A new method for congestion avoidance in wireless mesh networks

Son Pa\textsuperscript{1}, Satria Mandala\textsuperscript{2} and Adiwijaya\textsuperscript{3}

\textsuperscript{1} National Polytechnic Institute of Cambodia
\textsuperscript{2,3} School of Computing, Telkom University

E-mail:
\textsuperscript{1} misterpa007@gmail.com,
\textsuperscript{2} satriamandala@telkomuniversity.ac.id,
\textsuperscript{3} adiwijaya@telkomuniversity.ac.id

Abstract. Wireless Mesh Networks (WMNs) will play an important role in next-generation wireless communications involving wireless networks. The traffic in this network (WMN) often saturates on certain paths, causing congestion problems to occur. Currently, many proposed protocols have been created based on Ant Colony Optimization (ACO) to contribute towards solving this particular problem. Unfortunately, most of these methods disregard the congestion problem after an optimal path is found. In this paper, a New Congestion Avoidance Method (NCAM) is proposed. NCAM is designed to improve load balancing by solving congestion problems after the optimal path is found. There are three mechanisms proposed in NCAM: detection of congestion in each optimal node to prepare a suboptimal path, updating of suboptimal pheromone value, and transferring data packets to the suboptimal path. We implemented our method in Network Simulator Version 2 and measured its effective performance compared to a family of existing ACO approaches in terms of packet throughput, end-to-end delay, and packet loss. The result demonstrates that NCAM provided better throughput, decreased end-to-end delay, and less packet loss compared to AntNet and CACO.

1. Introduction

A Wireless Mesh Network (WMN) is a network in which the nodes in the network use a mesh topology. Each node in WMN acts as a part of the whole network; they are able to perform self-configuration and self-healing. Mesh routers connect mesh nodes to the Internet, and are therefore called gateway nodes [27]. Traffic flow in the mesh network is transmitted through these gateways. There are two kinds of nodes that are used in WMN: access points (packet forwarding/serving) and router (packet forwarding). The architecture of WMN is divided into three groups: infrastructure, client WMNs, and hybrid WMNs. In the infrastructure, the network consists of mesh routers/access points with a mesh topology that allows client nodes to transmit packets through the network. Meanwhile, the client WMNs provide a direct network connection between the client devices. Finally, the hybrid WMN is a combination of backbones and client networks. The hybrid WMNs are
frequently used to provide connectivity to the Internet such as Wi-Fi, WiMAX, cellular/cellular phone, and sensor services for users.

Recently, use of the Ant Colony Optimization (ACO) framework has been considered for improving the efficiency of network performance [1] [7] [13] [16] [17]. There are four factors for improving network performance, i.e. Static mesh nodes, Interference among multi-radios, Load balancing (data flows), and Channel diversity. Statistically, nodes in a Wireless Mesh Network are typically static and have no momentous constraints on power consumption. In interference among radios, nodes use the same path of the same flow at the same time and so their signals always compete with each other. In load balancing, almost all data flowing in WMN are accessed through Internet gateways, which force the link to experience unbalanced loading and resulting in saturated data on certain paths. Channel diversity is caused by the use of various means. Almost all nodes in a WMN can be equipped with more than one radio; smart channel assignment schemes can be a good solution to effectively increase the transmission data in the network.

ACO is a type of insect algorithm and is part of the swarm intelligence (SI) group. SI uses the natural behaviors of insects such as bees, ants, wasps and termites. The idea of the Ant algorithm is inspired by how ants move from their nest to their food sources, where there are many paths from the nest to the food source. Inside each ant, there is a chemical substance used as a form of communication to explore the food source. Ants release pheromones along the trail when they go back to their nest, allowing the next ant to follow the trail, which contains a high pheromone substance. ACO is known as an artificial algorithm for finding the shortest path; this takes the benefit of ant behaviors to solve discrete optimization problems. The basic principle of ACO is to route the shortest path. When the framework is applied in WMNs, many researchers are not only considering the shortest path distance but are also focusing on the optimal path for routing. In this case, the ACO framework has to fulfill the nature of WMNs with low delay, high load, low interference (inter/intra), and channel diversity efficiency.

