ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 5, Issue 4 (Oct. - Dec. 2017)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2017)

A Study on Quality of Service and Ant Colony Optimization in Routing of Mobile AD HOC Networks

[I]K.Sumathi, [II]Dr. J.K. Kanimozhi
[I]Research Scholar, Periyar University, Salem, Tamilnadu, India.
[II]Professor, PG & Research Dept. of Computer Science, Sengunthar Arts and Science College, Tiruchengode,Tamilnadu, India.

Abstract
Mobile Adhoc Network is a decentralized by crowd of mobile nodes which exchange data for the moment by means of wireless transmission. Routing in MANET is a dynamic optimization problem as the search space changes over time due to the time varying nature of the topology of the networks. Routing protocols also maintain connectivity when links on these paths break due to effects such as node movement, battery drainage, radio propagation, and wireless interference. In multihop networks, routing is one of the most important issues that have a significant impact on the performance of networks. In this paper, presented information of Routing Protocols, QoS of MANET and Ant Colony Optimization Algorithms.

Keywords
Mobile Ad Hoc Network, QoS, ACO, Routing Protocols, DSDV, DSR

I. Introduction

Mobile ad hoc networks (MANETs) consist of a collection of wireless mobile nodes which dynamically exchange data among themselves without the reliance on a fixed base station or a wired backbone network. All nodes are mobile and can be connected dynamically in an arbitrary manner. Multi hop, mobility, large network size combined with device heterogeneity, bandwidth and battery power constrain make the design of adequate routing protocols a major challenge. Because of these characteristics, path connecting source nodes with destination may be very unstable and go down at any time, making communication over ad hoc networks difficult [1]. Routing is an important issue in networks. In wired networks, dynamic routing approaches are prevalent among which distance vector routing and link state routing are two of the most popular models [2]. Distance vector routing is based on the Bellman-Ford algorithm in which each node maintains a routing table including the distance to reachable destinations. This routing information is advertised periodically. The source adopts the shortest route when it has packets for a destination. In link state routing, every node propagates its current status of links to all reachable nodes. Whenever a link status in one node changes, a corresponding advertisement will be broadcasted based on the routing table which is refreshed. In wired networks, both distance vector and link state routing behave well due to comparatively stable link quality and topology. However, properties such as link quality and topology in MANETs become unpredictable, degrading the performance of some distance vector and link state routes. As a consequence, protocols have been proposed and well studied, five of which are described below.

a) Destination Sequenced Distance Vector protocol (DSDV)

DSDV [3] is a typical proactive routing protocol in which each node has to maintain a routing table for all available destinations. Routing updates are broadcast periodically. DSDV relies on a sequence number to indicate the freshness of the corresponding item to guarantee loop-freedom. When a route breakage between two nodes, say A and B, is detected by node A, it increases the corresponding sequence number and sets the distance to node B as infinite and this information will be further broadcasted. In DSDV, the routing information broadcasts introduce a large number of control packets which increases the overhead. At the same time, it takes some time before a route can be used, the so called the convergence time [4]. In wired networks where the topology is comparatively stable, this convergence time is minor and it can be neglected. However, in a network where topology changes rapidly, the convergence time is Sufficiently long that there will likely be a lot of dropped packets.

b) Dynamic Source Routing (DSR)

DSR is a reactive protocol which establishes routes on demand [5]. It initializes a route request process when a route to the destination is not known in the route cache. Up on receiving a route request packet (RREQ) packet, intermediate nodes either generate a route reply packet (RREP) while it caches the corresponding route or it adds its own address to the RREQ and forwards the RREQ until it reaches the destination or the packet live time expires. Where bidirectional links exist, the reverse path will be used when the destination or intermediate node doesn't have a route to the source in the cache. In the case of a route breakage, an error packet is generated by the node which detects it and the corresponding item in the route cache is erased. Compared to DSDV, DSR doesn't use periodic broadcasts and thereby reduces routing overhead, saves energy and partly eases network congestion. However, each data packet carries routing information in DSR, increasing the overhead.

c) Ad hoc On-demand Distance Vector (AODV)

