Adaptation of the Ant Colony Algorithm to Avoid Congestion in Wireless Mesh Networks

Fadhil Mohammed Salman[1,*],
Ahssan Ahmmed Mohammed Lehmoud[1] and Fanar Ali Joda[2]

[1]*Ministry of Education, Babylon Education Directorate, Iraq*
[2]*Department of Air Conditioning & Refrigeration Engineering Techniques, Al-Mustaqbal University, Iraq. Ministry of Education, Babylon Education Directorate, Babylon, Iraq*
E-mail: fadhilmohammad2023@bab.epedu.gov.iq; ahssan_ahsan@bab.epedu.gov.iq; fanaralijoda@uomus.edu.iq
Corresponding Author

Received 02 May 2023; Accepted 27 June 2023;
Publication 12 August 2023

Abstract

Wireless mesh networks have recently presented a promising environment for many researchers to develop large-scale wireless communication. Traffic in WMNs often suffers from congestion due to heavy traffic load's saturation of certain routes. Therefore, this article proposes an efficient approach for congestion awareness and load balancing in WMNs, based on the Ant Colony Optimization (ACO) approach. The proposed approach aims to raise the performance of the WMN by distributing the traffic load between optimal routes and avoiding severe traffic congestion. The proposed approach relies on three basic mechanisms: detection of severe congestion within the ideal paths used for data transmission, creation of ideal secondary paths with updated pheromone values, and distribution of the traffic load (data packet flow) between the primary and secondary ideal paths. According to the results

Journal of Cyber Security and Mobility, Vol. 12_5, 785–812.
doi: 10.13052/jcsm2245-1439.1258

of the NS2 simulator, the suggested approach increased the WMN throughput by 14.8% when compared to the CACO approach and by 37% when employing the WCETT approach. The results also showed that the proposed approach achieved an average end-to-end delay closing of 0.0562, while WCETT and CACO approaches achieved an average end-to-end delay close to 0.1021 and 0.0976, respectively. The results indicated that the proposed approach achieved a lower percentage of dropped packets by 6.97% and 0.99% compared to the WCETT and CACO approaches. Thus, the proposed approach is effective in improving the performance of WMNs.

Keywords: WMNs, ACO, congestion, traffic load, NS2.

1 Introduction

The great development in wireless communications and the urgent need to use broadband Internet in various rural and urban environments led to the production of a promising technology that works as an infrastructure for Internet communications called wireless mesh networks (WMNs) [1]. WMNs are widely adopted in wireless communications because they are characterized by self-configuration, self-adaptive, dynamic self-organization, ease of implementation, and low cost [2]. Wireless mesh networks mostly connect three types of devices: mesh routers, mesh gateways, and mesh clients, as shown in Figure 1 [3]. Mesh routers with mesh gateways provide a direct connection to the Internet and support as an infrastructure meshing (backbone) to the mesh clients. WMN infrastructures can be based on different radio technologies, but IEEE 802.11 radio technology is often used [4].

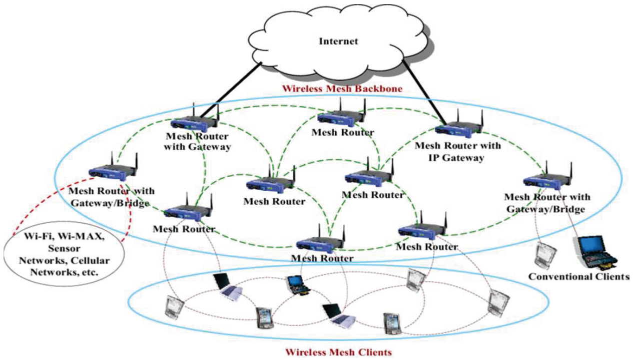

Figure 1 Wireless mesh network architecture [3].

Despite the many advantages of WMNs, they still need to improve their performance, such as heavy traffic, interference with communication channels, and congestion. Congestion is one of the problems that cause significant packet loss because it is caused each time the routing protocol chooses the same paths to send data (routing protocols often use shorter paths) [5]. Many routing algorithms based on the Ant Colony Optimization (ACO) approach have been developed to raise the performance of WMNs in terms of data flow load balancing, interferences between multiple radio devices, and communication channel diversity. ACO is a class of swarm intelligence (SI) groups that are inspired by the idea of creating routing algorithms based on observing the behavior of insects (such as wasps, aphids, bees, and termites) in how they navigate and follow many paths from their nests to food sources and vice versa. Each ant (insect) contains a chemical (called a pheromone) used to communicate with the rest of the swarm members when searching for food sources. When the insect (ant) releases the chemical (pheromones) along the path back to the nest, allowing the rest of the swarm members to follow the trail with the high pheromones. ACO is defined as an artificial intelligence algorithm that relies on choosing the shortest paths when routing and taking advantage of the behaviors of ants to find solutions to discrete optimization problems. When ACO algorithms are used in the field of wireless mesh networks, the focus should be not only on finding the shortest path but choosing the optimal path with the least cost (less traffic load, less delay, less interference (intra /inter) and effective channel diversity) [6, 7, and 8]. This work proposes an effective congestion detection and traffic redirection approach to equipoise the traffic load between optimal routes and avoid network congestion. The suggested technique is based on ant colony optimization (ACO) [9] mechanisms, exploiting intelligent ants to perform simultaneous and probabilistic data routing, forwarding data in cases of congestion by detecting ideal secondary paths, and distributing data flow between primary and secondary ideal paths.

In the proposed approach, the congestion conditions in the paths used for data transmission are detected by examining the buffers of the intermediate nodes and comparing them to a certain threshold; if the ratios exceed the threshold value, the torrent of new data packets will be redirected to an ideal secondary path that is determined using intelligent ants. Simulation results in Network Simulator version 2 show that the new approach increased WMN throughput, reduced end to end delay time, and reduced the number of dropped packet compared to other popular algorithms of the same family. The structure of this work is as follows: Section 1 presents WMNs. Section 2 summarizes an overview of some relevant technologies. Section 3 states the motivations for the work. Section 4 reviews an overview of the suggested approach. Section 5 provides details of the suggested algorithm. Section 6 explains the evaluation of experiments. Section 7 outlines the conclusions and future works.

2 Literature Survey

This section reviews the most important and common previous literature, which adopted the optimal routing approach for ant colony optimization (ACO) to overcome the routing problems. Table 1 illustrates the strengths and weaknesses of some relevant previous work.