In 2007, a previous work [13] proposed POSANT, which implemented ACO in a mobile ad-hoc network infrastructure. Two years later (2009), another study proposed AMIRA in order to improve the load balancing in WMN [1]; this method came with the idea of selecting lesser interference in each path in the network. At the end of 2010, a protocol named SARA [7] that was designed for MANET was proposed to address the problems of high computational resources. In 2012, AntMesh [3] and SmartAnt [4] were proposed to improve load balancing in the multi-radio infrastructure of WMN. In 2015, a plan to reduce the congestion problem in WMNs was proposed [16]. The idea behind the protocol was to forward packets through different paths when the packets met congested paths. Another similar concept named CACO [17] was proposed in 2016. This is a congestion prevention problem technique in which the author built a mechanism to route packets away from congested areas in the network.

2. Related works

In this section, we provide an overview of some Ant Colony Optimization (ACO) routing algorithms and congestion avoidance in wireless networks. The implementation of the proposed ACO family has its own strengths and weaknesses in applying each implementation of ACO, which are based on the characteristics and nature of each network.

In 2007, one study [13] proposed POSANT, which was based on the ant colony routing algorithm for mobile ad-hoc network infrastructures (MANETs). The idea of the study [13] centered on how to make use of a position that focused on electronic instruments, [13] where its method is applied with GPS receivers and the ACO technique. [13] The proposed method is a reactive routing algorithm, in which the author argues about the use of position information that could reduce a huge amount of ant generation and at the same time decrease route organizing time. Position information played the main role in the work [13], and was used in the heuristics maintained at each MANET node to assist ants to choose the next neighbor to move to during path routing. The drawback of POSANT is the
transmission time of the ant in moving to its next node. The study assumed all transmission times to be the same, meaning that this algorithm ignored the common problem in WMN, namely intra-flow/inter-flow interference and packet loss.

Another distributed routing algorithm for MANET named SARA was also proposed in a previous study [7]. SARA was designed in 2010 with the objective of lowering computational complications. The particular purpose of this algorithm was to create routes on insistence, establishing these to be a reactive protocol. The characteristics of agents (ants) in SARA are that they store only the node identity information to avoid redundant overheads. Ants make the decision based on pheromone values (probabilities) only, meaning that there is no mechanism to capture network dynamics or characteristic of network as interference, saturated path, etc. that often happen in a network with high nodal densities such as the Wireless Mesh Network (WMN).

Ant Mesh routing for Interference Avoidance (AMIRA) in a previous work [1] was introduced as an interference-aware routing protocol. It was designed to improve load balancing in WMN. AMIRA is designed to avoid interference problem in WMN. AMIRA was designed by combining the local heuristics technique and meta-heuristics approach to prevent interference within and among network flows. While performing path routing, the agents in AMIRA were used for selecting lesser interference paths. To select the lesser interference paths, each node in AMIRA used MAC level information gathered from link quality measurements. The methods that AMIRA uses to measure the link quality were explained in detail in the study. However, in actual fact, AMIRA was designed only for single-radio and single-channel WMNs.

Bokhari [3] and Zaruba [4] designed AntMesh and SmartAnt, respectively. Both studies used the ACO algorithm. They also implemented a load balancing advancement similar to the study mentioned above [1]; whilst also implementing a multi-radio infrastructure. Bokhari [3] and Zaruba [4] implemented two modules: link estimation (LE) and path estimation (PE). LE was used for estimating delay and load balancing, while measuring the cost of local link nodes. PE captured and measured interference and then used this information to select a low-interference path with high channel diversity. All nodes in both studies contained two kinds of tables i.e. the pheromone table (fitness of choosing the next node) and local estimation table (quality/strength of the outgoing links). Nevertheless, both studies lacked the mechanisms to avoid congestion problems when all packets flowed on optimal paths.

Grover [16] proposed an idea to reduce congestion problems in a network. In the study, buffers at each intermediate node were identified two times. First, the capacity of the buffer on the whole was identified, on whether or not the second identity would be operated (the capacity of packets in the contain path in the buffer for the available destinations). In case both conditions were true, the packet would be forwarded through alternate paths. However, the method in the study forwarded the data packets to an alternate path for a particular source destination pair that was chosen, despite the fact that the alternate path was not checked for optimality.

In another study, a mechanism was proposed to minimize the congestion problem. The method is called the Congestion Aware Ant Colony Optimization (CACO) approach [17]. CACO covers the design of a routing framework and congestion control algorithm in Wireless Mesh Networks (WMNs). It used AntMesh techniques previously presented in a previous work [3]. The main idea of CACO is to avoid packets from flowing into congested areas when the mechanisms are identified. The disadvantages of CACO, however, are the security of packets that have to route to other paths when optimal path is congested: while routing packets away from the congestion areas, it is not guaranteed whether or not the new path are optimal paths.