AODV [6] is a reactive protocol, based on the distance vector algorithm. The source in AODV originates a RREQ packet when a route to the destination is not available in the cache. The RREP packet is forwarded until it arrives at the destination or an intermediate node which has a fresh enough route. When a stale route is detected, the corresponding routing item is removed and a link failure message is sent out, triggering the route discovery process. HELLO messages are generated periodically to indicate the presence of a node to its neighbours. Compared to conventional distance vector protocols, the number of advertisement packets in AODV is largely reduced. Two main disadvantages of AODV are HELLO induced routing

Vol. 5, Issue 4 (Oct. - Dec. 2017)
ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

overhead increase and an assumption of bidirectional links.

d) Temporally-Ordered Routing Algorithm (TORA)

TORA [7] is a reactive MANET protocol, aimed at minimizing routing overhead by controlling the receiving scope of routing messages when the topology changes. In TORA, each node is assigned a height. All messages flow downstream like water, from a node with a higher height to another one with a lower height. When a node happens to have packets for a destination but it has no downstream links, it broadcasts a Query (QRY) packet which will then be forwarded until it reaches a node that either knows a valid route or is the destination. Such a node will broadcast an update (UPD) packet containing its own height. Other nodes receiving this UPD packet will set their own heights with higher values compared with that in the UPD packet and broadcast this new height. In this manner, the route is established. In In TORA, only one route will be discovered even if multiple routes are available because each node only has one height value that is initially based on the distance from 14 the destination.

e) Optimized Link State Routing Protocol (OLSR)

OLSR [8] is a proactive routing protocol which utilizes Hello messages and Topology Control (TC) messages to discover and exchange link state information based on which individual nodes are informed about the next hop node for destinations. Being a proactive routing algorithm, the route establishment time for OLSR is short since routes are known before use. Two disadvantages of OLSR are a potentially long convergence time, periodic information broadcast induced extra energy consumption and additional routing overhead.

II. Quality Of Service

As stated in last section, many routing protocols such as DSDV, DSR and AODV have paid little attention to QoS support in the early development of MANETs. However, QoS provision is becoming more important nowadays due to the rising popularity of real-time applications.

A) Rising need for QoS provision

In the past decades mobile traffic, which by definition refers to data generated by handsets, laptops and mobile broadband gateways, has been growing rapidly annually. According to a survey by Cisco, mobile data in 2010 was triple the volume of the entire global Internet traffic in 2000. The growth rate in the previous year was 159%, which is 10% higher than anticipated in 2009. This rapid growth in mobile data is forecast to continue for the next five years with an average annual growth of 92%. There are several reasons why mobile traffic has grown so quickly. Firstly, mobile video, which requires high bit rates, is considered to lead to the increase of mobile traffic. It is reported that mobile video reached as high as 49.8% of total mobile traffic in 2010 and will account for two thirds of mobile traffic by 2015. Moreover, Internet gaming, which consumes, on average, 63 PB per month in 2009, also results in a growth in mobile traffic and it is expected to achieve an annual growth of 37% in the coming five years [9]. Last but not the least, Voice over IP (VoIP) which includes phone-based VoIP services direct from or transported by a third party to a service provider, and software-based internet VoIP such as Skype, leads to the expansion of mobile traffic. Many of those applications described above are real-time applications which demand certain guarantees for performance metrics for acceptable operation. Those metrics specify the Quality of Service.

B) Metrics of QoS

QoS is usually defined as a set of services that should be supported during packet transmission. A QoS enabled protocol is expected to support several metrics in terms of end-to-end throughput, delay, and jitter as well packet delivery ratio.

End-to-End Throughput

End-to-End throughput, η, is defined as the ratio of the payload of effectively delivered data packets, $P_{ed,}$ over the elapsed time, $t_{elapsed}$.

the basic unit of η is b/s or B/s. Effectively delivered data packets refers to data packets that are successfully delivered, excluding any duplicated packets. Since the available bandwidth in a network is fairly well known, it is helpful to obtain the actual throughput achieved which reveals the bandwidth usage efficiency. The higher the average throughput is, the better the bandwidth is utilized.

Delay (or Latency)

Delay, τ, sometimes refers to as end-to-end delay, is the time between the originating node sending a packet and that packet reaching the destination. It may vary dramatically because of long queue time or a congested network environment.

