) 
               set str1= nextToken of token3 
               set parsestr1=converttoInteger(str1) 
               checking the parsestr1 if belong to Class1(1..126) or class2(128..191) or class3(191..223) to determine the iloop(1..3) 
               set str2= nextToken of token4 
               for tt=0 to iloop do 
                  set str2= nextToken of token4 
                  if str1= str2 then 
                     matchFlag = true 
                     set str002 =strb 
                  else 
                     matchFlag=false 
                     break; 
                  endif 
               set str1= nextToken of token3 
            end for 
         end for 
      end for 
   end for
Communication scheme is the unit that responsible for sending and receiving messages from/to other agents. The received messages sent to the message queue waiting for further processes by Agent behaviors. The Agent behavior normally is a software part that can do some action based on the interpretation of the incoming messages. Each agent in our suggested system use different massage types to interact with the other agents at different routers in the environment after configure IP addressing for all interfaces of each router in the network, each agent perceives this information from it environment, the agent broadcast INFORM message to other

Figure (3) adding adjacent routers algorithm

2- if matchFlag=true then
   set SQLquery="INSERT INTO lan_table (Agent ,IP,status)VALUES
   ("'"+ getsenderAgentNameID() +"", ""+str002+"", ""+ getAgentState ()+""")
call execute update function (SQLquery)
3- end algorithm
agents in other routers to determine neighbors and return also INFORM message from other agents at any shared router in same network in order to update the routing table at other agents. any router become out of the network , the related agent send CANCEL message to other agents .The REQUEST message is used to discover the paths to the destination and return an AGREE message that contains the paths to choose as the best paths if the destination is found based on the available bandwidth generated randomly as metrics , Figure(4) shows the message format or the content that dispatched in REQUEST message.

**destination IP ; agentID1 , traffic1 , ... , agentID n , traffic n ; totaltraffic ; hopcount**

where

- destination IP : the IP destination is requested.
- agentID1 , traffic1 : the first agentID name and traffic on it in path.
- agentID n , traffic n : the last agentID name and traffic on it in path.
- totaltraffic : set of all traffic on path (traffic1+...+traffic n).
- hopcount : number of router in path.

we use the Maximum Hop Count and Split Horizon to prevent the routing loop when the agents interact REQUEST message between them to find the path in AGREE message that contain the path .The figure (5) shows below REQUEST Algorithm for each agent. The Route can be found based on the agent cooperation, where the message is sent and traveled over network hops (routers) in REQUEST message and return the best routing path. Each agent uses REQUEST message to recognize and find the best to destination and in AGREE message based on the highest Bandwidth and the Active status of the path if the destination is found.

**REQUEST Algorithm**

**Input** : The message contain the destination (msg).

**Output** : AGREE message if destination is found, or resend REQUEST message to neighbors.

**Begin**

1- Tokenizer (cutter) the (msg) to return destIP , pathcost , totalcost , hopcount, nexthopID and metric

Set pathcost1=pathcost

Set agentAID=getAID() //return AID of this agent

Set primemetric = totalcost

2- SQL Query to sure if it has the destination(destIP)

Set resultset ="SELECT * FROM lan_table WHERE IP = destIP " // return records of table in resultset

3- If resultset = null then // Destination is not found, send to all other neighbor agents

Set record=0

repeat

Set neighbor=getobject (2) of resultset1

Set ss=convertAID(neighbor) // convert neighbor to AID

if ss<> agentAID then

if record=0 then // if first record

// setting massage to travel over other agents

Set metric = Random () // generate randomly
Figure (5) : REQUEST algorithm

The Agent behavior activation is the unit responsible for activate agent behavior, from the agent behavior’s pool based on the performative of the message arrived to the agent. The proposed system uses different performatives or types of message as shown in section 3.2 (communication Scheme) and it uses the format part from the contain as presented in figure (6).