Table 1 Shows the properties of some related algorithms

Algorithm	Description	Weakness	Strength
Smart ant (WMNs) [10] (2011)	– Designed to develop rules for updating pheromone tables.	– It did not take into account the problem of intra and inter-flow interference.	– It detects the routing path with the lowest traffic load, highest bandwidth, and lowest number of hops.
AntNet (WMNs) [11] (2012)	– Designed to determine the best routing routes between origin and target devices based on the total pheromone levels along the routes.	– It is made to work with just one radio and one channel. – It has no mechanism for dealing with congestion after using the perfect path.	– ACO algorithm mechanisms are adopted to reduce inter/intra interference. – It has few computational complications.
Smart ant (WMNs) [12] (2014)	– Designed to avoid congestion in WMNs. – uses ant colony optimization (CACO) mechanisms to detect network congestion.	– Not using intelligent agents to choose the ideal alternative paths. – Its mechanisms are integrated with the DSR, a well-known Ad-hoc protocol.	– Use of (CACO) interference aware, antmesh algorithm. – Strong against the problem of dynamic network routing.
AILCC (WMNs) [13] (2015)	– Designed for congestion control and appropriate bandwidth utilization.	– It has no mechanism for dealing with congestion after using the perfect path. – There is no packet security.	– An intelligent ant approach is used to control congestion conditions to suit appropriate bandwidth requirements.

(*Continued*)

Table 1 Continued

Algorithm	Description	Weakness	Strength
EAOMDV-LB (WMNs) [14] (2015)	– Designed for congestion avoidance based on a multi-path concept. – It has an interference control mechanism.	– There is no method to guess the path quality of neighbors. – It is made to work with just one radio and one channel.	– It reduced the end-to-end delay, increased the network throughput, and reduced the traffic load.
Smart ant (WSNs) [15] (2019)	– Designed to choose the lowest-cost routing path.	– It is not applied to WMNs. – Network congestion is not considered.	– Adaptation to changing network topologies. – Effective in increasing network throughput and reducing energy consumption.
Ant colony (WMNs) [16] (2019)	– Designed to improve ant colony algorithm. – It is based on an optimizing energy-saving strategy.	– It did not consider the possibility of congestion in the alternative ideal paths. – Each intelligent agent stores a list of visited devices.	– It uses the ACO algorithm's mechanisms to find the shortest path. – It considers the nodes' traffic congestion and residual power to extend the network's life.
CA-ACO (WMNs) [17] (2020)	– A congestion-aware multi-path routing algorithm based on the principle of fortress ants. – It is an improvement to the ACO approach.	– It does not check if the new sub-paths are congested or if there are interferences situations. – The number of agents required to discover the target.	– It uses ACO principles to find and arrange ideal paths. – It increases the speed of perfect routing based on the principle of elite ants.
ECA-HA WSNs [18] (2021)	– An effective model to avoid congestion in the ideal paths – Designed by combining resource-oriented and traffic-oriented optimization.	– Not appropriate for WMNs. – It does not consider computational complexities, interferences, network topology change, and network expansion.	– It uses ant colony optimization and the Huffman coding approach. – It takes advantage of ACO's approach to finding side paths and avoiding congestion.
Smart ant WMNs [19] (2021)	– Designed for efficient routing that relies on ant colony optimization.	– It uses a large number of smart agents, which consumes more energy and time. – Inter/intra interferences.	– Rely on the pheromone table to find the perfect paths. – It avoids the congestion of the ideal routing paths.

3 Motivations

The motive for completing this work is the importance of WMNs as promising wireless communication networks for different applications such as enterprise networks, wireless community and dynamic networks, building automation, broadband home wired networks, etc. The importance of WMNs lies in their many advantages, such as low cost, ease of installation, durability, ease of support and maintenance, wide coverage, reliable service, and adequate bandwidth. Therefore, WMNs allow anyone always to be connected to the Internet anytime and anywhere. In addition to the possibility of integrating WMNs with different wireless networks (such as wireless sensor networks, cellular networks, etc.) through bridges and gateways. Since a lot of traffic in WMNs is routed across mesh routers to access the Internet, most traffic is either coming from network clients going to gateways or from gateways going to clients. When many routers exploit the same optimal path to direct traffic to the gateways, this leads to overload on the optimal path and thus reduces network performance (The network throughput decreases, the packet delivery fraction decreases, the number of lost packets increases, the power consumption increases, etc.). Therefore, network congestion and load balancing have become the most important challenges affecting the quality of services in mesh networks. From this point of view, the main motivation for adapting to network congestion situations is to provide an efficient and reliable data transfer rate. That motivated the development of an efficient congestion detection and traffic redirection algorithm to balance the traffic load between optimal routes and avoid network congestion. The suggested approach mechanisms can help enhance the performance of the WMN by avoiding routing the traffic flow through congested paths and distributing the traffic load to other sub-optimal paths.

4 An Overview of the Proposed Model

In this part and subsequent subsections, we will explain the fine specifics of the suggested approach, designed to increase the throughput and performance of WMNs through congestion avoidance and improved network load balancing. The proposed approach uses the principle of intelligent agents to discover ideal data routing paths and secondary paths according to the ant colony optimization (ACO) algorithm. The standard operations of the proposed algorithm are designed and implemented according to the intelligent ants routing algorithms described in [20, 21, and 22]. The proposed algorithm generates intelligent ants as data packets that travel from the origin device to the target device through different network paths to detect and choose the optimal routing paths. The proposed algorithm generates six types of intelligent ants:

- Forward intelligent ants (FIAs) are the type that rushes from the source node toward the target node and is responsible for detecting the different paths.
- Backward intelligent ants (BIAs) are the type that bounces back from the target node back to the source node and are responsible for updating the routing tables of the nodes they pass through.
- Information intelligent ants (IIAs) in this class are responsible for populating the link estimation table by collecting local link quality information.
- Topical forward intelligent ants (TFIAs): They are the ants responsible for discovering the ideal secondary paths (from the sender node to the target node) when the number of data packets exceeds the queue limits in the primary ideal path nodes (congestion situation occurs in the data packets).
- Topical backward intelligent ants (TBIAs) are the type that bounces back from the target device to the origin device via the new paths and are responsible for updating the routing tables of the nodes in the ideal secondary path.
- Topical information intelligent ants (TIIAs) this class is responsible for populating the link estimation table by collecting local secondary links quality information.

4.1 Data Tables

The proposed algorithm used three types of data tables in each node of the network to perform ideal and secondary-ideal routing path selection operations, as shown below:

1. Probabilistic data structure (pheromone table): This table includes the probability values of neighboring nodes that will be selected as the next hop nodes to reach the target. In other words, this data structure includes pheromone track information to direct data from the source node to the target node, passing through the specified intermediate nodes. In addition, this data structure contains routing information (a list of nodes that create the path) and the pheromone values [23, 24, and 25]. So, this data structure includes the following information fields:
    - Target-ID: Contains information indicating the address of the target node.

- Next-hop-ID: This field contains information about the address of the next adjacent node used to reach the target node.
- Pheromone values field: This field contains the pheromone values the respective node relies on to calculate the probabilities of neighboring nodes to be elected as the next hop node to reach the target node.