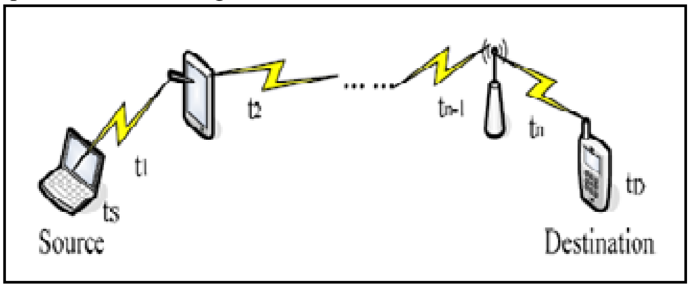

Fig.1 : Delay components

As shown in Figure 1, delay is additive in the sense that it is built up over relay nodes

$$\tau = tS + t1 + t2 + ... + tn\text{-}1 + tn + tD$$

Where *tS* and *tD* denote processing time at the source and destination respectively. The buffering time of a packet is of great importance for delay. If the buffering time in an individual node is set to a higher value, it could imply that packets could stay in the buffer for a long period of time when link breakages occur which will may reduce the packet dropping rate. In this case, the delay is higher. On the contrary, if the buffering time is shorter, the performance of delay will improve but the packet dropping rate will increase. Delay and packet delivery ratio are traded off in different applications. Delay can be computed in multiple layers (e.g., application layer, transport layer network layer and link layer) and thus it is layer-dependent. For the sake of synchronization, round trip delay is used in some literature while others use single trip delay. In this thesis one-way delay is computed in the application layer by using a time stamp in the packet header

$$\tau = Rt - St$$

Where *Rt* and *St* denote time at the source and destination for a given packet respectively, assuming suitably synchronized clocks in the transmitter and receiver. In some cases, excessive delay can render some time sensitive applications such as VoIP or online gaming unusable.

Jitter: Jitter was originally used in signal processing where it measures the deviation of some pulses in a digital signal and can be expressed in terms of phase, amplitude or width of the signal pulse. In the context of mobile ad hoc networks, the term jitter is defined as the average of difference between instantaneous delay and average delay [10]

$$t = \frac{\sum_{i=1}^{n} | (t_i - t)}{n}$$

where *n* denotes number of effective received data packets, τi symbolizes delays for different data unit and τ represents the average delay. It is reported that jitter can degrade live video quality nearly as much as packet loss rate [11].

Packet delivery ratio: The effective delivery ratio of data packets, α, is defined as:

$$a = \frac{ENDP}{TNTP}$$

Where *ENDP* and *TNTP* denote number of effectively received and total data packets respectively. Retransmission degrades the packet delivery ratio because it increases the denominator. A high packet delivery ratio is desirable, especially in MANETs, since the bandwidth available is limited for wireless links.

C) QoS routing in MANETs

The rapid growth of video in mobile traffic has resulted in a shift of research interests from best effort service to the provision of higher and better quality of service in MANETs. QoS routing algorithm design is challenging because it has to deal with unfavourable conditions such as time-dependent wireless links, dynamic topology and energy constraints. Considerable efforts have been devoted to this which leads to the emergence of a number of QoS routing techniques. Generally speaking, two schemes, new protocol design and QoS-aware extension, are adopted to implement QoS routing. New protocol design refers to developing an algorithm with a new methodology while QoS-aware extension means combining QoS guarantee schemes with some well-studied protocols (e.g., DSDV, DSR and AODV).

III. Ant Colony Optimization

A set of algorithmic concepts can be used to define heuristic methods applicable to a wide set of different problems. In other words, a Meta-heuristic is a general purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications. Examples of meta-heuristics include simulated annealing, iterated local search, evolutionary computation, and ant colony optimization. Ant colony optimization has been formalized into a meta-heuristic for combinatorial optimization problems by Dorigo. [12][13][14]

In order to apply ACO to a given a combinatorial optimization problem, an adequate model is needed:

A model P=(S, Ω , f) of a combinatorial optimization problem consists of:

- A search space S defined over a finite set of discrete decision variables Xi, i =1,...,n;
- A set Ω of constraints among the variables; and
- An objective function f: S→R to be minimized.