To extend the ACO algorithm, AntNet was proposed [19]. The idea of the study is to find the best path in the network depending on two types of ants: the forward ant (FA) and backward ant (BA). FAs can store useful information such as paths, traffic condition, or neighbor node status. In AntNet, FAs will die after they reach the destination node. All dead FAs are replaced by BAs. The BA is used to update the pheromone table information for each intermediate node. Each node will not maintain a routing table, but is instead replaced by a probability (pheromone table). This table contains the value
of the probability that the next nodes would be selected in a routing operation. Each node has its own pheromone table that usually stores the list of reachable nodes and their pheromone value. More details on how to compute the pheromone value in an ACO AntNet framework is detailed out in the study [19].

Table 1 outlines the strengths and weaknesses of the above researches that have implemented the idea of the Ant Colony Optimization algorithm.

| Protocol   | Describe                                    | Strength                                                                 | Weakness                                                                                   |
|------------|---------------------------------------------|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| SmartAnt (2012) | - It is an efficient data forwarding scheme.  
|             | - Designed for load balancing in multi-radio in WMNs. | - Has the capability to use space/channel diversity in multi-radio WMNs.  
|             | - Can discover high throughput paths.        | - It demonstrates the result under high-load network conditions.  
|             | - Less inter/intra-flow interference.        | - Has no mechanism to reduce congestion after the optimal path is found.                   |
| SARA (2010) | - It is a distributed ant routing algorithm.  
|             | - It is designed for ad hoc networks.        | - It is helpful in achieving seamless routing in heterogeneous networks.  
|             |                                              | - Low computational complexity.                                                   |
|            |                                              | - Cannot capture the link characteristics and interference network.               |
|            |                                              | - Not suitable for WMNs.                                                        |
|            |                                              | - Not suitable for traffic moving through gateways.                           |
| AMIRA (2009) | - It is an interference-aware routing protocol.  
|             | - Designed to improve load balancing in WMNs. | - It provides high throughput, low end-to-end delay.  
|             |                                              | - It works well under network congestion.                                     |
|             |                                              | - Is designed for single radio and single channel.                           |
|             |                                              | - No mechanism to estimate path quality of neighbor nodes.                    |
| AntHocNet (2001) | - It is a hybrid multi-path algorithm. 
|                | - Designed for wireless mobile ad-hoc networks. | - It consists of reactive and proactive methods.  
|                |                                              | - It can explore new paths and get up-to-date link quality information.         |
|                |                                              | - Number of ants that need to be sent to find the destination.                |
|                |                                              | - Each ant stores a list of visited nodes.                                   |
| POSANT (2007) | - It is a reactive routing algorithm (MANETs).  
|             | - Uses position-based instruments and the ACO technique. | - Can reduce the number of ant generations.  
|             |                                              | - Reduces route time.                                                          |
|             |                                              | - It helps ants decide what next hop to take in the path discovery phase.     |
|             |                                              | - Transmission time of ant to move to its neighbors is the same for all nodes. |
|             |                                              | - It ignores packet loss, interference, or delay on link and path.             |
| CACO (2016) | - It is a congestion avoidance & routing algorithm (WMNs).  
|             | - Based on the AntMesh idea.                 | - It avoids congested areas.                                                     |
|             |                                              | - Keeps forwarding packets to reach destination nodes.                       |
|             |                                              | - No security for packets.                                                     |
|             |                                              | - It does not check whether the new path is optimal or not.                   |
| CCS (2015)  | - It is a congestion reducing protocol (MANET)  
|             | - Based on the AODV routing protocol.        | - Has a mechanism to check the buffers at the intermediate nodes to avoid congestion. |
|             |                                              | - Outside the scope of our work                                               |
|             |                                              | - Alternates particular source destination pairs without checking whether or not the alternate path is optimal. |
3. New Congestion Avoidance Method

The New Congestion Avoidance Method (NCAM) is a congestion avoidance method designed to improve load balancing by avoiding the congestion problem after the optimal path is found, after which the data packets are split to a suboptimal path. The basic operation of NCAM is based on the routing protocol described in a previous study [4]. There are four types of ants that are used in NCAM: forward ants (FAs), which travel from source to destination to discover paths, backward ants (BAs) that travel from the destination to source to update the routing tables, local forward ants (LFAs) and local backward ants (LBAs), in which both ants are used for routing suboptimal paths (on the pair of optimal nodes) when each optimal nodal pair have gone beyond the limit of data packets.