4- assemble message to travel again to other agent and send ACLMessage with performative (type of message) REQUEST to each neighbor to return paths

Set msg2 = ACLMessage();
    Set receivermsg2 = ss  // AID of next hop    Set contentmsg2 = “destIP ; pathcost ; totalcost ; hopcount ”

// In order not to send to the same interface to prevent routing loop
If (ss<> getsenderAID of msg) and (hopcount<16) then
    send(msg2)
    set record = record +1
end if

until (resultset1.next() = null)
else

// find the path to destination and send AGREE message (type of message)
Set msg4=ACLMessage()  Set receivermsg4 = nexthopID  // AID of next hop  pathcost = pathcost + " ," + agentAID + " ,"  + convert to string(generate random (metric4))
set hopcount = hopcount +1
set totalcost = totalcost + metric4
Set contentmsg4 = "destIP ; pathcost ; totalcost ; hopcount ”
    send(msg4)
end if
4. Results

The proposed system has been implemented using Java language (NetBeans IDE Version 6.9.1) with JADE agent platform which works as a distributed system implemented on a computer network. Our system are used the topology in the figure (7) below each node represent router (node1 represent source node and node6 represent destination node), the agent is installed in each of router or node to configure the multi-agent environment to simulate this, we take a network consist of six laptop computer (each one the Processor: Intel® core ™ i5 2.30 GHz and 4GB main memory under the Microsoft Windows 7 Ultimate operating system) connected to (TP-LINK 150m wireless Lite N Router) each laptop computer represent node (router).
The Agents (Agent sup-router1, Agent sup-router2, Agent sup-router3, Agent sup-router4, Agent sup-router5 and Agent sup-router6) perceive the IP addresses of interfaces configured to each router (serial and FastEtherent IP address interfaces). After all agents perceive IP addresses of interfaces, each Agent automatically broadcast INFORM message to other agents in each router that contain the IP addresses interfaces information to determine the neighbors that share in the same network. For example, Figure (8) shown add the neighbors to routing table of Agent Sup-router1 (Agent Sup-router2 and Agent Sup-router3).

For requesting and finding the best path from first router (source node 1) to the destination in six router (node 6) by exchange the REQUEST message to return the best path in AGREE message, the result is shown in figure (9) below.
Figure (9): The Results

To show the interactions by messages between the agents to find the best path from source to destination in The Figure (10) bellow.

Figure (10): An Interactions as viewed by the Sniffer agent from JADE

The Figure (11) bellow show the activated agent in the platform, as we can see there are six active agentSuprouter1, agentSup-router2, agentSup-router3, agentSup-router4, agentSup-router5 and agentSup-router6 and the agents running on each container.
The paths list in figure(9) is shown in table (1) below, the path 1 (in last record) is best path because it has available high bandwidth.

Table (1) The sample path results

<table>
<thead>
<tr>
<th>ID</th>
<th>Destination</th>
<th>Path with Traffic</th>
<th>Total Path</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>192.168.15.1</td>
<td>AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter2@192.168.1.101:1099/JADE,16,AgentSuprouter4@192.168.1.101:1099/JADE,12,AgentSuprouter6@192.168.1.101:1099/JADE,43</td>
<td>AgentSuprouter1 --&gt; AgentSuprouter2 --&gt; AgentSuprouter4 --&gt; AgentSuprouter6 --&gt;</td>
<td>225</td>
</tr>
<tr>
<td>1</td>
<td>192.168.15.1</td>
<td>AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter2@192.168.1.101:1099/JADE,101,AgentSuprouter5@192.168.1.101:1099/JADE,286,AgentSuprouter3@192.168.1.101:1099/JADE,5,AgentSuprouter4@192.168.1.101:1099/JADE,329,AgentSuprouter6@192.168.1.101:1099/JADE,87</td>
<td>AgentSuprouter1 --&gt; AgentSuprouter2 --&gt; AgentSuprouter5 --&gt; AgentSuprouter3 --&gt; AgentSuprouter4 --&gt; AgentSuprouter6 --&gt;</td>
<td>962</td>
</tr>
<tr>
<td>1</td>
<td>192.168.15.1</td>
<td>AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter2@192.168.1.101:1099/JADE,101,AgentSuprouter5@192.168.1.101:1099/JADE,286,AgentSuprouter3@192.168.1.101:1099/JADE,87</td>
<td>AgentSuprouter1 --&gt; AgentSuprouter2 --&gt; AgentSuprouter5 --&gt; AgentSuprouter3 --&gt;</td>
<td>962</td>
</tr>
</tbody>
</table>