The probabilistic table at a given node *N* contains R_N rows (here $R_N = |S.N_N|$, ($S.N_N$ refers to the set of nodes adjacent to the given node)). Each row in the pheromone table has *C* columns (the columns represent the total possible paths to the target node (all population minus one)). Suppose $Pro_{L.T}$ is the probabilites of transmitting a data packets to target *T* across link *L*. The following relationship exhibits to each column in the pheromone tables at device *k*. Therefore, the following mathematical model can be formulated for each column in the probability table at the given node *N*:

$$\sum_{L \in S.N_N} Pro_{L.T} = 1 \quad \forall\, T \in \ [1 \ldots C] \tag{1}$$

2. Delay data structure: In this table, the rate delay time is saved for every trip between the origin device and the target device passing through the intermediate devices. Therefore, this data structure will contain *N* entries for each trip between the source and target node. The delay rates values kept in this table are taken from the delay rates of *M* number of received intelligent ants.
3. Link estimation table: This data structure contains information about the strength and quality of the links between the respective node and its neighboring nodes. The proposed algorithm exploits the information intelligent ants (IIAs) to measure the strength levels of links between nodes, taking into account the delay rate of sending data packets over the given link.

4.2 Next-hop Selection Rule

The proposed algorithm used the pseudo-random next-hop transmission rule to choose the optimal routing path that guarantees the improvement of WMN throughput. Suppose the intelligent forward and (*a*) at a certain intermediate node (*n*) selects the next hop node (*h*) in order to reach the specified target node (*t*); the selection process can be formulated according to the following relations [26, 27, and 28]:

$$h = \begin{cases} \arg Max_{h \in S.N_n}\{ph_a(t,h)\} & if\ r \leq r_0 \\ r_a(t,h) & Otherwise \end{cases} \tag{2}$$

$$r_a(t,h) = \frac{ph_a(t,h)}{\sum_{i \in S.N_n} ph_a(t,i)} \tag{3}$$

Here, (*r*) represents a random number that ranges from zero to one, while (r_0) denotes a constant number that ranges from zero to one. $Ph_a\ (t,h)$ represents the amount of pheromone on the link connecting node (*n*) to the next hop (*h*) for reaching the target node (*t*). According to Model (2), which refers to the state of ($r \leq r_0$), the node with the highest pheromone rate (compared to its neighbors) will be elected as the next hop node; otherwise, the next hop (*h*) will be elected proportionally among the neighbors according to the distribution probability of neighboring nodes $r_a(t,h)$, as indicated in Model (3).

5 The Proposed Congestion Awareness and Load-Balancing Algorithm

The proposed algorithm implements its basic operation based on the principles of the intelligent ants routing algorithms described in [20, 21, and 22]. According to the algorithms of intelligent ants, swarms of intelligent ants (FIAs, BIAs, and IIAs) will use to choose the primary ideal routing path, taking into account the intra and inter-flow interference in the communication channels. After determining the ideal routing path, the proposed algorithm will add three important operations for network congestion control: congestion detection in the ideal primary path, secondary ideal path selection (when a congestion condition is detected), data structure information update, and traffic load distribution between the primary and secondary ideal paths. The following flowchart (Figure 2) summarizes the three basic operations of the proposed algorithm.

The following Algorithm 1 outlines, in general terms, the basic stages of the proposed approach.

5.1 Congestion Detection Stage

In WMNs, data packet congestion is caused by many connected devices and the wide interlocking topology of the network [29]. WMN performance is

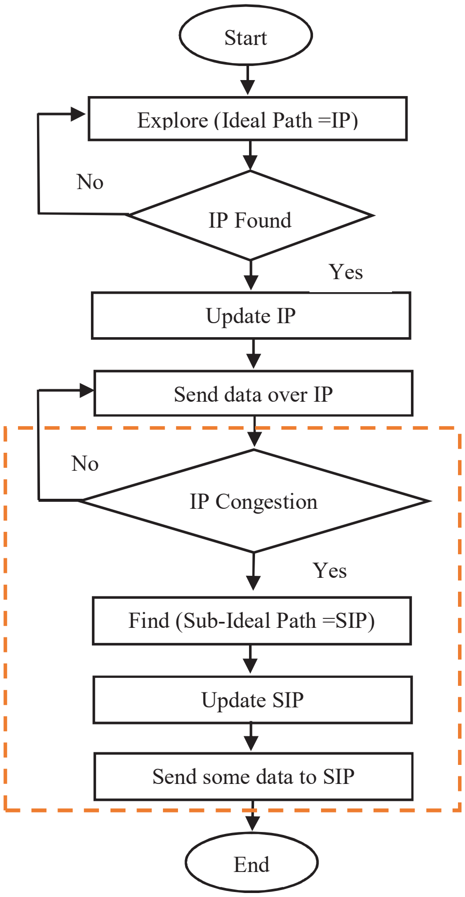

Figure 2 Flow chart of the proposed system.

negatively affected when congestion occurs on the intermediate devices of the routing path. In addition, the routing path bears a heavy load with congestion situations, which increases the amount of loss packets; thus, the network's throughputs and service quality decrease [30]. Sometimes, congestion occurs on the routing path because the routing protocol chooses the same ideal path to send data more than once, and fewer alternative paths are used, which leads to an increase in traffic load on the ideal path and an increase in dropped data packets [31, 32]. In the proposed algorithm, when intelligent ants elect the optimal path and data is exchanged, the congestion level of intermediate nodes is measured based on two parameters (queue length and transmission

Algorithm 1 Proposed algorithm

Step 1. *Detection of the optimal initial routing path using smart ant swarms (FIAs, BIAs, and IIAs), according to the mechanisms described in [20, 21, and 22].*

Step 2. *Update data structures with routing information (updating the pheromone table values of the selected intermediate nodes).*

Step 3. *Routing data packets through the primary ideal routing route from the sender device to the target device, passing through the intermediate devices.*

Step 4. *When a flow of data packets is transmitted along the ideal primary path, at the same time, the congestion states of the intermediate nodes are checked periodically depending on two parameters (queue length and transmission ratio).*

Step 5. *When a congestion situation is detected in the ideal path nodes, they flood the network with congestion information until they reach the WMN (mesh routers) edge to prepare for secondary ideal path detection.*

Step 6. *Detection of the secondary optimal initial routing path using smart ant swarms (TFIAs, TBIAs, and TIIAs), according to the mechanisms described in [20, 21, and 22].*

Step 7. *Update the values of the data structures of the secondary ideal path nodes according to the data structures of the primary optimal path. (Update the pheromone table values of the ideal secondary path depending on the pheromone table values of the ideal primary path to increase the probability that new nodes will choose for the next hop).*