3.1. How NCAM Work

The concept of NCAM is based on SmartAnt [4], after the optimal path is found. NCAM adds three mechanisms to work for congestion control, as illustrated in figure 1.

![Block diagram of NCAM](image)

Figure 1. Block diagram of NCAM.
Step 1: When an agent finds an optimal path, most data packets will follow the optimal path. At the same time, the level of congestion is detected to prepare a suboptimal path.

Step 2: When the buffer for each optimal nodal path goes over the threshold, local forward ants (LFA) and local backward ants (LBA) will be used to find the suboptimal path by following the rules of the algorithm process, as described in a previous work [4].

Step 3: When the suboptimal path is found, newly arrived data packets will follow the new path to the destination node.

Step 4: In case the suboptimal path goes over the threshold, Steps 2 and 3 will be repeated to explore other optimal paths.

3.2. Detect congestion

In WMN, congestion can occur at any intermediate node; thus, deteriorating the performance of the network. This is a situation when a link or a node carries excessive load that leads to the deterioration of quality of service. Furthermore, when the routing protocol determines the same best path due to the traffic load occurring on the same path while the other path is seldom used, packet loss will result; therefore degrading the performance of the network [15]. In this research, when all packets go forward through optimal paths, as explored by the agent, the congestion level in each intermediate node would be measured by detecting the queue space, i.e. if the number is bigger than or equal 75%. If the number of packets in the buffer is equal or more than a certain threshold, LFAs and LBAs will be used to explore the suboptimal path of that node pair. After the suboptimal path is found, the data packet would be separated and then forwarded through the new path, as explained in figure 2.

![Figure 2. Detect congestion.](image-url)
3.3. Transfer data to suboptimal path

Applying the behavior of real ants [4] [18] [20], these useful characteristics are useful resources for designing a mechanism to reduce data packets in a pair of optimal nodes when the load on the network is greater than the capacity of the network. In NCAM, strategies to reduce data packets from busy optimal pair nodes to a suboptimal path include assuming that while all packets are generated on the optimal path, a few of the nodal paths have spilled over the threshold, as per figure 3:

![Diagram of transfer data to suboptimal path]

**Figure 3.** Transfer data to suboptimal path.
**First step:** LFAs and LBAs are used to find the suboptimal nodal path.

**Second step:** When the suboptimal path is found, the probabilities (pheromone value) of each optimal node will be decreased by 30%.

**Third step:** All suboptimal intermediate nodes will update their pheromone table by copying the optimal intermediate node, so the optimal path and suboptimal path will have the same pheromone value (the probabilities would be 50% among both).

**Fourth step:** Newly arrived data packets will randomly select the previous optimal path and the new suboptimal path, but this depends on the width and how much lesser the data packets are on the suboptimal path, with all data packets moving faster on the suboptimal path than the optimal path. Thus, the probabilities of choosing the suboptimal path increases, while the probabilities for choosing an optimal path decreases.

3.4. Update pheromone table on suboptimal node

In NCAM, after the suboptimal path is found, each suboptimal intermediate node will update their pheromone table by copying the optimal node that has been stuck in congestion.

Let us denote $P_{i,d}$ as the pheromone values in the pheromone table corresponding to a particular destination $d$ via next hop $i$, whereas $P_{j,d}$ is the pheromone values in the pheromone table corresponding to a particular destination $d$ via other next hop $j$. The pheromone values in the suboptimal intermediate node ($P_{j,d}$) can be updated by equations (1) and (2):

$$P_{j,d} = P_{i,d}$$  \hspace{1cm} (1)

$$P_{i,d} = \frac{P_{i,d} + \Delta P}{1 + \Delta P}$$  \hspace{1cm} (2)

![Figure 4. Update pheromone.](image-url)
Where, $\Delta p$ is the reinforcement value that will be added to the pheromone table and depends upon how good $Trip_{i,d}$ is.

When the pheromone substance in suboptimal intermediate nodes is equal to the pheromone substance in the optimal intermediate node, the optimal path and suboptimal path will have the same probabilities (50%), so the newly arrived packets will randomly choose both paths. Because of the width of the suboptimal path, the probabilities of the suboptimal path increases while the optimal path probability decreases.

4. Performance evaluation

In this research, Network Simulator Version 2 (NS-2) was selected for the implementation of our scenarios. In our scenarios, the discussions mainly revolved around throughput performance, end-to-end delay, and packet loss. There are 12 nodes, with source nodes and destination nodes denoted by 3 and 11, respectively, and are part of an incomplete mesh network topology. Table 2 details our simulation setup.