The generic variable Xi takes values in Di ={v^1i ,...,$v^{|Di|}$i}. A feasible solution s ∈ Sis a complete assignment (i.e., an assignment in which each decision variable has a domain value assigned) that satisfies the constraints. If the set of constraints Ω is empty, then each decision variable can take any value from its domain independently of the values of the other decision variables. In this case we call P an unconstrained problem model, otherwise a constrained problem model. A feasible solution s ∈ Sis called a globally optimal solution (or global optimum),

If: f (s∗) ≤f (s) ∀s ∈S. The set of globally optimal solutions is denoted by S ∗⊆S. to solve a CO problem one has to find a solution s ∗ ∈ S ∗.

ACO works by intertwining three high-level procedures: Construct Solutions, Daemon Actions (optional), and Update Pheromones as shown in Algorithm 1. ACO algorithms are iterative. At each iteration, a number of solutions are built incrementally on the basis of stochastic decisions that are biased by pheromone and heuristic information. These solutions are used for updating the pheromone in order to bias future solutions towards promising regions of the search space. This pseudo code of a generic ACO algorithm is given in Algorithm 1. The constraints of the optimization problem are implemented by enumerating the set of solution components that can be added at each step. This set typically depends on the partial solution constructed so far.

The ACO Algorithm

Set parameters, set heuristic information, and initialize pheromone;

WHILE termination condition not met do (SCHEDULE_ACTIVITIES)

Construct solutions based on pheromone and heuristic information;

Improve solutions via local search - Daemon Actions; (optional)

Update pheromone;

End while (END_SCHEDULE_ACTIVITIES)

Initialize Pheromone Values (T).

At the start of the algorithm the pheromone values are all initialized to a constant value C >0.

Construct Solutions

This procedure implements the artificial ants' incremental construction of candidate solutions. In ACO, an instantiated decision variable $X_i \leftarrow v^j_i$ is called a solution component C_{ij} ∈ C, where C denotes the set of solution components. A pheromone trail value τ_{ij} is associated with each component c_{ij} ∈C. (More formally, each solution component has an associated pheromone variable that can take a value, the pheromone trail value, in a specific range.) A solution construction starts from an initially empty partial solution s^p. At each construction step, s^p is extended by appending to it a feasible solution component from the set of its feasible neighbors N(s^p)⊆C that satisfies the constraints in Ω .The choice of a solution component is guided by a stochastic decision policy, which is biased by both the pheromone trail and the heuristic values associated with c_{ij}. The

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

exact rules for the probabilistic choice of solution components vary across different variants of ACO. The best known rule is the one used first in the ant system algorithm.

$$p_{c_{ij}|s^p} = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{c_{il} \in N(s^p)} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta}, \quad (1)$$

Where τ_{ij} and η_{ij} are, respectively the pheromone trail value and the heuristic value associated with the component c_{ij} .The parameters α>0 and β>0 deter-mine the relative importance of pheromone versus heuristic information.

Deamon Actions (Local Search-optional)

This procedure, although optional, is important when state-of-the-art results are sought. It allows the execution of problem-specific operations, such as the use of local search procedures, or of centralized actions that cannot be performed by artificial ants. It is usually executed before the update of pheromone values in order to bias the ants' search toward high quality solutions.

Update Pheromones

This procedure updates the pheromone trail values associated with the solution components in the set C. The modification of the pheromone trail values is performed in two stages: (i) pheromone evaporation, which decreases the pheromone values of all components by a constant factor ρ (called evaporation rate) in order to avoid premature convergence, and (ii) pheromone deposit, which increases the pheromone trail values associated with components of a set of promising solutions S_{upd}. The general form of the pheromone update rule is as follows:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \rho \cdot \sum_{s \in S_{upd} | c_{ij} \in s} F(s), \quad (2)$$

Where $\rho \in (0, 1)$ is the evaporation rate, and $F: S \rightarrow R^+$ is a function such that $f(s) < f(s') \Rightarrow F(s) \geq F(s'), \forall s = s' \in S$. F (·) is called the fitness function. Different definitions for the set S_{upd} exist. Two common choices are $S_{upd} = S_{bsf}$, and $S_{upd} = S_{ib}$, where S_{bsf} is the best-so-far solution, that is, the best solution found since the start of the algorithm, and S_{ib} is the best solution of the current iteration. The specific implementation of the pheromone update mechanism differs across ACO variants.