Figure (11): view of agent platform
ntSuprouter3@192.168.1.101:1099/JADE,5,AgentSuprouter4@192.168.1.101:1099/JADE,329,AgentSuprouter6@192.168.1.101:1099/JADE,87

The Path:

- 2 192.168.15.1 AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter3@192.168.1.101:1099/JADE,145,AgentSuprouter5@192.168.1.101:1099/JADE,66,AgentSuprouter2@192.168.1.101:1099/JADE,41,AgentSuprouter4@192.168.1.101:1099/JADE,253,AgentSuprouter6@192.168.1.101:1099/JADE,21

The Best Path is:

- 1 192.168.15.1 AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter2@192.168.1.101:1099/JADE,16,AgentSuprouter4@192.168.1.101:1099/JADE,12,AgentSuprouter3@192.168.1.101:1099/JADE,9,AgentSuprouter5@192.168.1.101:1099/JADE,17,AgentSuprouter6@192.168.1.101:1099/JADE,37

The Path:

- 3 192.168.15.1 AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter2@192.168.1.101:1099/JADE,101,AgentSuprouter5@192.168.1.101:1099/JADE,286,AgentSuprouter3@192.168.1.101:1099/JADE,5,AgentSuprouter1@192.168.1.101:1099/JADE,212,AgentSuprouter2@192.168.1.101:1099/JADE,49,AgentSuprouter5@192.168.1.101:1099/JADE,67,AgentSuprouter6@192.168.1.101:1099/JADE,37

The Best Path is:

- 1 192.168.15.1 AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter2@192.168.1.101:1099/JADE,101,AgentSuprouter5@192.168.1.101:1099/JADE,101,AgentSuprouter3@192.168.1.101:1099/JADE,286,AgentSuprouter1@192.168.1.101:1099/JADE,212,AgentSuprouter2@192.168.1.101:1099/JADE,49,AgentSuprouter5@192.168.1.101:1099/JADE,67,AgentSuprouter6@192.168.1.101:1099/JADE,37

The Path:

- 4 192.168.15.1 AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter3@192.168.1.101:1099/JADE,145,AgentSuprouter5@192.168.1.101:1099/JADE,66,AgentSuprouter2@192.168.1.101:1099/JADE,41,AgentSuprouter1@192.168.1.101:1099/JADE,19,AgentSuprouter3@192.168.1.101:1099/JADE,146,AgentSuprouter5@192.168.1.101:1099/JADE,15,AgentSuprouter6@192.168.1.101:1099/JADE,48

The Best Path is:

- 1 192.168.15.1 AgentSuprouter1@192.168.1.101:1099/JADE,154,AgentSuprouter2@192.168.1.101:1099/JADE,101,AgentSuprouter5@192.168.1.101:1099/JADE,286,AgentSuprouter3@192.168.1.101:1099/JADE,5,AgentSuprouter1@192.168.1.101:1099/JADE,212,AgentSuprouter2@192.168.1.101:1099/JADE,49,AgentSuprouter5@192.168.1.101:1099/JADE,67,AgentSuprouter6@192.168.1.101:1099/JADE,37