Table 2. Simulation setup.

| No | Requirement          | Detail   |
|----|----------------------|----------|
| 1  | Ubuntu 14.04         | 32 bits  |
| 2  | Simulator            | NS-2     |
| 3  | Simulation time      | 300 seconds |
| 4  | Protocol agent       | TCP      |
| 5  | Link bandwidth       | 1 Mbps   |
| 6  | Traffic type         | CBR      |
| 7  | Packet size          | 1000 bytes |
| 8  | Number of nodes      | 12       |
| 9  | Buffer size          | 10-50 packets |

Table 3 shows the effective results of using the NCAM method compared with AntNet and CACO.

Table 3. Simulation result of AntNet, CACO, NCAM.

| Simulation result          | AntNet        | CACO         | NCAM        |
|---------------------------|---------------|--------------|-------------|
| Packet delivery radio (packet) | 200 (20%)     | 684 (68.4%)  | 726 (72.6%) |
| Number of packets lost (packet) | 15 (7.5%)     | 16 (2.5%)    | 11 (1.5%)   |
| Average delay (seconds)    | 0.1111 (800bts) | 0.0981 (800bts) | 0.0653 (800bts) |
Figure 5 below illustrates the throughput comparison graph of AntNet, CACO, and NCAM respectively represented by black, brown, and blue lines. The X-axis represents the duration of sequence simulation time in minutes (m) and the Y-axis represents number of packets that are successfully delivered to node 11. When the packets start forwarding from node 3 to node 11 at = 2 s, [19] [17] the NCAM is found to yield the same throughput performance. At t = 1.29 m when paths 7-11 become congested, AntNet does not have any mechanism to handle this situation [19] while in CACO [17], the packets are routed away from paths 7-11. Meanwhile, the NCAM will split some packets to paths 7-6-10-11. Because NCAM could select a path with less host nodes than AntNet [19] and the packets were transmitted through optimal path and suboptimal paths, NCAM resulted in more packet throughput than AntNet and CACO. The blue line of figure 5 illustrates the effectiveness of packet throughput using the NCAM method, which is better than AntNet [19] and CACO [17].

Figure 6 shows the different delays of AntNet, CACO, and NCAM. The X-axis represents the duration of sequence simulation time in minutes (m) and the Y-axis represents the end-to-end delay in seconds (s). The average end-to-end delay using AntNet [19], CACO [19], and NCAM [17] is 0.11 s, is 0.099 s, and 0.065 s, respectively. NCAM gave a lower end-to-end delay than AntNet [19] and CACO [17] based on the blue line statistical data showed in figure 6. When paths 7-11 became congested, all packets in AntNet [19] continually waited on paths 3-7, while in CACO, the packets were routed to neighbor node 7 and NCAM tried to measure the level of congestion on paths 7-11 before it explored a suboptimal path an then split newly arrived packets to that suboptimal path.

Figure 7 demonstrates the percentage of packet loss in each scenario. We calculated our packet loss from the statistical data taken from the ‘out.tr’ file. The percentage of packet loss of NCAM was found to be lower than that of AntNet and CACO. The result showed that NCAM provided better throughput compared to AntNet and CACO. CACO yielded no packet drops, similar to NCAM, while 10 packets were dropped in AntNet. The number of packets lost in NCAM was less than 4 and 5 compared to AntNet and CACO, respectively. Furthermore, end-to-end delay using the NCAM method was only 0.0653 s. The increase in packet throughput in NCAM is due to the capturing of level of packet data inside the queue and in front of the queue. Although routing algorithms proposed by previous works [1], [2], [3], [4], [8], [17], [19] were able to find the optimal path based on specific strategies, the congestion avoidance after a suboptimal path is found should be included with these algorithms to guarantee high-efficiency packet transmission.

Figure 5. Result of packet throughput performance.
5. Conclusion

In this paper, a method for avoiding the congestion problem after an optimal path is found in the Wireless Mesh Network (WMN) was proposed. The proposed method is called the New Congestion Avoidance Method (NCAM) and contains three mechanisms: detection of congestion level, transferring of packets to a new path, and updating the pheromone value. The result showed that NCAM outperformed AntNet and CACO. Packets lost in NCAM were 5.99% less than AntNet and
0.82% less than COCA. Furthermore, end-to-end delay using the NCAM method took only 0.0653 s while AntNet took 0.111 s and CACO took 0.098 s. The increase in packet throughput, lower end-to-end delay, and less packet loss using NCAM was due to its capture of packets inside the queue, incorporation of a mechanism to transfer packets to a suboptimal path, and implementation of a mechanism to update pheromone value in each sub-intermediate node.