Step 8. *Route new arrival data packets along the secondary ideal routing route from the origin device to the target device, passing through the intermediate devices.*

Step 9. *Detect congestion conditions in new nodes by comparing the ratio of parameters (queue length and transmission rate) with a certain threshold value.*

Step 10. *If there is congestion in the nodes of the ideal secondary path, a new optimal path will be selected by repeating steps 6, step 7, and step 8.*

rate). Each intermediate node checks the congestion level periodically, so if the percentage of the congestion level is higher than a certain threshold value, it will broadcast messages declaring the congestion status until it reaches the mesh routers. The level of congestion at the nodes of the routing path can be measured mathematically as follows:

$$C.l_n = \frac{Qu.l_n}{T.R_n} \tag{4}$$

Here, $(C.L_n)$ represents the congestion level at a given node (n). $(Qu.l_n)$ denotes the average length of the queue at a given node (n). $(T.R_n)$ refers to the transmission rate at a particular node (n) in the routing path. So, if $(C.L_n \geq Th$ (Congestion level threshold)), the mesh router will report the congestion condition; otherwise, the packet stream will continue to be sent along the routing path.

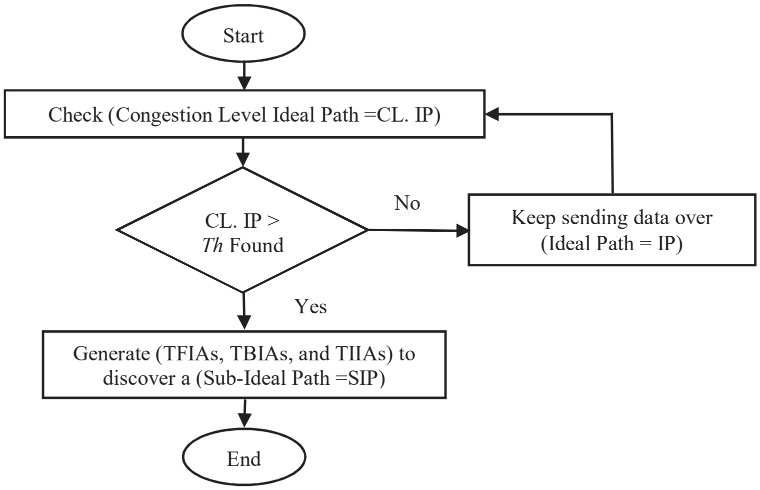
Start
Check (Congestion Level Ideal Path =CL. IP)
CL. IP > Th Found
No
Keep sending data over (Ideal Path = IP)
Yes
Generate (TFIAs, TBIAs, and TIIAs) to discover a (Sub-Ideal Path =SIP)
End

Figure 3 Flow chart of network congestion detection.

According to the previous one, the proposed algorithm will allow each node to check the congestion level periodically, so if the average congestion level is above or equal to the specified threshold, it will use the intelligent ants swarm (TFIAs, TBIAs, and TIIAs) to discover the secondary ideal routing path. After the ideal secondary path is determined, the pheromone table of the new path will be updated according to the ideal primary path. Then the received packets will be forwarded via the ideal secondary path. This method will equipoise the traffic load of network and reduce the exchange of periodic congestion information between nodes and their neighbors, conserving network bandwidth. The following flowchart (Figure 3) summarizes the network congestion detection.

The following Algorithm 2 describes the network congestion detection and load balancing process.

5.2 Transmit Data to Secondary Ideal Routing Path

When there are congestion conditions in some nodes of the ideal path, the suggested model will exploit the behavior of real ants described in [33, 34, and 35] to create a mechanism that reduces the traffic load of data packets when there is a load larger than the capacity of nodes in the ideal path. Therefore, the mechanism adopted in the suggested approach is to balance the flow of data packets between the ideal primary and secondary paths, assuming that some nodes of the ideal primary path exceed the congestion

Algorithm 2 Congestion detection algorithm

Step 1. *Each node in the ideal primary path periodically checks the congestion level (according to the ratio between the queue length and the transmission rate) using information stored in the congestion level table.*

Step 2. *If the congestion level in all nodes is below a certain threshold, the packet traffic will continue through the ideal main path; otherwise, go to the next step.*

Step 3. *Generate intelligent agent swarms (TFIAs, TBIAs, and TIIAs) to discover a secondary ideal routing path.*

Step 4. *Update the data structures of the secondary ideal path nodes according to the data structures of the primary ideal path nodes.*

Step 5. *Balance the traffic load between the chosen paths bypassing the flow of newly received data packets over the new ideal secondary path.*

Step 6. *Repeat steps (step 1, step 2, step 3, step 4, and step 5) on the secondary ideal routing path.*

level threshold. The following flowchart (Figure 4) summarizes the process of transmitting data over the secondary ideal routing path.

The following Algorithm 3 can achieve the last objective.

5.3 Update Probabilities Tables on Secondary Ideal Path

After the ideal secondary path is determined, the pheromone tables from the intermediate nodes of the ideal primary path will be copied to update the pheromone tables of the intermediate nodes of the ideal secondary path. Subsequently, the pheromone tables of the ideal secondary path will be updated by intelligent ants (TBIAs) after probabilities in secondary path nodes are upgraded and downgraded in primary path nodes. The information is stored in the routing table (*R.T*) like probability values are stored in the distance-vector algorithm [24, 36]. Therefore, the routing table (*R.T*) of the target-neighbors pair (*T, N*) stores the probability values (Pro_{nt}) for the adjacent nodes to determine the next hop (*n*) and reach the target node (*t*) as follows:

$$\sum_{n \in N_a} Pro_{nt} = 1,\ t \in [1, N],\ N_a = \{adjacent\ nodes(a)\} \tag{5}$$

Now, to illustrate the process of updating the probabilities values (pheromone values of the routing path nodes), we assume ($Ph_{n,t}$) represents the pheromone value (stored in the pheromone table) of the next hop node (*n*) to reach the target node (*t*), within the ideal primary path. Whereas suppose ($Ph_{m,t}$) represents the pheromone value of the next hop node (*m*) to reach the target node (*t*) within the ideal secondary path. Therefore, in congestion