IV. ACO in Mobile Ad HOC Network

The simple ant algorithm shown in the previous section illustrates different reasons why this kind of algorithms could perform well in mobile multi-hop adhoc networks. We discuss some by relating them to important properties of mobile ad-hoc networks.

- **Dynamic topology:** This property is responsible for the poor performance of many 'classical routing algorithms in mobile multi-hop adhoc networks. The ant algorithm is based on autonomous agent systems imitating individual ants. This allows a high adaptation to the current topology of the network.

- **Local work:** In contrast to other routing approaches, the ant algorithm is based only on local information, i.e. no routing tables or other information blocks have to be transmitted to other nodes of the network.

- **Link quality:** It is possible to integrate the connection/link quality into the computation of the pheromone concentration, especially into the evaporation process. This will improve the decision process with respect to the link quality. It is important to note that the approach can be modified so that nodes can also manipulate the pheromone concentration independent of the ants, e.g. if a node detects a change of the link quality.

- **Support for multi-path:** Each node has a routing table with entries for all its neighbors which also contains the pheromone concentration. The decision rule for selection of the next node is based on the pheromone concentration at the current node which is provided for each possible link. Thus, the approach supports multi-path routing.

In Ant based route discovery, the transmission delay of each link, processing delay at each node, the available bandwidth capacity of each link, and the number of hops visited are collected by the ant agents to estimate the path preference probability. Then the route with higher preference probability is established.

The following set of core properties characterizes ACO instances for routing problems:

- Providing traffic-adaptive and multipath routing.
- Relying on both passive and active information monitoring and gathering.
- Making use of stochastic components.
- Not allowing local estimates to have global impact,
- Setting up paths in a less selfish way than in pure shortest path schemes favoring load balancing.
- Showing limited sensitivity to parameter settings [15].

The route from source node to destination node changes in the MANET since it consists of mobile nodes. Detection of dynamic topology, generation of path between nodes and handling route failures are performed by the routing algorithm. It has three phases.

Route Discovery Phase: The route discovery and maintenance is done by flooding the network with ants. Both forward and backward ants are used to fill the routing tables with probabilities. These probabilities reflect the likelihood that a neighbor will forward a packet to the given destination. Also multiple paths between source and destination are created. First of all, neighbors are discovered using HELLO messages, but entries are only inserted in the routing table after receiving a backward ant from the destination node. Each neighbor receives a probable value for destination. This value is increased as a backward ant comes from that node, establishing a path towards destination. As ants are flooded, the algorithm uses sequence numbers to avoid duplicate packets. Only the greater sequence number from the same previous hop is taken into account. Forward ants with a lower sequence number are dropped [16]. All feasible paths from source node to destination node are found in this phase.

Route Maintenance Phase: The route maintenance phase is responsible for the improvement of the routes during the communication. A route can be invalid due to nodes along the route moving away or a link being broken. The broken link will conduct a local repair procedure, trying to find an alternative path to the destination while buffering all the packets it receives.

If the node successfully finds a new path to the destination, it will send all the buffered packets to the destination via the newly found route, meanwhile, a notification ant will be sent to the source to let the source node know the change of route. All nodes on the path that the notification ant visits will update their routing table to remove any invalid routes. The source will replace the related path with the path value in the notification ant. If such an alternative path can not be found, an error ant will be sent to the source node. After receiving the error ant, if the source node still needs a route to the destination, it will initiate a new forward ant to find a route to the destination [17].