NCAM was based on the SmartAnt protocol and was able to perform better compared to other protocols. It also yielded a higher throughput, less packet loss, and lower end-to-end delay. However, there are several aspects that could not be handled in this study due to:

- Input data traffic rate that went over the capacity of the output lines
- Abilities of the mesh node
- Security of network attack

If this study were able to control the rate of input data traffic compared to the capacity of output lines, congestion on certain paths would decrease. In addition, if the mesh nodes were to have better ability to perform book-keeping tasks (queuing buffers, updating tables, etc.), the packets would reach destination nodes more effectively. On the other hand, if the data packets were kept safe from different types of network attacks, positive effects such as improved throughput, reduced packet loss, and cutting down of end-to-end delay in the network could result.

References

[1] Bokhari F and Zaruba G 2009 April AMIRA: interference-aware routing using ant colony optimization in wireless mesh networks In Wireless Communications and Networking Conference 2009 WCNC 2009 IEEE (pp. 1-6) IEEE

[2] Zhu J Liu F Wang X and Rong L 2008 November A congestion and interference aware routing metric for Wireless Mesh networks In Communication Technology 2008 ICCT 2008 11th IEEE International Conference on (pp. 62-65) IEEE

[3] Bokhari F and Zaruba G 2010 November AntMesh: an efficient data forwarding scheme for load balancing in multi-radio infrastructure mesh networks In Mobile Adhoc and Sensor Systems (MASS) 2010 IEEE 7th International Conference on (pp. 558-563) IEEE

[4] Bokhari F and Zaruba G 2012 On the use of smart ants for efficient routing in Wireless Mesh Networks arXiv preprint arXiv:1209.0550

[5] Akyildiz I F Wang X and Wang W 2005 Wireless mesh networks: a survey Computer networks 47(4) pp.445-487

[6] Berliner B Clark B and Hartono A QoS Requirements of Multimedia Applications

[7] Correia F and Vazão T 2010 Simple ant routing algorithm strategies for a (Multipurpose) MANET model Ad Hoc Networks 8(8) pp.810-823

[8] Di Caro G and Dorigo M 1998 AntNet: Distributed stigmergetic control for communications networks Journal of Artificial Intelligence Research 9 pp.317-365

[9] Di Caro G Ducatelle F and Gambardella L M 2005 AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks European Transactions on Telecommunications 16(5) pp.443-455

[12] Dorigo M Di Caro G and Gambardella L M 1999 Ant algorithms for discrete optimization Artificial life 5(2) pp.137-172
[13] Kamali S and Opatrny J 2007 March Posant: A position based ant colony routing algorithm for mobile ad-hoc networks In *Wireless and Mobile Communications 2007 ICWMC’07 Third International Conference on* (pp. 21-21) IEEE

[14] Gopalakrishnan P Spasojevic P Greenstein L and Seskar I 2004 September A method for predicting the throughput characteristics of rate-adaptive wireless lans In *Vehicular Technology Conference 2004 VTC2004-Fall 2004 IEEE 60th* (Vol. 6, pp. 4528-4532) IEEE

[15] Grover P Singh P and Rani S 2015 Congestion Reduction in Wireless Mesh Networks *International Journal of Computer Applications* 124(13)

[16] Grover Payal Paramjeet Singh and Shaveta Rani "*Congestion Reduction in Wireless Mesh Networks*" *International Journal of Computer Applications* 124.13 (2015)

[17] Mani S and Ponraj R Optimization With Congestion Aware Routing In Mesh Topology Optimization With Congestion Aware Routing In Mesh Topology

[18] Ma L Xu Y Meng W and Li C 2011 December An Ant Colony Based Congestion Elusion Routing Scheme for MANET In *Global Telecommunications Conference (GLOBECOM 2011)* 2011 IEEE (pp. 1-5) IEEE

[19] Gupta A K Sadawarti H and Verma A K 2012 Computation of pheromone values in antnet algorithm *International Journal of Computer Network and Information Security* 4(9) p.47

[20] Dorigo M Birattari M and Stutzle T 2006 Ant colony optimization *IEEE computational intelligence magazine* 1(4) pp.28-39