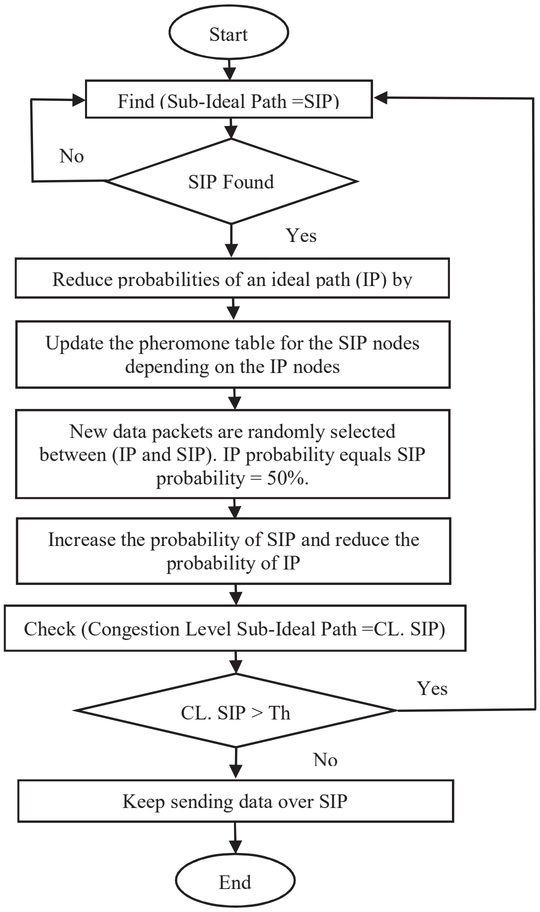
Start
Find (Sub-Ideal Path =SIP)
No
SIP Found
Yes
Reduce probabilities of an ideal path (IP) by
Update the pheromone table for the SIP nodes depending on the IP nodes
New data packets are randomly selected between (IP and SIP). IP probability equals SIP probability = 50%.
Increase the probability of SIP and reduce the probability of IP
Check (Congestion Level Sub-Ideal Path =CL. SIP)
Yes
CL. SIP > Th
No
Keep sending data over SIP
End

Figure 4 Flowchart of sending data to the secondary ideal routing path.

situations, the pheromone values of intermediate nodes in the ideal secondary path will be determined and updated according to the following mathematical models:

$$Ph_{m,t} = Ph_{n,t} \tag{6}$$

$$Ph_{n,t} = \frac{Ph_{n,t} + \Delta Ph}{1 + \Delta Ph} \tag{7}$$

Here, (ΔPh) represents the amount of reinforcement that intelligent ants will add (when they pass through the node) to the pheromone values,

Algorithm 3 Transmit data to the secondary ideal routing path

```
Step 1. Detection of the secondary ideal routing path through the generation of intelligent ant swarms (TFIAs, TBIAs, and TIIAs) in case of detecting congestion sites at the nodes of the primary ideal routing path.
Step 2. Reduce the probabilities (pheromone values) of the nodes in the ideal primary path to 30% (in the case of choosing an ideal secondary path) by matching the data structures between the nodes of the primary and secondary ideal paths.
Step 3. Update the probabilities table (pheromone table) of all intermediate nodes of the secondary route by copying the pheromone tables from the intermediate nodes of the primary routing path.
Step 4. Determine the probabilities values of 50% between the primary and secondary ideal routing paths after the two paths have the same values in the pheromone tables.
Step 5. After matching the pheromone values (probabilities) between the primary and secondary routing paths, new incoming data packets can randomly choose which routing path they will use to reach their destination.
Step 6. Since the secondary ideal routing path has a wider bandwidth and fewer data packets passing through it, the probability of selecting the secondary ideal path nodes will increase. In contrast, the probability of the ideal primary path will decrease.
```

which depend on how well the trip is to reach the target. The amount of reinforcement added to the pheromone schedule of a given node can be estimated according to the following mathematical model:

$$\Delta Ph = \frac{1}{2} \times \left[\frac{ATT_{n,t}}{Trip_{n,t}} \right] \tag{8}$$

Where $(ATT_{n,t})$ denotes the average trip time to reach a specific target node (*n*), defined in the delay table of the intermediate node (*t*). $(Trip_{n,t})$ indicates the trip time to reach the target node (*n*) and pass through the node (*t*). It should be noted that formulas ((6) and (7)) satisfy the mathematical models ((1) and (5)).

According to the previous mathematical models, the probability values of the two ideal paths (primary and secondary) will equal 50%, so the data packet will randomly choose the appropriate routing path. However, since the bandwidth of the ideal secondary path is greater than that of the ideal primary path, the probability of the ideal secondary path will increase while the probability of the ideal primary path gradually decreases.

6 Simulation and Evaluation of Results

This part will evaluate the performance of our suggested ant-based intelligent routing approach by presenting three important performance metrics

Table 2 Network settings

Parameters	Values
Simulation software	NS2
Simulation environment size	1000×1000 m^2
No. of simulation devices	12
Packet size	1000 Bytes
The bandwidth of the connection link	1 Mbps
Traffic kind	CBR
Protocol agent	TCP
Time of simulation experiments	300 Sec
Buffer capacity	10 – 50 packets
Transmitting range	250 m
Diffusion model	Two-ray ground

computed from several experiments conducted using Network Simulator Version 2 (NS2) software [37]. The proposed algorithm used intelligent agents (ant swarms), represented by small fixed-size control packets, which travel periodically on the network to discover the optimal paths and update the data structures (pheromone and routing tables). The experimental results of the proposed algorithm will be compared with the results of the famous algorithms that use the intelligent ant-colony approach (WCETT [38] algorithm and CACO [39] algorithm) to prove the effectiveness of the suggested algorithm in increasing the performance of WMNs. The performance experiments of the three approach were carried out using the same WMN of (12) devices, assuming that device (2) is the sender of data (source) and device (12) is the receiver of the data (target). Table 2 lists all the network settings for WMN with their values as follows:

The simulation experiments involved the use of (3) influential performance measures to evaluate the performance of the presented approach by comparing its performance to that of the ACETT and CACO algorithms, as illustrated in the sub-sections below.

(1) Network throughput: It represents the total number of data packets transmitted in real-time to the simulation that have successfully reached the target node and acknowledged. The network throughput can be found according to the following form [40]:

$$N.Thr = \frac{T.No_P}{S.T} \tag{9}$$

Here $(T.No_p)$ denotes the actual number of data packets that arrived without fail on the target device. (*S.T*) denotes the actual simulation time.

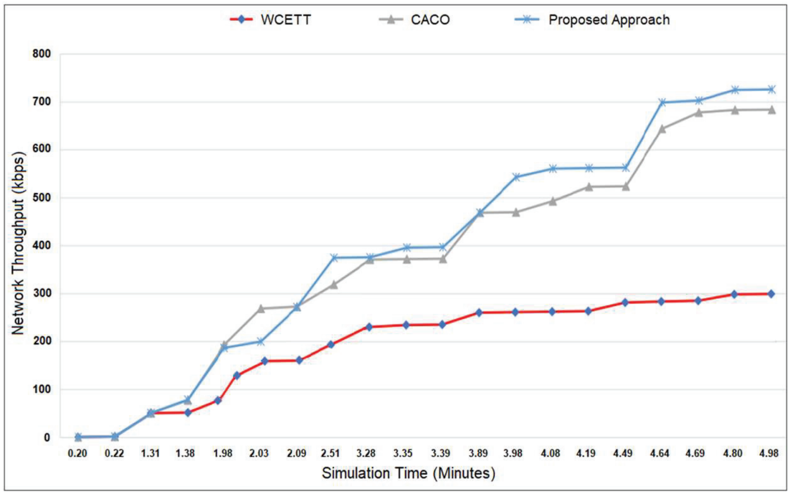

Figure 5 Network throughput.