Route Failure Handling: This phase is responsible for generating alternative routes in case the existing route fails. Node mobility in Ad Hoc Network may cause certain links to fail. Every packet is associated with acknowledgement, hence if a node does not receive an acknowledgement, it indicates that the link is failed. On detecting a link failure the node sends a route error message to the previous node and deactivates this path by setting the pheromone value to zero. The previous node then tries to find an alternate path to the destination. If the alternate path exists, the packet is forwarded on to that path else the node informs its neighbors to relay the packet towards source. This continues till the source is reached. On reaching the source, the source initiates a new route discovery phase. Ant algorithm provides multiple paths. If the optimal path fails, it leads to choosing next best path. Next best path will be that path with links having next highest pheromone value (second best path). Hence ant algorithm does not break down on failure of optimal path [18].

V. Conclusion

QoS routing for MANET focuses on guarantees with respect to bandwidth, cost and delay. Several techniques were proposed in ACO, but it had certain drawbacks, The mathematical and engineering problems can be solved in the existing ant based approaches. Though the nodes in the networks have different transmission ranges, ARA can find routing paths that are close to the shortest path. But here energy is not taken into account. Node connectivity and end to end delay is increased using Multi agent routing algorithm but complex optimization problems are not considered. The total overhead is reduced to some extent in the efficient ant routing protocol. To discover a route to the destination node it resumes its route discovery process from the place where it ended in the last round following a failure. More energy is consumed because of this routing process.

References

[1]. *P. Jacquet and T. Clausen, "Optimized Link State Routing Protocol", IETF Internet Draft, draft-ietf-manet-olsr-11.txt, July 2003.*

[2]. *D.B. Johnson, D.A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", Ad Hoc Networking, pp. 139-172, 2001.*

[3]. *E. Perkins and P. Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers, in Proceeds of SIGCOM'94, 1994, pp. 234-244.*

[4]. *T. Larsson and N. Hedman, Routing protocols in wireless ad hoc network – a simulation study, master thesis, 1998.*

[5]. *D. Johnson et al., The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC 4728.*

[6]. *Ad hoc On-demand Distance Vector (AODV) Routing, RFC 3561, tools.ietf.org/html/rfc3561*

[7]. *Park, S. Corson, Temporally-Ordered Routing Algorithm (TORA) Version 1, Functional Specification, Internet Draft, IETF MANET Working Group, 2001.*

[8]. *Optimized Link State Routing Protocol, RFC 3626, tools.ietf.org/html/rfc3561.*

[9]. *Cisco visual networking index: forecast and methodology, 2009-2014, http://www.cisco.com/.*

[10]. *M. Claypool and J. Tanner, The effects of jitter on the perceptual quality of video, http://web.cs.wpi.edu/~claypool/papers/jitter-pq/jitter-pq.pdf.*

[11]. *C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.*

[12]. *Marco Dorigo, Mauro Birattari, and Thomas St¨ utzle Universit´ eLibre de ruxelles, BELGIUM, Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique.*

[13]. *Dorigo M, Maniezzo V, Colorni A. Ant System: optimization by a colony of cooperating agents. IEEE Trans Syst Man and Cybern-Part B 1996;26(1):29–41*

[14]. *Dorigo M, Gambardella LM. Ant Colony Sys-tem: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1997;1(1):53–66.*

[15]. *Mesut Gunes, Udo Sorges, and Imed Bouazizi. ARA - The Ant-Colony Based Routing Algorithm for MANETs. In Stephan Olariu, editor, Proceedings of the 2002 ICPP Workshop on Ad Hoc Networks (IWAHN 2002), pages 79.85. IEEE Computer Society Press, August 2002.*

[16]. *P.Deepalakshmi1 and Dr.S.Radhakrishnan, Ant Colony Based QoS Routing Algorithm For Mobile Ad Hoc Networks, International Journal of Recent Trends in Engineering, Vol. 1, No. 1, May 2009, 459-462.*

[17]. *F.J.Arbona Bernat, Simulation of Ant Routing Protocol for Ad-hoc networks in NS-2, Delft University of Technology, November 2006.*

[18]. *Jianping Wang, Eseosa Osagie, Parimala Thulasiraman and Ruppa K. Thulasiram, HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network, Elsevier Ad Hoc Networks 7 (2009) 690–705*

[19]. *Cauvery N. K., and K. V. Viswanatha, Enhanced Ant Colony Based Algorithm for Routing in Mobile Ad Hoc Network, World Academy of Science, Engineering and Technology 2008.*