AgentSuprouter1 --

- 2 AgentSuprouter2 -

AgentSuprouter3 -

- 3 AgentSuprouter4 -

AgentSuprouter5 -

AgentSuprouter6 -->

AgentSuprouter1 --

- 4 AgentSuprouter2 -

AgentSuprouter3 -

AgentSuprouter5 -

AgentSuprouter6 -->

AgentSuprouter1 --

- 5 AgentSuprouter2 -

AgentSuprouter3 -

AgentSuprouter4 -

AgentSuprouter6 -->

AgentSuprouter1 --

- 6 AgentSuprouter2 -

AgentSuprouter3 -

AgentSuprouter4 -

AgentSuprouter5 -

AgentSuprouter6 -->
<table>
<thead>
<tr>
<th>Path</th>
<th>Best Path</th>
<th>AgentSuprouter1[@192.168.1.101:1099/JADE,49,AgentSuprouter5@192.168.1.101:1099/JADE,67,AgentSuprouter6@192.168.1.101:1099/JADE,37]</th>
<th>AgentSuprouter1 --&gt; AgentSuprouter3 --&gt; AgentSuprouter5 --&gt; AgentSuprouter2 --&gt; AgentSuprouter1 --&gt; AgentSuprouter3 --&gt; AgentSuprouter4 --&gt; AgentSuprouter6 --&gt; AgentSuprouter2 --&gt; AgentSuprouter5 --&gt; AgentSuprouter3 --&gt; AgentSuprouter4 --&gt; AgentSuprouter6</th>
<th>742</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>192.168.15.1</td>
<td>AgentSuprouter1[@192.168.1.101:1099/JADE,23,AgentSuprouter3@192.168.1.101:1099/JADE,7,AgentSuprouter5@192.168.1.101:1099/JADE,50,AgentSuprouter6@192.168.1.101:1099/JADE,153]</td>
<td>AgentSuprouter1 --&gt; AgentSuprouter3 --&gt; AgentSuprouter5 --&gt; AgentSuprouter6 --&gt; AgentSuprouter2 --&gt; AgentSuprouter5 --&gt; AgentSuprouter3 --&gt; AgentSuprouter4 --&gt; AgentSuprouter6</td>
<td>233</td>
</tr>
<tr>
<td>7</td>
<td>192.168.15.1</td>
<td>AgentSuprouter1[@192.168.1.101:1099/JADE,154,AgentSuprouter3@192.168.1.101:1099/JADE,145,AgentSuprouter5@192.168.1.101:1099/JADE,66,AgentSuprouter2@192.168.1.101:1099/JADE,41,AgentSuprouter4@192.168.1.101:1099/JADE,6,AgentSuprouter3@192.168.1.101:1099/JADE,5,AgentSuprouter1[@192.168.1.101:1099/JADE,145,AgentSuprouter6@192.168.1.101:1099/JADE,302]</td>
<td>AgentSuprouter1 --&gt; AgentSuprouter3 --&gt; AgentSuprouter5 --&gt; AgentSuprouter2 --&gt; AgentSuprouter4 --&gt; AgentSuprouter3 --&gt; AgentSuprouter5 --&gt; AgentSuprouter6</td>
<td>924</td>
</tr>
<tr>
<td>8</td>
<td>192.168.15.1</td>
<td>AgentSuprouter1[@192.168.1.101:1099/JADE,154,AgentSuprouter2@192.168.1.101:1099/JADE,67,AgentSuprouter5@192.168.1.101:1099/JADE,286,AgentSuprouter6@192.168.1.101:1099/JADE,37]</td>
<td>AgentSuprouter1 --&gt; AgentSuprouter2 --&gt; AgentSuprouter5 --&gt; AgentSuprouter3 --&gt; AgentSuprouter4 --&gt; AgentSuprouter6</td>
<td>911</td>
</tr>
</tbody>
</table>
V. Conclusions

1- Due to the massive increase in network connection over the internet now days, the status of network segment is rapidly changed which presents a serious challenge for traditional routing protocols, which relies on the information gathered by querying the network devices before start delivering data. The information gathered will be used along delivering session especially in streaming data over the network where connection has to be valid over the entire session. This approach inhibits the performance of the network due to loss connections and compulsory demand on re- connect.

2- Intelligent software Agent technology provides a cooperative environment to find best path in computer network, where network device can publish its segment capabilities to the neighbors. The capabilities can be used as metrics to determine best path.

3- Agent-based provides an environment where the network can be self-organized and self-administering network due to social behaviors possessed by the Agent. The metrics used by agents to manage their behaviors include dynamic network topologies, bandwidth-constrains, variable capacity links.

4- Deploying agent within routing domain needs efficient ontology designed for this task to accommodate emerged network terminologies and infrastructures. Network events, protocols, identification scheme should involved within the routing ontology.

References