Figure 5 displays the network throughput of WMN when using the proposed algorithm compared to the network throughput using the WACTT and CACO algorithms. Comparing the results, we can easily notice that the number of data successfully transmitted from device 2 to target device 12 when using the proposed algorithm is greater than the number of packets sent using WACTT and CACO algorithms. When the data packets start moving between the two nodes (2 and 12) at 2 minutes of the simulation time, we observe the stability of the network throughput with the proposed algorithm. Where, at 1.30 minutes, there is a congestion condition on the routing path (between node 7 and node 12), in this case, we note that the WACTT algorithm does not contain any effective mechanism to avoid congestion of the routing path, while the CACO algorithm will avoid the path (7-12) by directing data packets away to an uncongested path. On the other hand, the proposed algorithm solves the congestion problem by dividing the flow of data packets into secondary ideal routing paths (7-5-9-12). Thus, it will avoid data packet loss (due to congestion), leading to increased network throughput. The outstanding performance of the proposed approach came as a result of electing the ideal secondary path with the lowest cost (shortest distance) to avoid the congestion problem in the ideal primary path. Table 3 outlines the gross throughput of the WMN when implementing the suggested approach, WACTT and CACO algorithms.

Table 3 WMN throughput per time

Algorithms	No. of Packets Received at the Target	Network Transmission Ratio (%)
Proposed	731	73.4%
CACO	689	63.8%
WACTT	300	30%

(2) End-to-end delay: This metric indicates the time wasted for data packets to travel between the sending node and the target node, passing through the intermediate nodes. This time includes the time consumed to send control packets across the network to establish the appropriate routing path. The average end-to-end delay (*A.De*) in wireless networks can be calculated according to the following mathematical models [41]:

$$A.De = \frac{\sum TDe_p}{No.P_r} \tag{10}$$

Where $(No.P_r)$ denotes the actual number of packets that reached the target device. (TDe_p) denotes to the time actually used to pass each received packets at the target device, and this metric is calculated based on the following model:

$$TDe_p = Tr_p - Ts_p \tag{11}$$

Here, (Tr_p) denotes the precise second the target device received the data packet. (Ts_p) indicates to the time when the packet is sent from the arigin device.

Figure 6 shows the rate of end-to-end delay time of packets transmitted over network when using the proposed algorithm, that is much less than the delay time for data packets when WCETT and CACO algorithms are used. That means the proposed algorithm has better performance towards reducing delivery time, which increases the quality of service of WMN. The reason why the proposed algorithm has the lowest delay time compared to WCETT and CACO algorithms can be explained according to the following scenario:-

Using the WCETT algorithm, when the congestion of the routing path (7-12) occurs at 1.30 minutes, all the data packets sent through the path will wait continuously on the congested path to pass through and reach the target. Whereas, with the CACO algorithm, the data packets will be directed away from the congested path (7-12) (to the neighboring nodes of node 7); thus, the data packets will take more delay time to reach the target node. With the proposed algorithm, the congestion level will be

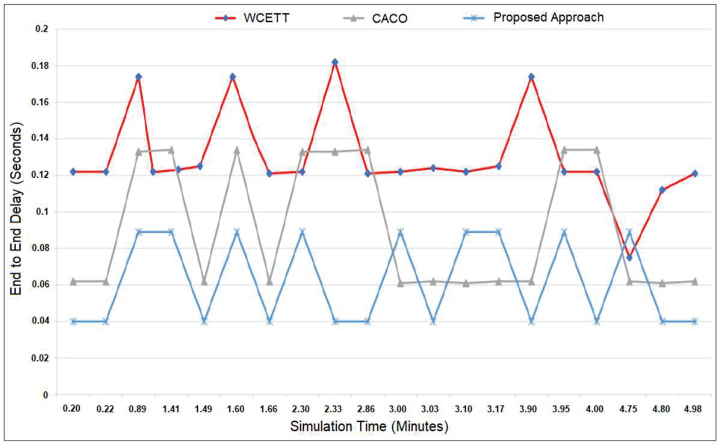

Figure 6 Average end-to-end delay.

Table 4 WMN end-to-end delay, per time

Algorithms	Average End-to-End Delay (Seconds)
Proposed	0.0562
WACTT	0.1021
CACO	0.0976

measured periodically, and before it reaches its peak, an ideal secondary path will be detected, and new data packets will be routed through it, thus reducing the delay time. Table 4 presents the average end-to-end delay of the WMN when using the proposed WACTT and CACO algorithms per time.

(3) Drop packets: Refers to the actual number of unsuccessful packets exchanges through the network per time. The number of dropped data packets ($No.\ Dro_p$) in wireless networks can be calculated according to the following model [42]:

$$No.Dro_p = No.Tr_p - No.Re_p \tag{12}$$

Where ($No.\ Tr_p$) represents the total data packets the source device sends during the simulation time. ($No.Re_p$) refers to the total data packets the target device receives during the simulation time.

Figure 7 shows that the actual number of dropped data that did not swimmingly reach the target node when implementing the proposed algorithm is very less than the total number of data packets when WCETT and CACO algorithms are used. That mean the suggested approach achieved higher

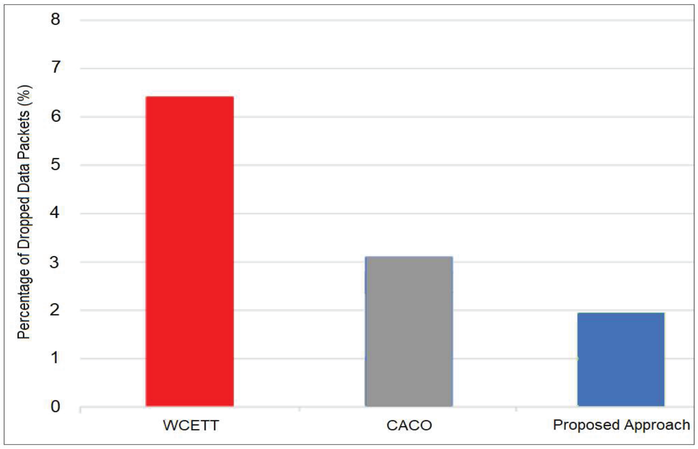

Figure 7 Percentage of dropped packets.

Table 5 Drop packets per time

Algorithms	Percentage of Dropped Packets (%)
Proposed	1.9%
WACTT	6.4%
CACO	3.1%

throughput than other algorithms. The proposed algorithm's good performance came from early avoidance of congestion situations across the routing paths by selecting new ideal paths and continuously checking congestion situations in new secondary ideal paths, thus reducing the number of dropped data packets. Table 5 shows the packet loss percentage in WMN when using the proposed WCETT and CACO algorithms per time.

7 Conclusions and Future Works

This work proposed an efficient routing algorithm for WMNs, which detects and avoids congestion situations in the primary ideal routing path by balancing a load of data packets with other secondary ideal paths. The proposed algorithm is based on its basic operations on the mechanisms of Ant Colony Optimization (ACO) algorithms. In general, the proposed algorithm relied in its basic operations on three important techniques: detecting heavy congestion within the ideal paths currently used for data transmission, creating ideal secondary paths with updated pheromone values, and distributing the traffic load (data packets) between the primary and secondary ideal paths. The NS2 software is used as a simulator to prove the effectiveness of the suggested approach for WMNs by evaluating its performance with that of WCETT and CACO algorithms. Simulation experiments demonstrated that the suggested approach has a good performance that improves throughput, end-to-end delay rate, and packet loss rate of WMN compared to other approachs. The results displayed that the suggested approach achieved a network throughput ratio of 73.4%, while the WCETT and CACO algorithms achieved a network throughput ratio of 30% and 63.8%, respectively. The experiment results also showed that the proposed algorithm achieved an average end-to-end delay close to 0.0562.

In contrast, WCETT and CACO algorithms achieved an average end-to-end delay close to 0.1021 and 0.0976, respectively. Furthermore, the results showed that the proposed algorithm achieved a lower percentage of dropped packets by 6.97% and 0.99% compared to the WCETT and CACO algorithms, respectively. That means the proposed algorithm achieved the best performance by increasing network throughput, decreasing end-to-end delay, and reducing the number of lost packet. The suggested algorithm's good performance is due to effective mechanisms to avoid network congestion situations by detecting the secondary ideal paths, updating the pheromone tables appropriately, and dividing the flow of data packets between the ideal routing routes.

As a promising future work, we are looking forward to developing the suggested approach by adding effective mechanisms to identify numerous paths while considering multiple radios' intra/inter flow interferences and appropriate channel space utilization. We also plan to evaluate the performance of our suggested approach with other well-known intelligent ant-based routing algorithms according to other performance measurements. Other plans to develop the proposed algorithm include adding mechanisms to raise the security level to avoid potential network attacks that negatively affect network performance.

References

[1] H. Q. Abdulrab et al., "Optimal coverage and connectivity in industrial wireless mesh networks based on Harris' hawk optimization algorithm," IEEE Access, vol. 10, pp. 51048–51061, 2022.

[2] S. Mahajan, R. Harikrishnan, and K. Kotecha, "Adaptive Routing in Wireless Mesh Networks Using Hybrid Reinforcement Learning Algorithm," IEEE Access, vol. 10, pp. 107961–107979, 2022.

[3] S. M. Taleb, Y. Meraihi, A. B. Gabis, S. Mirjalili, A. Zaguia, and A. Ramdane-Cherif, “Solving the mesh router nodes placement in wireless mesh networks using coyote optimization algorithm,” IEEE Access, vol. 10, pp. 52744–52759, 2022.

[4] Y. Tian and T. Yoshihiro, “Traffic-demand-aware collision-free channel assignment for multi-channel multi-radio wireless mesh networks,” IEEE Access, vol. 8, pp. 120712–120723, 2020.

[5] B. Shin and D. Lee, “An efficient local repair-based multi-constrained routing for congestion control in wireless mesh networks,” Wireless Communications and Mobile Computing, vol. 2018, pp. 1–17, 2018.

[6] S. K. Narayana and N. T. Hosur, “Priority-based trust efficient routing using ant colony optimization for IoT-based mobile wireless mesh networks,” International Journal of Intelligent Engineering and Systems, vol. 15, no. 2, pp. 99–106, 2022.

[7] X. Deng et al., “An ant colony optimization-based routing algorithm for load balancing in Leo satellite networks,” Wireless Communications and Mobile Computing, vol. 2022, 2022.

[8] V. de Figueiredo Marques, J. Kniess, and R. S. Parpinelli, “An energy-efficient mesh LNN routing protocol based on ant colony optimization,” in 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), 2018: IEEE, pp. 43–48.

[9] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” IEEE computational intelligence magazine, vol. 1, no. 4, pp. 28–39, 2006.

[10] C. Liu, G. Chang, J. Jia, L. Jin, and F. Li, “A load balanced routing protocol based on ant colony algorithm for wireless mesh networks,” in 2011 Fifth International Conference on Genetic and Evolutionary Computing, 2011: IEEE, pp. 295–298.

[11] A. K. Gupta, H. Sadawarti, and A. K. Verma, “Computation of pheromone values in antnet algorithm,” International Journal of Computer Network and Information Security, vol. 4, no. 9, p. 47, 2012.

[12] S. Mani and R. Ponraj, “Optimization With Congestion Aware Routing In Mesh Topology Optimization With Congestion Aware Routing In Mesh Topology,” IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr–May, 2014.

[13] C. P. Reddy and P. Venkata Krishna, “Ant-inspired level-based congestion control for wireless mesh networks,” International Journal of Communication Systems, vol. 28, no. 8, pp. 1493–1507, 2015.

[14] K. N. Kapadia and D. D. Ambawade, “Congestion aware load balancing for multi-radio Wireless Mesh Network,” in 2015 International

conference on communication, information & computing technology (ICCICT), 2015: IEEE, pp. 1–6.

[15] Z. Zou and Y. Qian, "Wireless sensor network routing method based on improved ant colony algorithm," Journal of Ambient Intelligence and Humanized Computing, vol. 10, pp. 991–998, 2019.

[16] S. Harikishore and V. Sumalatha, "Ant colony optimization based energy efficiency for improving opportunistic routing in a multimedia wireless mesh network," Indonesian Journal of Electrical Engineering and Computer Science, vol. 16, no. 3, pp. 1371–1378, 2019.

[17] Q. Q. Li and Y. Peng, "A wireless mesh multi-path routing protocol based on sorting ant colony algorithm," Procedia Computer Science, vol. 166, pp. 570–575, 2020.

[18] S. L. Yadav, R. Ujjwal, S. Kumar, O. Kaiwartya, M. Kumar, and P. K. Kashyap, "Traffic and energy-aware optimization for congestion control in next-generation wireless sensor networks," Journal of Sensors, vol. 2021, pp. 1–16, 2021.

[19] J. Peng, Z. Cao, and Q. Huang, "Routing with Ant Colony Optimization in Wireless Mesh Networks," in International Conference on Parallel and Distributed Computing: Applications and Technologies, 2021: Springer, pp. 15–26.

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

[21] F. Bokhari and G. Zaruba, "AMIRA: interference-aware routing using ant colony optimization in wireless mesh networks," in 2009 IEEE Wireless Communications and Networking Conference, 2009: IEEE, pp. 1–6.

[22] F. Bokhari and G. Zaruba, "AntMesh: an efficient data forwarding scheme for load balancing in multi-radio infrastructure mesh networks," in The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010), 2010: IEEE, pp. 558–563.

[23] M. Chandana and S. Thakur, "Ant-Net: An adaptive routing algorithm," in 2016 IEEE 1st international conference on power electronics, intelligent control, and energy systems (ICPEICES), 2016: IEEE, pp. 1–4.

[24] A. K. Gupta, H. Sadawarti, and A. K. Verma, "Computation of pheromone values in the ancient algorithm," International Journal of Computer Network and Information Security, vol. 4, no. 9, p. 47, 2012.

[25] N. Girme, "Load balancing in network to increasing performance ratio of packet delivery using ant colony optimization," International Journal of Computer Applications, vol. 110, no. 13, pp. 16–20, 2015.

[26] X. Wang and D. Chen, “Link design for wireless optical communication network based on ant colony algorithm,” Journal of Communications and Networks, vol. 24, no. 2, pp. 184–191, 2022.

[27] Q. Luo, H. Wang, Y. Zheng, and J. He, “Research on path planning of mobile robot based on improved ant colony algorithm,” Neural Computing and Applications, vol. 32, pp. 1555–1566, 2021.

[28] M. Umlauft and W. Elmenreich, “Ant algorithms for routing in wireless multi-hop networks,” The Application of Ant Colony Optimization, p. 41, 2021.

[29] S. Sheikh, R. Wolhuter, and H. A. Engelbrecht, “An adaptive congestion control and fairness scheduling strategy for wireless mesh networks,” in 2015 IEEE symposium series on computational intelligence, 2015: IEEE, pp. 1174–1181.

[30] K. Matsuo, S. Sakamoto, T. Oda, A. Barolli, M. Ikeda, and L. Barolli, “Performance analysis of WMNs by WMN-GA simulation system for two WMN architectures and different TCP congestion-avoidance algorithms and client distributions,” International Journal of Communication Networks and Distributed Systems, vol. 20, no. 3, pp. 335–351, 2018.

[31] E. Pertovt, K. Alic, A. Svigelj, and M. Mohorcic, “Cancar-congestion-avoidance network coding-aware routing for wireless mesh networks,” KSII Transactions on Internet and Information Systems (TIIS), vol. 12, no. 9, pp. 4205–4227, 2018.

[32] D. J. David, V. Jegathesan, and T. J. Jebaseeli, “Distributed optimal congestion control and channel assignment in wireless mesh networks,” TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 19, no. 2, pp. 414–420, 2021.

[33] D.-Y. Qin, H.-W. Li, L. Ma, H.-B. Ma, and Q. Ding, “An ant colony algorithm based congestion elusion routing strategy for mobile ad hoc networks,” Journal of Harbin Institute of Technology, vol. 3, 2013.

[34] A. Nayyar and R. Singh, “Ant colony optimization—computational swarm intelligence technique,” in 2016 3rd International Conference on Computing for sustainable global development (INDIACom), 2016: IEEE, pp. 1493–1499.

[35] M. Dorigo and K. Socha, “An introduction to ant colony optimization,” in Handbook of Approximation Algorithms and Metaheuristics, Second Edition: Chapman and Hall/CRC, 2018, pp. 395–408.

[36] M. Khan, M. F. Majeed, and S. Muhammad, “Evaluating radio propagation models using destination-sequenced distance-vector protocol for

MANETs," Bahria University Journal of Information & Communication Technologies (BUJICT), vol. 10, no. 1, 2017.

[37] K. Fall and K. Varadhan, "ns notes and documentation. The VINT Project," UC Berkeley, LBL, USC/ISI, and Xerox PARC, 1997.

[38] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in Proceedings of the 10th annual international conference on Mobile Computing and Networking, 2004, pp. 114–128.

[39] S. Mani and R. Ponraj, "Optimization with Congestion Aware Routing In Mesh Topology," IJREAT International Journal of Research in Engineering & Advanced Technology, vol. 2, no. 2, 2014.

[40] A. Vinitha and M. Rukmini, "Secure and energy-aware multi-hop routing protocol in WSN using Taylor-based hybrid optimization algorithm," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 5, pp. 1857–1868, 2022.

[41] M. A. Moridi, Y. Kawamura, M. Sharifzadeh, E. K. Chanda, M. Wagner, and H. Okawa, "Performance analysis of ZigBee network topologies for underground space monitoring and communication systems," Tunnelling and Underground Space Technology, vol. 71, pp. 201–209, 2018.

[42] S. M. Bozorgi and A. M. Bidgoli, "HEEC: A hybrid unequal energy efficient clustering for wireless sensor networks," Wireless Networks, vol. 25, pp. 4751–4772, 2019.

Biographies

Fadhil Mohammed Salman received his BS in computer science from the University of Anbar in Iraq in 2005. From 2010 to 2012, he completed his master's in Computer Science from the Department of Computer Science

University of Babylon, Iraq. He received his Ph.D. thesis in information technology from the Department of Software College of Information Technology University of Babylon, Iraq, in 2019. He has been a lecturer at the Babylon Education Directorate since 2007 till now. His research interests include computer networks, Network Security, Data compression, Image processing, and Data mining.

Ahssan Ahmmed Mohammed Lehmoud received his BS in computer science from the University of Babylon in Iraq in 2006. From 2009 to 2011, he completed his master's in Computer Science from the Department of Computer Science and Information Technology at Dr. BabaSahib Ambedkar Marthwada University, India. He received his Ph.D. thesis in information technology from the Department of Software College of Information Technology University of Babylon, Iraq, in Abril 2018. He has been a lecturer at the Babylon Education Directorate since 2008 till now. His research interests include information security, Network security, Cryptography, Steganography, Image processing, and Data mining.

Fanar Ali Joda in Hilla, Babylon City, Iraq, on November 27, 1980. He received a BSc degree in computer science in 2002 from the University of

Babylon/Computer Science Dept.-Iraq. He received the MSc in Data Compression 2015 from Babylon University – Iraq. He received his Ph.D. degree in Computer vision in 2019 from Babylon University-Iraq. Currently, Joda is Lecturer at Al-Mustaqbal University – Air Conditioning & Refrigeration Engineering Techniques Dept. His research interests include computer vision and information hiding.